50 лет модулярной арифметике

Юбилейная Международная научно-техническая конференция

(В рамках V Международной научно-технической конференции «Электроника и информатика 2005»)

Сборник научных трудов

Москва, Зеленоград, 2005

Министерство образования и науки Российской Федерации.
Федеральное агентство по образованию.
Московский государственный институт электронной техники (технический университет).
Открытое акционерное общество "Ангстрем".
Институт проблем проектирования в микроэлектронике (ИППМ) РАН.
Государственное унитарное предприятие "Научно-производственный центр СПУРТ".
Еженедельник PC-Week RE и Виртуальный компьютерный музей http://www.computer-museum.ru.
НИИ прикладных физических проблем им. А.Н.Севченко, г. Минск, Республика Беларусь.
Научно-внедренческая фирма "КРИПТОН", г. Киев, Украина.
Институт математики Национальной Академии Наук РК, Алма-Ата, Республика Казахстан.
Computer Algorithm & Program Development Corp., New York, USA.

Юбилейная Международная научно-техническая конференция

50 лет модулярной арифметике

(В рамках V Международной научно-технической конференции "**Электроника и информатика – 2005**")

(*Зеленоград, 23 – 25 ноября 2005 г.)*

СБОРНИК НАУЧНЫХ ТРУДОВ

ББК 32.973-04
П99
УДК 004.312 (066)

П99 «**50 лет модулярной арифметике**». Юбилейная Международная научно-техническая конференция (В рамках V Международной научно-технической конференции "Электроника и информатика – 2005"). Сборник научных трудов. – М.: ОАО «Ангстрем», МИЭТ, 2006, - 775 с.
ISBN 5-7256-0409-8

В сборнике научных трудов Юбилейной конференции «50 лет модулярной арифметике» представлены результаты научных исследований в области Системы счисления остаточных классов (СОК) и модулярной арифметики на ее основе за пятидесятилетний период с момента зарождения СОК. В связи с намечающимся синтезом модулярной и троичной арифметик, представлены также материалы по троичной арифметике и трехзначной диалектической логике.

Тексты научных трудов представлены в авторской редакции

ISBN 5-7256-0409-8

Введение

В 2005 году исполняется 50 лет модулярной арифметике на основе системы счисления остаточных классов.

За истекшие 50 лет модулярная арифметика пережила периоды и бурного развития, и серьезных спадов. Ведущую роль в этом направлении сыграла группа зеленоградских специалистов, возглавляемая крупными учеными в области вычислительной техники, профессорами И.Я. Акушским, Д.И. Юдицким и В.М. Амербаевым. В настоящее время наблюдается прогрессирующий рост интересов к модулярной арифметике среди разработчиков сложных систем, связанных с обработкой сигналов и изображений, с криптографией, с вычислениями в больших диапазонах и т.п.

50-летний юбилей – хороший повод для подведения итогов и оценки перспектив модулярной арифметики. В связи с этим группа предприятий России, Казахстана, Белоруссии, США и Украины приняла совместное решение о проведении специальной юбилейной Международной научной конференцией "**50 лет модулярной арифметике**".

Конференция проводится в два этапа:

- Заочная Internet-Конференция в период март – октябрь 2005 г.,
- Завершающая Очная Конференция в Зеленограде (МИЭТ) 23-25 ноября 2005 г.
- Издание трудов конференции – сентябрь 2006 г.

Учитывая намечающийся синтез двух перспективных научных направлений развития вычислительной техники – модулярной арифметики и троичной системы, оргкомитет пригласил для участия в конференции ученых, разрабатывающих теории применения троичной арифметики и трехзначной диалектической логики при построении средств вычислительной техники.

В сборнике трудов конференции статьи участников размещены в следующем порядке: обзорно-исторические, общие вопросы, научные разработки, прикладные разработки.

Оргкомитет благодарит Б.М. Малашевича за неоценимый вклад в организацию, подготовку и проведение конференции, выпуск сборника ее трудов, а также за восстановление зеленоградских страниц истории развития отечественной вычислительной техники.

Оргкомитет

Израиль Яковлевич Акушский

Основоположник отечественной модулярной арифметики

30 июля 1911 – 2 апреля 1992 гг.

Трудовую деятельность Израиль Яковлевич Акушский начал в 1931 г. вычислителем НИИ математики и механики МГУ. С 1956 г. он в СКБ-245, где встречается с Д.И. Юдицким и они впервые узнают об идее М. Валаха о Системе остаточных классов (СОК). Далее они совместно разрабатывают эту идею, воплотив ее в реальные модулярные ЭВМ Т-340А, К-340А, Алмаз и 5Э53. И.Я. Акушскому принадлежит ведущая роль в разработке эффективных методов машинной арифметики и реализации этих методов для повышения производительности ЭВМ. Им была построена теория самокорректирующихся арифметических кодов в СОК, позволяющая резко повысить надежность ЭЦВМ.

В 1953 г. при Президиуме АН Казахстана И.Я. Акушский образовал Лабораторию машинной и вычислительной математики при Президиуме АН Каз. ССР, а в 1966 г. в МИЭТ – кафедру Вычислительной математики (ныне «Информатики и программного обеспечения вычислительных систем»), которую возглавлял до 1974 г.

И.Я. Акушский опубликовал более 200 научных трудов, в т.ч. 12 монографий, имеет более 90 изобретений и ряд зарубежных патентов. Им подготовлено более 80 кандидатов и 10 докторов наук.

УДК. 681.3

В.М. Амербаев, И.Т. Пак

МОДУЛЯРНОЙ АРИФМЕТИКЕ – 50 ЛЕТ

(ГУП Научно-производственный центр «СПУРТ», РФ, Зеленоград, Институт математики НАН Республики Казахстан, г. Алма-Ата)

Рассмотрена краткая история зарождения и развития системы счисления остаточных классов (модулярной арифметики). Показано, что современные модели компьютерных арифметик по своей природе являются архимедовыми. Дан анализ понятия «модулярная арифметика». Ее перспективу составляет разработка различных приемов комплексирования (сближения) достоинств позиционной и модулярной арифметик с целью достижения сверхпроизводительности и сверхнадежности вычислительных структур, спроектированных на элементной базе, предельно реализующих возможности технологии.

Прошло 50 лет с тех пор, как чешский учёный-инженер Миро Валах (в 1955 году) предложил для кодирования целых чисел в математических машинах использовать кольцо вычетов по составному модулю с попарно взаимно-простыми основаниями. После первых публикаций Антонина Свободы и Миро Валаха (1955г.) эта идея была с энтузиазмом подхвачена мировой научной общественностью в области компьютерных технологий (в большей степени – американской) и к 1959 году из открытых публикаций можно было получить достаточно чёткое представление о базовых принципах конструирования модулярной арифметики, её достоинствах и не-

достатках. Сформировалось новое движение научной мысли в науке о вычислениях и о компьютерной арифметике. Начальные вехи этого движения за рубежами СССР таковы:

- Свобода и Валах (1954-1958гг [1-6]) – исследовали базовые свойства арифметики в остатках (модулярной арифметики).
- Айкен и Семон (1959г [7-8]) – показали преимущества модулярной арифметики.
- Гарнер (1959г [9-10]) – исследовал арифметические свойства модулярной арифметики с целью построения самокорректирующихся арифметичных кодов, исследовал принципы синтеза систем счисления.
- Чини (1961г [11]) сконструировал цифровой коррелятор в остатках.
- Сабо (1962г [12]) – доказал, что знак любого числа в остаточных кодах является функцией всех остатков (что, собственно, сродни свойству самокоррекции кодов в остатках, основанной на высокой чувствительности этих кодов к малейшему изменению любого остатка по любому основанию).
- Шапиро (1964г [13]) – исследовал принципы организации параллельных вычислений в модулярной арифметике.

В последующем работы в области модулярной арифметики разрастаются снежным комом.

В Советском Союзе первая открытая публикация по модулярной арифметике состоялась в 1964 году в широко известном сборнике переводов под редакцией А.А. Ляпунова и О.Б. Лупанова – «Кибернетический сборник» №8 – серией статей ведущих зарубежных специалистов в области модулярной арифметики.

Далее:

- Андрианов Е.С. (1964г [14,15]) – исследовал вопросы реализации важнейших немодульных операций.
- Акушский И.Я., Юдицкий Д.И., Машинная арифметика в остаточных классах, –М.:1968 (популярная в России монография).

- Торгашов В.А., Система остаточных классов и надёжность ЦВМ, М.:1973 (известная в России монография).

Однако, значительно раньше упомянутого выше «Кибернетического сборника», работы в области модулярной арифметики в СССР были начаты группой учёных и инженеров – И.Я. Акушским, Д.И. Юдицким, Е.С. Андриановым, перед которыми была поставлена задача разработать модулярный процессор [27]. Результатами теоретических исследований и практического использования стала докторская диссертация И.Я. Акушского и кандидатские диссертации Д.И. Юдицкого и Е.С. Андрианова, защищённые в 1964-65 годах, а так же модулярные ЭВМ Т-340А (экспериментальная) и К-340А (производилась серийно).

В 1964 г. возглавляемый Д.И. Юдицким и И.Я. Акушским коллектив приверженцев модулярной арифметики переместился в создаваемый тогда Центр микроэлектроники в Зеленограде. На базе этого коллектива был образован Специализированный вычислительный центр (СВЦ) с задачей создания высокопроизводительной модулярной ЭВМ 5Э53 [27].

Публикация работ в «Кибернетическом сборнике», а затем монографии Акушского И.Я и Юдицкого Д.И., стимулировали независимые исследования в конце 60-х гг.: в Ленинграде – Торгашов В.А.; В Киеве – Вышенский В.А., Мороз И.Г., Петушак В.Д.; в Алма-Ате – Амербаев В.М., Пак И.Т., Казангапов А.Н., Бияшев Р.Г.; в Тбилиси – Хацкевич В.Х.; в Минске – Коляда А.А. и Чернявский А.Ф., активно развивающие оригинальные идеи в области модулярных вычислений по настоящее время.

В семидесятых и последующих годах высокий имидж модулярной арифметики в Союзе (вне стен СВЦ) поддерживался блестящим талантом И.Я. Акушского, увлекающего идеями модульной арифметики молодых учёных и привлекающего их к разработкам в этой области. Не имея возможности встречаться с молодыми «соковцами» в стенах закрытого учреждения (СВЦ), он широко открыл перед ними двери своей квартиры, где Галина Петровна (жена Израиля Яковлевича) и Елизавета Соломоновна (мать его жены) радушно встречали всех посетителей. Трудно представить себе личностное влияние И.Я. Акушского на развитие модулярной арифметики в Союзе без тёплого и внимательного соучастие его семейного окружения. В последующем очаги модульной арифметики возникли в

Нижнем Новгороде, Ставрополе, Воронеже, Харькове, Днепропетровске, Дагестане. Были созданы новые большие школы, возглавляемые Чернявским А.Ф. и Колядой А.А., Червяковым Н.И., Краснобаевым В.А. и др.

Несмотря на столь широкое распространение модулярной арифметики в технике вычислений, само понятие «модулярная арифметика» заслуживает более подробного анализа. Мы настолько привыкли к позиционной арифметике, что понятие позиционный код отождествляется нами с понятием числа. В действительности это не так. Дело в том, что в понятии «числа» присутствует некая тройственность:

- число, как некий объект (возникший исторически в результате осмысления таких процессов человеческой деятельности, как счёт и измерение),
- способ представления числа , т.е. синтаксис объекта,
- смысл числа (количество, величина), т.е. семантика объекта. Семантика числа в полной мере отражается системой аксиом вещественных чисел **R.**

Как известно, таких аксиоматик, дающих полное описание понятия числа несколько: дедекиндова, канторова, вейерштрассовская:

- Вейерштрассовская трактовка числа существенно использует позиционное представление чисел, и тем самым, как бы, универсализируется сам факт позиционного представления чисел.
- Канторова трактовка базируется на всюду плотности рациональных чисел **Q** в **R** или, точнее на пополнении **Q** до **R**.
- Дедекиндово понимание числа эксплуатирует факторизацию – разбиение **Q** на классы эквивалентности, так называемые сечения Дедекинда.

Все эти аксиоматики порождают (с точностью до изоморфизма, и это доказано) одно и то же множество объектов, называемых числами. Тем самым показано, что позиционное представление числа равнозначно всем другим средствам описания чисел. Требования к способу представления чисел чётко определено Леонардом Эйлером: «Всякий способ изображения чисел требует к арифметическим действиям особых правил, которые подлежит производить от свойств оных чисел, кои употребляются» [28]. Эйлеровские требования к способу изображения чисел можно рассматривать как научную программу построения компьютерных моделей чисел. Как

следует из аксиоматического определения числа – целые числа составляют конструктивную основу, с помощью которых строятся вещественные числа. Понятие же целого числа, возникло в результате практики счёта. «Все числа, как ты знаешь, состоят из некоторого количества единиц» (Диафант). Целые числа образуют бесконечное упорядоченное кольцо, базовые операции которого (сложение, вычитание, умножение) согласованы с отношением естественного порядка чисел. С помощью целых чисел (т.е. пар целых чисел) определяется также понятие дробного (рационального) числа. Необходимость в определении множества рациональных чисел возникла в результате обобщения измерительных процессов. Опыт измерений отражен в, так называемой, аксиоме Архимеда, которая в наиболее рафинированной форме формулируется следующим образом: любое вещественное число x единственным образом представимо в виде суммы целой части числа, обозначим её временно символом $W(x)$, и дробной части того же числа, обозначим её символом $f(x)$:

$$\forall x \in R \quad \exists! \quad W(x) \in Z; \quad x = W(x) + f(x)$$

Наибольшее распространение имеет имеют два случая выбора дробной доли:

- Наименьшая неотрицательная дробная часть, т.е.

$$0 \le f(x) < 1 \quad \forall x \in R$$

этот случай соответствует экстремальному выбору целой части, как наибольшего целого числа, не превосходящего x; такую целую часть принято обозначать символом $\lfloor x \rfloor$ [16].

- Абсолютно наименьшая дробная часть:

$$-\frac{1}{2} \le f(x) < \frac{1}{2} \quad \forall x \in R$$

которая соответствует экстремальному выбору целой части, как ближайшего к x целого числа: такую целую часть принято обозначать символом $\lfloor x \rceil$ [16].

Итак, аксиома Архимеда записывается в двух формах и утверждает она единственность разложений вида:

$$\forall x \in R \quad \exists! \ \lfloor x \rfloor \in Z : \ x = \lfloor x \rfloor + f(x),$$

где $0 \le f(x) < 1$

$$\forall x \in R \quad \exists! \ \lfloor x \rfloor \in Z : \ x = \lfloor x \rceil + \bar{f}(x),$$

где $-\frac{1}{2} \le \bar{f}(x) < \frac{1}{2}$

Связь между этими двумя разложениями устанавливается соотношениями

$$\lfloor x \rceil + \left\lfloor x + \frac{1}{2} \right\rfloor, \ \bar{f}(x) = f\left(x + \frac{1}{2}\right) - \frac{1}{2}.$$

Из аксиомы Архимеда следует, что любое вещественное число с любой наперёд заданной точностью может быть приближено рациональным числом.

Пусть M – достаточно большое число (натуральное), тогда

$$\forall x \in R \qquad Mx = \lfloor Mx \rceil + \bar{f}(Mx) \text{ или}$$

$$x = \frac{1}{M} \lfloor Mx \rceil + \frac{\bar{f}(Mx)}{M}$$

отсюда следует, что абсолютная величина ошибки представления числа x дробью $\lfloor Mx \rceil / M$ не превышает величины $\frac{1}{2M}$. Величина M называется масштабным множителем. Если M – постоянный масштаб, то представление чисел x дробями вида $\lfloor Mx \rceil / M$ имеет равномерно-распределённую абсолютную ошибку, что удобно для больших значений x и не удобно для малых. Для того, чтобы устранить этот недостаток режима фиксированной запятой вводят понятие относительной погрешности. Требуется, чтобы уже относительная, а не абсолютная ошибка, пред-

ставления вещественных чисел рациональными была равномерно распределённой. Это требование обусловливает введение переменного масштабного множителя и соответствующего представления чисел с плавающей запятой. Теперь абсолютная ошибка растёт с увеличением числа x. Балансирование между этими представлениями чисел определяется стратегией вычислений и относится к категории метатеории вычислений (т.е. семантики вычислений). Важно отметить то, что масштабирование регулирует информационную ёмкость представления чисел рациональными дробями.

Из сказанного следует, что если строить модели компьютерных чисел, то в первую очередь необходимо построить модель арифметики целых чисел и, далее, используя принцип масштабирования, добиваться нужных точностных характеристик. Этот вывод касается, так сказать, семантического, т.е. метрического, количественного, точностного аспекта представления чисел, тогда как реализация арифметических операций связана со способом кодирования (т.е. изображения) чисел. Как будет показано ниже, реализация арифметических операций также связана с масштабирующими множителями, но уже не столько с их величинами, сколько с их структурной (мультипликативной) характеристикой. Варьируя величиной и структурой масштабирующих множителей можно получить разнообразные структуры компьютерных арифметик. Современные технологии проектирования вычислительных средств на различной элементной базе микроэлектроники позволяют адаптировать вычислительные среды к различным арифметическим структурам с целью эффективного сочетания таких показателей вычислительных процессов, как точность, параллелизм, надёжность, реконфигурируемость и т.п. В этом смысле, чем богаче набор различных арифметических структур, тем более успешно может быть решена проблема оптимизации вычислений при решении того или иного класса задач на данной вычислительной среде или данной элементной базе.

Возвращаясь к вопросу о компьютерной модели целых чисел Z следует заметить, что сама модель должна быть конечной и максимально точно представлять все арифметические операции над целыми числами и должна сохранять естественный порядок целых чисел.

Наилучшим образом такая модель, состоящая из N элементов,

представляется набором целых чисел полной системы абсолютно наименьших вычетов множества целых чисел Z по модулю N:

$$Z_N^- = \left\{ x \in Z \mid -\frac{N}{2} \le x < \frac{N}{2} \right\}$$

В ряде случаев, по чисто техническим причинам, бывает удобно за конечную модель принять наименьшие неотрицательные вычеты по модулю N, т.е.

$$Z_N = \{ x \in Z \mid 0 \le x < N \}$$

При построении колец Z_N^- и Z_N, имитирующих на компьютерном уровне кольцо Z мы вновь не обходимся без аксиомы Архимеда. Действительно, согласно аксиоме Архимеда справедливо:

$$\forall x \in Z \qquad \frac{x}{N} = \left\lfloor \frac{x}{N} \right\rfloor + f\left(\frac{x}{N} \right)$$

отсюда

$$\forall x \in Z \qquad x = \left\lfloor \frac{x}{N} \right\rfloor N + f\left(\frac{x}{N} \right) N$$

Так как $x, \left\lfloor \frac{x}{N} \right\rfloor N \in Z$, то $f\left(\frac{x}{N} \right) N \in Z$. Обозначим это целое число символом

$$|x|_N := f\left(\frac{x}{N} \right) N$$

В теории чисел величина $|x|_N$ называется наименьшим неотрицательным вычетом целого числа x по модулю N. Область зна-

чений функции $y = |x|_N$ составляет числа $\{0,1,2,..., N-1\}$. Это множество является кольцом наименьших неотрицательных вычетов по модулю N и обозначается символом Z_N, с кольцевыми операциями $\oplus, \otimes$, определяемыми следующим образом:

$$\forall x, y \in Z_N \quad x \oplus y := |x + y|_N$$

$$x \otimes y := |x \cdot y|_N$$

Функция $y = |x|_N$ определена на Z и называется функцией вычетов по модулю N.

Пусть $|x|_N = r \qquad r \in Z_N$

Очевидно, целые числа $z \in Z_N$ вида $z = r + kN \quad \forall k \in Z$ отображаются функцией вычета в точку r. Множество всех этих целых чисел называется классом вычетов и обозначается так (r)

$$(r) = \{z \in Z \mid z = kN + r, \forall k \in Z\}$$

Два числа $\forall z_1, z_2 \in Z$, принадлежащие одному классу вычетов, называются сравнимыми по модулю N, что записывается следующим образом: $z_1 \equiv z_2 (\mathrm{mod}\, N)$.

Таким образом, сравнение $z_1 \equiv z_2 (\mathrm{mod}\, N)$ равносильно двум равенствам:

$(z_1) = (z_2)$ – равенство классов вычетов по модулю N,

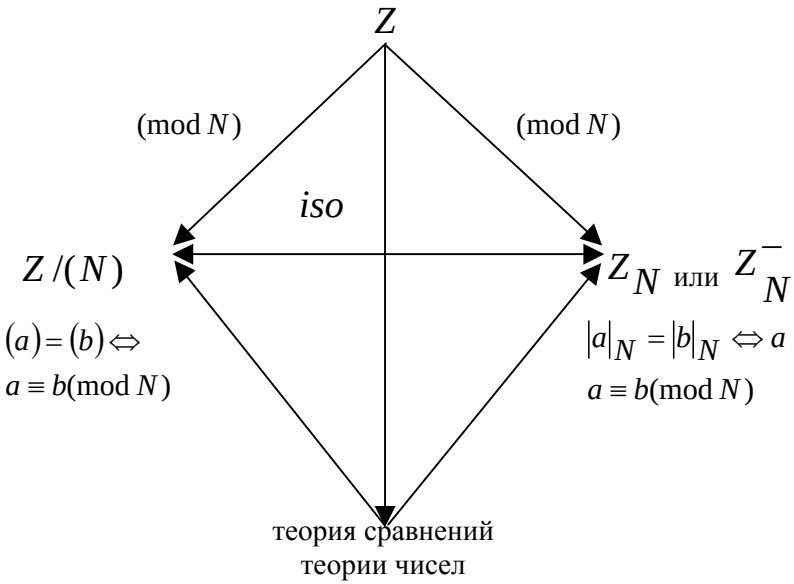
$|z_1|_N = |z_2|_N$ – равенство вычетов чисел z_1 и z_2 по модулю N.

Множество всех классов вычетов по модулю N обозначается сим-

волом $Z/(N)$ и называется фактор-кольцом кольца Z по модулю N. Оно изоморфно кольцу вычетов Z_N; пишут $Z/(N) \cong Z_N$.

Таким образом, с функцией вычетов связана следующая схема гомоморфных (т.е. сохраняющих кольцевые свойства) отображений (рис.1).

При этом только правая ветвь $Z \to Z_N$ отвечает требованиям сформулированным Л. Эйлером к конструированию конечной модели кольца Z.



Из этой схемы явно просматривается неточность термина «система счисления остаточных классов», идущая от А. Свободы [4], т.к. «остаточные классы» или «классы вычетов» образуют элементы фактор-кольца $Z/(N)$, тогда компьютерная модель целых чисел

является кольцом вычетов (остатков) Z_N . Возможно, этой замысловатости термина СОК мы обязаны тому, что один из ведущих в Союзе разработчиков ЭВМ М.А. Карцев в своей книге [17] назвал СОК – «экзотической системой счисления» и тем самым углубил недопонимание значимости модулярной арифметики у многих разработчиков вычислительных средств.

В действительности, никакой экзотичности нет. Модулярная арифметика, как и всякая другая компьютерная арифметика, на регистровом уровне является кольцом вычетов Z_N , что существенно важно, т.к., во-первых, Z_N образует отрезок из множества Z , точнее подмножество множества Z , непосредственно идущих друг за другом в порядке возрастания целых чисел , общее количество которых равно N , и во-вторых, указана связь между моделируемыми арифметическими операциями над целыми числами и их модельными аналогами (кольцевыми операциями), которая выражается следующим утверждением: для любых $x, y \in Z_N$ имеют место равенства

$$x + y = |x + y|_N \,,\ x \cdot y = |x \cdot y|_N$$

тогда и только тогда, когда результаты соответствующих арифметических операций $+,\bullet$ над вычетами, как целыми числами, не выходят за «диапазон» $Z_N = \{x \in Z \mid 0 \le x < N\}$. Последнее утверждение назовём условием согласования. Таким образом, любая компьютерная арифметика на регистровом уровне должна сопровождаться «мониторингом» условия согласования и, если это условие нарушается, то система слежения должна соответствующим образом реагировать. Реакция зависит от режима вычисления: в целых числах или с дробями, с фиксированной или с плавающей точкой, в режиме обнаружения и исправления ошибок и т.п. Здесь умышленно использовано слово «мониторинг», чтобы подчеркнуть фундаментальность этой операции в сигнатуре любой компьютерной модели чисел, часто опускаемой «по умолчанию».

Таким образом, компьютерная модель целых чисел – это арифме-

тика кольца вычетов по некоторому модулю N, плюс мониторинг условия согласования. Сам мониторинг реализуется опять же посредством аксиомы Архимеда. Например, если контролируется выход результата аддитивной операции за диапазон, то этот контроль осуществляется функцией:

$$\eta = \left\lfloor \frac{x+y}{N} \right\rfloor = \begin{cases} 1 & \text{åñëè } x+y \notin Z_N \\ 0 & \text{åñëè } x+y \in Z_N \end{cases}$$

Легко видеть, что указанный контроль не может быть осуществлен посредством модульных операций кольца вычетов Z_N, так как:

$$\forall x, y \in Z : \left\lfloor \frac{|x+y|_N}{N} \right\rfloor = 0$$

Иначе говоря, контроль аддитивного переполнения является немодульной операцией.

Отсюда, в частности, следует, что поскольку аксиома Архимеда является составной частью семантики вещественных чисел (т.е. всех аксиоматических систем **R**), то любая компьютерная модель чисел неизбежно является архимедовой и любые реальные вычисления с вещественными числами являются архимедовыми.

Многообразие компьютерных арифметик зависит от структуры модуля N и от способа масштабирования. Синтаксис компьютерной арифметики, т.е. её система правил, зависит только от выбора модуля N. Вообще говоря, следует отметить, что перечисленной выше спецификой компьютерной арифметики целых чисел обладают не только вычислительные структуры с числовым модулем N, но и с модулями более сложной природы (кольцо главных идеалов) [18]. Возвращаясь к числовому модулю N, отметим, что не трудно установить связь между Z_N и Z_N^-:

$$\forall x \in Z \qquad \left|x\right|_N^- = \left|x + \left\lfloor \frac{N}{2} \right\rfloor\right|_N - \left\lfloor \frac{N}{2} \right\rfloor.$$

Вывод. Всякая компьютерная арифметика целых чисел на регистровом уровне (будем говорить кратко «регистровая арифметика») является модулярной, т.е. условно разбивается на модульную арифметику по модулю N регистра и систему мониторинга за переполнением результата вычислений за динамический диапазон регистра.

Далее, если всё сказанное выше характеризует смысл (семантику) регистровой арифметики, то вопросы кодирования её элементов относятся, так сказать, к синтаксису регистровой арифметики. Здесь уже существенное значение приобретает природа модуля N кольца вычетов [19].

При $N = 2^n$ имеет место двоичный код; при $N = 3^n$ – троичный, при $N = p_1, \dots p_n$, где основания попарно взаимно-просты – модулярный код (код в остатках). Кодирование вычетов называется позиционным, если лексико-графический порядок этого кода совпадает с естественным порядком вычетов, как целых чисел. В противном случае код называется непозиционным. Позиционному коду присуща простая организация мониторинга переполнения – сравнение чисел по величине, и сложнее – реализация арифметических операций над ними. Вычеты по составному модулю $N = p_1, p_2, \dots p_n$ могут кодироваться двояко – позиционным (полиадический код), непозиционным (кодом в остатках); при этом непозиционный код эффективен в реализации операций кольца вычетов (модульные операции) и мало эффективен в процессах мониторинга переполнения (немодульные операции), а полиадический – эффективен в процессах мониторинга и не эффективен в арифметических операциях кольца вычетов. Интерес к арифметике в остатках, которую только и принято называть модулярной арифметикой, вызван двумя обстоятельствами, присущими коду в остатках – параллелизмом и свойством арифметической самокоррекции.

К настоящему времени с точки зрения современной технологии микроэлектроники и особенно технологии проектирования наибо-

лее «обкатанной» является регистровая арифметика по модулю $N = 2^n$, т.е. двоичная арифметика. Немалый интерес представляет регистровая арифметика по модулю $N = 3^n$ [20]. Преимущественный интерес этих арифметик при решении массовых задач обусловлен, главным образом, гибкостью мониторинга условия согласования. При решении же некоторых типов индивидуальных задач (в специальных приложениях), когда возникает необходимость в получении на элементах «текущей технологии» такого сверхкачества на регистровом уровне, как повышение производительности средствами распределения и распараллеливания вычислительных нагрузок, повышение эксплуатационной надёжности средствами сочетания свойств арифметичной самокоррекции, реконфигурации и резервирования, преимущественный интерес приобретает модулярная арифметика. В связи с этим в разработках модулярной арифметики возникли две тенденции:

- привязка систем автоматизации проектирования, разработанной для двоичной арифметики к задачам модулярной арифметики, посредством адаптации модулей модулярной арифметики к удобному двоичному представлению [22], а также посредством сближения и комбинирования двоичной и модулярной арифметики [23],
- посредством разработки дополнительного программного обеспечения оптимального выбора модулей, средствами целочисленного программирования, которые позволяют осуществить эффективную реализацию как модульных, так и немодульных операций.

Эти тенденции возникли ещё при проектировании ЭВМ 5Э53. Ярким примером тому служит программа, разработанная Я.Н. Кобринским, который решил задачу оптимального выбора модулей модулярной арифметики, удовлетворяющих техническим ограничениям. Эти ограничения описаны в работе [19] и связаны они со специфическим видом модуля динамического диапазона и модуля избыточного диапазона, что обеспечивало возможность совмещения операции округления с операцией обнаружения и исправления ошибок, причём операция округления осуществлялась на основе алгоритма встречного расширения с использованием неточного ранга, позволяющего распараллелить операции встречного расширения. Таким образом, достигалась эффективная реализация опера-

ции округления, совмещённая с операцией обнаружения и исправления ошибок, при вычислениях с дробями в режиме фиксированной запятой.

Как показал опыт проектирования 5Э53, успешная разработка модулярного процессора возможна только при слаженной, взаимодополняющей и взаимоувязанной работе большого коллектива технологов, схемотехников, программистов, математиков и при гибком централизованном управлении демократичным и энергичным архитектором проекта, таким, каким был Д.И. Юдицкий. Это объясняется тем, что в области проектирования модулярных процессоров до сих пор отсутствуют законченные рецепты и стандарты проектирования. Современные технологии микроэлектроники и технологии проектирования при соответствующей их адаптации могут использоваться с успехом при проектировании высокоэффективных модулярных процессоров, что собственно и составляет основу перспективного развития модулярной арифметики. Современная теория чисел способна за короткий срок дать ответы на все вопросы разработчиков, усилия которых направлены на сближение достоинств модулярной и двоичной арифметик и создания вычислительных структур и вычислительных сред двоично-модулярного типа на новых технологиях [21-27].

Модулярная арифметика заняла достойное место в теории вычислений, в компьютерной алгебре [30-34]. Как показывает обзор цитированных работ, перспективу модулярной арифметики представляют:

- Теоретические исследования вопросов ускорения и повышения надежности вычислений, связанные с выбором оснований арифметики в классе задач модулярной алгоритмики;
- Создание вычислительных структур позиционно-модулярного типа, направленных на сближение достоинств модулярной и позиционной арифметик с целью комплексного достижения сверхпроизводительности и сверхнадежности вычислений на современных вычислительных средствах, спроектированных на элементной базе, предельно реализующей возможности технологии:
- Разработка интегрированных режимов варьирования масштабами вычислений;
- Исследования и расширение классов задач, допускающих эффек-

тивную реализацию на гибридных модулярно-позиционных вычислительных средах.

Литература

1. Valach M., Vznik kodu a čiselene soustavy zbyitkovychtžid, Stroje na Zpracovani Informaci, Sbornik III, Praha, 1955, 211-255.
2. Valach M., Převodčiselze soustavy sbyitkovich třid do polyadicke soustavy změnon měřitka periody, Stroje na Zprancovani Informaci, Sbornik VI, Praha, 1956, 53-64.
3. Svoboda A., Valach M, Operatorove obovody, Stroje na Zpracovani Informaci, Sbornik III, Praha, 1955, 247-295.
4. Svoboda A., Rational numerical system of residual classes, Stroje na Zpracovani Informaci, Sbornik V, Praha, 1957, 9-37.
5. Svoboda A., The numerical system of residual classes in mathematical machines, Proc. Congreso Informational de Automatica, Madrid, October, 1958. 11-12.
6. Svoboda A., The numerical system of residual classes in mathematical machines, Information Progressing (Processings of UNESCO Conference, June, 1959) 1960, 419-422.
7. Aiken H.H. Theory of switching, Computation Lab., Harvard Univ., Cambrige., Mass. Rep N BL-23, June, 1959.
8. Aiken H.H., Semon W., Advanced Digital computer logic, WADC TR-59-472, July, 1959.
9. Garner H.L. Error checking and the structure of binary addition, Ph. D. Diss., Univ. Mitchigan, Ann Arbor, Mich., 1958.
10. Garner H.L., The residue number system, JRE Trans. On Electronic Computers, EC-8 (1959), June, 140-147.
11. Cheney P.W., A digital correlator based on the residue number system JRE Trans. on Electronic Computers, EC-10 (1961), March, 63-70.
12. Szabo N. Sign detection in non redundant residue systems, JRE Trans. on Electronic Computers, EC-14 (1962), August, 494-501.
13. Shapiro H.S. Some remarks on Modular Arithmetics and Parallel Computation, Mathematics of computation (MTAC) v.16, no 78, 1962, 218-222.
14. Андрианов Е.С., О методе определения знака в системе остаточных классов. – Вопросы радиоэлектроники, 1964, серия XIII, вып. 2.
15. Андрианов Е.С. О некоторых методах организации округления в системе остаточных классов. – Вопросы радиоэлектроники, 1964, серия XIII, вып.8.
16. Кнут Д. Искусство программирования. Т.2. Получисленные алгоритмы. Вильямс: М-СП/б-Киев. 2000.
17. Карцев М.А., Арифметика цифровых машин. М., 1969.
18. Амербаев В.М., О построении систем счисления в остаточных классах в кольце главных идеалов. – Труды М.О.,1967, №75, с.61-81.

19. Амербаев В.М., Теоретические основы машинной арифметики, Алма-Ата, издат. «Наука», 1976, 323 стр.
20. Виноградов И.М., Основы Теории чисел, М.: 1972.
21. Брусенцов Н.П., Вычислительная машина «Сетунь» Московского государственного университета – В кн.: Новые разработки в области вычислительной математики и вычислительной техники. – Киев, 1960, с. 226-234.
22. Стемпковский А.Л., Корнилов А.И., Семёнов М.Ю., Особенности реализации устройств цифровой обработки сигналов в интегральном исполнении с применением модулярной арифметики // Информационные технологии – 2004г. с.2-8.
23. Евстигнеев В.Г. Недвоичная машинная арифметика и специализированные процессоры – М.: МИФИ СЕРВИС и АО «ИНСОФТ», 1992.
24. Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А., Модулярные параллельные вычислительные структуры нейропроцессорных систем – М.: Физматлит, 2003 – 288с.
25. Инютин С.А. Модулярные вычисления в сверхбольших компьютерных диапазонах , Известия вузов. Электроника – 2001 – N6, с. 34-39.
26. Коляда А.А. Пак И.Т., Модулярные структуры конвейерной обработки цифровой информации. Мн.: Университетское, 1992, 256с.
27. Малашевич Б.М., Малашевич Д.Б. Отечественные модулярные и троичные ЭВМ. // Труды Юбилейной Международной научно-конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, стр 101-148.
28. Л. Эйлер. Руководство к арифметике. Ч. 1., СПб, 1740.
29. Малашевич Б.М., Малашевич Д.Б. Модулярная арифметика – взгляд изнутри. // Труды Юбилейной Международной научно-конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, стр. 47-100.
30. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М. Мир, 1979.
31. Акпитас А. Основы компьютерной алгебры с приложениями. – М. Мир, 1994.
32. Лидл Р., Пильц Г. Прикладная абстрактная алгебра. – Екатеринбург, изд. Уральского ун-та, 1996.
33. Глухов М.М., Елизаров Е.П., Нечаев А.А. Алгебра, т. 1, 2. – М. Гелиос АРВ, 2003.
34. Ноден П., Ките К. Алгебраическая алгоритмика. – М. Мир, 1999.
35. Корман Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М. МЦНМО, 2002.

УДК 621.29

Коляда А.А., Чернявский А.Ф.

Модулярные вычислительные структуры: Вчера, сегодня, завтра

(НИИ Прикладных физических проблем им. А.Н. Севченко БГУ, г.Минск)

В статье дан аналитический обзор основных этапов, особенностей и путей развития модулярных вычислительных структур (МВС). Выделяются два главных периода в исследованиях по теоретическим основам модулярной арифметики (МА) – периоды становления (1955–1970гг.) и оптимизации (1970–1990гг.). При этом раскрывается сущность разработанных вариантов (полиадического и рангового) классической МА и версий МА (минимально избыточной и позиционно-модулярной), возникших в процессе оптимизационных исследований. Даются оценки вкладов в решение фундаментальных проблем МА отечественных и зарубежных научных школ. Проблематика приложений МВС освещается с позиций критерия максимума пределов действия режима модульных вычислений, преимущественно, на примерах конкретных применений минимально избыточных МВС.

In article the analytical review of the basic stages, features and ways of development modular computing structures (MCS) is given. Two main periods in researches on theoretical bases modular arithmetics (MA) – the periods of becoming (1955–1970) and optimization

(1970–1990) are allocated. Thus the essence of the developed variants (polyadic and rank) classical MA and versions MA (minimally redundant and positionally-modular), arisen in process optimization researches is opened. Ratings of contributions to the decision of fundamental problems MA of domestic and foreign scientific schools are given. The problematic of applications MCS is covered from positions of criterion of a maximum limits actions of a mode of modular calculations, mainly, by the example of concrete applications minimally redundant MCS.

Как известно, в процессе развития компьютерных наук и их многочисленных приложений фундаментальную роль играют параллельные вычислительные структуры (ВС); и особое место среди таких структур занимают модулярные ВС (МВС). Обладая максимальным уровнем внутреннего параллелизма, МВС представляют собой уникальное средство декомпозиции вычислительных процессов на независимые друг от друга субпроцессы, определённые на математических моделях с элементами которые изменяются в диапазонах небольшой (в сравнении с исходным диапазоном) мощности [1–6].

Естественный параллелизм МВС обеспечивает им ряд фундаментальных преимуществ над позиционными структурами:

- выполнение всех модульных операций – сложения, вычитания и умножения без контроля переполнения за одно и то же время;
- табличную природу компьютерных алгоритмов модулярной арифметики (МА);
- простоту конвейеризации вычислений на уровне цифр модулярных кодов (МК) [4];
- широкие возможности модулярных систем счисления (МСС) для проведения численно-аналитических расчётов с повышенной и со сверхвысокой точностью (в режиме модульных вычислений (РМВ)) [7–9];
- исключительно высокую производительность при обработке данных, представляющих собой элементы более сложных, чем вещественные диапазоны, математических моделей таких, как множества комплексных чисел, кватернионов, полиномов и тому подобных [10–17];
- эффективность модулярных кодовых конструкций с обнаружением и исправлением ошибок;
- способность МВС к гибкой реконфигурации, позволяющей адаптировать (перестраивать) архитектуру вычислительных систем

адекватно классам решаемых задач, динамике внутренней логики выполняемых процессов, внешним условиям, сбойным ситуациям и прочим факторам;

- идеальную приспособленность компьютерных процедур МА к реализациям на СБИС, в том числе на нетрадиционных перспективных СБИС таких, в частности, как оптоэлектронные СБИС, СБИС на многозначной логике и других [18–21].

Благодаря как отмеченным, так и некоторым другим важным достоинствам, МВС идеально согласуются с концепциями передовых компьютерных идеологий, в том числе положенных в основу супер-ЭВМ, мультипроцессорных, многомашинных, транспьютерных, нейронносетевых и тому подобных систем параллельной обработки информации [22–27]. Именно этим обстоятельством и объясняется постоянный живой интерес специалистов, как теоретиков так и практиков, к МА на протяжении всего 50-летнего периода развития. В настоящее время со всей определённостью можно говорить как о вполне сложившихся модулярных направлениях в информатике, цифровой обработке сигналов (ЦОС), криптологии, других областях.

Анализ направлений теоретических и прикладных исследований по МВС показывает, что они, фактически, всегда были подчинены одной и той же стратегической цели – реализации в рамках текущего состояния вычислительной техники фундаментальных преимуществ МА как можно в более полной мере. При этом характер осуществлявшихся разработок и реальные успехи технологий модулярной обработки информации (ТМОИ) на конкретных исторических этапах, естественно, в первую очередь определялись уровнем элементной базы.

Начальный период развития МА – период становления (1955–1980), характеризовался интенсивными исследованиями в области создания теоретических основ компьютерной арифметики МСС. Здесь следует отметить фундаментальные труды как зарубежных специалистов (Szabo, Tanaka, Garner, Jullien), так и советских учёных (И.Я. Акушкий, В.М. Амербаев, И.Т. Пак и др.). Вклад в теорию МВС специалистов московско-алма-атинской школы, родоначальниками которой являются И.Я. Акушский и В.М. Амербаев, представляется особенно значимым. Разработанные ими методы выполнения немодульных операций, базирующиеся на так называемых ранговых и ядерных интегральных характеристик МК

(ИХМК), сыграли основополагающую роль в последующих исследованиях по оптимизации МВС. Что же касается теоретических разработок, выполненных В.М. Амербаевым и И.Т.Паком в области создания версий МА, которые предназначены для быстрых параллельных вычислений в комплексных диапазонах, то они остаются востребованными по сей день. Лишь в середине 80-х годов XX века по мере совершенствования интегральных технологий указанные комплексные варианты МА, главным образом в виде так называемой квадратичной версии (G.A.Jullien, M.A. Bayoumi, M.A.Soderstrand, F.J.Taylor, W.K.Jenkins, A.Skavantzos, G.Alia, E.Martinelli) стали широко применяться для реализации процедур ЦОС и, в частности, ДПФ. Исследования В.М. Амербаева, относящиеся к проблемам модулярных вычислений на произвольным математических моделях – множествах гиперкомплексных чисел, полиномов и тому подобных, по существу, составили основу интенсивно развивающегося в течение последних 25 лет нового направления – модулярной компьютерной алгебры. Кроме комплексных и квадратичных данное направление охватывает также полиномиальные и другие МВС [28,29].

В зависимости от типа базовых ИХМК наиболее употребительные подходы к построению МА можно разделить на два принципиально различных класса. Первый характеризуется применением в качестве ИХМК цифр полиадического кода, а второй – использованием ранговой характеристики [1–4]. В МСС с модулями $m_1, m_2,\ldots, m_k$ нормированный ранг $\rho_k(X)$ целого числа X определяется равенством

$$|X|_{M_k} = \sum_{i=1}^{k} M_{i,k} \left| M_{i,k}^{-1} \chi_i \right|_{m_i} - M_k \rho_k(X) , \qquad (1)$$

где $M_k = \prod_{j=1}^{k} m_j$; $M_{i,k} = M_k / m_i$; через $|a|_m$ обозначается элемент множества $\boldsymbol{Z}_m = \{0,1,...,m-1\}$, сравнимый с величиной a (в общем случае рациональной) по натуральному модулю m; $\chi_i = |X|_{m_i}$. При попарно простых $m_1, m_2,\ldots, m_k$ в рамках соотношения (1) устанавливается изоморфное отображение между вычетами

X кольца $\boldsymbol{Z}_{M_k}$ (диапазона МСС) и их МК (χ_1, χ_2,…, χ_k)$\in \boldsymbol{Z}_{m_1} \times \boldsymbol{Z}_{m_2} \times ... \times \boldsymbol{Z}_{m_k}$. При этом вычисление $\rho_k(X)$ даёт возможность реализовать указанное отображение.

Известно, что расчёт ИХМК, как коэффициентов полиодического представления, так и ранга $\rho_k(X)$ числа $X \in \boldsymbol{Z}_{M_k}$ по его МК (χ_1, χ_2,…, χ_k) фактически сводится к суммированию *k*-1 различных наборов вычетов с формированием количеств выходов сумм за пределы колец $\boldsymbol{Z}_{m_2}, \boldsymbol{Z}_{m_3}, ..., \boldsymbol{Z}_{m_k}$. При максимальном распараллеливании вычислений соответствующая компьютерная процедура выполняется за $\lfloor \log_2 k \rfloor + 1$ модульный такт. Поскольку сложность и быстродействие того или иного варианта МА находятся в прямой зависимости от одноименных показателей базовой процедуры формирования ИХМК, то одним из возможных путей совершенствования МВС является применение альтернативных, более эффективных по отношению к классическим, ИХМК. Период развития ТМОИ с 1970 по 1990 годы во многом можно характеризовать как оптимизационный. Это относится не только к теоретическим исследованиям, но и к конкретным приложениям МА.

Ещё в конце 60-х годов XX века в целях упрощения алгоритмов немодульных операций некоторыми исследователями (Sasaki, Rao) была предложена модулярная кодовая конструкция, которая на ряду с остатками по модулям явно включала и ИХМК типа ранга. Это позволяет уменьшить сложность определения ИХМК суммы, разности и произведения чисел. При этом, однако, ввиду неоднородности кода затрудняется помодульная организация вычислений (РМВ). К тому же авторами выдвинутой идеи не была решена проблема ограничения и регуляризации рабочего диапазона. Как отмеченные, так и ряд других недостатков систем счисления Сасаки удалось устранить в рамках так называемого минимально избыточного модулярного кодирования [4].

Минимально избыточная МСС (МИМСС) в качестве рабочего диапазона использует множество $\boldsymbol{D} = \{-M, -M+1, ..., M-1\}$, где $M = m_0 M_{k-1}$; m_0 - вспомогательный модуль, удовлетворяющий условиям $m_0 \geq \rho = \max_X \{\rho_{k-1}(X)\}$ и $m_k \geq 2m_0 + \rho$. Произвольный

элемент X диапазона **D** восстанавливается по своему минимально избыточному МК ($\chi_1, \chi_2, \ldots, \chi_k$) согласно формуле

$$X = \sum_{i=1}^{k-1} M_{i,\,k-1} \left| M_{i,\,k-1}^{-1} \; \chi_i \right|_{m_i} + M_{k-1} I(X), \quad (2)$$

где I(X) – целочисленная величина, названная интервальным индексом (ИИ) числа X.

В отличие от ранга $\rho_k(X)$ вычисление ИИ $I(X)$ требует сложения по модулю m_k только k вычетов. Благодаря данному обстоятельству на базе интервально-модулярной формы (2) построена так называемая минимально избыточная МА (МИМА), которая существенно проще классических аналогов. Это преимущество достигается как на вещественных диапазонах, так и на более сложных математических моделях [13,14,16,17].

К проблематике оптимизации МВС относятся также разработки по созданию версий компьютерной арифметики, адаптированных к существующей элементной базе, которая ориентирована на вычисления в позиционных системах счисления (ПСС). Начиная с 80-х годов XX века в приложениях МА по ЦОС и криптографии широко применяются, например, МВС, синтезированные для специальных наборов модулей. Наибольшее распространение получили, в частности МСС с основаниями 2^n-1, 2^n, 2^n+1 (n – натуральное число).

В русле оптимизационного направления находятся также исследования, нацеленные на реализацию фундаментальных преимуществ МА и арифметики ПСС в рамках гибридных ВС каскадного типа. Речь идёт о так называемых позиционно-модулярных ВС (ПМВС), которые порождаются композицией позиционного и модулярного кодирований [30]. В рамках определяющего отображения позиционно-модулярной системы счисления (ПМСС) цифры внешнего (позиционного) кода представляются в МСС. За счёт применения на внутреннем уровне МК результирующая конструкция позволяет уменьшить сложность формирования аддитивных и мультипликативных переносов в разрядах внешнего уровня, повысить скорость выполнения арифметических операций, расширить возможности в части проведения вычислений с повышенной точностью, унификации процессорных блоков, контроля ошибок, оптимизировать ряд

других показателей.

Впервые для построения вычислительной машины ПМВС была применена в середине 60-х годов прошлого столетия группой учёных МИЭТ (Зеленоград), которую возглавляли И.Я. Акушский и Д.И. Юдицкий. В базовой ПМСС на внешнем уровне использовалась ПСС с основанием p=10, а на внутреннем – МСС с модулями m_1=2 и m_2=5. По сравнению с ЭВМ, функционирующей в двоично-десячином коде, реализованный проект обеспечил не только экономию оборудования, но и увеличение скорости выполнения арифметических операций, особенно умножения. Однако по мере развития интегральных технологий полученное преимущество фактически исчезло. С другой стороны, благодаря крупным успехам, достигнутым на протяжении последних 15–20 лет в области производства БИС, вполне естественным и логичным стал следующий важный шаг на пути создания качественно новой позиционно-модулярной вычислительной технологии, характеризующийся применением на внешнем уровне ПСС с большими основаниями: от 2^8 до 2^{12} и более. Соответственно это потребовало использования на внутреннем уровне МСС с числом модулей от 3 до 6 с разрядностью 3,4 бита. Теоретические и конкретные разработки по обозначенному направлению плодотворно проводились в течение 1980–1990 годов как за рубежом, так и в СССР, главным образом, в МИИГА и НИИ точного электронного прибора строения под руководством В.Г. Евстигнеева. Исследования и результаты их опытной апробации позволяют заключить, что каскадные ПМВС, занимая промежуточное положение между позиционными и модулярными ВС, представляют собой весьма привлекательную альтернативу по отношению к каждому из этих классов структур. Проведенные недавно в НИИ прикладных физических проблем БГУ (Минск) исследования по оптимизации ПМВС показали, что если на внутреннем каскаде вместо МСС использовать МИМСС, то эффективность данных ВС существенно повышается [8,9].

Говоря о конкретных приложениях ТМОИ, прежде всего, отметим, что подлежащие реализации целевые функции обычно сводятся к вычислительной модели, которая состоит из набора модульных сегментов выполняемых в РМВ. При этом диапазон применяемой МСС, естественно, должен включать конечные результаты счёта на всех сегментах. Корректность РМВ обеспечивается с помощью операций масштабирования. Описанная организация модулярных

вычислений позволяет использовать МА с минимальным набором немодульных процедур, включающим лишь кодовые преобразования с масштабированием. Для МИМСС в настоящее время разработаны эффективные процедуры требуемого класса [4,5,31,32]. Поэтому МИМА-реализации вычислительных процессов, укладывающиеся в рассмотренную схему, оказываются существенно проще неизбыточных аналогов.

В течение периода с 1980 по 1990 годы в НИИ ПФП (Минск) на базе МИМА разработано целое семейство периферийных процессоров (ПП), предназначенных для реализации в составе ПЭВМ ряда трудоёмких процедур ЦОС. Приведём краткую информацию о некоторых из выполненных разработок.

1. На основе алгоритма Винограда создан ПП для выполнения ДПФ, объёмы которых являются произведениями натуральных степеней попарно простых чисел из множества {2,3,…,9} [33–35]. Процессор выполнен в виде платы и имеет следующие характеристики:
   - вычисления проводятся в режиме с блочной плавающей запятой;
   - вещественные и мнимые части отсчётов входных сигналов представляются 12-битовыми, а выходных – 16-битовыми дополнительными двоичными кодами;
   - абсолютная погрешность вычислений на стадиях быстрой процедуры ДПФ не превышает 2^{-16};
   - результирующая абсолютная погрешность ограничена порогом 2^{-12};
   - тактовая частота составляет 10МГц;
   - выполнение одного 1260-точечного ДПФ занимает не более 2мс;
   - суммарное количество используемых интегральных микросхем – 91.
2. Разработан ПП для адаптивной КИХ-фильтрации, который выполнен в виде платы [36,37]. Процессор имеет характеристики:
   - непосредственно в ПП выполняются процедуры КИХ- фильтрации над вещественными сигналами, длины которых не превышают 16 отсчётов;
   - базовый набор импульсных характеристик реализуемых фильтров может включать до 8 вариантов;

- отсчёт входных и выходных сигналов представляются 16-битовыми двоичными кодами;
- абсолютная погрешность ограничена сверху порогом 2^{-16};
- производительность ПП составляет 32млн арифметических операций в секунду;
- пропускная способность ПП адекватна скорости обмена данными ПЭВМ с периферийными устройствами;
- количество используемых микросхем – 78.

3. На основе технологии параллельных вычислений в пространствах ортогональных проекций создан ПП [38–40] для системы, предназначенной для оперативного управления космическими средствами наблюдения. По сравнению с традиционно применяемыми компьютерно-арифметическими, алгоритмическими и программно-аппаратными технологиями предложенный процессор обеспечивает более чем 100-кратное повышение производительности при 50-кратном сжатии входных сигналов.

Анализ публикаций по приложениям МА показывает, что применение высокоскоростных систем модулярной обработки информации (СМОИ) на основе параллельно-конвейерных БИС- и СБИС-архитектур во многих областях, безусловно, оказывается оправданным. Вместе с тем, в настоящее время интенсивно ведутся многообещающие исследования и конкретные проектные разработки (K. Pltssmann, J.Wollert, С.А. Инютин и др.), которые нацелены на реализацию фундаментальных преимуществ МА на вычислительных системах позиционного типа. Речь, в частности, идёт о внедрении многомашинной и особенно мультипроцессорной ТМОИ, причём преимущественно на программном уровне [7,23,24]. Обозначенный подход позволяет синтезировать принципиально новые варианты МА, которые характеризуются несоизмеримо большей свободой выбора значений модулей МСС и объёмов рабочих таблиц, чем в случае применения чисто аппаратного подхода. Это открывает исключительно широкие возможности для расширения пределов действия РМВ и, что особенно важно, без использования дорогостоящих специализированных средств. Следует особо подчеркнуть, что по мере расширения сферы распространения ПЭВМ мультипроцессорного типа (а уже сейчас этот процесс резко интенсифицируется) потребительский рынок программного продукта, созданного на основе указанной ТМОИ, также будет расти.

Литература

1. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. – М.: Сов. Радио, 1968. – 440 с.
2. Торгашов В.А. Система остаточных классов и надежность ЦВМ. – М.: Сов. Радио, 1973. – 118 с.
3. Амербаев В.М. Теоретические основы машинной арифметики. –Алма-Ата: Наука, 1976. – 320 с.
4. Коляда А.А., Пак И.Т. Модулярные структуры конвейерной обработки цифровой информации. Мн.: Университетское, 1992. 256 с.
5. Чернявский А.Ф., Данилевич В.В., Коляда А.А., Селянинов М.Ю. Высокоскоростные методы и системы цифровой обработки информации. Мн.: Белгосуниверситет, 1996. 376 с.
6. Ирхин В.П. Проектирование непозиционных специализированных процессоров. Воронеж: Изд. Воронеж. ун-та, 1999. 136 с.
7. Инютин С.А. Модулярные вычисления в сверхбольших компьютерных диапазонах // Известия вузов. Электроника. 2001. № 6. с. 34–39.
8. Рекурсивные минимально избыточные интервально-модулярные системы счисления/ А.Ф. Чернявский, А.А. Евдокимов, А.А. Коляда, В.В. Ревинский// Доклады НАН Беларуси. – 2004. –Т.48, №1. – С.10–14.
9. Коляда А.А., Кравцов В.К., Чернявский А.Ф. Основы минимально избыточной интервально-модулярной арифметики с рекурсивной кодовой структурой// Информатика. – 2004. – №1. – С. 112–120.
10. Амербаев В.М., Пак И.Т. Параллельные вычисления в комплексной плоскости. – Алма-Ата: Наука, 1984. – 182 с.
11. Silverman R.D. Parallel polynomial arithmetic over finite ring // J. Parallel. and Distribut. Comput. – 1990. – Vol. 10, N 3. – P. 265–270.
12. Skavantzos A., Taylor F.J. On the polynomial residue number system // IEEE Trans. Signal Process. – 1991. – Vol. 39, N 2. – P. 376–382.
13. Коляда А.А., Ревинский В.В., Селянинов М.Ю., Чернявский А.Ф. Применение минимально избыточного модулярного кодирования для быстрого умножения комплексных чисел в системах цифровой обработки сигналов // Весці Акадэміі навук Беларусі. Сер. фіз.-тэхн. Навук. – 1996. – № 1.
14. Теоретические основы минимально избыточных квадратичных модулярных систем счисления / А.Ф. Чернявский, А.А. Коляда, В.К. Кравцов и др. // Доклады НАН Беларуси. 1998. Т.42. №1. С. 5 – 12.
15. Alia G., Martinelli E. Optimal VLSI complexity design for high speed pipeline FFT using RNS // Comput. and Elec. Eng. 1998. Vol. 24, N3. P.167–182.
16. Коляда А.А., Селянинов М.Ю., Чернявский А.Ф. Минимально избыточные полиномиально-скалярные модулярные системы счисления// Актуальные проблемы социально-гуманитарных и естественных наук. Тез. научн. конф., посвящённой 70-ю Белгосуниверситета. – Т.1. – Минск, 1996. – С. 181–183.

17. Минимально избыточные полиномиально-скалярные модулярные системы счисления/ А.А. Коляда, В.В. Ревинский, А.Ф. Чернявский// Весці НАН Беларусі. Сер. фіз.-мат. навук. – 1998. – №3. – С.103–107.
18. Papachristou C.A. Associative table look up processing for multioperand residue arithmetic // J. Assoc. Comput. 1987. Vol. 34, N 2. P. 376 – 396.
19. Mirsalehi M.M., Shamir J., Caulfield N.J. Residue arithmetic processing utilizing optical Fredkin date arrays // Applied optic. 1987. Vol. 26, N 19. P. 3940 – 3946.
20. Optical processing with residue LED/LD lookup tables/ A.P. Goutzoulis, E.C. Malarkey, D.K. Davies et al. // Applied optic. 1988. Vol. 27, N9. P. 1674 – 1681.
21. Волоконная оптика в измерительной и вычислительной технике / А.Н. Казангапов и др. Алма-Ата: Наука, 1989, 245с.
22. Компьютеры на СБИС / Т.Мотоока, Х.Хорикоси, М. Сакаути и др. – М.: Мир, 1988. – Кн. 2.
23. Plessmann K. A parallel highly modular object-oriented computer architecture // 10 юбил. Международн. Симп. по пробл. модулярных инф.-выч. сист. и сетей. – Санкт-Петербург, Россия, 13–18 сент., 1993. – Пленар. докл. – М., 1996. – С.97–109.
24. A modular multi-PC system for real-time applications / K. Plessmann, J. Wollert and others // там же. – С. 110–119.
25. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А. – М.: Физматлит, 2003. – 288 с.
26. Нейрокомпьютеры в остаточных классах / Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н. Учебное пособие для вузов – М.: Радиотехника, 2003. – 272 с.
27. Нейрокомпьютеры в системах обработки сигналов. Коллективная монография./ Н.И.Червяков, Л.Б.Копыткова, Е.Н.Непретимова, П.А. А.В.Шапошников и др. Под редакцией Гуляева Ю. Лушкина А.И. – М: Радиотехника, 2003. – 224 с.
28. Теоретические основы модулярных вычислительных структур на конечных математических моделях / А.А. Коляда, В.В. Ревинский, М.Ю. Селянинов и др. // Современные вопросы оптики, радиационного материаловедения, информатики, радиофизики и электроники: Сборник научных трудов НИИ прикладных физических проблем им. А.Н.Севченко. – Мн.: Белгосуниверситет, 1996. – Ч. 2. – С. 4 – 9.
29. Селянинов М.Ю. Теоретические основы модулярной кодификации алгебраических систем // Весці НАН Беларусі. Сер. фіз.-мат. навук. 2002. № 1. С. 114119.
30. Евстигнеев В.Г. Недвоичная машинная арифметика и специализированные процессоры. – М.: МИФИ СЕРВИС и АО «ИНСОФТ», 1992.
31. Теоретические основы мультипликативных процедур для минимально избыточных модулярных систем счисления / А.Ф. Чернявский, А.М.

Аксенов, А.А. Коляда, М.Ю. Селянинов // Доклады АН Беларуси. 1995. Т.39. №6. С. 5 – 10.
32. Методы масштабирования минимально избыточной модулярной арифметики / А.Ф.Чернявский, А.А.Коляда, В.В.Ревинский, М.Ю.Селянинов, Е.В.Шабинская // Весці НАН Беларусі. Сер. фіз.-мат. навук. – 1998, № 4. С. 132 – 137.
33. А.с. №1732353 СССР. Устройство для вычисления ДПФ / Л.Н. Василевич, И.И. Гунько, А.А. Коляда // Откр. Изобр. – 1992. – №17.
34. Василевич Л.Н., Коляда А.А. Структура арифметических устройств модулярных процессоров БПФ конвейерного типа // Электронное моделирование. 1989, Т. 2, № 6. С. 15-20.
35. Модулярные принципы построения процессоров для дискретного преобразования Фурье / Л.Н. Василевич, А.А. Коляда, М.Ю. Селянинов, А.Ф. Чернявский // Весці НАН Беларусі. Сер. фіз.- тэхн. навук. – 2001. – № 4. – С.108–117.
36. Василевич Л.Н., Коляда А.А., Ревинский В.В. Высокоскоростная модулярная реализация адаптивных цифровых фильтров с конечной импульсной характеристикой // Весці АН Беларусі. Сер. фіз.-мат. навук. 1997. №1. С. 126 – 131.
37. Селянинов М.Ю. Минимально избыточная модулярная архитектура адаптивного КИХ-фильтра // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. 2002. № 2. С. 7988.
38. A. Kolyada, E. Otlivanchik, V. Revinsky, L. Vasilevitch // 5th Work-shop on DIP'94: Image Processing and Computer Optics. Samara, Russia, Aug. 22 – 26, 1994. Proc. SPIE. Vol. 2363. Washington, D.C., 1994. P. 147–151.
39. Синтез компьютерной процедуры и архитектуры высокопроизводительного процесса ортогональных проекций дискретных сигналов на базе минимально избыточных модулярных систем счисления / А.А.Коляда, А.М.Аксенов, Е.А.Отливанчик и др. // ММРО-7: Математические методы распознавания образов. – Пущино, Россия, 25–30 сентября 1995 г. – Тез. Докл. – Москва. – 1995. – С. 105–106.
40. Селянинов М.Ю. Применение численно-аналитической модулярной вычислительной технологии для выполнения аддитивных и мультипликативных операций над сигналами в пространствах ортогональных проекций // Доклады НАН Беларуси. – 2002. – Т. 46, № 2. – С. 62–66.

УДК. 681.3

В. Х. Хацкевич, Л. Хэскелл

ФУНКЦИОНАЛЬНАЯ ИЗБЫТОЧНОСТЬ В МОДУЛЯРНОЙ АРИФМЕТИКЕ И СОПРЯЖЕННЫЕ ЗАДАЧИ

Functional Redundancy in Modular Arithmetic and Related Problems

Vice President & President of
Computer Algorithm & Program Development Corp.
New York, USA

Предложены методы введения в структуру модулярных представлений чисел избыточности, предназначенной для нахождения алгоритмов обработки информации, декомпозиции и декодирования, ориентированных на построение эффективных систем программного обеспечения.

Proposed methods of redundancy introduction into structure of modular representation of numbers designed for construction of data processing, decomposition and decoding algorithms leading to creation of effective software solutions.

Система счисления в остаточных классах (СОК), адаптированная М. Валахом в 1955 году [1] для машинных вычислений, положила начало развитию современной модулярной арифметики. Первона-

чально модулярная арифметика была предназначена для повышения производительности электронных вычислительных машин (ЭВМ) первого поколения, построенных на технической базе электронно-вакуумной аппаратуры. Однако, ее не постигла участь широко применявшейся в то время, но ныне забытой т.н. «алгебры Айкена» – специального операционного исчисления для синтеза и анализа формализованных операторов электровакуумных приборов. Во многом благодаря трудам И. Я. Акушского и его учеников в бывшем Советском Союзе была создана и плодотворно работала школа модулярной арифметики и непозиционных арифметических кодов. Были обнаружены и исследованы корректирующие возможности непозиционных модулярных представлений чисел, т. н. «живучесть» и ряд других свойств структур модулярной арифметики, расширен круг ее применения [3, 5, 6, 7, 8, 9, 29, 31]. Это позволило модулярной арифметике и непозиционному кодированию пережить все очередные изменения технической базы вычислительной техники и при этом использовать новые технологические достижения. Параллельно аналогичные работы проводились в США [49, 55].

Поэтому представляется упрощенным сведение проблематики непозиционных вычислительных систем только к задачам построения специальных сегментов машинной арифметики. К проблемам, которыми занимается теория модулярной арифметики и к ее достижениям могут быть отнесены некоторые методы построения современных криптографических систем с открытыми ключами. В частности, известная система RSA [47, 48, 50, 51] и ее модификации в настоящее время являются самыми применяемыми криптографическими системами в Интернете и в локальных сетях. Достижения модулярной арифметики используются в задачах цифровой обработки сигналов, в целочисленном линейном программировании.

Специфическая декомпозиция цифровых данных при непозиционном кодировании, обеспечивая возможность независимой обработки образуемых компонент файла, вместе с тем приводит к потере некоторых мультипликативных характеристик кода, содержащих в явном виде информацию о числе в целом. Поэтому, одну из основных задач непозиционного кодирования составляет искусственное введение таких характеристик для непозиционных представлений чисел. Вычисление мультипликативных характеристик кодовых слов является самым существенным звеном алгоритмов выполне-

ния немодульных операций и основными операциями (кроме модульных) в непозиционных системах. К ним сводятся любые другие немодульные операции, и от времени, затрачиваемого на выполнение этих вычислений и методики их выполнения, в конечном счете, зависят производительность и надежность соответствующих устройств. Известные алгоритмы вычисления непозиционных характеристик кодовых слов не всегда отвечают требованиям, предъявляемым к набору технических средств при конструировании распределенных вычислительных структур. Некоторые методы определения мультипликативных характеристик требуют значительных временных и аппаратурных затрат. Другие – связанные с альтернативными соотношениями, допускают существования в цифровом диапазоне областей неопределенности, и требуют привлечения для ее раскрытия дополнительной информации.

Еще в 1962 г. Н. Сабо [2] показал, что удовлетворительное преодоление этих трудностей принципиально невозможно в неизбыточных непозиционных системах. Однако, длительное время исследования, направленные на построение избыточных непозиционных числовых представлений были ориентированы на нахождение корректирующих, а не на оптимизацию алгоритмических свойств непозиционных файлов. В настоящее время построение избыточных непозиционных представлений для расширения алгоритмических возможностей модулярной арифметики становится все более актуальным.

Найдены специальные решения [7, 8, 9, 10, 15, 33] системы сравнений

$$N \equiv \alpha_i (\mathrm{mod}\, p_i) \qquad (i = \overline{1,n})$$

вида

$$\begin{aligned}
&\lambda_1 \equiv \gamma_1 \alpha_1 \pmod{p_1 p_2 \cdots p_n} \\
&\lambda_2 \equiv \gamma_2 \alpha_2 - \lambda_1 a_{12} \pmod{p_2 p_3 \cdots p_n} \\
&\lambda_3 \equiv \gamma_3 \alpha_3 - \lambda_2 a_{23} \pmod{p_3 p_4 \cdots p_n} \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&\lambda_n \equiv \gamma_n \alpha_n - \sum_{i=1}^{n-1} \lambda_i a_{i(i+1)} \pmod{p_n}
\end{aligned} \qquad (1)$$

Для неортогональных базисных чисел:

$$\begin{aligned} B_1 &= (\gamma_{11},0,\ldots,0) \\ B_2 &= (\gamma_{21},\gamma_{22},0,\ldots,0) \\ &\ldots\ldots\ldots \\ B_n &= (\gamma_{n1},\gamma_{n2},\ldots,\gamma_{nn}) \end{aligned} \qquad (2)$$

Число N кодируется в виде последовательности $\lambda_1,\lambda_2,\ldots,\lambda_n$, определяемой соотношениями (1). Система остаточных классов (Residue Number System) соответствует частному случаю, когда числа (2) являются ортонормированными базисами и все $\lambda_i = \alpha_i$ $(i=\overline{1,n})$. На этой основе получено много модификаций непозиционных систем, а также систем занимающих по своим свойствам промежуточное положение между непозиционными и позиционными системами. Некоторые из них, в частности система с базисами вида $B_i = (0,\ldots,0,\gamma_{i(i-1)},\gamma_{ii},0,\ldots,0)$ $(i=\overline{1,n})$, обладают полезными арифметическими и корректирующими свойствами [19, 10, 34].

Разработаны методы построения т.н. «псевдонепозиционных» и «псевдодвоичных» представлений чисел [18, 19]. Поскольку неизбыточные модулярные системы обладают упомянутыми выше недостатками, о которых говорил Н. Сабо, одним из направлений исследований было нахождение методов введения избыточности в модификации систем остаточных классов для улучшения их арифметических свойств [24, 28]. Доказано что коэффициент r_N из соотношения

$$N = \sum_{i=1}^{n} B_i\lambda_i - r_N\prod_{i=1}^{n} p_i \quad (\lambda_i \equiv f_i(\alpha_i) \bmod p_i;\ \alpha_i \equiv N \pmod{p_i};\ i=\overline{1,n})$$

называемый «рангом числа» [3] может быть введен в код числа в виде вычета по основанию δ если $B_i \equiv 0 \pmod{\delta}$ $(i=\overline{1,n})$. Примером того, какие применения могут найти такие избыточные системы, может служить найденная модификация быстрого умножения Шонхаге [4]. Основу метода составляет получение специ-

альных решений системы сравнений $w \equiv w_i \pmod{m_i}$ $(i = \overline{1,n})$ для $w \in [0, M)$ путем применения рекуррентных вычислений, при условии, что основания m_i представляют собой параметрическую систему попарно взаимно простых чисел Мерсенна. Однако, введя в систему оснований избыточное основание вида $m_0 = 2^\nu$ и оперируя соответствующими вычетами сомножителей u и v по основанию m_0, можно исключить из последнего этапа алгоритма рекуррентные вычисления, на долю которых приходятся здесь основные временные затраты [41]. Имеет место следующая

Теорема. Пусть

$$u \cdot v = w \in [0, M) \qquad (M = \prod_{i=1}^{n} m_i) .$$

Основания

$$m_i = 2^{\nu_i} - 1 \qquad (i = \overline{1,n})$$

попарно взаимно просты между собой и введено избыточное основание $m_0 = 2^\nu$ такое, что $2n - 2 < 2^\nu$.

Тогда w может быть вычислено из соотношения

$$w = M_i w_i' + \cdots + M_{i1} \overline{w'}_{i1} + \cdots + M_{ir_w} \overline{w'}_{ir_w} + \cdots + M_n w_n'$$

где

$$M_i = M^{-1} m_i; \quad w_i' \equiv w_i M_i^{-1} \equiv u_i v_i M^{-1} \pmod{m_i};$$

$$0 \le w_i' < m_i; \quad \overline{w'}_i = w_i' - m_i$$

$$r_w = \begin{cases} r_w' & , \text{при } r_w' \le n-1 \\ 2^\nu - r_w' & , \text{при } r_w' > n-1 \end{cases}$$

$$r_w \equiv w_0 \cdot r_{e1} - \sum_{i=1}^{n} \delta_i \pmod{2^\nu} \quad (0 \le r_w' < 2^\nu)$$

$w_0 \equiv v_0 u_0 \pmod{2^\nu}$ – число, определяемое ν младшими разрядами двоичного представления произведения w; δ_i – числа, определяемые ν младшими разрядами двоичных представлений величин w'_i $(i = \overline{1,n})$; r_{e1} – целочисленная константа. Применение избыточного непозиционного представления с модульным заданием значения ранга кодируемого числа, позволило помимо распараллеливания вычислительных процедур, значительно упростить алгоритм декодирования и построить модификацию метода, превосходящую по быстродействию и достоверности исходный алгоритм Шенхаге. Умножение на константы $\left\{ M_i^{-1} \right\} m_i$, в случае табличной реализации, не влечет за собой временные и аппаратурные затраты. Так же получение u_ν, v_ν – вычетов сомножителей u, v по основанию $m_0 = 2^\nu$ и величин δ_i $(i = \overline{1,n})$ не требует дополнительных временных затрат и реализуется на стандартном оборудовании с минимальной аппаратурной избыточностью.

Определение w без предварительных рекуррентных вычислений приводит к существенному (n-кратному) временному выигрышу. Не менее существенными представляются, открываемые предлагаемой модификацией, дополнительные возможности распараллеливания и табличной реализации операций над очень большими числами. Построение на этой основе соответствующего специального программного обеспечения, ориентированного на использование ресурсов больших компьютерных сетей, таких как Интернет, может найти важное применение для решения научных и криптографических задач, связанных с нахождением делителей гиперчисел.

Одной из известных криптографических систем с открытыми ключами, где могут быть использованы возможности модульной арифметики, является система Меркля-Хеллмэна [52, 53], вычислительная стойкость которой обеспечивается трудностями решения в общем виде так называемой «задачи об укладке ранца». В этой системе в качестве открытого ключа публикуется n-мерный вектор $a = a_1, a_2, \ldots, a_n$, порождаемый вектором a' из сравнений

$$a_i \equiv a'_i w \pmod{r} \quad (0 \le a_i < r;\ \sum_{i=1}^{n} a'_i < r;\ i = \overline{1,n}) .$$

Компоненты вектора a' подобраны таким образом, что соответствуют легко решаемому частному случаю задачи об укладке ранца. Кодирование n-битных двоичных блоков $x = x_1, x_2, \ldots, x_n$ происходит путем образования сумм вида

$$S = \sum_{i=1}^{n} a_i x_i ,$$

декодирование – посредством вычисления

$$S' \equiv S_w^{-1} \pmod{r} \quad (w \cdot w^{-1} \equiv 1 \pmod{r}) .$$

и последующего решения частного случая задачи об укладке ранца при $S' = a'x$. Параметры a', r, w – составляют комплекс личного ключа пользователя. Рассматривается возможность использования для декодирования цифрового сообщения S специальных процедур перевода в одну из предложенных непозиционных систем счисления [31, 36]. Декодирование после нахождения S', состоит в отожествлении этого сообщения с его непозиционным представлением в виде последовательности наименьших неотрицательных вычетов $x = x_1, x_2, \ldots, x_n$ соответственно по попарно взаимно простым основаниям, причем

$$x_i \equiv S' k_i^{-1} \pmod{p_i}, \quad i = \overline{1,n}, \tag{3}$$

где $k^{-1} \equiv 1/k_i \pmod{p_i}$, а левая часть принимает значения 0 или 1. В частности, нахождение двоичных разрядов x_i может осуществляться параллельно на n вычислительных блоках, реализующих вытекающее из (3) соотношение

$$x_i = \begin{cases} 0 & , \text{при } S' \equiv 0 \pmod{p_i} \\ 1 & , \text{при } S' \not\equiv 0 \pmod{p_i} \end{cases} \qquad i = \overline{1,n}$$

Такая структура алгоритма позволяет увеличить производительность декодирующей аппаратуры, избежать на данном этапе вычислений, операций над гипероперандами, требующих кратной точности. Не менее важна возможность повышения достоверности вычислений по сравнению с алгоритмом последовательного нахо-

ждения x_i, так как здесь устраняется распространение ошибок на соседние разряды.

Вопросы применения теории модулярной арифметики для построения алгоритмов решения некоторых задач целочисленного линейного программирования составляют предмет отдельного рассмотрения.

Литература

1. Valach M. *Vznik kodu a ciselne zbytkoych tzid Stroje na zprakovant informaci.* sbornik III Nak I. CSAV, 1955.
2. Szabo N. *Sign detection in nonredundant residue system*, volume EC-11, N4. “IRE Trans.”, 1962.
3. Акушский И. Я., Юдицкий Д. И. *Машинная арифметика в остаточных классах.* «Советское радио», Москва, 1967.
4. Knuth D. E. *The art of computer programming*, volume 2. Addison-Wesley, 1969.
5. Акушский И. Я., Амербаев В. М., Пак И. Т. *Основы машинной арифметики комплексных чисел.* «Наука», Алма-Ата, 1970.
6. Амербаев В. М. Теоретические основы машинной арифметики. «Наука», Алма-Ата, 1976.
7. Хацкевич В. Х. *Об арифметических возможностях одного класса непозиционных кодов.* Сб. «Управл. выч. Машины». Энергия, М. – Л., 1967.
8. Акушский И. Я., Хацкевич В. Х. *Инверсные представления чисел в системе остаточных классов.* Сб. «Цифровая вычислительная техника и программирование, вып. 2», Москва, 1967.
9. Акушский И.Я., Хацкевич В.Х. *О безранговых непозиционных представлениях чисел для одного класса оснований.* Сб. «Математическая и техническая кибернетика», Тб. Мецниереба, 1975.
10. Хацкевич В. Х. *Об одном классе неортогональных базисных систем в СОК*. «Сборник научных трудов МО #75», Москва, 1967.
11. Хацкевич В. Х. *Вопросы работы с приближенными числами в системе остаточных классов*. Сообщ. АН ГССР., Тбилиси, 1967.
12. Хацкевич В. Х. *Модульная арифметика в системе остаточных классов*. Труды ТНИИСА, IX т., Тбилиси, 1969.

13. Хацкевич В. Х., Даниленко П. Я. *К вопросу о повышении эффективности работы электронных цифровых систем путем применения методов непозиционного кодирования*. ВРЭ #9, Москва, 1969.

14. Габашвили Н. В., Хацкевич В. Х. *К вопросу о матричной арифметике непозиционных систем*. Сообщ. АН ГССР, Тбилиси, 1971.

15. Хацкевич В. Х. *О неортогональных базисных системах непозиционных представлений чисел*. Сообщ. АН ГССР., Тбилиси, 1971.

16. Гегелия Г. Д., Хацкевич В. Х. *Об одном методе построения систем автоматического контроля*. «Тр. III научно-технической конференции по надежности», Ленинград, 1970.

17. Габашвили. Н. В., Хацкевич В. Х. *К вопросу о мультипликативных характеристиках непозиционных систем*. «Тр. Проблем. лабор. автоматики и вычислительной техники ГПИ», Тбилиси, 1971.

18. Габашвили Н. В., Хацкевич В. Х. О контроле по модулю машинных операций над псевдодвоичными словами. «Тр. Проблем. лабор. автоматики и вычислительной техники ГПИ», Тбилиси, 1971.

19. Хацкевич В. Х. *К вопросу о достоверности передачи информации между компонентами гибридных вычислительных комплексов*. Сб. «Гибридные вычислительные машины и комплексы». «Наукова думка», Киев, 1973.

20. Хацкевич В. Х., Шакарян Р. А. *О применении элементов теории вычетов для измерения линейных величин*. «Материалы второй II научной республиканской конференции по метрологии. т. I», Тбилиси, 1974.

21. Хацкевич В. Х. *О расширении диапазона представления чисел для одного класса непозиционных кодов*. Сб. «Математическая и техническая кибернетика», Мецниереба, Тбилиси, 1975.

22. Хацкевич В. Х. *Минибазисные модели непозиционных систем*. Сб. «Математическая и техническая кибернетика», «Мецниереба», Тбилиси, 1977.

23. Х. Хацкевич В, Чачанашвили А. Р. *Базисные представления для одного класса непозиционных систем*. Сб. материалов «Математическая и техническая кибернетика», «Мецниереба», Тбилиси, 1977.

24. Хацкевич В. Х. *Вопросы построения избыточных непозиционных кодов*. Сб. «Вопросы вычислительной техники и управления». Изд.

«Мецниереба», Тбилиси, 1978.

25. Хацкевич В. Х., Чачанашвили А. Р. *Вопросы построения однопараметрических двуступенчатых непозиционных кодов*. Сб. «Математическая и техническая кибернетика», «Мецниереба» Тбилиси, 1979.

26. Хацкевич В. Х., Чачанашвили А. Р. *Избыточное представление чисел для матричных компонент вычислительных сетей*. Сб. «Вопросы кибернетики, кодирования и передачи информации в вычислительных сетях». АН СССР. Выпуск 42., Москва, 1978.

27. Хацкевич В. Х. *Базисные представления непозиционных систем в метрологических задачах*. Материалы пятой республиканской научно-технической конференции по метрологии, Тбилиси, 1978.

28. Хацкевич В. Х., Шакарян Р. А. *О применении одной модификации циклического AN-кода для построения модульной арифметики непозиционных систем*. Сб. «Вопросы вычислительной техники», Тбилиси, 1979.

29. Хацкевич В. Х. *О решении целочисленных линейных оптимизационных задач в остаточных классах*. Сб. «Математическая и техническая кибернетика». Изд. «Мецниереба», Тбилиси, 1981.

30. Хацкевич В. Х., Мгебришвили М. Н. *О некоторых кодовых произведениях в вычислительных сетях*. Сб. «IV Всесоюзная школа-семинар по вычислительным сетям», Москва-Ташкент, 1979.

31. Хацкевич В. Х. *Модификация криптографической системы для вычислительных сетей*. Сб. «V Всесоюзная школа-семинар по вычислительным сетям», Москва-Владивосток, 1980.

32. Хацкевич В. Х., Шакарян Р. А. *Применение нормальных инверсных представлений в задачах цифрового регулирования*. Сб. «Материалы VI конференции по метрологии», Тбилиси, 1982.

33. Хацкевич В. Х. *Некоторые методы декодирования в системе счисления в остаточных классах*. Сб. «III Международный симпозиум по теории информации», Москва-Тбилиси, 1979.

34. Хацкевич В. Х. *Корректирующие возможности некоторых базисных представлений непозиционных систем*. Сб. «IV Всесоюзная школа-семинар по вычислительным сетям», Москва-Ташкент, 1979.

35. Хацкевич В. Х. *Адаптивный метод нахождения интегральных характеристик при непозиционных параллельных вычислениях*. Сб. «Вопросы разработки и применения средств вычислительной техники». Материалы республиканской конференции, Тбилиси, 1982.

36. Хацкевич В. Х., Ревазишвили Г. Г. Система Меркля-Хэллмэна с распараллеливанием вычислений. Сб. «VIII Всесоюзная школа-семинар по вычислительным сетям», Москва, 1983.

37. Хацкевич В. Х. *Специальные базисные представления непозиционных систем для метрологических задач*. Труды института вычислительной математики. «Математическая и техническая кибернетика», 1984.

38. Хацкевич В. Х. *Избыточность для оптимизации специальных арифметических свойств кода*. Труды IX симпозиума по проблеме избыточности в информационных системах, Ленинград, 1986.

39. Хацкевич В. Х., Чачанашвили А. Р. *Устройство для определения знака числа в системе остаточных классов*. Авторское свидетельство на изобретение #1254480, Государственный реестр изобретений СССР, Бюллетень изобретений от 1.5.1986.

40. Хацкевич В. Х. *О использовании структуры представления информации для защиты арифметической обработки в сетевых системах*. Труды XII Всесоюзной школы-семинара по вычислительным сетям, Москва-Одесса, 1987.

41. Хацкевич В. Х. *Модификации метода быстрого умножения для высокопроизводительных систем*. Сб. «Математическая и техническая кибернетика», «Мецниереба», Тбилиси, 1987.

42. Хацкевич В. Х. *Избыточность для декодирования на основе алгоритмов с модулярной арифметикой*. В сб. «Шестой международный симпозиум по теории информации». Часть II, Москва-Ташкент, 1984.

43. Хацкевич В. Х. Сопряженные по рангу числовые представления. Сб. «IX Всесоюзная конференция по теории кодирования и передачи информации», часть I, Одесса, 1988.

44. Хацкевич В. Х. *Переходные параметры ступенчатых непозиционных систем*. Сб. «Математическая и техническая кибернетика», т. 24, вып. 2, Тбилиси, 1984.

45. Хацкевич В. Х. *Корректирующие возможности сопряженных последовательностей парных произведений*. Сб. «X симпозиум по проблеме избыточности в информационных системах», Ленинград, 1989.

46. Хацкевич В. Х. *Быстрое умножение для высокопроизводительных систем*. Труды Международной конференции «Высокопроизводительные вычислительные системы в управлении и научных исследованиях», Москва, 1991.

47. Shamir A. *A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem*, volume IT-30, pages 699–704. IEEE Trans. Inform. Theory, September 1984.

48. Shamir A. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

49. Tanaka R. I. *Some options in the design of a residue arythmetic*, volume III. “Proc.Nat.El.Conf”, Chicago, 1963.

50. Rivest R. L., Shamir A., Adleman L. Some options in the design of a residue arithmetic. *Commununications of ACM*, 21(2):120–126, April 1978.

51. Rivest R. L., Shamir A., Adleman L. *On Digital Signatures and Public Key Cryptosystems*, pages 120–125. Technical Report, MIT/LCS/TR-212, January 1979.

52. Hellman M. E. The mathematics of public key cryptography. *Scientific American*, 241:146–157, February 1979.

53. Merkle R., Hellman M. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, IT-24:525–530, September 1978.

54. Cho G. Y., Johnson L. G., Soderstrand M. A. New complex-arithmetic heterodyne filter. In *ISCAS (3)*, pages 593–596, 2004.

55. Soderstrand M. A., Jenkins W. K., Jullien G. A., Taylor F. J. *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*. IEEE Press, 1986.

УДК 658.512.011.56

Б.М. Малашевич, Д.Б. Малашевич

Модулярная арифметика – взгляд изнутри

(ОАО «Ангстрем»,
Московский государственный институт электронной техники)

Произведён краткий обзор состояния и развития модулярной арифметики на основе экспертного анкетирования участников международной юбилейной конференции «50 лет модулярной арифметике».

За 50 лет развития модулярной арифметики (МА) о ней говорили много и многие и очень по-разному. Диапазон мнений простирается от крайнего оптимизма, рассматривающего модулярную арифметику как светлое будущее вычислительной техники, до крайнего пессимизма, считающего, что у нее нет будущего. Как обычно бывает, наиболее категоричны те, кто МА не занимается и слабо представляет ее особенности. В связи с этим особый интерес представляет, что же о модулярной арифметике думают ее активные приверженцы, которые, несмотря ни на что, тратят на нее свои интеллект, силы и время. Что бы получить такой «взгляд изнутри», оргкомитет Международной юбилейной научно-технической конференции изначально задумал сделать общий обзор истории, состояния и прогноз перспектив модулярной арифметики на основе экспертного опросы, в котором в качестве экспертов выступают

участники конференции. В статью не включены материалы обзоров А.А. Коляды с А.Ф. Чернявским и В.М. Амербаева, т.к. они полностью представлены в трудах конференции.

Для подготовки обзора была разработана специальная анкета, которую предложили заполнить всем участникам конференции – экспертам. Откликнулись они на эту инициативу оргкомитета поразному. Кто-то ответил сразу, из кого-то ответы пришлось буквально вытягивать, а кто-то так и не ответил. Сказалась и неопытность составителей анкеты. Полученные ответы были написаны в произвольной форме и практически не поддавались обобщению. Пришлось пойти на второй шаг: формализовать полученные ответы, составить новые анкеты экспертной оценки и разослать их для вторичного заполнения. Это тем более не вызвало энтузиазма у многих участников конференции, но, после многочисленных напоминаний, используя ответы и на первую, и на вторую анкеты, удалось все же получить некоторый материал. Всего в ходе обоих опросов анкеты заполнило 41 человек, причем многие ответили не на все вопросы. Ряд вопросов подразумевал возможность нескольких вариантов ответов, поэтому число ответов в таких случаях превышает число экспертов. При обработке результатов авторы убедились, что и второй вариант анкеты оказался далеким от удачного, поэтому часть ответов пришлось интерпретировать по ходу обработки, стараясь при этом не вносить искажения.

Результаты опроса сведены в диаграммы и каждый может самостоятельно их интерпретировать в соответствии со своим пониманием. Любые интерпретации собранных данных, если они этим данным не противоречат, имеют право на существование. Ниже приведены некоторые интерпретации авторов, которые не является догмой, но так же имеет право на существование.

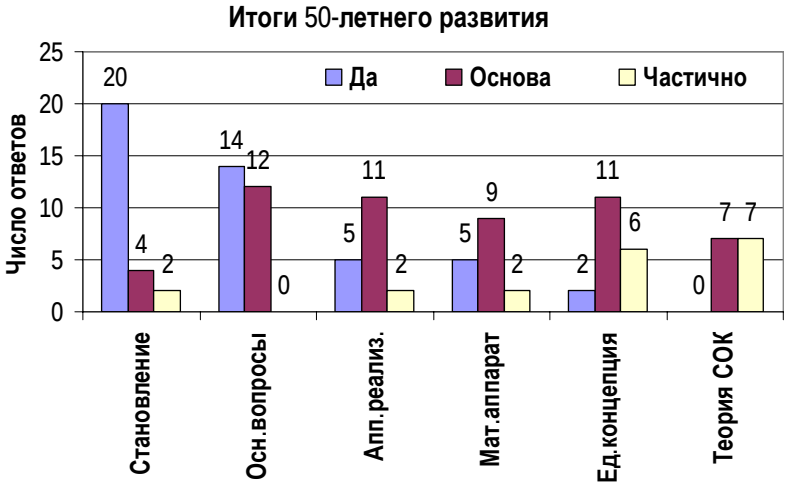
Итоги развития МА

50 лет в науке это и много, и мало. Вопросом: «*Как Вы оцениваете 50-летний этап развития МА?*» экспертов попросили оценить результаты этого периода. Из ответов образовалось шесть видов оценки (здесь и далее выделенное шрифтом и подчеркиванием – форма надписи на диаграммах):

1. Этап **становления** теории модулярных вычислений.
2. Решены **осн**овные теоретические **вопросы** МА.

3. Решены основные вопросы **апп**аратной **реализации** устройств для выполнения модулярных операций
4. Развит **мат**ематический **аппарат** модулярной арифметики
5. Создание **ед**иной **концепции** построения средств обработки информации в целочисленной арифметике
6. Завершено создание **теории** применения **СОК**.

Варианты оценки: **да**, в **основа**, **частично**.

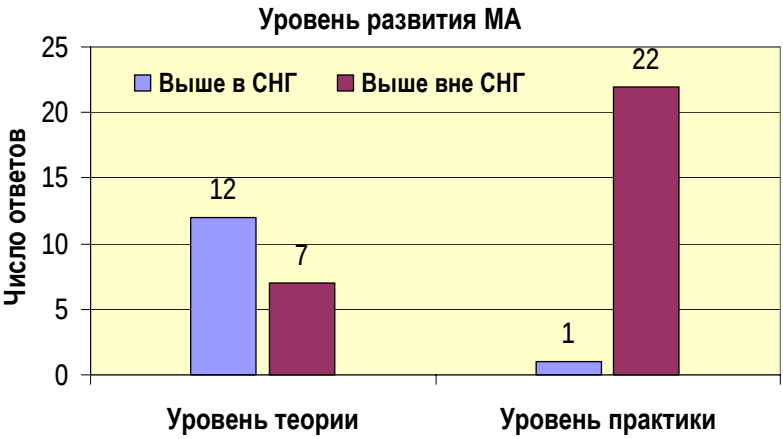


На этот вопрос ответило 28 экспертов, причем не все высказались по каждому виду оценки. Общий итог – большинство экспертов считает, что МА завершила этап становления как самостоятельное направление науки, решены ее основные теоретические проблемы, но многое еще предстоит сделать.

Современное состояние МА

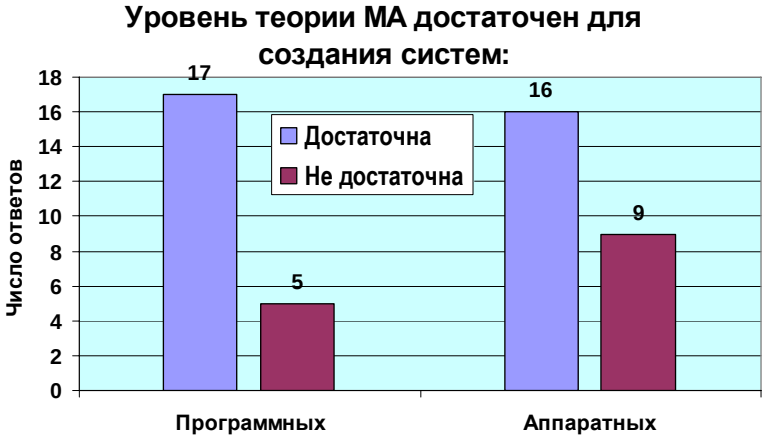
Ответы 34 экспертов на вопрос: «*Как Вы оцениваете современное состояние МА в стране и за рубежом*» вылился в четыре группы оценок.

1. Сравнительная оценка состояния теоретических и практических разработок в странах СНГ и дальнем зарубежье представляется несколько странной. Эту оценку дали 23 эксперта. Большинство из оценивших уровень теоретических разработок (63%) сочли, что у нас он выше, чем в дальнем зарубежье.



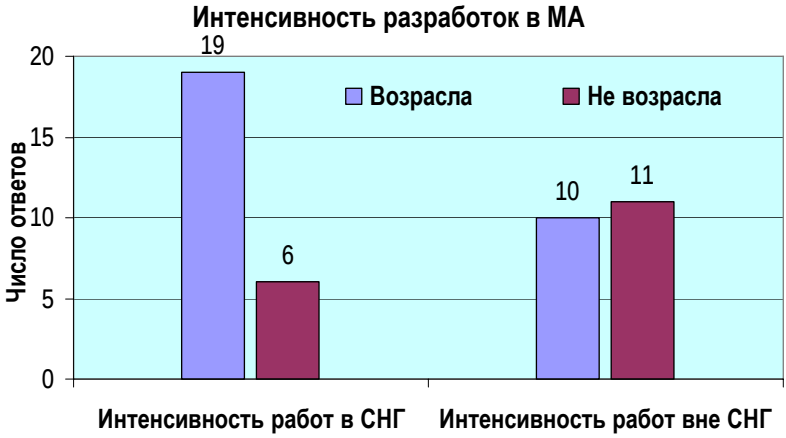
Это можно объяснить недостатком информации о зарубежных исследованиях. Но практически единодушно (96%) эксперты зарубежный уровень практических работ оценили выше. По-видимому, это чисто интуитивная оценка: информации о фактах применения МА в странах дальнего зарубежья, несмотря на многочисленные «приставания» к ним, они так и не дали, т.е. ее у них нет. Таких примеров разработок вне СНГ, как наши Т-340А, К-340А, Алмаз, 5Э53, изделия Б.С. Гаспера, Е.К. Лебедева, А.А. Коляды и т.п. экспертам не известно. И в то же время такая оценка.

2. Вторая оценка рассматривает достаточность уровня теоретической разработанности МА для построения прикладных аппаратных и программных систем. Большинство экспертов считает, что этот



уровень достаточен, следовательно не является препятствием для практического применения МА.

3. Интересна динамика разработок в МА. 25 экспертов оценили интенсивность разработок в странах СНГ и вне его.

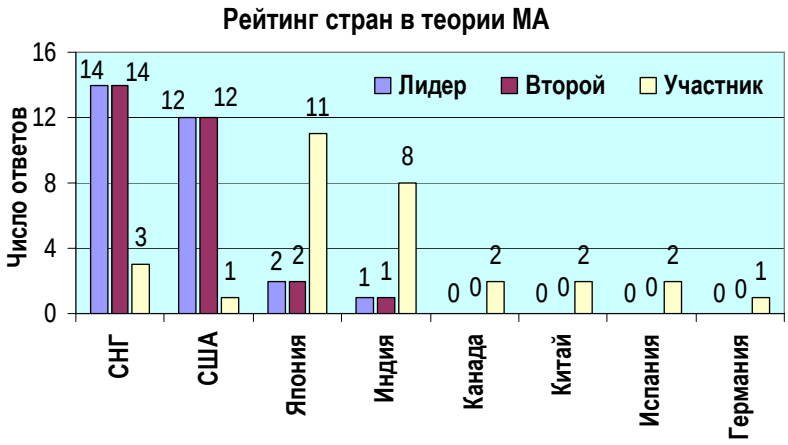


78% экспертов считают, что у нас интенсивность разработок растет. О динамике вне СНГ оценки противоречивы, почти половина экспертов считает, что интенсивность растет, немного более половины – что падает. Опят непонятно и противоречит оценке уровня развития МА. Что же так энергично разрабатывают наши ученые, если в результате нет реального применения. А вне СНГ топчутся на месте, отстают в теории, но в практике, по нашим же оценкам, оставили нас далеко позади.

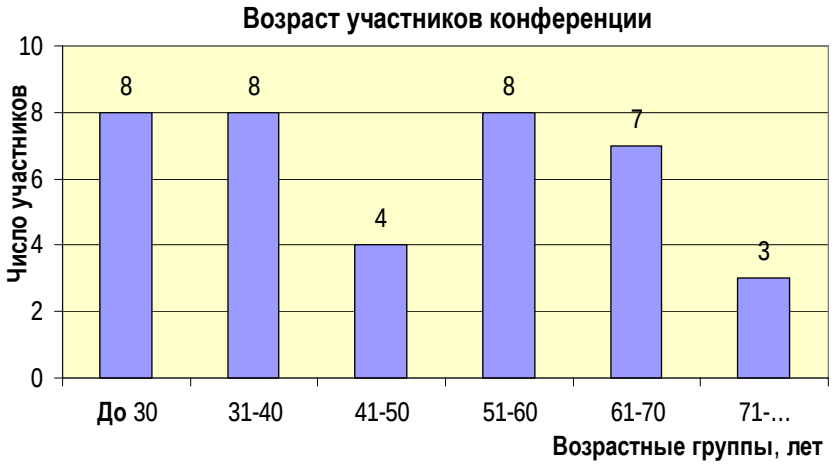
4. Не дает ответа на этот вопрос и четвертая оценка 31 экспертов современного состояния МА. В ней сделана попытка оценка уровня развития МА в различных странах. Выявлено два явных лидера: СНГ (45%) и США (39%). Интересно, что для обоих лидеров подано по равному числу голосов, отдающих им первое или второе место в рейтинге. Минимальное число экспертов отдают лидерство Японии (6%) и Индии (3%).

Возраст

Одним из верных признаков перспективности того или иного научного направления представляется возраст активно работающих в нем людей.



Естественно ожидать, что в продуктивном стабильном направлении все возрастные группы ученых должны быть представлены примерно одинаково. В неперспективные направления молодежь, как правило, не идет. Но она активно идет в те развивающиеся направления, в будущее которых верит.



Распределение по возрастным группам получилось до неприличного красивым. Оно неоспоримо демонстрирует, что СОКу все возрасты одинаково покорны, что интерес к модулярной арифметике стабилен. Заметный спад в группе от 41 до 50 лет является исключением, подтверждающим закономерность – это «эхо» негативного

влияния «провала» проекта ЭВМ 5Э53: истинные его причины научной общественности были неведомы, но сам факт получил широкую известность, был неверно истолкован и подорвал доверие молодежи тех лет к модулярной арифметике. Но вскоре интерес к ней восстановился.

Причина интереса к МА

Естественно возникает вопрос, почему этот интерес появляется, почему люди занимаются проблемами модулярной арифметики, направлением науки не только не популярным долгое время, но и в какой-то мере дискредитированным в современных научных кругах. На вопрос «*Что явилось первопричиной Вашего интереса к модулярной арифметике?*» ответило 39 экспертов. Полученные ответы укладываются в четыре группы:

1. **Интерес** возник в результате самостоятельной научной деятельности и изучения специальной литературы,
2. Необходимость решения производственной **проблемы,**
3. Влияние **авторит**етного ученого или научного руководителя,
4. Интерес возник в процессе **учебы.**

Все ответы (здесь и далее, если не оговорено особо) распределены по трем уровням приоритета: **первично**, **вторично** и несущественно (**неважно**).

Опрос показывает, что большинство экспертов (22 из 39) заинтере-

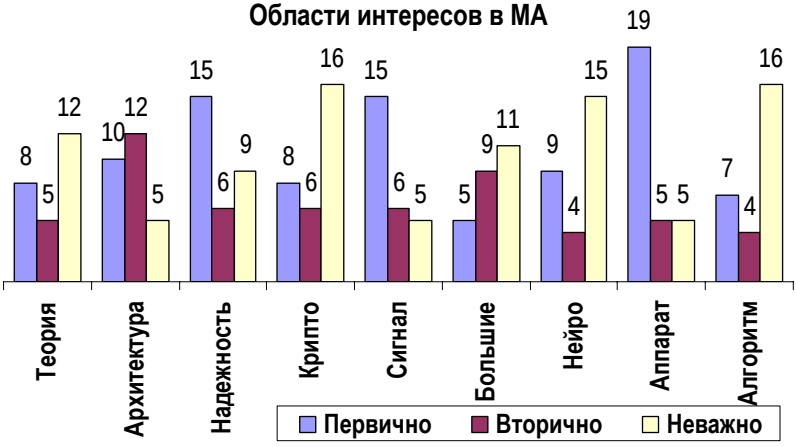
совались модулярной арифметикой под влиянием авторитетного ученого, в т.ч. 14 из них были вовлечены в МА научным руководителей при поступлении в аспирантуру (адъюнктуру).

Половина экспертов (18 из 39) к первичным причинам, побудившим их заниматься МА, отнесли интерес к ее специфичным свойствам, у половины из них (9 человек) этот интерес возник в результате решения производственных проблем, которые с помощью МА удалось решить изящнее. Парадоксально, но 46 ответов относят эти же причины к несущественным.

Области интересов в МА

Естественно спросить, что же привлекает ученых в этом научном направлении? На вопрос «*Круг Ваших интересов в модулярной арифметике?*» ответило 40 экспертов. Но поскольку многие из них назвали по два и более интересных для них направления в модулярной арифметике, то ответов получилось 57. Все ответы неплохо группируются в девять направлений:

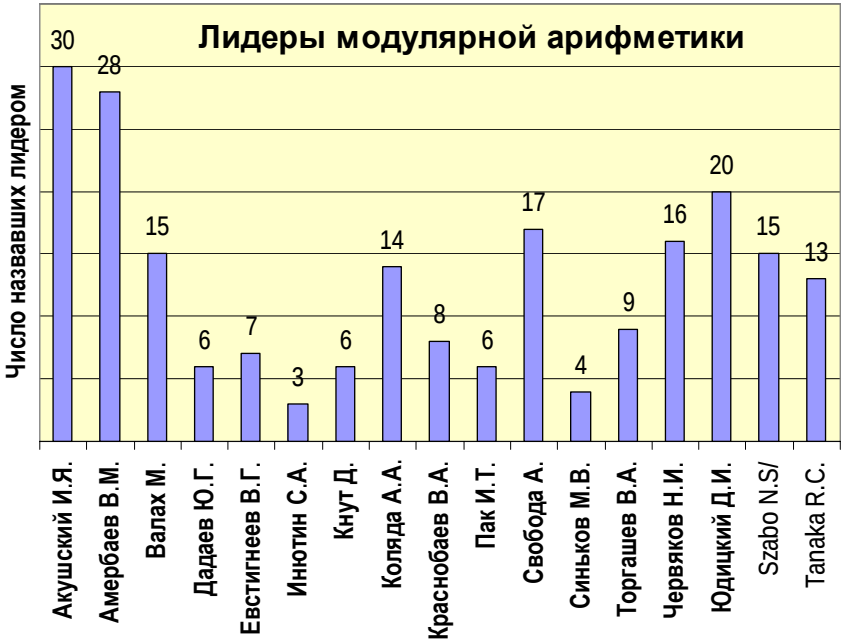
- Развитие **теории** модулярной арифметики.
- Развитие **архитектур** вычислительных средств и спецпроцессоров.
- Обеспечение **надежности**, живучести, обнаружения и исправления ошибок.
- **Крипто**графия, шифрование, защита информации.
- Обработка **сигнал**ов и изображений.



• Обработка многоразрядных данных, (**большие** числа).
• Модулярные **нейро**структуры.
• **Аппарат**ная реализация модулярных устройств.
• Программная реализация модулярных **алгоритм**ов.

Из диаграммы следует, что спектр интересов участников конференции в МА весьма широк, но все же превалируют проблемы надежности и обработки сигналов, т.е. используются основные преимущества СОК – арифметичность и параллельность. И очевидно предпочтение аппаратной реализации модулярных устройств по сравнению с программной реализацией модулярных алгоритмов.

Лидеры в модулярной арифметике

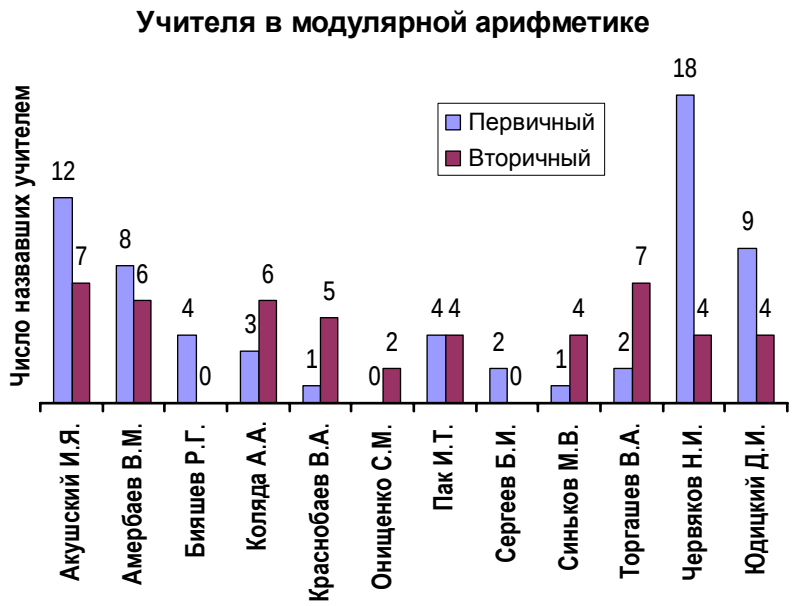


Ответы на вопрос «*Чей вклад в развитие модулярной арифметики Вы считаете наиболее весомым (в стране и за рубежом)?*» ответило 36 экспертов, ответы сложились следующим образом (по ординате указано число отметивших особый вклад лидеров в развитие модулярной арифметики).

Бесспорными лидерами признаны основоположники отечественной МА – И.Я. Акушский, Д.И. Юдицкий и В.М. Амербаев, а также авторы идей применения СОК в вычислительной технике чехи М. Валах и А. Свобода, хотя вряд ли многие знакомы с их трудами, по-видимому это знак признательности первопроходцам. Еще помнят, но уже меньше, СОК-овцев первой волны, многое сделавших в свое время для развития модулярной арифметики, но впоследствии отошедших от нее – И.Т. Пака, М.В. Синькова, В.А. Торгашова (они даже отказались от участия в конференции). Отмечены и активно работающие в настоящее время СОК-овцы второй волны: А.А. Коляда, Н.И. Червяков, В.А. Краснобаев, С.А. Инютин и др.

Учителя и ученики

Лидеры – это явление общественное, но у каждого есть свой учитель, который, как мы уже отмечали, привел многих участников конференции в модулярную арифметику и помог занять в ней оп-



ределенное положение. Что бы определить активных подвижников МА, был задан вопрос: «*Кого Вы считаете своим учителем в модулярной арифметике?*». Ответило на него 36 экспертов, в качестве учителей было названо 15 фамилий. Многие в качестве своего учи-

теля назвали более одной фамилии, в т.ч. людей, известных им только по трудам. Причем выявилось две категории учителей, первичные и вторичные. В результате общее число ответов достигло 122, в т.ч. 68 первичных и 54 вторичных. Естественно эта статистика далеко не полная, т.к. в ней приведены данные опроса только участников конференции, а это далеко не все, кто в настоящее время активно работает в МА, и без учета тех, кто по каким-то причинам прекратил эту работу. Из-за высокой плотности диаграммы из нее пришлось опустить учителей, имеющих менее двух учеников.

Из диаграммы можно сделать два вывода. Во-первых благодарные ученики помнят своих учителей, и не только непосредственных, но и тех, по чьим трудам они учились без личного контакта. И во-вторых, о необыкновенном подвижничестве в настоящее время Н.И. Червякова, которого в той или иной мере своим учителем считает 22 участника конференции, т.е. больше половины.

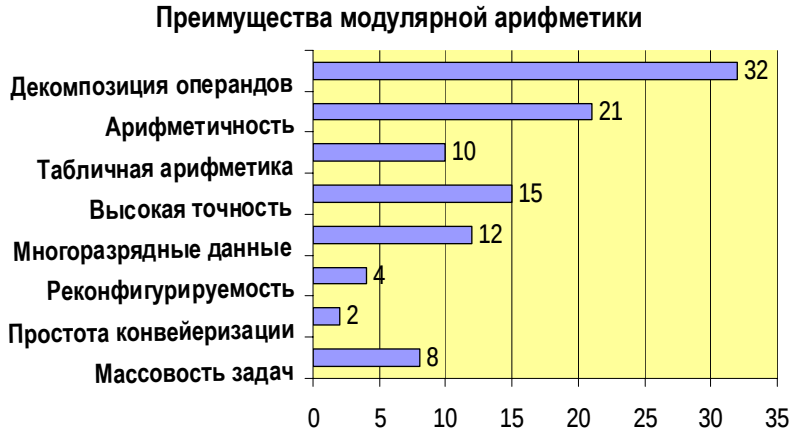
Участников конференции просили также назвать своих учеников. Откликнулось 15 человек, назвавших в совокупности 63 фамилии. Таким образом, с учетом участников конференции (некоторые из них и учителя, и ученики), в настоящее время выявлено 102 человека, занимающиеся проблемами модулярной арифметики (это без учета СОК-овцев первой и второй волн, о которых авторам известно, что по различным причинам они прекратили эти занятия). Все ли они в настоящее время активно занимаются МА, осталось не выясненным, но, по-видимому, не все, т.к. более половины из них к конференции не проявила никакого интереса.

Преимущества модулярной арифметики

Интересно было узнать, какие из свойств модулярной арифметики ее приверженцы считают наиболее полезными. На вопрос: «*Что Вы считаете важнейшими положительными качествами модулярной арифметики?*» ответило 34 эксперта, в совокупности отметивших 8 положительных качеств МА:

- Распараллеливание вычислений на уровне **декомпозиции операндов**, влекущее за собой резкое сокращение времени выполнения модульных операций.
- **Арифметичность** МА, обеспечивающая возможность обнаружения и исправления ошибки во время выполнения модульных операций.

- Возможность реализации **табличной арифметики** модульных операций и полиномных функций.



- **Высокая точность** вычислений в целочисленном диапазоне.
- Высокая эффективность при обработке **многоразрядных** (сотни и тысячи бит) **данных.**
- Гибкая **реконфигурируемость** структуры вычислителя и системы.
- **Простота конвейеризации** вычислений.
- **Массовость задач**, на которых МА эффективна.

Диаграмма показывает, что наиболее ценится в МА распараллеливание операндов и арифметичность. Несколько удивляет, что 24 эксперта (70%) не считают важной возможность реализации табличной арифметики.

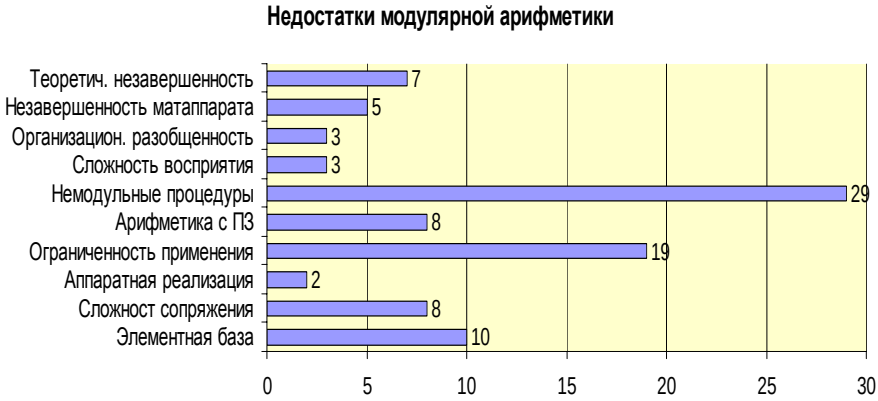
Недостатки модулярной арифметики

Особый интерес представляет оценка приверженцев модулярной арифметики ее недостатков. На вопрос «*Что Вы считаете важнейшими недостатками модулярной арифметики, как Вы оцениваете возможность их преодоления?*» ответило 34 эксперта, в совокупности отметивших 10 недостатков:

- **Теоретич**еская **незавершенность** МА,
- **Незавершенность мат**ематического **аппарата**,
- **Организацион**ная **разобщенность** исследований и разработок,

- Математическая **сложность** МА для **восприятия** инженерам, отсутствие специальной для них литературы, облегчающей это восприятие,
- Сложность выполнения **немодульных процедур**,
- **Сложность арифметики с П**лавающей **З**апятой,
- Ограниченность **областей применения**
- Сложность **аппаратной реализации**,
- **Сложность сопряжения** с двоичной индустрией,
- Отсутствие специальной **элементной базы**.

Как видим, почти единодушно (85%) участников конференции главным недостатком МА считают сложность выполнения немодульных операций (вычитание и деление), резко сокращающую область эффективного применения СОК. Более половины (56%) вторым по важности недостатком видят ограниченность круга задач, эффективно решаемых МА. Однако если учесть, что сами эти задачи в последние годы перешли в категорию массовых, (криптография, обработка сигналов и изображений, топонимика и т.п.), то роль этого недостатка оказывается не столь важной: поле для применения МА огромное.



В связи с этим представляется несколько странным, что 10 человек (30%) третьим по важности недостатком считает отсутствие специальной элементной базы. Возникает вопрос, что они под этим понимают. Если отсутствие модулярных процессоров в виде БИС или

IP-модулей, то это не элементная база, в настоящее время для таких изделий принято иное название - электронная компонентная база (ЭКБ), где компонентом может быть и процессор. А элементная база – это, на современном уровне, библиотеки стандартных элементов в системах автоматизации проектирования БИС. Так вот, эта элементная база имеется в широком разнообразии для проектирования и заказных, и полузаказных БИС и на основе различных ПЛИС. И никто, кроме приверженцев модулярной арифметики, а наиболее активные среди них и являются участниками конференции, модулярные процессоры разрабатывать и не сможет, и не захочет. Так что 10 человек посетовали сами на себя.

Направления в МА

За 50 лет МА развилась в серьезную школу со своими научными направлениями. В результате первого опроса было выявлено 17 претендентов (тем и групп людей) на выделение в самостоятельное направление в МА:

1. **Основы** модулярной арифметики (Акушский И.Я., Юдицкий Д.И., Амербаев В.М.)
2. **Спец**иализированные **в**ычислительные **у**стройства (Акушский И.Я., Юдицкий Д.И., Амербаев В.М.)
3. **Конвейер**ная обработка цифровой информации (Коляда А.А., Пак И.Т.)
4. **Обобщенные** системы остаточных классов (Коляда А.А.)
5. **Нейро**математика и модулярные нейрокомпьютеры (Червяков Н.И., G. Jullien, M. Bayoumi)
6. **Ускоренная** модулярная арифметика СОК (Акушский И.Я., Юдицкий Д.И.)
7. Вычисления в **комплекс**ной плоскости (Амербаев В.М.)
8. Теория «**безошибоч**ных вычислений» (Е. Кришнамурти, Индия)
9. **Надежные** модулярные вычислительные структуры (Торгашев В.А., Краснобаев В.А.)
10. **Поз**иционно-**остаточные** представления (Евстигнеева В.Г.)
11. Параллельные логические вычисления на основе модулярных **ар**ифметико-**логическ**их форм (Финько О.А.)
12. **Корректир**ующие свойства кодов СОК (Дадаев Ю.Г., Торгашев В.А.)
13. Представление комплексных и **гиперкомплекс**ных чисел в СОК (Онищенко С.М., Синькова М.В., Губарени Н.М.)

14. **Мин**имально **избыточные** модулярные вычислительные структуры (А.А. Коляда, A. Kumarisan)
15. Компьютерная модулярная **алгебра** (А.А.Коляда, A. Scavantzos, F. Jenkins, G. Jullien, M. Bayoumi)
16. Многомашинная и **мультипроцесс**орная технологии модулярной обработки информации (А.А.Коляда, С.А.Инютин, K. Plessman)
17. Системы **защиты** информации с применением МА (В.П.Ирхин и др.)

В конце статьи приведены характеристики некоторых направлений, данные соответствующими участниками конференции.

Авторам представляется, что не все эти претенденты можно относить к самостоятельным направлениям. Одни не годятся по масштабности, вернее недостаточностью токовой. Другие являются вариациями на тему уже признанных направлений. Но авторы сочли, что приговор должны вынести участники конференции.
Для этого им был задан вопрос, какие из этих претендентов действительно можно отнести к самостоятельным направлениям МА, предлагалось четыре варианта ответов:

1. **Важнейшие** сформировавшиеся направления.
2. **Перспективные** сформировавшиеся направления.
3. Перспективность направления **не очевидна**.
4. Как самостоятельное направление **не сформиров**алось.

Рассмотрев гистограмму, авторы из этических соображений решили воздержаться от своих комментариев.


Направления в МА
Обобщенные
Спец.ВУ
Конвейер
Обобщенные
Нейро
Ускоренные
Комплексные
Безошибочные
Надежные
Поз.остаточные
Ар.логические
Корректирующие
Гиперкомплексные
Мин.избыточные
Алгебра
Мультипроцесс
Защита
Важнейшее
Перспективное
Не очевидно
Не сформиров.
0
5
10
15
20
25
Число ответов

Области эффективности МА

На вопрос «*Ваша оценка перспектив МА, области ее эффективного применения*» содержательно ответило 36 экспертов. Образовалось три категории областей эффективного применения МА: **первостепенные**, **второстепенные** и третьестепенные.

При этом было рассмотрено 13 классов задач, при решении которых применение МА целесообразно:

1. Цифровая обработка сигналов – **ЦОС**,
2. Цифровая обработка изображений – **ЦОИ**,
3. **Крипто**графия,
4. Целочисленная арифметика высокой точности – **ЦАВТ**,
5. Обработка **длинных** (многоразрядных, сотни и тысячи бит) целочисленных данных,
6. **Нейросет**евые системы обработки данных,
7. Высоконадежные **самокоррект**ирующие системы,
8. Пространственно-**распределен**ные высоконадежные системы,
9. Системы повышенной **производит**ельности,
10. **Некрипто**графические системы обработки, передачи и хранения секретной информации,
11. Системы искусственного **интеллект**а,
12. **Кодеры** и декодеры помехозащищенных кодов,
13. Высокоточные **ЦАП и АЦП**.

Гистограмма показывает, что по некоторым областям применения МА эксперты единодушны во мнениях (верхняя часть гистограммы), а по другим у них полный разлад во мнениях. Из этого можно сделать вывод, что комплексного изучения различных классов задач на эффективность их решения средствами МА не проводилось, во всяком случае результаты такого исследования экспертам незнакомы.

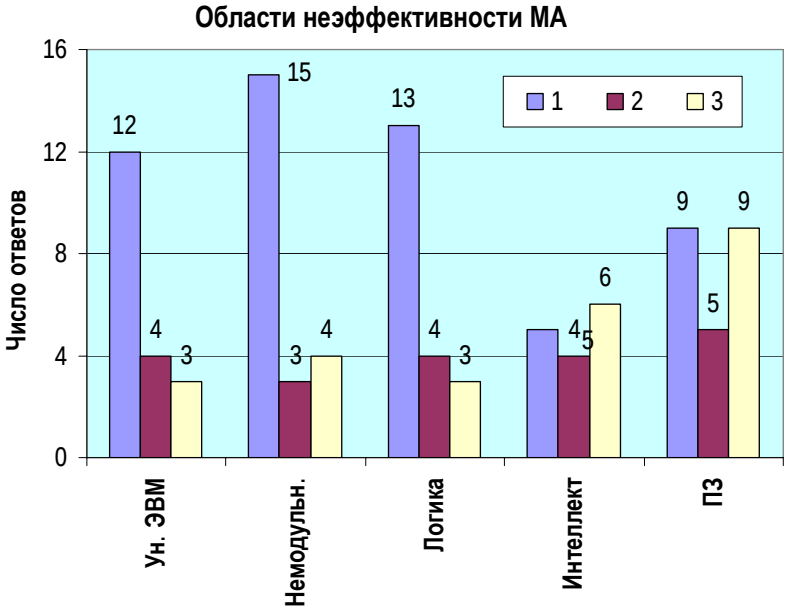

Области эффективного применения МА
ЦОС
ЦОИ
Крипто
ЦАВТ
Длинные
Нейросети
Самокоррекцмя
Распределенные
Производительные
Некрипто
Интеллект
Кодеры
ЦАП, АЦП
26
2
3
21
5
2
21
2
0
18
1
3
13
8
2
7
11
2
12
7
3
6
9
3
11
6
5
1
8
4
2
8
5
4
7
8
2
6
7
Первостепенные
Второстепенные
Третьестепенные
0
5
10
15
20
25
30
Число ответов

Области неэффективности МА

На вопрос «*Ваша оценка классов задач, на которых МА не эффективна*» содержательно ответило только 25 экспертов, т.е. 11 человек, оценивших области эффективности МА о сферах ее неэффективности высказаться не захотели, и это тоже показатель.

По степени неприменимости МА образовалось так же три категории оценки: **1** (самая жесткая), **2** и **3**. При этом на первом этапе экспертного опроса было выявлено только 5 категорий применения, в которых МА не эффективна, правда первая из этих категорий весьма обширна:

1. **Ун**иверсальные **ЭВМ**,
2. Задачи, с высокой долей **немодульн**ых операций,
3. Задачи, с высокой долей **логических** операций и операций сравнения,
4. Системы искусственного **интеллект**а,
5. Арифметика с плавающей запятой – **ПЗ**..



Полученная гистограмма подтверждает вывод, вытекающий из гистограммы областей эффективности МА. То же единодушие в одной части классов задач (в обоих случаях очевидных) и тот же

разброд мнений в другой части (с первого взгляда не очевидных). .

Библиография по МА

102 человека в науке – это много, целая армия. Каковы же результаты деятельности этой армии? Эффективность научной деятельности косвенно может быть оценена двумя показателями:

- Количеством публикаций по рассматриваемому направлению (для фундаментальных исследований).
- Количеством реально внедренных результатов (для прикладных исследований).

Модулярная арифметика содержит оба эти аспекта. Рассмотрим сначала публикации.

Оргкомитет изначально решил ввести в сборник трудов полную (насколько удастся) библиографию по МА. Участников конференции попросили представить всю имеющуюся у них информацию. Несмотря на то, что ответили не все, результат превзошел ожидаемое. Объединенная библиография включает 1336 публикаций разных видов, в т.ч. 981 (73,4 %) на русском языке и 355 (26,6 %) – на английском. Авторы понимают, что это далеко не полная библиография, особенно в ее англоязычной и иной иностранной части. Но это уже довольно большая выборка, из которой можно делать определенные выводы.

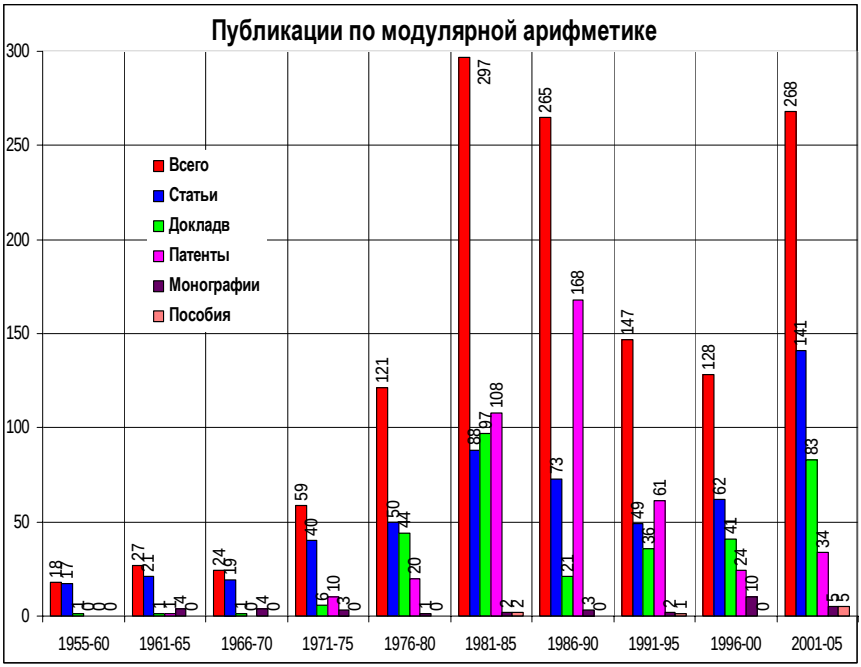
Все публикации были распределены на 5 групп:

- **Статьи**.
- **Доклады** на различных конференциях и симпозиумах.
- Авторские свидетельства СССР и **патенты**.
- **Монографии**.
- Учебные **пособия**.

Результаты исследования сведены в гистограмму.

С 1955 г. и до первой половины восьмидесятых годов общее количество публикаций неуклонно растет, затем следует устойчивый спад в течение около 15 – 20 лет. Можно предположить, что спад связан с эйфорией от успехов микропроцессоров, которые, казалось, решают все проблемы. Но на грани тысячелетий ситуация в корне изменяется и число публикаций за одно пятилетие увеличивается более, чем вдвое. По-видимому, успехи микропроцессоров вывели в область массовой продукции ранее экзотические приме-

нения: криптографию, обработку сигналов и изображений и т.п., в которых возможности традиционных микропроцессоров не всегда удовлетворяют. Это вызвало интерес к иным способом обработки информации, в т.ч. и к МА.



Эти соображения относятся к общему числу публикаций, а также к их литературной форме (статьи, доклады, монографии). Аналогичное распределение отмечается и для патентов, но здесь пик сдвинут более, чем на пятилетие и падает на вторую половину восьмидесятых годов, совпадая с минимумом активности на конференциях. Возможно, чувствуя спад интереса к МА, исследователи старались, на всякий случай, закрепить за собой приоритет по ранее выполненным наработкам.

Внедрение результатов

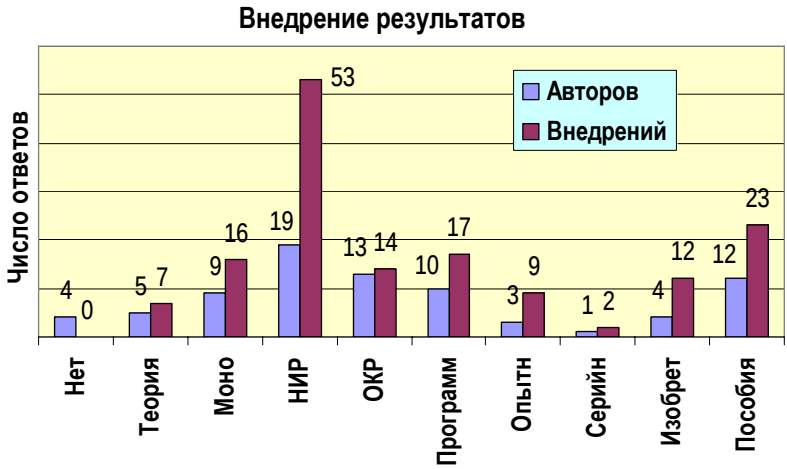
Итак. Теоретических работ по МА было достаточно много. Каковы же практические результаты? Для ответа на этот важнейший вопрос участников конференции попросили сообщить «*Реальные применения результатов Ваших трудов и трудов Ваших учеников по модулярной арифметике (внедренные и не внедренные проек-*

ты)». Три участника конференции честно признались, что у них нет реальных применений. Семь человек, ответив на анкеты, этот вопрос оставили без внимания. А 31 человек, сообщив, что применения есть, не сообщили конкретную информацию о реальных применениях результатов своих работ. Они ограничидись сообщениями типа «использованы в стольких-то НИР или ОКР», «использованы в изделии предприятия такого-то», «внедрение принесло такую-то прибыль», «применение в специзделии» и т.п. Авторам же хотелось наглядно показать реальные применения модулярной арифметики понятным любому специалисту образом. В результате многократных общих и индивидуальных обращений, кое-что удалось буквально «наскрести». Результаты сведены в диаграмме и в размещенном ниже разделе **«Внедрение результатов научных разработок в странах СНГ»**. В диаграмму включены и ответы типа выше приведенных, но исключены ответы типа «результаты реально применены в диссертации», были и такие. Судить о степени достоверности отраженного в диаграмме обилия реальных применений авторы обзора не берутся, но сочли целесообразным опубликовать полученный результат без особых комментариев.

В диаграмме рассмотрено 8 вариантов внедрения модулярной арифметики:

1. В научных концепциях, **теория**х и т.п.
2. В **моно**графиях.
3. В прикладных **НИР**.
4. В прикладных **ОКР**.
5. В нашедших применения пакетах **программ**.
6. В макетных или **опытн**ых приборах.
7. В **серийн**ой аппаратуре.
8. Использование **изобрет**ений в реальных разработках.
9. В учебных материалах и **пособия**х для ВУЗ-ов.

В диаграмму не включены несколько известных авторам старых разработок модулярных ЭВМ, выполненных не участниками конференции Д.И. Юдицким (ЭВМ Т-340А, К-340А, Алмаз и 5Э53), Б.С. Гаспером (бортовая ЭВМ) и Е.К. Лебедевым (Вычет-1 и Вычет-2). Но в разделе «Внедрение результатов научных разработок в странах СНГ» они, в меру имеющейся информации, представлены.

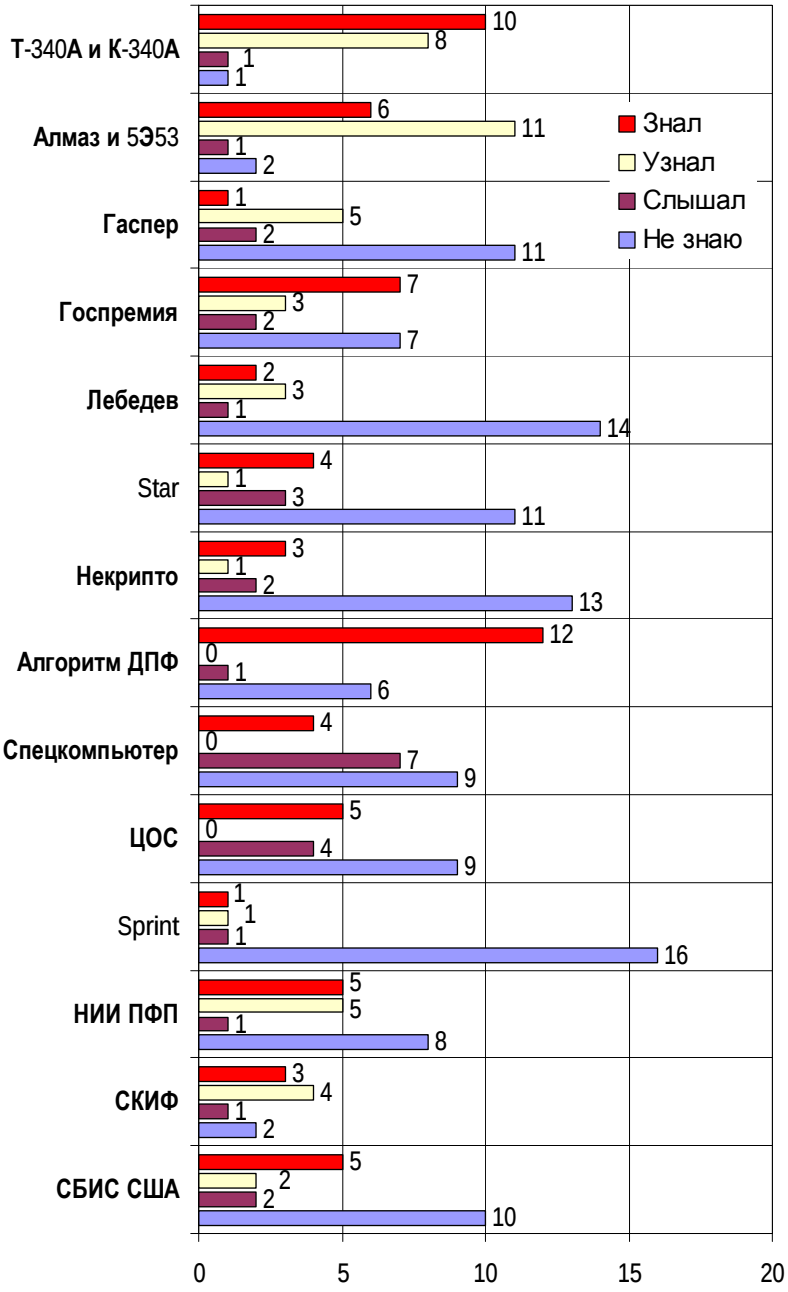


Информированность о применениях МА

В качестве показателя степени благополучия в научном направлении можно рассматривать уровень осведомленности работающих в этом направлении о результатах деятельности коллег и конкурентов. Участникам конференции был задан вопрос: «*Какие факты реального применения модулярной арифметики (внедренные и не внедренные проекты, аппаратура, микросхемы, изобретения, алгоритмы, программы и т.п.) в стране и в мире Вам известны (в т.ч. те, достоверность которых требует проверки)? Источники информации*». Как и в ответах по реальному применению результатов собственных работ, здесь было много «пустых» ответов, типичный из них: «*известны факты реализации алгоритмов ДРФ*», но о самих фактах ни слова. На этот вопрос вообще ответило только 20 экспертов. Но все-таки по результатам первого анкетирования удалось выявить 14 эпизодов, которые с разной степенью вероятности можно отнести к случаям реального применения МА:

1. Модулярные ЭВМ **Т-340А и К-340А**.
2. Модулярные ЭВМ **Алмаз и 5Э53**.
3. Бортовой компьютер управления авиационным двигателем Б.С. **Гаспер**а.
4. Проект А.А. Коляды, А.Ф. Чернявского, В.М. Амербаева и В.Г. Евстигнеева, отмеченный **Госпремией**.
5. Модулярные цифровые фильтры Е.К. **Лебедев**а.

Информированность о применениях МА



6. Бортовой компьютер ракеты «**Star**».
7. **Некрипто**графические методы защиты информации.
8. Теоретический аппарат реализации **алгоритма ДПФ**.
9. Военные **спецкомпьютер**ы США.
10. Спецпроцессоры **ЦОС** в США.
11. Компьютеры для роботов США, таких, как **Sprint**.
12. Модулярные спецпроцессоры ДПФ и др. (**НИИ ПФП**, Минск).
13. Программная реализация 1008-точечного ДПФ для суперкомпьютера **СКИФ** (НИИ ЭВМ, Минск).
14. Создание **СБИС** для модулярный устройств в **США**.

О первых в мире модулярных ЭВМ А. Свободы (Чехословакия) **ЭПОС 1** и **ЭПОС 2** вопроса не задавалось.

График неопровержимо показывает, что уровень осведомленности участников конференции весьма низок. В целом ответы распределились так:

- Знал – 70,
- Узнал – 45,
- Слышал – 32,
- Не знаю – 125.

И объяснить такую неосведомленность закрытостью работ не удается, т.к. о самых из закрытых в свое время ЭВМ Т-340А и К-340А заранее знала половина ответивших. По-видимому дело в разобщенности.

Причины незначительности применения МА

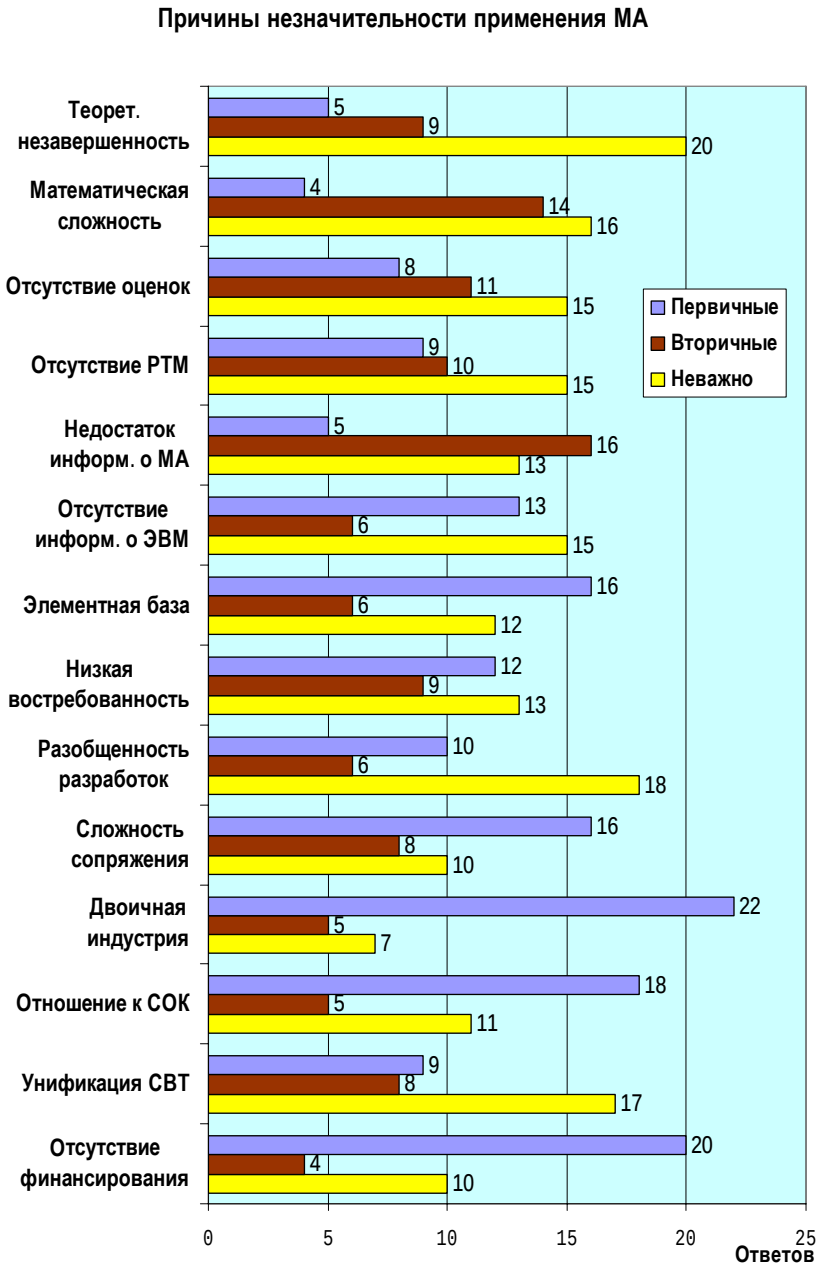
После триумфального старта в ЭВМ Т-340А, К-340А, Алмаз и 5Э53 модулярная арифметика практически утратила все свои позиции и в течение многих лет практического применения в СССР не находила. Объяснять это исключительно кознями чиновников бессмысленно, следует понять действительные объективные причины резкого и устойчивого спада применения МА. В первом анкетировании эксперты назвали много причин тому, которые удалось обобщить в 14 пунктах:

- **Теоретич**еская **незавершенность** МА.
- **Математическая сложность** модулярной арифметики, отсутствие популярной “легкой” литературы в этой области.

- **Отсутствие** сравнительных **оценок** вычислительной сложности алгоритмов модулярной и двоичной арифметик и их практических программных и аппаратных реализаций.
- **Отсутствие** соответствующих информационных и руководящих технических материалов (**РТМ**) по проектированию модулярных устройств, обобщающих накопленный опыт.
- **Недостаток** доступной **информ**ации **о МА** и выполняемых в этой области работах.
- **Отсутствие**, до недавних пор, доступной **информ**ации **о** реальных разработках модулярных устройств, **ЭВМ** и систем.
- Отсутствие специальной **элементной базы**.
- **Низкая востребованность**, до недавних пор, в задачах эффективного применения МА (ЦОС, обработка сигналов и т.п.).
- **Разобщенность разработок** в области МА.
- **Сложность сопряжения** модулярных устройств и программ с двоичными.
- Подавляющее распространение **двоичной индустрии**.
- **Отношение** специалистов, руководителей и ВУЗ-ов **к СОК** как к некоторой экзотической системе счисления, не имеющей практического значения.
- Высокий уровень **унификации** и стандартизации современных средств вычислительной техники (**СВТ**), препятствующий всему новому.
- **Отсутствие финансир**ования фундаментальных исследований и прикладных разработок в области МА.

В сумме на обе анкеты ответило 34 эксперта, результаты представлены в диаграмме. Представляется, что это одна из самых интересных и полезных диаграмм. Действительно, если понятны причины недостатков – понятны и методы борьбы с ними. Диаграмма дает большое поле для размышлений, остановимся на некоторых аспектах, которые авторам представляются интересными.

Бросается в глаза преобладание мнений о малой значимости для большинства названных причин, над мнениями об их первостепенном влиянии на применяемость МА. Исключением являются только причины, связанные с влиянием традиционной вычислительной техникой, и с финансированием. Не очень волнуют экспертов причины, затрудняющие другим специалистам вхождение в МА.


Причины незначительности применения МА
Теорет. незавершенность
5
9
20
Математическая сложность
4
14
16
Отсутствие оценок
8
11
15
Отсутствие РТМ
9
10
15
Недостаток информ. о МА
5
16
13
Отсутствие информ. о ЭВМ
13
6
15
Элементная база
16
6
12
Низкая востребованность
12
9
13
Разобщенность разработок
10
6
18
Сложность сопряжения
16
8
10
Двоичная индустрия
22
5
7
Отношение к СОК
18
5
11
Унификация СВТ
9
8
17
Отсутствие финансирования
20
4
10
Первичные
Вторичные
Неважно
0
5
10
15
20
25
Ответов

Интересно, что на элементную базу, т.е. на себя, уже сетуют вдвое больше, 19 экспертов, 16 из них относят эту причину к важнейшим, 6 – к вторичным. 13 экспертов к важнейшим причинам отнесли неосведомленность о других реальных разработках. Интересно, что 7 из них указавши, что применения МА у них имеются, так и не раскрыли их, т.е. опять на себя же и посетовали.

Авторам представляется, что отсутствие четких и понятных любому инженеру данных о классах задач, на которых МА эффективна с количественной оценкой этой эффективности, является одним из решающих тормозов применения МА. Ведь пока потребители не поймут, какие выгоды они получат от МА, они не закажут соответствующий прибор. Эту точку зрения разделило только 8 экспертов (23%), еще 11 (32%) оценили эту причину, как вторичную. А 15 экспертов (44%) сочли ее несущественной. А ведь именно отсутствие таких общедоступных оценок является причиной низкой востребованности МА, которую признали 21 эксперт (62%). И именно поэтому в научной и технической общественности существует стойкое предубеждение против МА, которое отметило 23 эксперта (68%).

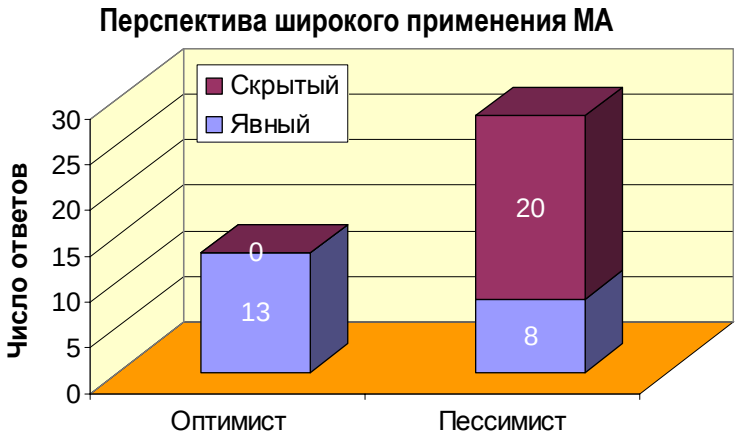
Перспективы широкого применения МА

То, что реальное применение МА в СНГ весьма незначительно, это факт, установленный в т.ч. и в результате экспертного опроса участников конференции. Так почему же они продолжают ею заниматься? Интересно, как они оценивают перспективы широкого применения МА. Экспертам был задан прямой вопрос о времени начала широкого применения МА. Рассматривалось два варианта ответа:

1. **Пессимист**ический – перспективы в ближайшее время невысоки,
2. **Оптимист**ический – ситуация для широкого применения МА созрела.

Всего на этот вопрос ответило 25 человек, но определенно высказалось только 21 и оптимистов среди них не так уж и много, всего 13, т.е. 32% всех экспертов. Если тех, кто уклонился от ответа на этот вопрос, отнести к скрытым пессимистам (а иначе уклонение от

ответа объяснить трудно), то пессимистов окажется 28 (41-13=28), т.е. 68% экспертов. Интересное отношение людей к перспективности дела, которым они занимаются.

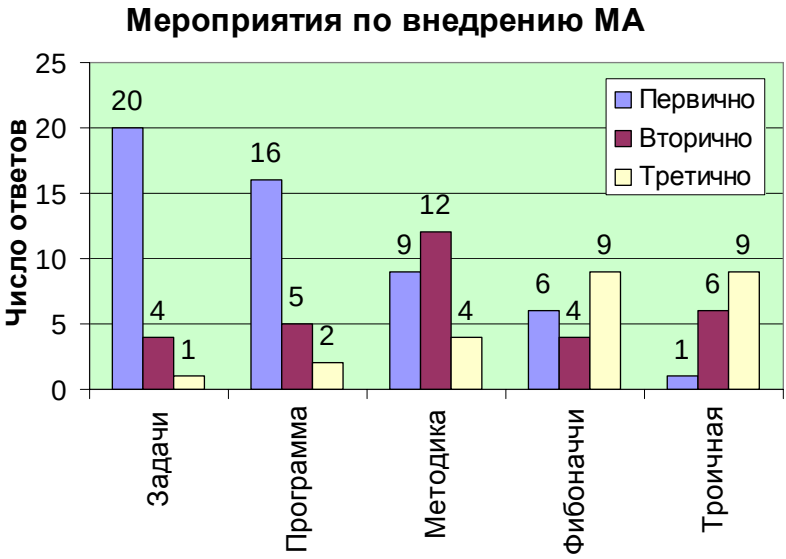


Мероприятия для расширения применения МА

Интересно было узнать, что, по мнению участников конференции, необходимо сделать, что бы МА получила достойное применение. По результатам первого анкетирования было сформулировано 6 мероприятий:

1. Выделение эффективно решаемых МА классов **задач** и целенаправленное совершенствование алгоритмов их решения.
2. Объединение всех работ по МА в единую **программу** с четкой координацией работ.
3. Подготовка учебно-**метод**ических материалов для ВУЗов.
4. Синтез МА с системами счисления с иррациональными основаниями (коды **Фибоначчи** и золотой пропорции)
5. Синтез МА с **троичной** системой счисления.

На этот вопрос ответило 25 экспертов. 80% из них к первостепенной относят задачу целенаправленного совершенствования алгоритмов решения выигрышных для МА задач, а 64 % большое значение придают четкой координации работ по МА. Большинство экспертов не придают большого значения синтезу МА с другими системами счисления. Возможно из-за недостаточной изученности этого вопроса, возможностей и последствиях такого синтеза.



А изучить эти варианты имеет смысл. Во-первых это «не паханая целина», а во-вторых это стык наук, а именно на стыках создается что-то новое (хотя можно и «провалиться»). Например, в симметричной троичной системе счисления знак числа представляется естественным образом, так же естественно выполняются и операции, без дополнительных кодов и прочих проблем. Возможно, это позволит перевести вычитание из немодульных операций в модульные. А вслед за ней и деление. Если это удастся (что не факт), то положение МА в корне изменится – ведь проблемы с немодульными операциями, по оценке этих же экспертов, являются главным недостатком МА, главным ограничителем сфер ее эффективности. Стоило бы попробовать.

Значимые труды

На вопрос «*Какие из трудов других ученых по модулярной арифметики Вы считаете наиболее значимыми (в порядке убывания приоритета)?*» содержательно ответило только 5 экспертов. Еще 14 просто перечислили фамилии видных ученых, т.е. по существу второй раз ответили на другой вопрос (о лидерах). 11 экспертов на этот вопрос не ответили, ограничились прочерком. Авторы воздержатся от комментариев по этому вопросу. А наиболее значимые

труды приведены в отдельном разделе в начале библиографии.

О CD-сборнике трудов по МА

Оргкомитет выступил с предложением подготовки сборника трудов по модулярной арифметике на CD-ROM и задал участникам конференции простой вопрос: «*Какие из трудов (статей и монографий, Ваших и других авторов) Вы считаете целесообразным включить в сборник на компакт-диске*?». Ответило всего 4 человека, причем один из них сказал, что предложений нет, другой – что нужно включить «труды всех без исключения авторов», третий предложил список трудов, цитированных в его статье, а четвертый предложил один из своих трудов. Остальные эксперты, заполняя анкету, этот вопрос оставили без внимания. Иными словами идея выпуска CD-сборника трудов по модулярной арифметике интереса участников конференции не вызвала и оргкомитетом оставлена.

•

ВНЕДРЕНИЕ РЕЗУЛЬТАТОВ РАЗРАБОТОК В СТРАНАХ СНГ

Рассмотрим четыре вида внедрения научных разработок в области модулярной арифметики:

- Технологии построения программных и аппаратных модулярных средств обработки информации.
- Аппаратная реализация модулярных средств обработки информации.
- Программная реализация модулярных средств обработки информации.
- Учебные пособия для высшей школы.

По каждому виду внедрения ниже приведена краткая информация участников конференции о полученных результатах.

Технологии построения программных и аппаратных модулярных средств обработки информации

Многофункциональные модульные средства

Коляда А.А., Чернявский А.Ф.
НИИ ПФП БГУ, г. Минск

Технология построения параллельно-кольцевых многофункциональных модульных средств нового поколения с перестраиваемой архитектурой на основе минимально избыточных модулярных вычислительных структур.

Многомашинная технология модулярной обработки

Коляда А.А., Чернявский А.Ф.
НИИ ПФП БГУ, г. Минск

Технологии модулярной обработки информации на основе многомашинных высокопроизводительных систем для автоматизации научных исследований и производственно-технологических процессов на базе минимально избыточных модулярных вычислительных структур.
Новая гибкая модулярная технология параллельных вычислений на позиционных ЭВМ.

Основы построения процессоров БПФ

Коляда А.А., Чернявский А.Ф.
НИИ ПФП БГУ, г. Минск

Теоретико-методологические, алгоритмические и структурные основы построения процессоров БПФ на основе композиционных процедур с применением минимально избыточной модулярной арифметики.

Библиотеки VHDL-описаний функциональных блоков модулярных нейромультипроцессорных ЦОС

Евдокимов А.А.
Невинномысский технологический институт

Библиотека содержит VHDL-описания основных функциональных блоков модулярного нейросопроцессора: один слой нейронной сети конечного кольца (НСКК) на основе комбинационного сумматора (FRNN_Perform), НСКК на основе однобитных нейронов (слой

frnnmain и сеть frnnfull), табличный преобразователь чисел из позиционного двоичного представления в СОК (pns2rns), преобразователь чисел из СОК в полиадическую систему счисления на основе табличной арифметики (rns2mrs).
Библиотека предназначена для построения процессоров цифровой обработки сигналов на основе теоретико-числовых преобразований, процессоров, функционирующих в системе остаточных классов, и процессоров других приложений модулярной арифметики на базе ПЛИС типа FPGA фирмы Xilinx.
На основе библиотеки разработан криптографический нейропроцессор, предназначенный для построения систем пролонгированной безопасности на базе модулярной пороговой схемы и принципа «блуждающих» ключей.

Методы кодирования структурированных цифровых данных для защиты от техногенных и природных помех

Инютин С.А.
Сургутский государственный педагогический институт

В рамках хозяйственных договоров с НПО «Геофизика», НИИ «Проблем моделирования в энергетике» Украины разрабатывались методы кодирования структурированных цифровых данных для защиты от техногенных и природных помех при передаче по каналам связи и долговременном хранении на магнитных носителях последовательного доступа, а также для борьбы со сбоями в автоматизированной системе обработки данных в режиме «off line». В рамках хозяйственных договоров с проектным институтом «Сейсморазведка» Казахстана разрабатывались методы и математическое обеспечение для систем передачи аналоговой и цифровой информации в условиях природных и техногенных помех с целью защиты в режиме «on line» СПД прогностического полигона с сетью взаимосвязанных автоматизированных пунктов наблюдения за изменением во времени вариаций геофизических полей и сейсмического режима. В основе методов лежит кодирование разделимыми слабо-арифметическими модулярными кодами, с контрольной частью формируемой линейными формами по модулю. Для методов доказана эффективность в условиях пакетных асимметричных ошибок в каналах передачи данных. Методы защищены изобретениями СССР.

Методы перепрограммирования нескольких типов или поколений «однородных вычислительных сред»

Инютин С.А.
Сургутский государственный педагогический институт

В рамках хозяйственных договоров с НПО «Астрофизика», «Алмаз» разрабатывались методы перепрограммирования нескольких типов или поколений «однородных вычислительных сред», содержащих множество элементарных процессорных элементов, для задач распознавания образов на основе модификаций быстрого преобразования Фурье в процессорах летательных аппаратов в режиме «on line» с жесткими ограничениями на время реакции. Методы и программное обеспечение, отлаженное на эмуляторах «однородных вычислительных сред» на ЕС ЭВМ, оформлены как технические предложения при проектировании специализированных процессоров. Обоснован технический эффект методов - повышение скорости сравнения образов оригинала и образца за счет меньшей алгоритмической сложности обработки в модулярном варианте, учитывающем архитектурные особенности «однородных вычислительных сред»

Обоснование модулярных арифметико-логических форм

Финько О.А.
Краснодарское высшее военное училище

Выполнен ряд НИР, посвященных обоснованию модулярных арифметико-логических форм для обеспечения технической реализации криптографических алгоритмов защиты информации в перспективных автоматизированных системах управления.

Анализ тенденций развития защищенных АСУ выявил конфликтную ситуацию, когда требования к существенному возрастанию объемов и скорости обрабатываемой и передаваемой информации сталкиваются с низкой производительностью и функциональной гибкостью средств криптографических преобразований. АСУ и средства криптографической защиты как правило разрабатываются независимо друг от друга и поэтому часто оказываются несогласованными по требуемым техническим возможностям. Средства криптографической защиты как правило существенно ОТСТАЮТ от перспективных технических требований. Однако условие использования их согласно тех. задания ОБЯЗАТЕЛЬНО.

Существующие средства криптографический преобразований яв-

ляются своего рода «пробками» в каналах связи, не позволяющими развить потенциальные возможности, уже заложенные в, как правило, более прогрессивную аппаратуру, входящую в состав проектируемой АСУ. Ярким примером этому является массовое внедрение средств съема, обработки и передачи видеоинформации, требующих широкополосных каналов связи и средств цифровой обработки. Данное обстоятельство отягощается так же тем, что анализ структуры защищенных АСУ выявил, что криптографические преобразования относятся к самому массовому виду преобразований. Вследствие прохождения через различные звенья управления информация криптографическом циклам преобразованиям подвергается многократно. Кроме того использование различными ведомствами собственных (разных) криптоалгоритмов, постоянное совершенствование (замена) и необходимость модернизации в будущем требует высокой функциональной гибкости от средств технической реализации криптоалгоритмов.

Использование модулярных арифметико-логических форм в перспективных защищенных АСУ позволяет обеспечить:

1) высокий параллелизм (производительность) криптопреобразований,
2) высокую функциональную гибкость (инвариантность к реализуемому криптоалгоритму),
3) необходимый уровень достоверности функционирования, что обеспечивает безопасную эксплуатацию таких средств, а так же согласованность технических характеристик перспективной аппаратуры и каналов связи АСУ и средств криптографической защиты информации.

•

Аппаратная реализация модулярных средств обработки информации

Экспериментальная модулярная ЭВМ «Т-340А»

Юдицкий Д.И., Акушский И.Я.
НИИ-37 (ныне ОАО «НПК НИИДАР»), Москва

В 1960-1963 гг. в НИИ-37 была разработана первая в стране (а возможно и в мире, единственный конкурент – ламповая ЭВМ «ЭПОС 1» А. Свободы создана в 1958-63 гг.) реально работавшая полупроводниковая модулярная ЭВМ Т-340А, гл. конструктор Д.И. Юдиц-

кий. Она предназначалась для полигонного варианта РЛС Дунай-3УП системы противоракетной обороны (ПРО) А-35. Теория и практика варианта модулярной арифметики, принципы построения ЭВМ на их основе были разработаны И.Я. Акушским, Д.И. Юдицким и Е.С. Андриановым. Это была экспериментальная ЭВМ, изготовленная, отлаженная и решающая реальные задачи по своему назначению. Т-340А проработала несколько лет в эксплуатационном режиме в составе полигонного варианта системы А-35 – комплекса «Алдан» до его демонтажа.

Модулярные ЭВМ «EPOS» и «EPOS 2»

В. Грегор, А. Свобода.
Центральный институт математики, Прага

ЭВМ «ЭПОС» (Электронный почитач) была разработана в Чехословакии под руководством В. Грегора и А. Свободы в 1958-1963 гг. Она была построена на основе ламповых усилителей, логики на германиевых диодах, регистров на никелевых линиях задержки и памяти на ферритовых сердечниках.

Основное назначение ЭПОСа – обработка данных. ЭПОС имел модульную структуру, состоящую из базового компьютера, который мог быть оснащен различными периферийными устройствами, в соответствии с требованиями конкретного пользователя.

ЭПОС представлял собой десятичный одноадресный последовательно-параллельный компьютер, оперирующий со словами в 12 десятичных разрядов.. В центральном процессоре, отличающемся высокой надежностью вычислений (с автоматическим нахождением и исправлением ошибок) операции десятичного сложения и умножения выполнялись в системе остаточных классов СОК, а для деления использовался новый алгоритм А. Свободы. ЭПОС выполнял сложение за 52 мкс, умножение за 208 мкс и деление за 1196 мкс, с учетом времени обращения к памяти. ЭПОС мог быть оснащен сопроцессором, выполняющим операции с плавающей запятой. Набор запоминающих устройств содержал расширяемое ЗУ на ферритовых сердечниках с базовой емкостью 2000 слов, а так же внешние ЗУ на магнитных барабанах до 50000 слов, на магнитных лентах и на магнитной проволоке.

Система была оснащена встроенными аппаратными средствами для выполнения пяти полностью независимых программ, распределенных во времени.

ЭВМ «ЭПОС 2», разработанная в 1964-65 гг., была транзисторной версией ЭПОСа с незначительными архитектурными модификациями.

Модулярная ЭВМ «К-340А»

Юдицкий Д.И., Акушский И.Я.
НИИ-37 (ныне ОАО «НПК НИИДАР»), Москва

В 1963-1966 гг. в НИИ-37, на основе опыта проектирования и применения экспериментальной ЭВМ Т-340А, была разработана первая в стране (а возможно и в мире) серийно выпускаемая модулярная ЭВМ Т-340А, гл. конструктор Д.И. Юдицкий. Она предназначалась РЛС Дунай-3У системы ПРО А-35. ЭВМ была освоена в серийном производстве и стала базовой для всех РЛС, разрабатываемых в те годы в НИИ-37. Опытным заводом при НИИ-37 и Свердловским заводом радиоаппаратуры было выпущено более 50 ее комплектов. Благодаря высочайшей надежности и уникальным характеристикам ЭВМ К-340А до сих пор (2006 г., 40 лет !!!) находятся в эксплуатации, демонстрируя значительно более высокую живучесть, чем работающие рядом с ними другие, современные им, электронные системы.

В ЭВМ Т-340А и К-340А впервые в стране был реализован принцип независимых каналов памяти команд и данных. Оперативная память была выполнена в виде 16 блоков емкостью по 1К слов. Каждый блок имел по два порта для ввода–вывода информации: с абонентами (с возможностью параллельного обмена с любым числом блоков) и с процессором. Для увеличения быстродействия было реализовано программное расслоения оперативной памяти с чередованием обращения процессора к блокам. Кроме того, была применена многовходовая буферная память для двухоперационных команд (в каждой команде выполнялось по две операции, каждая из которых в других ЭВМ того времени выполнялась в виде отдельной команды). Эти особенности построения системы памяти обеспечили высокую эффективность ЭВМ К-340А: задержек при обращении к памяти большого объема (бич ЭВМ тех лет) практически не было.

ЭВМ Т-340А и К-340А обладали невиданным в те времена быстро-

действием в 1,2 млн. двойных оп/с. или 2,4 млн. обычных оп/с. Типовое быстродействие ЭВМ в те времена измерялось десятками или сотнями тысяч оп/с. Это первая в мире ЭВМ с быстродействием более 1 млн. оп/с. И это была ЭВМ с самой низкой стоимостью единицы производительности – 25 коп за 1 оп/сек. Опытным заводом при НИИ-37 и Свердловским заводом радиоаппаратуры было выпущено более 50 ее комплектов. Только в РЛС "Дунай-3У" работало 10 ЭВМ "К-340А". Благодаря высочайшей надежности и уникальным характеристикам ЭВМ К-340А до сих пор (2005 г., 40 лет!!!) находятся в эксплуатации, демонстрируя значительно более высокую живучесть, чем работающие рядом с ними другие, современные им, электронные системы.

Экспериментальная модулярная ЭВМ «Алмаз»

Юдицкий Д.И., Акушский И.Я.
Центр микроэлектроники, Москва, Зеленоград

В 1965 г. трем организациям: ЦМ (МЭП, Ф.В. Лукин), ИТМ и ВТ (МРП, С.А. Лебедев) и ИНЭУМ (Минприбор, М.А. Карцев) было дано конкурсное задание на разработку и для системы ПРО «Аврора», и для Многоканального стрельбового комплекса «Аргунь» (МКСК) второй очереди системы ПРО «А-35» было дано конкурсное задание на разработку эскизных проектов высокопроизводительной ЭВМ со сроком окончания 30 марта 1967 года.

Проект ЭВМ был разработан с изготовлением экспериментального образца. Это был первый в стране проект ЭВМ на основе ИС, поскольку первые ИС разрабатывались здесь же и первые их образцы шли на изготовление Алмаза.

Алмаз имел процессоры трех типов: процессор предварительной обработки сигналов (тогда это было новинкой), модулярный процессор обработки радиолокационных сигналов, и двоичный процессор, в основном, для управления работой ЭВМ. Общая производительность Алмаза составляла около 7,5 млн. алгоритмических оп/с или до 30 млн оп/с в общепринятом тогда исчислении (одна алгоритмическая операция на задачах МКСК соответствовала примерно $3 \div 4$ простейшим операциям ЭВМ).

Конкурс выиграл проект «Алмаз», в результате Научный центр (Минэлектронпром) получил заказ на разработку ЭВМ «5Э53» с организацией серийного производства в Загорском электромеханическом заводе (Минрадиопром).

Модулярная ЭВМ «5Э53»

Юдицкий Д.И., Акушский И.Я.

Специализированный вычислительный центр, Москва, Зеленоград

В 1969-1971 гг. в Специализированном вычислительном центре НЦ (Зеленоград) была разработана модулярная ЭВМ «5Э53» с изготовлением опытного образца. Это была 8-процессорная ЭВМ (4 модулярный и 4 двоичных процессора) с общей производительностью около 10 млн. алгоритмических (40 млн. обычных) оп/с.

Архитектура 5Э53 имела много принципиально новых решений:

- Разделение команд на управленческие и арифметические. Арифметические команды (в т.ч. предварительная и основная обработка сигналов) выполнялись на модулярных процессорах, управленческие – на двоичных.
- 8-уровневая конвейерная организация.
- Аппаратная блочная реализация арифметики: блок сложения/вычитания, блок умножения, блок управления адресами и т.п.),
- Разделение памяти на оперативную данных и полупостоянную команд,
- Разделение шин команд и данных,
- Аппаратное расслоение памяти на 8 блоков с чередующейся адресацией по блокам и т.п.

ЭВМ была разработана, изготовлена, настроена, проведены типовые испытания ЭВМ и всех ее ячеек, субблоков и блоков, проведена корректировка документации по результатам испытаний. Документация была передана на Загорский электромеханический завод, проведена подготовка производства более, чем на 70% и начато изготовление отдельных устройств. Но в это время работы над второй очередью системы ПРО А-35, для которой 5Э53 разрабатывалась, были прекращены, финансирование работ по 5Э53 остановлено. Другого заказчика и изготовителя для 5Э53 не нашлось, а проект уникальной ЭВМ был погублен.

Спецпроцессоры прямого вычисления АФК «Вычет-1, -2»

Лебедев Е.К.

Чувашский государственный университет по информатизации, г. Чебоксары

Модулярные спецпроцессоры автокорреляционной функции (АКФ) «Вычет-1» и «Вычет-2», разработанные под руководством

Е.К. Лебедева, предназначены для применения в системах обработки сигналов и изображений. Для вычисления АКФ в них используется прямое и обратное дискретное преобразование Фурье.

Спецпроцессоры «Вычет-1» и «Вычет-2» – это устройство со своим модулем АЛУ и умножителя; шинами данных и микрокоманд; управляющих и операционных устройств; АЦП; интерфейсов ввода и вывода. Операционное устройство спецпроцессора "Вычет-1" построено на основе ПЛМ с применением БИС К1518ВЖ1 и К1804ВС1.

Спецпроцессоры «Вычет-1» и «Вычет-2» были удостоены серебряной медали ВДНХ СССР в 1989 г. Эти изделия нашли применение в системах радиолокации.

Спецпроцессоры «Вычет-1» и «Вычет-2» использовались для решения задач распознавания изображений в системах радиолокации (вычисление корреляционной функции).

Оргкомитет Конференции с прискорбием сообщает, что ведущий разработчик спецпроцессоров «Вычет 1» и «Вычет 2», проректор ЧувГУ Лебедев Евгений Константинович 25 мая 2005 г. безвременно ушел из жизни, не успев завершить подготовку материалов для конференции о результатах своих работ в модулярной арифметике. А это далеко не только указанные спецпроцессоры. Оргкомитет неоднократно обращался в ЧувГУ с просьбой о подготовке для конференции материалов по работам Е.К. Лебедева, в т.ч. трижды лично к ректору Л.П. Куракову. Но вместо того, чтобы предпринять некоторые усилия для достойного представления результатов научных трудов своего коллеги, ректор, не сообщив об этом оргкомитету, свалил проблему на вдову Е.К. Лебедева, которая еще ранее любезно передала всю известную ей информацию.

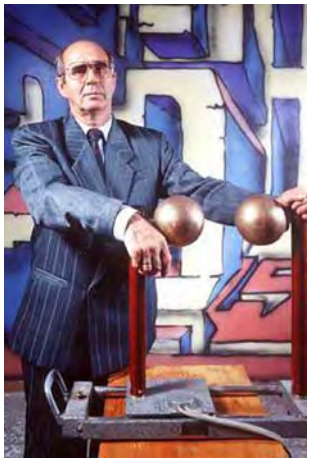
Модулярная бортовая управляющая ЭВМ

Гаспер Б.С.

Пермский государственный технический университет, г. Пермь

Примерно в 1973-1975 гг. в г. Перми, по-видимому на Пермском моторной заводе, коллективом во главе с Борисом Сер-

геевичем Гаспером была разработана модулярная бортовая управляющая ЭВМ для управления режимами работы авиационными двигателями. Ее образец был изготовлен, настроен и прошел испытания на одном из авиадвигателей, но по каким-то причинам не был принят в серийное производство. Борис Сергеевич в конце 2004 г. горячо откликнулся на просьбу автора и обещал дать информацию об этой разработке для истории ращвития модулярной арифметики, но не успел.



Оргкомитет Конференции с прискорбием сообщает, что ведущий разработчик модулярной бортовой ЭВМ, декан электротехнического факультета Пермского государственного технического университета Гаспер Борис Сергеевич 16 января 2005 г. безвременно ушел из жизни. Оргкомитет неоднократно обращался в ПГТУ с просьбой о подготовке для конференции материалов по работам Б.С. Гаспера в модулярной арифметике, в т.ч. дважды лично к ректору В.Ю. Петрову. Наконец был получен ответ от проректора ПГТУ по учебной работе Н.Н. Матушкина, что институт «*не располагает возможностями в подготовке материалов по результатам его работ*».

Средства криптографической защиты сигналов на основе модулярного преобразования

Коряков И.В.
ООО НВФ «КРИПТОН», г. Киев

Метод защиты сигналов на основе модулярного преобразования реализован как один из режимов в изделиях «КРИПТОН-4М7» и «СЕКМОД-К» предприятия ООО НВФ «Криптон».

Достоинством модулярного преобразования, по сравнению с вокодерными системами, является восстановление на приемной стороне “чистого” речевого сигнала, т. е. не подвергнутого сжатию, что позволяет получить наивысшие показатели разборчивости и естественности речи.

Помимо речевых сигналов, изделие «КРИПТОН-4М7» закрывает при помощи модулярного преобразования сигналы модемных и

факсимильных передач со скоростью до 9600 бит/с.

Измеритель частоты импульсных сигналов на основе системы остаточных классов

Коряков И.В.
ООО НВФ «КРИПТОН», г. Киев

Метод измерения частоты сигнала на основе системы остаточных классов реализован в приемнике импульсных сигналов в диапазоне 40 – 3094 МГц с разрешением по частоте 1 МГц. Применение СОК позволило радикально уменьшить число каналов обнаружения и перенести обработку в низкочастотную область спектра, что снизило требования к быстродействию элементов и в десятки раз уменьшило объем оборудования по сравнению с существующими параллельными приемниками.

Двумерный процессор ДПФ

Коляда А.А., Чернявский А.Ф.
НИИ ПФП БГУ, г. Минск

Двумерный процессор для выполнения (128x128)-точечного ДПФ, построенный на основе минимально избыточной модулярной арифметики и соответствующей структуры арифметического устройства, включая блоки немодульных операций.
Макетный образец используется в качестве периферийного процессора ПЭВМ для обработки информации томографа.

Процессор адаптивной КИХ-фильтрации

Коляда А.А., Чернявский А.Ф.
НИИ ПФП БГУ, г. Минск

Процессор для адаптивной КИХ-фильтрации, построенный на основе минимально избыточной модулярной арифметики и структуры арифметического устройства, включая блоки немодульных операций.
Используется в системе обработки измерительной информации (НИИ ПФП))

Процессор ДПФ

Коляда А.А., Чернявский А.Ф.
НИИ ПФП БГУ, г. Минск

Процессор для реализации 1260-точечного ДПФ по алгоритму Винограда, построенный на основе минимально избыточной моду-

лярной арифметики и структуры арифметического устройства, включая блоки немодульных операций.
Используется в системе обработки измерительной информации (НИИ ПФП)

Процессор дискретных сигналов

Коляда А.А., Чернявский А.Ф.
НИИ ПФП БГУ, г. Минск

Процессор для вычисления чебышевских ортогональных проекций дискретных сигналов, построенный на основе минимально избыточной модулярной арифметики и структуры арифметического устройства, включая блоки немодульных операций.
Применяется в системе управления, предназначенной для оперативного управления космическими средствами наблюдения (ИМПБ РАН).

Элементы процессора ЦОС

Тынчеров К.Т.
Ставропольский филиал Краснодарской академии МВД России

Все три ОКР выполнялись по специальному заказу Воронежского научно-исследовательского института (ВНИИС) по закрытой тематике. Суть разработки заключается в синтезе сверхскоростных СБИС, выполняющих обработку широкополосных сигналов на основе КИХ-фильтров. Таким образом, были решены задачи, связанные с разработкой аналогов СБИС серии TMS-320, удовлетворяющих скоростным и другим специальным требованиям цифровой обработки сигналов в аппаратуре автоматизированных систем боевого управления и связи.

•

Программная реализация модулярных средств обработки информации

Многомашинная система модулярной обработки информации

Коляда А.А., Чернявский А.Ф., Коляда Н.А.
НИИ ПФП БГУ, г. Минск

Разработанные на основе минимально избыточной модулярной арифметики системный программный модуль для четырёхмашинной системы модулярной обработки информации.

Используются в пакете прикладных программ на супер-ЭВМ “СКИФ” (НИИ ЭВМ; г. Минск), созданной в рамках НТП Союзного государства (2000 – 2003гг.).

Многомашинная система модулярной обработки информации

Коляда А.А., Чернявский А.Ф., Коляда Н.А.
НИИ ПФП БГУ, г. Минск

Программа для реализации 1008-точечного ДПФ по алгоритму Винограда.

Используются в пакете прикладных программ на супер-ЭВМ “СКИФ” (НИИ ЭВМ; г. Минск), созданной в рамках НТП Союзного государства (2000 – 2003гг.). А также в дактилоскопической системе информационной безопасности (НИИ ПФП) для классификационного анализа отпечатков пальцев на этапе детектирования поля плотностей папиллярных линий.

Программный комплексе тестирования абитуриентов

Бияшев Р.Г., Нысанбаев Р.К., Егай Р.В.
Институт проблем информатики и управления МОН РК, Алма-Ата

Программа модулярного шифрования в комплексе тестирования абитуриентов осуществляет шифрование файлов с правильными ответами для их записи на магнитные носители и расшифровки при их считывании с обнаружением ошибок. Доступ к процедуре дешифрования производится в случае совпадения трех паролей. Алгоритмы шифрования и дешифрования построены на базе полиномиального варианта кодов Лагранжа.

Программный комплекс использован Республиканским центром тестирования Министерства образования и науки Республики Казахстан.

Программа для порогового разделения файла

Червяков Н. И., Евдокимов А. А.
Невинномысский технологический институт

Программа реализует способ порогового разделения и восстановления электронных документов в соответствии со схемой пространственного порогового разделения секрета на базе Китайской Теоремы об Остатках. Файл разделяется на n частей, по k частям из которых можно восстановить документ. Каждая часть файла содержит информацию об основаниях и величине порога, необходи-

мую при восстановлении файла. Программа позволяет выбрать общее n и пороговое k числа схемы, содержит генератор взаимно простых чисел из заданного диапазона, определяемый разрядностью оснований, который задается пользователем (максимальная разрядность равна 31 бит). Программный продукт предназначен для использования в системах криптографического порогового разделения секретных документов, системах динамического распределения нагрузки передачи данных между компьютерами в сети, системах надежного хранения информации. Использование программы для пространственного разделения критической информации позволило увеличить (с возможностью варьирования) надежность хранения при объеме ЗУ, требуемого резервным копированием, за счет того, что каждый остаток приблизительно в k раз меньше разделяемого документа. Так объем информации при резервировании с архивацией данных 10-ти копий потребовал емкость ЗУ в три раз меньше, чем без сжатия. При задании данной программой n = 18 и k = 6 для хранения был востребован тот же общий объем памяти, однако при потере 10 частей информации остается 8, из которых для восстановления данных требуется всего 6 частей. Наличие порога при восстановлении позволило сохранить данные от несанкционированного доступа при хранении отдельных частей информации на общедоступных ЭВМ.

Программа использована в ООО «Арнест – информационные технологии» для пространственного разделения критической информации.

Программный эмулятор модулярной арифметики

Макоха А.Н., Ионисян А.С.
Ставропольский государственный университет

Создана программа для ЭВМ для моделирования арифметических операций над целыми и рациональными числами, представленными в СОК. Разработана библиотека программ основных алгоритмов обработки данных, которая включает в себя ряд подпрограмм, эмулирующих: операции сложения, вычитания, умножения; алгоритмы деления с остатком; нахождение ортогональных базисов и их весов; вычисление следа и ранга числа; перевод чисел ППС в СОК и обратно; сравнение двух целых чисел в СОК; деление на 2; формальное деление; нахождение НОД и НОК двух чисел; вычисление факториала числа; арифметику обыкновенных дробей; перевод числа из СОК с одними основаниями в СОК с другими основания-

ми.
Используется в Ставропольском государственном университете.

Программа сравнения реализаций логических функций

Финько О.А.
Краснодарское высшее военное училище

Для исследования типовых узлов (блоков) криптоалгоритмов в среде программирования Delphi была разработана программа, позволяющая сравнивать время выполнения одних и тех же функций, реализованных логическими выражениями и арифметическими полиномами. Программа позволяет производить исследование коммутаторов и нелинейных устройств усложнения (блоков нелинейной замены).
В качестве *показателя эффективности* вычисления значений системы логических функций выбирается время, затраченное на вычисление значений функций на полном наборе аргументов. При этом функции задаются случайным образом с равномерным законом распределения.
Условия проведения эксперимента: ПЭВМ на базе процессора Intel Pentium III 650 МГц (Частота шины 100 МГц, кэш-память первого уровня — 32 Кбайт, кэш-память второго уровня — 512 Кбайт), ОЗУ 128 Мбайт.
Сравнение производилось с реализацией системы булевых функций полиномом Жегалкина. При числе реализуемых булевых функций при количестве булевых аргументов 4–6 выигрыш во времени вычислений составил приблизительно 2,5 раза.

Библиотека классов вычисления значения элементарных функций от аргументов, представленных обыкновенными дробями, в системе остаточных классов

Мезенцева О.С., Рыцев П.А.
ГОУ ВПО "СевКавГТУ"

Библиотека классов вычисления значения элементарных функций в системе остаточных классов включает в себя следующие компоненты:

) программную библиотеку, реализующую функции арифметического базиса, функционирующего в системе остаточных классов;
) программную библиотеку, реализующую вычисление значений следующих элементарных функций от аргументов, выра-

женных в системе остаточных классов: sin(x), cos(x), tg(x), sqrt(x), exp(x) , ln(x);

) прикладную программу, предназначенную для вычисления перечисленных выше функций с использованием указанных выше программных библиотек;

Программная библиотека, реализующая арифметический базис, позволяет оперировать с числами, представленными в системе остаточных классов, включая операции сложения, умножения, деления, вычитания выполняющихся в СОК. Для их реализации используется табличный метод выполнения арифметических операций. Программная библиотека предоставляет возможность как целочисленного, так и дробного представления чисел.

Прикладная программа позволяет вычислять значения выбранных элементарных функция с использованием разработанной программной библиотеки, имеет дружественный и интуитивно понятный интерфейс.

№ ГОС регестрации 2005610954

Программа безошибочного решения системы линейных уравнений методом Гаусса с рациональными коэффициентами

Оцоков Ш.А.
Московский энергетический институт

Разработанная на основе рациональной модулярной арифметики программа для решения системы линейных уравнений без ошибок округления в процессе вычислений. (Свидетельство гос. регистрации №2003610764 в реестре программ для ЭВМ / Оцоков Ш.А, Шухман И.М)

Используется при решении задач расчета установившихся режимов в ОАО «Дагэнерго» (г. Махачкала, 2004 г.)

•

Учебные пособия для высшей школы

Лабораторный практикум по теории электрической связи

Смирнов А.А., Баркетов С.В.
Ставропольский государственный университет

Применение модулярной арифметики в лабораторных работах по линейному кодированию по циклическим кодам и быстрому пре-

образованию Фурье. Общая публикация в журнале «Информационные технологии» 2005, № 6. В 2003 и 2004 годах и отмечен серебряными медалями ВВЦ.
Применяется в Ставропольском государственном университете в учебных курсах «Спецглавы алгебры и теории чисел», «Обработка данных в системе остаточных классов» и «Модулярные нейрокомпьютерные технологии».

Модулярные вычислительные структуры нейропроцессорных систем

Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А.
Ставропольский государственный университет

М.: Физматлит, 2003, 288 с. (грант Российского фонда фундаментальных исследований для проведения фундаментальных научных исследований по области знаний: «Параллельные вычислительные системы» шифр 07-211, издательский проект № 02-07-95001).
Применяется в Ставропольском государственном университете в учебных курсах «Спецглавы алгебры и теории чисел», «Обработка данных в системе остаточных классов» и «Модулярные нейрокомпьютерные технологии».

Алгебраические и теоретико-числовые основы модулярного кодирования

Макоха А.Н., Сахнюк П.А., Червяков Н.И.
Ставропольский государственный университет

Дискретная математика. Учебное пособие. – М: Физматлит, 2005, 358 с. (гриф УМО «Прикладная математика и физика», 5 глава – «Алгебраические и теоретико-числовые основы модулярного кодирования»).
Применяется в Ставропольском государственном университете в учебных курсах «Спецглавы алгебры и теории чисел», «Обработка данных в системе остаточных классов» и «Модулярные нейрокомпьютерные технологии».

Нейроматематика

Сахнюк П.А., Червяков Н.И. и др.
Ставропольский государственный университет

Учебное пособие / под ред. Галушкина А.И. – М.: ИПЖР, 2002, 447с.
Применяется в Ставропольском государственном университете в

учебных курсах «Спецглавы алгебры и теории чисел», «Обработка данных в системе остаточных классов» и «Модулярные нейрокомпьютерные технологии».

Нейрокомпьютеры в остаточных классах

Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н.
Ставропольский государственный университет

Учебное пособие. - М.: ИПРЖ “Радиотехника”, 2003, 272 с. (гриф УМО «Прикладная математика и физика»).

Применяется в Ставропольском государственном университете в учебных курсах «Спецглавы алгебры и теории чисел», «Обработка данных в системе остаточных классов» и «Модулярные нейрокомпьютерные технологии».

Применение нейрокомпьютеров для обработки сигналов

Червяков Н.И., Сахнюк П.А., Копыткова Л.Б. и др.
Ставропольский государственный университет

Коллективная монография/ под ред. Гуляева Ю.В. и Галушкина А.И. - М.: ИПРЖ “Радиотехника”, 2003, 224 с.

Применяется в Ставропольском государственном университете в учебных курсах «Спецглавы алгебры и теории чисел», «Обработка данных в системе остаточных классов» и «Модулярные нейрокомпьютерные технологии».

•

Информация участников конференции о применении СОК в странах дальнего зарубежья

СБИС для универсального преобразования, построенная по КМОП-технологии (1,25 мкм), имеет 132 вывода, оперирует с 16-битовыми числами. В статье: Meehan S.J., O’Neil S.D., Vascaro J.J. An universal input and output RNS converter // IEEE Trans. Circuits and syst. – 1990.- vol. 37, N6. – P. 799 – 803

Семейство СБИС для модулярных вычислительных устройств конвейерного типа и в частности, преобразователь (СБИС) 64-

битового кода в код МСС с 9 основаниями разрядностью 8 бит для использования в процессоре БПФ. В статье: G. Alia, E. Martinelli On the lower bound to the VLSI complexity of number conversion from weighted to residue representation // IEEE Trans on comput. – 1993. – vol.42, N8.- P.962 – 967

В памяти некоторых участников конференции сохранилась информация:

- О сообщениях в специальной печати 70-х годов прошлого века об использовании МА в бортовой ЭВМ аэрокосмической ракеты «STAR». Ракета STARS представляет собой модификацию баллистической ракеты морского базирования Polaris A3 с добавленной верхней ступенью Orbus-1. Доработку ракеты провела компания Lockheed Martin Missiles & Space.
- О создании СБИС для кодовых преобразований.

•

Интернет о модулярной арифметике

Авторы решили посмотреть, что о применении модулярной арифметики скажет Интернет. Ниже приведены результаты краткого исследования.

ИС для SmartCard

Одним из первых попалось сообщение ВНИИКИ: http://www.vniiki.ru/iso_details.asp?id=12379

Номер ISO:	**ISO/IEC 10118-4:1998**
Заглавие русское:	Информационные технологии. Методы обеспечения защиты. Хеш-функции. Часть 4. Хеш-функции с применением арифметики в остаточных классах
Заглавие английское:	Information technology - Security techniques - Hash-functions - Part 4: Hash-functions using modular arithmetic
Статус:	Действует

Регистрация/введение:	01.02.1999/01.12.1998
Язык оригинала:	английский
ТК, разработавший стандарт:	JTC 1
Страниц оригинала:	27

Наличие международного стандарта обнадежило и стимулировало направленный поиск, который выдал мгновенный результат.

T8 USB
Smart Token

STMicroelectronics

Интернет полон информации о целом семействе БИС фирмы STMicroelectronics (ранее SGS Thomson) для интеллектуальных карт, ключей доступа и иных средств криптозащиты с применением модулярных арифметических процессоров (Modular Arithmetic Processor – MAP).

Применение MAP фирмой началось в первой половине 90-х годов. Судя по публикациям сначала был разработан «512 бит MAP» в виде IP-блока (фирма проектирует БИС по технологии «система на кристалле» (System-on-Chip – SoC), см. статью «Элементная база модулярных и троичных ЭВМ» в трудах конференции), который применялся в БИС семейства ST16. В публикациях 1996 г. об этом говорится, как о хорошо известном подходе. В материалах 1997 г. уже говорится о более мощном IP-блоке «1088 бит MAP», который широко применяется в семействе БИС ST19.

На момент написания статьи из Интернета удалось получить информацию о 20 БИС с MAP ф. STMicroelectronics:

№	ИС	Modular Arithmetic Processor (MAP)
1	ST16CF54	512 bit MAP: – Fast modular multiplication and squaring using Montgomery method. – Software Crypto Libraries for efficient algorithm coding using a set of advanced functions. – Software selectable operand length up to 1024 bits.
2	ST19CF68	

3	ST16WL18	1088 bit MAP with library support for asymmetrical algorithms: – Fast modular multiplication and squaring using Montgomery method – Software Crypto libraries in separate ST ROM area for efficient algorithm coding using a set of advanced functions – Software selectable operand length up to 2176 bits.
4	ST19CF68	
5	ST19KF16	
6	ST19WI18	
7	ST19WK08	
8	ST19WK08-A	
9	ST19WL18	
10	ST19WL34	
11	ST19WL66	
12	ST19WP18	
13	ST19WR66	
14	ST19XL34	
15	ST19XL34V2	
16	ST19XT34	
17	ST19WP18-TPM-A	1088-bit Modular Arithmetic Processor providing Full support for Asymmetric operations
18	ST19WP18-TPM-B	
19	ST19WP18-TPM-C	
20	T8 USB Smart Token	

Эти БИС используются в защищенных интеллектуальных картах с контактным и бесконтактным считыванием и в ключах с USB-портом для систем доступа к оборудованию и информации.

M-Systems

Компания M-Systems в 1999 г. выпустила на рынок технологию «**SuperMAP**» (Superscalar Modular Arithmetic). Технология основана на выполненном в виде поддерживаемого компиляторами VHDL, ModelSim, Altera FLEX10K и Synopsys IP-блока многофункционального криптографического сопроцессора **SuperMAP**, получившего широкую известность благодаря своей высокой производительности и малым размерам.

Компания M-Systems объявила о новинке на рынке – накопителях для встраиваемых систем серии uDiskOnChip (uDOC), объединившей достоинства технологии флэш-памяти, интерфейса USB и технологии **SuperMAP**. Новые накопители предназначены для применения в телекоммуникационной сфере, торговых терминалах, промышленной ав-

томатике, игровых автоматах - т.е. везде, где требуется надежность хранения данных и высокая скорость обмена. Основные технические характеристики uDOC: подключение по интерфейсу USB 2.0; режим "plug-and-play"; установившаяся скорость чтения - 9 Мб/с; записи - 7 Мб/с; объем хранимой информации - от 16 Мб до 1 Гб (6 Гб к концу 2004 года). Технология **SuperMAP** позволяет реализовывать защиту части или всей находящейся на «диске» информации, идентификацию пользователя, цифровую подпись, криптографическую защиту данных. Спецификация нового продукта M-Systems поддержана целым рядом компаний (в частности - Wyse, TECO, MontaVista), занимающих существенную долю рынка встраиваемых систем.

Emosyn LLC

Полупроводниковая фирма Emosyn LLC (подразделение компании ATMI, Inc.) и компания M-Systems заключили соглашение о сотрудничестве, предполагающее оптимизацию криптографического сопроцессора **SuperMAP** компании M-Systems для использования последнего в продуктах фирмы Emosyn. Первыми решениями фирмы Emosyn, в которые будет интегрирована технология SuperMAP, станут представители семейства защищенных микропроцессоров Theseus Platinum. Продукты Theseus Platinum являются ведущими флэш-решениями для рынка смарт-карт и обеспечивают безопасность транзакций в мобильной телефонии, банковских операциях, системах идентификации и общественном транспорте.

"Мы рады заключить соглашение о сотрудничестве с фирмой Emosyn, сказал Дэн Дариэл (Dan Dariel), глава подразделения компании M-Systems. Эта фирма производит одни из самых миниатюрных, недорогих и малопотребляющих продуктов на сегодняшнем рынке. Наша технология SuperMAP добавит к этим характеристикам наличие современного многофункционального криптографического сопроцессора, получившего широкую известность благодаря своей высокой производительности и малым размерам". "Мощная технология SuperMAP компании M-Systems является признанным стандартом шифрования на рынке смарт-карт, сказал Фил Барнетт (Phil Barnett), директор подразделения фирмы Emosyn. Вместе мы сможем дать индустрии смарт-карт и защищенных интегральных схем проверенные эффективные решения, обеспечивающие высочайший уровень безопасности".

Hifn Inc.

Микросхема **Hifn 6500** фирмы Hifn Inc. – 1024-битный модулярный арифметический процессор ассиметричного шифрования с 2048-битной поддержкой, реализует алгоритмы шифрования с открытым ключом: RSA, DSA и DH, а также основные функции, оперирующие с большими числами. Предназначен для построения высокоскоростных защищенных линий связи в компьютерных сетях.

По сравнению с программной реализацией аналогичных алгоритмов на базе процессора Intel Pentium II – 266, показывает в 15 раз большее быстродействие.

Выполнен в виде БИС в 160-выводном корпусе.

Sun Microsystems

В конце 2005 г. ф. Sun Microsystems объявила о выпуске семейства принципиально новых процессороф UltraSPARC T1 (Niagara), созданного по технологии «система на кристалле». В одном чипе, соответствующем архитектуре SPARC v9 и выполненным по технологии 0.09 микрон, сосредоточено 8 четырехпоточных CPU ядра, каждое из которых работает на частоте 1.2 Ghz. В CPU интегрирован контроллер памяти и RSA-акселератор, позволяющий увеличить скорость выполнения некоторых операций связанных с шифрованием в 10 раз.

УДК. 681.3

Б.М. Малашевич, Д.Б. Малашевич

Отечественные модулярные и троичные ЭВМ

(ОАО «Ангстрем»,
Московский государственный институт электронной техники)

В статье рассмотрена история зарождения и развития модулярной и троичной арифметик. Кратко описана история создания модулярных ЭВМ Т-340А, К-340А, «Алмаз» и 5Э53, а так же троичных ЭВМ «Сетунь» и «Сетунь-70». Приведены их технические характеристики, информация о серийном производстве и некоторых применениях. Дан краткий анализ причин прекращения практических работ в этих направлениях, повлекших за собой так же и спад теоретических исследований в стране в областях модулярной арифметики и троичной информатики.

Истоки модулярной арифметики

Впервые о Системе остаточных классов (СОК) международная научная общественность узнала из статьи Миро Валаха, "Origin of the code and number system of remainder classes", опубликованной в 1955 г. в сборнике "*Stroje Na Zpracovani Informaci*", vol. 3, Nakl. CSAV, в Праге. А первый импульс в этом направлении дал чехословацкий ученый Антонин Свобода, специализирующийся в области логического проектирования средств вычислительной техники, с 1948 г. доцент Чешского института технологии в Праге. В 1950 г., читая курс лекций по цифровой и аналоговой вычисли-

тельной технике и объясняя теорию построения умножителей, он обратил внимание, что в аналоговой технике нет принципиальных структурных отличий между сумматором и умножителем (разница наблюдается только в применении масштабов на входах и выходах), в то время, как в цифровой реализации сумматор и умножитель имеют коренные отличия. Он предложил своим студентам попытаться найти цифровую реализацию, которая могла бы выполнять сложение и умножение со сравнимой легкостью. Немного позже один из студентов, Миро Валах предложил идеею цифрового кодирования, которая позже получила название «Система остаточных классов». Идея оказалась продуктивной и, после многолетних исследований А. Свободы, М. Валаха, Н. Сзабо, Р. Танаки и др., нашла воплощение в теории и практике построения первой электронной (ламповой) ЭВМ в Чехословакии «ЭПОС» (Электронный почитач), первой в мире модулярной ЭВМ. Позже была построена и «Эпос 2» – полупроводниковый вариант Эпоса.

В 1957 г. А. Свобода выступал с курсом лекций по логическому проектированию в китайской Академии наук в Пекине. С такими лекциями он выступал в Москве, Киеве, Дрездене, Кракове, Варшаве и Бухаресте. Его визиты в западные страны были сильно ограничены. Ему удалось выступить на конференциях в Дармштадт (1956), Мадриде (1958), Намур (1958), но не был допущен в Кембридж (1959) и на многие другие западные научные конференции.

В 1959 г. А. Свобода переходит в Центральный институт математики в Праге, где создает отделение "Математических машин", которое позже стало "Исследовательским институтом математических машин" АН – основное место исследований в цифровой и аналоговой вычислительной техники в Чехословакии.

В 1963 г. А. Свободе не позволили принять приглашение возглавить кафедру прикладной математики университета в Гренобле. В результате в 1964 г. семье А. Свободы удалось эмигрировать из Чехословакии, в 1965 г. они приехали в США. В 1966 г. Свобода поступил на факультет Калифорнийского университета в Лос-Аржелесе и в 1968 г. стал профессором. Он читал курс логического проектирования, компьютерной архитектуры и компьютерной арифметики. В 1968 г. он получил премию IEEE за вклад в логическое проектирование, механическое проектирование и за фундаментальную работу по СОК.

Первым в СССР в конце 50-тых годов на систему счисления остаточных классов (СОК) обратил внимание главный инженер КБ-1 Федор Викторович Лукин. Один из ведущих теоретиков в области СОК и активных участников ее практического применения, доктор технических наук, профессор, академик НАН Казахстана Вильжан Мавлютинович Амербаев вспоминает: "*Израиль Яковлевич Акушский рассказывал мне, что первую информацию о СОК он получил от Ф.В. Лукина в виде закрытой справки о работах в США. По словам Израиля Яковлевича, Федор Викторович считал СОК очень перспективным направлением развития вычислительной техники*". Последующие его действия подтверждают это – именно стараниями Ф.В. Лукина модулярная арифметика получила столь бурное и успешное развитие в стране, а с его уходом из жизни совпадает начало спада ее разработок.

Сопоставляя отрывочную информацию из разных источников, можно реконструировать эту историю следующим образом. Первым мысль о возможности применения СОК в вычислительной технике в 1955 г. в краткой статье высказал чехословацкие ученые А. Свобода и М Валах. Они и стали первопроходцами СОК. Их работами заинтересовались американцы, где работы над модулярной арифметикой (основанной на СОК) были развернуты широким фронтом.

СОК

В системе остаточных классов каждое число, многоразрядное в позиционной системе счисления, представляется в виде нескольких малоразрядных позиционных чисел, являющихся остатками от деления исходного числа на взаимно простые основания. В обычной позиционной двоичной системе выполнение операций (например, сложение двух чисел) производилось последовательно по разрядам, начиная с младшего. При этом образуются переносы в следующий старший разряд, что и определяет поразрядную последовательность обработки. В СОК появилась возможность распараллелить этот процесс: все операции над остатками по каждому основанию выполняются отдельно и независимо (параллельно), следовательно, в связи с их малой разрядностью, легко и быстро. Малая разрядность остатков обеспечивает возможность реализации табличной

арифметики, при которой результат операции не вычисляется каждый раз, а, однажды рассчитанный, помещается в запоминающее устройство (ЗУ) и, при необходимости, считывается из него. Т.е. операция в СОК при табличной арифметике и конвейеризации выполняется за один период синхронизирующей частоты (машинный такт). Проблемы возникают при переполнении диапазона представления чисел и округлении результатов, на их решение и потребовалось масса сил и интеллекта математиков.

Табличным способом, и тоже за один машинный такт, в СОК можно выполнять не только простейшие операции, но и представимые в виде полинома сколь угодно сложные функции, если результат не выходит за переделы диапазона представления чисел. Этим определяется одно из парадоксальных свойств модулярной арифметики: эффективная производительность модулярной ЭВМ может быть значительно, в разы, в десятки и сотни раз выше, чем у позиционной ЭВМ с той же тактовой частотой. Действительно, если операция, которая в обычной ЭВМ выполняется за 100 тактов, в модулярной ЭВМ выполняется за 1 такт, то и ее эффективная производительность на этих операциях, при прочих равных условиях, в 100 раз выше.

Введя дополнительные основания, получаем избыточность, обеспечивающую контроль и исправление ошибок в процессе выполнения операций. Это еще одно из важнейших преимуществ СОК (арифметичность) перед всеми позиционными системами: ни одна из них не позволяет находить и, тем более, исправлять ошибки в процессе выполнения арифметических операций. Наоборот, в арифметическом устройстве они, раз возникнув, бесконтрольно размножаются. В результате в ЭВМ всех времен и народов, работающих в традиционных позиционных системах счисления, контроль и исправление ошибок (контроль на четность, избыточное кодирование, мажорирование и т.п.) обеспечиваются только в системах хранения и передачи информации. Арифметико-логические устройства – один из основных источников сбоев и ошибок в ЭВМ – остаются бесконтрольными. Сейчас, когда весь процессор размещается в одном кристалле БИС, это не столь критично. В те времена процессор занимал шкаф или несколько, содержал многие тысячи отдельных элементов и паяных контактов, а так же километры проводников – гарантированный источник различных

помех и сбоев, причем бесконтрольных. Взяв под контроль эти источники сбоев и ошибок в процессоре, СОК резко повысил общую надежность ЭВМ К-340А и 5Э53 (о них ниже) по сравнению с современными им машинами.

Примерно в 1959 г. в КБ-1 (ныне ОАО "НПО "Алмаз"") по закрытым каналам поступила справка об этих работах. Ф.В. Лукин, имеющий личный опыт разработки счетно-решающих устройств и, особенно, их применения в крупнейших военных системах, сразу оценил перспективность этого направления. Но КБ-1 разработкой ЭВМ не занималось, и Федор Викторович направил заинтересовавшую его справку в СКБ-245 (в 1953 г. он был там председателем Госкомиссии по приемке ЭВМ "Стрела", первый экземпляр которой был установлен в КБ-1). Справка заинтересовала математика И.Я. Акушского и его начальника, ведущего разработчика ЭВМ Д.И. Юдицкого, ставших впоследствии основоположниками модулярной арифметики в СССР. Примерно в это же время поступила информация и из открытого источника. Вот как об этом вспоминает участник тех событий В.С. Линский*: "Примерно в 1957-58 г.* (скорее всего в 1959 г.) *начальник отдела НИЭМ"* (СКБ-245) *Э.А. Глузберг получил из Реферативного журнала АН СССР для подготовки реферата копию статьи чехословацких ученых А. Свободы и М. Валаха о представлении натуральных чисел группой вычетов по различным модулям и операциях с ними, позже названном СОК. Статья была написана на чешском языке и далека от научных интересов Э.А. Глузберга. Поэтому он поручил разобраться с ней И.Я. Акушскому, а он, в свою очередь, попросил меня ознакомиться со статьей. Я перевел статью, для чего мне пришлось купить чешско-русский словарь (хранится у меня до сих пор) и изучил ее. Я пришел к выводу о нецелесообразности использования СОК в большинстве ЭВМ из-за низкой эффективности операций в ней с плавающей точкой. Однако И.Я. Акушский со мной не согласился и приступил к научным исследованиям СОК*". По-видимому, информация о работах в США и вызвала запоздалый интерес в АН СССР к статье, вышедшей в Праге еще в 1955г.

Полученная таким образом исходная информации, весьма краткая и поверхностная, дала старт научным исследованиям И.Я. Акушского и Д.И. Юдицкого. Первая в стране попытка осмыслить принципы построения модулярной ЭВМ (на основе СОК) была предпринята в 1959-1960 гг. в СКБ-245 Ю.Я. Базилевским, Ю.А. Шрей-

дером, И.Я. Акушским и Д.И. Юдицким, но не получила единого понимания – не все ее участники прониклись сутью СОКа. И когда в 1960 г. Ф.В. Лукин, недавно назначенный директором НИИ-37 ГКРЭ (позже НИИ ДАР), пригласил Д.И. Юдицкого и И.Я. Акушского для разработки ЭВМ, они согласились. Д.И. Юдицкий стал начальником отдела НИИ-37, а И.Я. Акушский – начальником лаборатории в этом отделе. Первой задачей Д.И. Юдицкого в НИИ-37 было завершение неудачной разработки ЭВМ "А-340А" для создаваемых предприятием радиолокационных станций (РЛС), И.Я. Акушский, как ученый-теоретик, сразу занялся разработкой научных основ построения модулярной ЭВМ.

После успешного завершения работ над А-340А, возглавляемый Д.И. Юдицким коллектив в 1960-1963 гг. разработал первую в стране реально работавшую модулярную ЭВМ Т-340А для полигонного варианта РЛС Дунай-ЗУП системы противоракетной обороны (ПРО) А-35. Теория и практика варианта модулярной арифметики, принципы построения ЭВМ на их основе были разработаны И.Я. Акушским, Д.И. Юдицким и Е.С. Андриановым. Это была экспериментальная ЭВМ, изготовленная, отлаженная и реально проработавшая много лет в полигонной РЛС. Полученные результаты были использованы при проектировании ЭВМ К-340А, которая была освоена в серийном производстве и стала базовой для всех РЛС, разрабатываемых в те годы в НИИ-37. В этих ЭВМ впервые в стране был реализован принцип независимых каналов памяти команд и данных. Оперативная память была выполнена в виде 16 блоков емкостью по 1К слов. Каждый блок имел по два порта для ввода–вывода информации: с абонентами (с возможностью параллельного обмена с любым числом блоков) и с процессором. Для увеличения быстродействия было реализовано программное расслоения оперативной памяти с чередованием обращения процессора к блокам. Кроме того, была применена многовходовая буферная память для двухоперационных команд (в каждой команде выполнялось по две операции, каждая из которых в других ЭВМ того времени выполнялась в виде отдельной команды). Эти особенности построения системы памяти обеспечили высокую эффективность ЭВМ К-340А: задержек при обращении к памяти большого объема (бич ЭВМ тех лет) практически не было.

ЭВМ Т-340А и К-340А обладали невиданным в те времена быстродействием в 1,2 млн. двойных оп/с. или 2,4 млн. обычных оп/с. Ти-

повое быстродействие ЭВМ в те времена измерялось десятками или сотнями тысяч оп/с. Это первая в мире ЭВМ с быстродействием более 1 млн. оп/с. И это была ЭВМ с самой низкой стоимостью единицы производительности – 25 коп за 1 оп/сек. Опытным заводом при НИИ-37 и Свердловским заводом радиоаппаратуры было выпущено более 50 ее комплектов. Только в РЛС "Дунай-3У" работало 10 ЭВМ "К-340А". Благодаря высочайшей надежности и уникальным характеристикам ЭВМ К-340А до сих пор (2005 г., 40 лет!!!) находятся в эксплуатации, демонстрируя значительно более высокую живучесть, чем работающие рядом с ними другие, современные им, электронные системы.

1. ЭВМ «Т-340А» и «К-340А»

Разработка принципов построения ЭВМ в СОК и способов их реализации – И.Я. Акушский и Д.И. Юдицкий.

Главный конструктор:

- Т-340А – Д.И. Юдицкий,
- К-340А – Д.И. Юдицкий, позже Л.В. Васильев.

Разработка, НИИ-37:

- Т-340А – 1960-1963гг.,
- К-340А – 1963-1966 гг.

Изготовители: опытный завод при НИИ-37 и Свердловский завод радиоаппаратуры, в 1966-1973 гг. выпущено более 50 комплектов.

Разрядность данных и команд – 45 бит.

Трехадресная, две операции в одной команде.

Система счисления – СОК с дополнительным основанием.

СОК – основания и занимаемые ими разряды слова:

Основания:

2; 5; 23; 63; 17; 19; 29; 13; 31; 61.

Разряды слова:

1; 2-4; 5-9; 10-15; 16-20; 21-25; 26-30; 31-34; 35-39; 40-45.

Производительность – 1,2 млн. двухоперационных команд/с (в общепринятом исчислении – 2,4 млн. оп/с).

Обнаружение ошибки в слове при выполнении операций в

арифметическом устройстве.
Многовходовая буферная память – 16х45 бит.
ОЗУ данных – 16К 45-разрядных слов (720К бит),
ПЗУ команд – 16К 45-разрядных слов (720К бит).
Стоимость ЭВМ: – опытной – 1,2 млн. руб.,
– серийной – 0,6 млн. руб.
Стоимость единицы производительности – 25 коп/оп.
Элементная база – транзисторы, диоды, ферриты и т.п.
Потребляемая мощность – 33 кВт.
Размер шкафа – 600х700х1800 мм.
Количество шкафов – 12.

Зеленоградский Центр микроэлектроники

В начале 1963 г. Ф.В. Лукин был назначен директором организуемого в строящемся Зеленограде Центра микроэлектроники (ЦМ, позже Научный центр – НЦ). Оказавшись на переднем рубеже отечественной электроники, он решил соединить ее новые возможности с передовой для того времени мыслью в области вычислительной техники, проверенной им в НИИ-37 созданием модулярных супер-ЭВМ (под супер-ЭВМ будем понимать ЭВМ с рекордно высокими для своего времени характеристиками). Для этого Федор Викторович пригласил в ЦМ хорошо известный ему коллектив создателей ЭВМ Т340А и К340А во главе с Д.И. Юдицким и И.Я. Акушским. К этому времени ЭВМ Т-340А была разработана, изготовлена и настроена. Разработка проекта серийной ЭВМ К-340А, ее изготовление и отладка на опытном заводе НИИ-37 были завершены после ухода группы специалистов в Зеленоград оставшимся коллективом сотрудников под руководством Леонида Викторовича Васильева. А перешедшие в 1964 г. образовали отдел перспективных ЭВМ в предприятии п/я 2014 (позже НИИ Физических проблем – НИИ ФП, зам. директора Д.И. Юдицкий).

Вспоминает М.Д. Корнев: «*Однажды Д.И. Юдицкий сказал мне: «К завтрашнему утру нужен проект системы команд мощной ЭВМ в остаточных классах. Будем обсуждать его все вместе». Это было уже конкретное дело. Я просидел весь день и почти всю ночь и наутро принес готовый вариант системы команд. Давлет Исламович собрал в своем кабинете всех спецов, и началось под-*

робнейшее обсуждение каждой команды. Обсуждения продолжались несколько дней, в результате появилась система команд мощной ЭВМ, работающей в СОК». Так началась работа по созданию высокопроизводительной модулярной ЭВМ. И когда появился заказчик, коллектив уже был готов к конкретному разговору.

Заказчик

В 1953 г. начались работы по созданию отечественной системы противоракетной обороны (ПРО), вылившиеся в разработку боевой Системы "А-35" для защиты московского промышленного района (генеральный конструктор Григорий Васильевич Кисунько, ОКБ «Вымпел», МРП). Но когда А-35 была уже практически разработана и в значительной степени изготовлена, в США появились межконтинентальные баллистические ракеты (БР) с разделяющимися боеголовками. А-35 бороться с такими БР не могла – в свое время ее заказчики не смогли предвидеть их появление. Было принято решение о модернизации А-35 и о создании ее второй очереди, т.е. о дополнении А-35 тремя принципиально новыми многоканальными стрельбовыми комплексами (МКСК) и была начата разработка и изготовления его полигонного варианта – МКСК "Аргунь". Главным конструктором (ГК) МКСК “Аргунь” Г.В. Кисунько назначил Николая Кузьмича Остапенко.

По предварительным оценкам для МКСК требовалась ЭВМ с производительностью около 3,0 млн. алгоритмических оп/с. Как вспоминает Н.К. Остапенко: «*Одна алгоритмическая операция на задачах МКСК соответствовала примерно 3 ÷ 4 простейшим операциям ЭВМ*», т.е. в обычном тогда понимании требовалась ЭВМ с быстродействием около 10 (9 ÷ 12) млн. оп/с. Такой ЭВМ тогда ни где не было. Лучшие на конец 1966 г. ЭВМ США обладали быстродействием в 4 – 12 раз меньшим требуемого для МКСК (таблица подготовлена в 1968 г. Д.И. Юдицким):

Фирма	Модель	Быстродействие ЭВМ, млн. сложений/с	Быстродействие элементов, нс
IBM	360/75	1,0	5
CDC	6600	2,5	10
Philco	2000/212	1,5	5
Burroughs	B 5500	0,3	20
Sparry Rand	1108	1,2	5

Когда требования к ЭВМ прояснились, встал вопрос, где ее взять. В это время готовилось постановление ЦК КПСС и СМ СССР, вышедшее 5 ноября 1965 г., о создании эскизного проекта территориальной системы ПРО страны “Аврора” (как утверждает Г.В. Кисунько в своей книге, навязанного ему вопреки его позиции о несвоевременности такого проекта). Но, для пользы дела, Григорий Васильевич включил в это же постановление и дополнительные поручения по созданию МКСК "Аргунь". В результате трем предприятиям: ЦМ (МЭП, Ф.В. Лукин), ИТМ и ВТ (МРП, С.А. Лебедев) и ИНЭУМ (Минприбор, М.А. Карцев) было дано конкурсное задание на разработку и для Авроры, и для Аргуни эскизных проектов высокопроизводительной ЭВМ со сроком окончания 30 марта 1967 года.

Так в Зеленограде началась разработка эскизного проекта супер-ЭВМ "Алмаз", главный конструктор (ГК) – Давлет Исламович Юдицкий.

Проект "Алмаз"

В соответствии с исходными данными Генерального конструктора ПРО к ЭВМ предъявлялись следующие требования: разрядность данных 45 бит, производительность 2,5 ÷ 3,0 млн. алгоритмических оп/с, сложные функции в одной команде, работа со словами переменной длины, объем памяти 2^{17} 45-разрядных слов (5,625М бит) и т.п. Весьма не простые для тех времен требования.

Нельзя не отметить дружественный характер поведения конкурентов. У главных конструкторов М.А. Карцева и Д.И. Юдицкого были прекрасные человеческие отношения, распространившиеся и на их коллективы. Как вспоминает М.Д. Корнев: "*У нас и у Карцева проходили регулярные заседания НТС* (научно-технический совет), *на которых специалисты обсуждали пути и проблемы построения ЭВМ. На эти заседания мы обычно приглашали друг друга: мы ездили к ним, они – к нам. И активно участвовали в обсуждении*". Это не мешало, а помогало им. Выбрав изначально разные стратегии в построении ЭВМ, они, таким образом, помогали друг другу в тактике их реализации.

К созданию ЭВМ "Алмаз" были привлечены все силы Зеленограда. На НИИ ФП возлагалась разработка архитектуры и процессора ЭВМ, на НИИ ТМ – базовой конструкции, системы питания и сис-

темы ввода/вывода информации, на НИИ ТТ – интегральных схем: в этом отношении проект Алмаз имел неоспоримое преимущество по сравнению с проектами С.А. Лебедева и М.А. Карцева, т.к. первые в стране интегральные схемы – новейшая элементная база, создавалась здесь же, в Зеленограде и на процесс ее создания можно было влиять. Но элементная база всех проблем не решала. Огромное значение имело применение СОК. Вот что писал Д.И. Юдицкий в итоговой справке по проекту "Алмаз" в марте 1968 г.:

"*В результате проведенных исследований было установлено, что в непозиционных системах могут быть построены самокорректирующиеся коды, позволяющие восстанавливать истинные результаты вычислений по цепи элементарных операций, если во время этих вычислений имели место какие-либо искажения. Была построена теория специального кодирования в непозиционных системах, позволяющая введением минимальной избыточности в представление слова, осуществлять исправление возникающих ошибок методами, близкими к исправлению по смыслу на основе анализа последовательно получающихся слов в процессе обработки. Применение методов специального кодирования значительно увеличивает функциональную надежность вычислительных машин и позволяет создавать "живучие" машины, сохраняющие работоспособность при выходе из строя значительной части оборудования.*

Инженерный пульт управления ЭВМ "Алмаз"

Таким образом, требования Генерального Конструктора оказалось возможным удовлетворить:

1. За счет использования разработанной в Научном Центре теории непозиционных систем исчисления, позволяющей добиваться высокой производительности на основе широкого распараллеливания выполнения элементарных операций и максимальной надежности в силу специфических самокорректирующих способностей непозиционных систем;

2. За счет использования микроэлектронной технологии изготовления системы логических элементов и основных блоков и узлов вычислительной машины, удачно сочетающейся со спецификой не-

Ф.В. Лукин докладывает
Председателю Совета министров СССР А.Н. Косыгину
о разработке супер-ЭВМ «Алмаз» на основе модулярной арифметики.
(Слава направо: А.И. Шокин, Д.И. Юдицкий, А.Н. Косыгин, Ф.В. Лукин.
Сзади слева – шкаф макета «Алмаз»)

позиционных систем. Разработка машины проведена на основе системы логических элементов типа "Посол" со средним временем распространения порядка 25-30 наносекунд".

Наряду с применением модулярной арифметики был найден еще один архитектурный способ значительного увеличения общей производительности ЭВМ. Это было решение, широко применяемое позже в системах обработки сигналов – введение в систему процессора предварительной обработки сигнала. Но тогда это было новым словом в науке и технике. В состав ЭВМ Алмаз было введено три типа вычислительных процессоров.

- Узко специализированный непрограммируемый процессор предварительной обработки радиолокационной информации, названный в Алмазе Преобразователем информации (ПИ, в нынешней терминологии ПОС или ЦОС – процессор цифровой обработки сигналов).

- Программируемый модулярный процессор, выполняющий основную обработку данных.
- Программируемый двоичный процессор, выполняющий не модулярные операции, в основном, связанные с процедурами управления работой ЭВМ.

Информация от антенн радиолокатора (поток 30 тыс. 100-разрядных слов в секунду) подается на ПИ, проходит предварительную обработку в реальном темпе ее поступления, что исключает необходимость ее промежуточного хранения. Результаты этой обработки (их объем многократно меньше исходного) поступают на модулярный процессор. Расчеты показали, что предлагаемый ПИ имеет производительность, эквивалентную примерно 4,0 млн. алгоритмических оп/с и позволяет сэкономить около 3 миллионов бит памяти. Модулярный процессор ЭВМ Алмаз имеет производительность 3,5 млн. алг. оп/с. В результате эффективная производительность ЭВМ Алмаз составляет 3,5 + 4,0 = 7,5 млн. алг. оп/с., т.е. в два-три раза выше требуемой. Эти расчетные данные были подтверждены результатами моделирования на универсальной ЭВМ.

Так в рамках единого проекта общими усилиями специалистов предприятий ЦМ под руководством и при непосредственном участии Ф.В. Лукина, Д.И. Юдицкого и И.Я. Акушского многие проблемы построения высокопроизводительной ЭВМ были решены и проверены на макетном образце ЭВМ "Алмаз".

ЭВМ «Алмаз»

Эскизный проект – март I968 г.

Главный конструктор Д.И. Юдицкий, научный руководитель И.Я. Акушский.

Разработчик: Центр микроэлектроники МЭП, Зеленоград.

Разрядность данных и команд – 45 бит.

Диапазон представления чисел -2^{30}.

Производительность – 7,5 млн. алг. оп/с (в общепринятом исчислении – до 30 млн. оп/с).

- Система счисления остаточных классов (СОК) с дополнительным основанием.

- СОК – основания и занимаемые ими разряды слова:

  Основания:

  2; 5; 23; 63; 17; 19; 29; 13; 31; 61.

  Разряды слова:

  1; 2-4; 5-9; 10-15; 16-20; 21-25; 26-30; 31-34; 35-39; 40-45.

Обнаружение двойных и исправление одиночных ошибок при выполнении операций в арифметическом устройстве.

Адресность – двухадресная.

Вычисления значения специальных функций в качестве элементарной команды.

Работа со словами переменной длины.

Параллельная обработка малоразрядной информации.

Режим с плавающим диапазоном.

Объем памяти – 128К 45-разрядных слов (5,898 Мбит)

Быстрая буферная память – 32 55-разрядных слов.

Вероятность безотказной работы в течение 15 мин – 0,999.

Коэффициент готовности в установившемся режиме – 0,999.

Размер шкафа – 550х800х1750 мм.

Объем оборудования – 11 шкафов, инженерный пульт управления, внешние устройства.

Занимаемая площадь – 80-100 м^2.

Потребляемая мощность – 5 кВт.

Расчетная стоимость: – опытного образца – 4,2 млн. руб,
– серийного образца – 2,6 млн. руб.

Эскизный проект был разработан и 30 марта 1967 г. представлен заказчику. Распоряжением Д.Ф. Устинова, в то время председателя Военно-Промышленной комиссии при Совмине СССР (ВПК), под председательством главного конструктора МКСК Н.К. Остапенко была создана Государственная комиссия для оценки эскизных проектов. Академик С.А. Лебедев, ИТМ и ВТ которого был и без того перегружен работами по Эльбрусу и БЭСМ, ознакомившись с другими проектами, снял свой вариант с рассмотрения. Осталось два проекта: "Алмаз" Д.И. Юдицкого и "М-9" М.А. Карцева.

Конкурс выиграла ЭВМ "Алмаз". К этому времени проект территориальной системы "Аврора" был отвергнут, но задача создания МКСК "Аргунь" осталась. 20 мая 1967 г. ОКБ "Вымпел" и НЦ заключили договор на разработку высокопроизводительной ЭВМ "5Э53" и 5-машинного комплекса на ее основе с организацией серийного производства в Загорском электромеханическом заводе (ЗЭМЗ) и сдачей комплекса на противоракетном полигоне. Главным конструктором 5Э53 был назначен Д.И. Юдицкий. В октябре 1969 г. коллектив разработчиков ЭВМ был выделен в самостоятельное предприятие – Специализированный вычислительный центр (СВЦ), директор Д.И. Юдицкий, зам. по науке И.Я. Акушский.

Д.И. Юдицкий докладывает 1-му Зам. Председателя Госплана СССР В.М. Рябикову и первому секретарю МГК КПСС Н.Г. Егорычеву о разработке ЭВМ «Алмаз».
(Слева на право: Н.Г. Егорычев, В.И. Трифонов, Г.Я. Гуськов, В.М. Рябиков, В.В. Савин, Д.И. Юдицкий, А.И. Шокин. Сзади слева – шкаф ЭВМ «Алмаз»)

У проекта ЭВМ "М-9" М.А. Карцева была иная судьба. Он не победил в конкурсе и не был признан в родном Минприборе, отказавшемся от продолжения работ по созданию мощных ЭВМ. Коллективу М.А. Карцева было предложено перейти в МРП, что он в середине 1967 г. и сделал. Еще с 1958 г. М.А. Карцев тесно сотруд-

ничал с академиком А.Л. Минцем (Радиотехнический институт – РТИ), разрабатывая для его Систем Предупреждения о Ракетном Нападении (СПРН) высокопроизводительные ЭВМ М-4, М4-2М и комплексы на их основе, серийно выпускавшиеся ЗЭМЗом (МРП). В это время наступила очередь создания нового поколения СПРН и 16.10.1969 г. М.А. Карцев получил заказ на разработку для нее мощной ЭВМ 5Э66 (фирменное наименование – М-10), в которой были использованы наработки по М-9. Далее оба проекта (5Э53 и 5Э66) развивались независимо, их производство планировалось на одном заводе – ЗЭМЗ.

Супер-ЭВМ 5Э53

5Э53 была предназначена для решения следующих основных задач:

- обнаружение и сопровождение целей,
- селекция реальных целей среди ложных,
- наведение противоракет на цели,
- управление системами МКСК и др.

Пока разрабатывался Алмаз, в ОКБ "Вымпел" шла работа над МКСК "Аргунь", требования к ЭВМ были уточнены. Для второй очереди Системы А-35 требовалась общая производительность до 0,6 млрд. оп/с. Эту вычислительную мощность должны были обеспечивать 15 ЭВМ (по 5 в каждом МКСК) производительностью на задачах ПРО по 10 млн. алгоритмических оп/с (около 40 млн. обычных оп/с), ОЗУ 7,0 Мбит, ППЗУ 2,9 Мбит, ВЗУ 3 Гбит, аппаратура передачи данных на сотни километров. Т. е. 5Э53 должны быть существенно мощнее Алмаза.

В составе Аргуни планировалось использовать 5 комплектов ЭВМ 5Э53 (в РЛС цели "Истра" – 2, в РЛС наведения противоракет – 1 и в командно-вычислительном пункте – 2), объединенных в единый комплекс.

В 5Э53 был реализован целый букет новых и прогрессивных для того времени идей, изобретений и решений. Вот некоторые примеры.

Применение СОК обеспечивало два основных бесспорных преимущества:

- Повышенную производительность и простоту аппаратной реа-

лизации арифметического устройства за счет малоразрядности оснований.

- Повышенную надежность системы благодаря свойствам СОК, обеспечивающим обнаружение и исправление ошибок, возникающих при выполнении операций в арифметическом устройстве (двоичные ЭВМ этого никогда не умели).

Архитектура 5Э53 имела много принципиально новых решений:

- Разделение команд на управленческие и арифметические. Арифметические команды (в т.ч. предварительная и основная обработка сигналов) выполнялись на модулярных процессорах, управленческие – на двоичных.
- 8-уровневая конвейерная организация.
- Аппаратная блочная реализация арифметики: блок сложения/вычитания, блок умножения, блок управления адресами и т.п.),
- Разделение памяти на оперативную данных и полупостоянную команд,
- Разделение шин команд и данных,
- Аппаратное расслоение памяти на 8 блоков с чередующейся адресацией по блокам.

Конструктор Г.А. Кириллова
около шкафа 5Э53

Для 5Э53 было разработано ОЗУ на основе интегрального носителя – цилиндрических магнитных пленок (ЦМП). По быстродействию, габаритам, массе, энергопотреблению, технологичности и стоимости (1 коп/бит) оно было гораздо привлекательнее применявшихся тогда ОЗУ на ферритовых сердечниках. Физика работы ЗУ на ЦМП довольно сложная, сложнее, чем у ферритов, поэтому оставим ее для

специалистов, ограничимся констатацией факта, что многие научные и инженерные проблемы были решены и ОЗУ на ЦМП работало. Сравнение реальных устройств показывает, что преимущества ЦМП перед ферритами составляют: по физическому объему в 15 раз, по быстродействию в 5 раз.

Еще одной из главных проблем было построение полупостоянной памяти для хранения программ и констант. В системах ПРО задачи меняются не часто, поэтому требовалась достаточно простая и быстрая постоянная память, но допускающая смену информации. Для 5Э53 разработано ППЗУ с индукционной связью. На печатной плате реализовалась система ортогональных адресных и разрядных шин. На их пересечение накладывался или не накладывался замкнутый виток связи. Если виток наложен – индукционная связь есть, при подаче адресного импульса в разрядной шине индуцируется импульс, соответствующей информации «0». Если витка нет – нет и разрядного импульса, значит записан «1». Все эти витки связи размещаются на тонкую печатную плату – интегральную карту, которая плотно прижимается к матрице адресных и разрядных шин. Меняя вручную карту (не выключая ЭВМ), меняем информацию.

В качестве внешней памяти большой емкости было разработано ЗУ на оптической ленте. Оно имело много общего с основными в то время ВЗУ на магнитных 35 мм лентах (подобные конструкция, привод, электроника), но отличалось носителем информации и методами записи/чтения информации – фото/светодиоды через оптоволокно на фотопленку. В результате емкость ВЗУ при тех же габаритах повышалась на два порядка и достигала 3 Гбит. Образец накопителя был изготовлен и работал в составе макетного образца 5Э53.

Повышенная надежность 5Э53 обеспечивалась самокорректирующимися свойствами СОК в арифметическом устройстве, полным мажорированием (2 из 3) всех других систем машины, технологией монтажа межячеечных и межсубблочных соединений методом накрутки и другими средствами.

Супер-ЭВМ 5Э53

Технический проект – февраль I971 г.
Главный конструктор Д.И. Юдицкий.

Разработчик: Специализированный вычислительный центр, МЭП, Зеленоград.

Назначенный завод-изготовитель – Загорский электромеханический завод, МРП.

Разрядность:

– данных – 20 и 40 бит,

– команд – 72 бит.

Система счисления – СОК с дополнительным основанием.

Основания:

17; 19; 26; 31; 23; 25; 27; 29.

Разряды слова:

1-5; 6-10; 11-15; 16-20; 21-25; 26-30; 31-35; 36-40.

Тактовая частота –6,0 МГц.

Производительность:

– 10 млн. алгоритмических операций в секунду на задачах ПРО (40 млн. коротких оп/с),

– 6,6 млн. коротких оп/с на одном модулярном процессоре,

Формат алгоритмической операции – 3-4 коротких.

Время выполнения модулярных операций – 1 такт = 166 нс.

Число процессоров – 8 (4 модулярных и 4 двоичных).

ППЗУ команд:

– емкость: - общая – 2,8М бит,
- шкафа – 573К бит
- блока – 1024х72 бит = 73 728 бит = 72К бит,

– время цикла – 332 нс,

– темп выборки – 166 нс,

– число блоков – 40,

– число шкафов – 5.

ОЗУ данных:

– емкость: - общая – 7,0М бит,
- шкафа – 1,0М бит
- блока – 4096х64 = 262 144 бит = 256К бит,

– время цикла 700 нс,

– темп выборки – 166 нс,

– число блоков – 28,

– число шкафов – 7.

– стоимость – 1 коп/бит в ценах 1972 г.

Объем оборудования ЭВМ:

– типов шкафов – 7 и Инженерный пульт управления.

– число шкафов – 24

Размер шкафа, HxBxL: – 1800x800x600 мм

Потребляемая мощность – 60 кВт.

Среднее время безотказной работы – 600 часов.

Занимаемая площадь (со стендовым и ремонтным оборудованием) – 120м^2.

Проектирование

Разработка 5Э53 была проведена в рекордно короткий срок. Весь коллектив предприятия работал с необыкновенным подъемом не щадя себя, по 12 часов в день и более. Руководитель военной приемки в СВЦ В.Н. Каленов вспоминает реплику одного из ведущих разработчиков В.М. Радунского: «*Вчера до того доработался, что, входя в квартиру, предъявил жене пропуск*». Е.М. Зверев, возглавлявший группу по наладке макетного образца 5Э53, вспоминает другой пример: «*В то время были нарекания на помехоустойчивость ИС серии 243. Как-то часа в 2 ночи на макет пришел Давлет Исламович, взял*

Е.М. Зверев за наладкой макета 5Э53

щупы осциллографа и долго сам просматривал наиболее сложные места в схемах, разбираться в причинах помех». Ночью работали и инженеры, и директор. Такой напряженный труд хорошо оплачивался, для активных участников проекта применялась аккордная оплата, по завершении этапов работы выплачивались премии, применялись различные меры морального стимулирования.

В.С. Кокорин, М.Д. Корнев, М.Н. Белова, Л.Г. Рыков, В.С. Хайков.
Сначала в НИИ-37, а затем в НИИ ФП и СВЦ они разрабатывали модулярные ЭВМ Т-340А, К-340А, Алмаз и 5Э53
22 сентября 2004 г. на встрече в Зеленограде, посвященной 75-летию Д.И. Юдицкого

В ходе разработки продолжались теоретические исследования с целью совершенствования методов обработки информации в СОК: операции типа умножения, деления, определение знака и т.п. к тому времени не имели удовлетворительных алгоритмов решения в СОК. В результате напряженной работы В.М. Амербаева и его команды проблема была решена, алгоритмы разработаны и реализованы в проекте. Разработка алгоритмов проводилась одновременно с их аппаратной реализацией. Вспоминает М.Д. Корнев: «*Ночью Вильжан Мавлютинович думает, утром результаты приносит В.М. Радунскому. Схемотехники просматривают аппаратную реализацию нового варианта, задают Амербаеву вопросы, он уходит думать опять и так до тех пор, пока его идеи не поддадутся хорошей аппаратной реализации*». Это характерный пример взаи-

модействия подразделений и специалистов СВЦ в ходе разработки 5Э53.

При проектировании 5Э53 в СВЦ широко применялось машинное проектирование, в основном собственной разработки. В начале 1971 г. разработка документации была завершена. У В.Н. Каленова сохранились записи об ее объемах: 160 типов ячеек, 325 типов субблоков, 12 типов блоков питания, 7 типов шкафов, инженерный пульт управления, масса стендов. Были проведены все необходимые испытания ячеек и субблоков, изготовлен и испытан макетный образец 5Э53. 27 февраля 1971 г. 8 комплектов конструкторской документации (по 97 272 листа каждый) колонной машин были доставлены на ЗЭМЗ. Началась подготовка производства. Закончить ее, к сожалению, не удалось.

Супер-ЭВМ 5Э53 пала жертвой бескомпромиссной борьбы в МРП и МО вокруг ПРО. Это другая интересная и трагичная история и писать о ней ее участникам. Ограничимся лишь констатацией факта, что в 1971 г. началось планомерное сворачивание работ по созданию полигонного МКСК "Аргунь", а в 1972 г. они были практически прекращены. В связи с этим в 1971 г., когда подготовка серийного производства 5Э53 подходила к концу и началось изготовление ее устройств, было прекращено финансирование работ по ЭВМ в СВЦ и ЗЭМЗ. Главного инициатора и основной опоры проекта Ф.В. Лукина уже не было среди живущих. Его преемник А.В. Пивоваров вспоминает: "*Я обратился к заместителю министра МРП В.И. Маркову. Владимир Иванович объяснил мне, что загорский завод перегружен, что он уже выпускает аналогичную ЭВМ разработки МРП, их вполне удовлетворяющую* (5Э66) *и что 5Э53 Минрадиопрому для ПРО не нужна*". Два завода, в Выборге и Днепропетровске, были готовы выпускать 5Э53, но оба они принадлежали МРП, которое ни разрешения на это, ни средств, необходимых для организации производства, естественно, не дало. В.И. Марков лукавил. Во-первых ЭВМ 5Э53 и 5Э66 совершенно не "аналогичны", а во-вторых разработчики ПРО не получили ни той, ни другой. И в момент прекращения почти завершенной организации производства 5Э53 в ЗЭМЗ работы по 5Э66 только начинались, на нее на заводе еще даже не было полного комплекта документации. А новый гигантский корпус выпускного цеха 14 еще стоял полупустой, что автор лично наблюдал в середине 1971 г. Проблемы с мощностями для выпуска 5Э66 действительно возникли в ЗЭМЗ к

концу 1972 г., но после того, как 5Э66 получила второе применение (в спутниковой системе обнаружения стартов ракет академика А.И. Савина) и потребность в ней резко возросла. Но в 1971 г., когда работы по 5Э53 в ЗЭМЗ были остановлены, об этом еще известно не было. И решили возникшую проблему просто, подключив другие заводы.

Невостребованной 5Э53 оказалась и в МЭП – задач для нее еще не было. Тогда МЭП разрабатывал ИС низкой интеграции, которые вполне поддавались ручному проектированию. Время мощных систем автоматизированного проектирования еще не наступило. Если бы 5Э53 появилась в эпоху микропроцессоров и других сложных БИС, а на таких задачах модулярная арифметика весьма эффективна, скорее всего, ее ожидала бы иная судьба.

ЭВМ четвертого поколения

На этом работы по созданию модулярной высокопроизводительной ЭВМ в СВЦ не закончились. Когда теоретические подразделения закончили свою часть работы и центр тяжести по созданию 5Э53 переместился на схемотехников и конструкторов, в СВЦ началась работа по созданию образа новой мощной вычислительной системы – ЭВМ четвертого поколения (ЭВМ-IV). Задумывалась модульная реконфигурируемая система с аппаратно-микропрограммной реализацией языка программирования высокого уровня типа PL-1 и IPL, считавшихся тогда наиболее перспективными. ЭВМ включала подсистемы центральной обработки (до 16 центральных процессоров – ЦП), ввода-вывода (до 16 процессоров ввода-вывода – ПВВ), ОЗУ (до 32 секций ОЗУ 32Кх64 бит) и мощную модульную систему динамичной коммутации перечисленных модулей по сложному графу (любой ЦП мог быть соединен с любым ПВВ и любой секцией ОЗУ). Общая производительность ЭВМ оценивалась в 200 млн. оп/с. В ЦП планировалась табличная реализация СОК: результат не вычисляется, а считывается из ПЗУ – в СОК это возможно. При этом любая непрерывная функция одной или двух переменных может выполняться за один машинный такт. Предполагалось использовать парадоксальное свойство СОК – эффективная производительность модулярной ЭВМ может быть многократно выше ее физического быстродействия или производительности позиционной ЭВМ с таким же быстродействием.

Для реализации табличной ЭВМ требовалось компактное постоян-

ное ЗУ большой емкости. Его разработкой в СВЦ уже несколько лет занималось подразделение С.А. Гаряинова. Суть этой работы заключалась в создании бескорпусных диодных матриц, а так же конструкции и технологии изготовления устройств на их основе.

К этому времени в подразделении С.А. Гаряинова была разработана диодная 256-битная матрица на диэлектрической подложке – ДМР-256, на заводе «Микрон» осваивалось ее производство. На основе этой матрицы была разработана соответствующая оригинальная конструкционная система:

- кристаллы ДМР-256 монтировались на ситаловую плату,
- платы собирались в семиэтажную этажерку (МФБ – многофункциональный блок) с межплатным монтажом по четырем ее граням. Этажерки устанавливались на большую печатную кросс-плату.
- несколько кросс-плат с МФБ монтировались в металлический, герметичный корпус блока, заполняемый фреоном. Для вывода тепла из блока в него устанавливались тепловые трубки. В коллективе этот корпус получил название «чемодан».

Таким образом, на фоне бурных событий, связанных, сначала с разработкой, а потом с борьбой за выживание 5Э53, в спокойной обстановке создавался задел для реализации следующего проекта. В это время все внимание Д.И. Юдицкого было сконцентрировано на событиях вокруг 5Э53, но он регулярно интересовался и перспективными проработками, доверяя, в то же время, их руководителям. Как впоследствии выяснилось, не все они оправдали доверие.

Аванпроспект ЭВМ-IV был закончен в начале 1973 года. Эта ЭВМ задумывалась как прототип для последующих разработок СВЦ. Однако еще до его завершения ЭВМ-IV, ей, казалось, нашлось хорошее применение.

Супер-ЭВМ "41-50", "Лидер"

В начале 1972г. СВЦ получил заказ ГРУ МО на разработку эскизного проекта супер-ЭВМ для обработки векторных и структурированных данных, получившей условное наименование 41-50, ОКР «Лидер». 64-разрядная ЭВМ должна была обладать быстродействием в 200 млн. оп/с, иметь ОЗУ емкостью 16М байт, развитую пе-

риферию. В то время за рубежом уже были известны ЭВМ такого типа, например фирмы Burroughs (США), но они были заметно слабее. Это многопроцессорные машины, обрабатывающие одиночным потоком команд множественный поток данных. Основная задача заключалась в распараллеливании данных между процессорами, которую обычно решали на основе традиционных скалярных процессоров, со скалярными системами команд, на программном уровне. В СВЦ строили изначально векторную архитектуру ЭВМ с векторной системой команд, работающих над массивами и ориентированной на реализацию алгоритмов заказчика. Задача динамического распараллеливания при этом решалась на аппаратно-микророграмным уровне, на основе внутренних алгоритмов, что приводило к резкому повышению эффективности системы в целом.

Эскизный проект 41-50 СВЦ выполнял совместно с Институтом Кибернетики (ИК) АН Украины, директор ИК академик В.М. Глушков был научным руководителем проекта. В связи с этим в ИК было создано 2 специальных подразделения (филиал СВЦ) во главе с З.Л. Рабиновичем и Б.Н. Малиновским. Главным конструктором проекта был Д.И. Юдицкий, активное участие в его реализации принимали Н.М. Воробьев, М.Д. Корнев, В.Г. Сиренко, В.А. Савеличев, В.С. Петровский, В.М. Елагин, И.П. Селезнев, П.Н. Казанцев, Ю.М. Сокол, Ю.Г. Бобошко, Ж. Мамаев, В.Ф. Лукин, Т.Г. Родкина и др.

Первоначально планировалось ЭВМ строить на основе задела, выполненного в рамке проекта ЭВМ IV поколения. Этого, по ряду причин, не получилось.

Проектирование 41-50 начинается с изучения алгоритмов решения задач заказчика. Поэтому в первую очередь начали просматривать реализацию специфичных алгоритмов заказчика на основе разработанного варианта табличной реализации модулярной арифметики. Работу возглавили В.М. Амербаев в качестве математика и основного автора модулярной арифметики, и Л.Г. Рыков в качестве схемотехника, реализующего эти алгоритмы. Этот хорошо сработавшийся дуэт дал возможность трезво оценить ситуацию. Вспоминает Л.Г. Рыков: «*И.Я. Акушский был больше математиком и теоретиком и до таких понятий, как время задержки, гонка импульсов и других схемотехнических неприятностей, не опускался. Вильжан Мавлютинович – совершенно другой человек. Он не гнушался наших проблем и всегда старался найти такой вариант*

математического решения, который наиболее удачно реализуется аппаратно». Результаты этого напряженного труда были аккумулированы в Руководящем техническом материале РТМ У10.012.003 «Машинные алгоритмы двухступенчатой непозиционной арифметики». Проведенный анализ показал, что на алгоритмах заказчика (процент логических операций, не выполнявшихся тогда в СОК, в них был значительно выше обычного) эффективная производительность модулярной ЭВМ не превышает производительности обычной двоичной позиционной ЭВМ. Оставалось некоторое преимущество по надежности за счет арифметичности СОК, но в табличной арифметике и это мало что давало, т.к. табличная арифметика реализуется в памяти, в которой хорошо работают традиционные методы обнаружения и исправления ошибок. Таким образом применительно к задачам 41-50 преимущества СОК практически не срабатывали. В результате оправдать применение СОК могло только более удачные конструктивно-технологические решения реализации табличной арифметики на основе полупроводниковой постоянной памяти. Они обещали существенное сокращение объема аппаратуры по сравнению с традиционной двоичной позиционной арифметикой.

Но своевременно задуманный конструктивно-технологический задел не оправдал надежд. Когда он потребовался, выяснилось, что он еще весьма далек от возможности практического применения. Все это в совокупности привело к отказу от применения СОК в проекте 41-50. Начался второй этап реализации проекта на основе традиционной двоичной арифметики, но это уже другая история.

Система 41-50 была последней разработкой в СВЦ высокопроизводительных многоразрядных супер-ЭВМ. Проект был выполнен и блестяще сдан Госкомиссии. Заказчик, фактически соисполнитель, внес в него все, что ему было нужно, высоко оценил проект и верил в успех. Но в МРП, где изначально планировалось и заказчиком было согласовано производство 41-50, изготовителя для нее не нашлось и продолжения работ не последовало. Далее СВЦ занимался созданием изделий, которые можно было производить своими силами, – 16-разрядных мини- и микро-ЭВМ, микропроцессоров и систем на их основе. А в малоразрядных системах преимущества модулярной арифметики не существенны и работы по ее практическому применению в СВЦ были свернуты.

Судьба СОК

В шестидесятых-семидесятых годах прошлого века в связи с разработками ЭВМ К-340А, 5Э53 и ЭВМ-IV в СВЦ и в сотрудничающих с ним предприятиях производились серьезные научные исследования в области модулярной арифметики и было много публикаций на эту тему в открытой печати, в т.ч. и в виде монографий. Они возбудили серьезный интерес у иностранных специалистов. Вот что вспоминает академик В.М. Амербаев: "*В 1970-71 гг. большой интерес к модулярной арифметике проявили банковские структуры США. Им требовались высокопроизводительные средства для высоконадежных вычислений с самокоррекцией – именно этим и характерна модулярная арифметика. По данным открытой печати (статьи, книги, патенты) они оценили результаты работы И.Я. Акушского и Д.И. Юдицкого как передовые в мире и обратились в МЭП с предложением о закупке модулярных алгоритмов (предложили около 20 млн. долларов США). Начавшиеся переговоры были пресечены «компетентными органами»*». Об этом же случае, а возможно и о другом, вспоминает уже цитированный нами В.С. Линский: "*Во время работы в НИИ ФП - СВЦ в 1966-70 гг. я открыто выражал негативное отношение к СОК, вплоть до обращения в Военно-промышленную комиссию при СМ СССР (ВПК). С моим мнением был ознакомлен В.С. Бурцев, выразившийся в том смысле, что однозначный ответ о СОК преждевременен. На вопрос сотрудников ВПК о том, почему американцы хотят закупить результаты И.Я. Акушского и Д.И. Юдицкого, я ответил, что по-видимому это им выгоднее, чем самим проводить исследования в этой области*". А.В. Пивоваров вспоминает другой случай: "*У Юдицкого был контакт с французской фирмой, не помню ее название, которая пожелала купить проект ЭВМ. Д.И. Юдицкий пришел ко мне за разрешением на такую сделку, но я отказал ему по двум причинам. Во-первых для выполнения такой сделки необходимо изготовление образца ЭВМ для полной отработки технологии, а сделать то его было не где. Во-вторых – зачем нам вооружать французов, тогда наших потенциальных военных противников. Да если бы я и согласился, нам все равно бы это не позволили сделать вышестоящие органы*". Были и другие примеры интереса зарубежных фирм к работам СВЦ по СОК, но все они были пресечены "в установленном порядке".

Прекращение работ по 5Э53 вызвало определенный психологиче-

ский шок у сторонников СОК, их научная активность существенно снизилась, число открытых публикаций резко сократилось. Имеются свидетельства, что этот факт был замечен зарубежными учеными и их "компетентными органами", сделавшими вывод о засекречивании этих работ в СССР (истинных причин они не знали). Некоторые страны, например США, последовали этому "примеру" и засекретили работы по модулярной арифметике у себя.

Таким образом, печальная судьба 5Э53 стала причиной пресечения нового, перспективного направления развития отечественной вычислительной техники, превосходящего все имевшееся и в стране, и за рубежом – модулярной арифметики. Истинных причин остановки ЭВМ 5Э53 практически никто не знал. Но сам факт, получив широкую огласку в кругах специалистов, начал самостоятельную жизнь и стал почти непреодолимым барьером на дальнейшем пути внедрения СОК в отечественную вычислительную технику. Далее модулярной арифметикой в нашей стране занимались только отдельные энтузиасты, в основном, в теоретическом плане.

Уровень элементной базы шестидесятых-семидесятых годов прошлого века (электронные лампы, транзисторы и диоды, интегральные схемы низкой и средней интеграции) не позволял создавать ЭВМ с характеристиками, полностью удовлетворяющими потребителя. Каждая ЭВМ того периода была результатом компромисса между желаемым и возможным. Именно поэтому разработчики ЭВМ искали самые разнообразные методы повышения их производительности и надежности. Одним из таких методов была модулярная арифметика, и именно поэтому и именно тогда она вызывала повышенный к себе интерес и получила интенсивное развитие. В восьмидесятые годы с появлением микропроцессоров и других интегральных схем все возрастающей интеграции, существенно сгладились проблемы и производительности, и надежности ЭВМ (исчезли километры проводов и миллионы паек). В настоящее время подавляющее число потребителей использует лишь малую часть возможностей своих ЭВМ и не подозревают, что проблемы производительности и надежности были когда-то очень актуальны и часто непреодолимы. И поиски путей их преодоления существенно сократились. Последние 20-30 лет в мире почти не появилось новых архитектурных решений и других системных новаций в принципах построения ЭВМ – практически используется задел шестидесятых-семидесятых годов. Колоссальный прогресс вычислитель-

ной техники определяется, в основном, микроэлектроникой.

Но в настоящее время развитие вычислительной техники, похоже, подходит к очередному кризису. Вызвано это многими следующими причинами:

- Во-первых, ее широкое проникновение во все сферы жизнедеятельности человека резко повысило актуальность решения таких, ранее редких, а теперь массовых задач, как обработка сигналов, изображений, распознавания образов, криптографии, обработка многоразрядной информации и т.п. Все они требуют огромных вычислительных ресурсов, часто превышающих возможности.
- Во-вторых, традиционная микроэлектроника подходит к пределу своих технологических возможностей, размеры ее элементов измеряются нанометрами, числом атомов. А идущие ей на смену наноэлектроника, молекулярная электроника, микромеханика, биоэлектроника и т.п. находятся в "эмбриональном" состоянии, еще далеки от промышленного применения и их перспективы оцениваются по-разному. Старшее поколение специалистов помнит радужные прогнозы оптимистов об "ошеломляюще высоких" возможностях оптических ЭВМ – молодежи о них и не рассказывают: оптические ЭВМ не состоялись.
- В-третьих – остро встает проблема безопасности. Об этом еще далеко недостаточно говорят, но для России это проблема национальной. Применение зарубежной электроники в стратегически важных системах таит в себе огромную скрытую потенциальную угрозу. Современный уровень микроэлектроники, когда в кристалле одной интегральной схемы содержатся миллионы транзисторов, функционально законченные устройства и системы, обеспечивает и возможности введения диверсионных "закладок". Компьютер с такой "закладкой" может многие годы прекрасно работать, а "закладка" будет спать. Но в нужный кому-то момент, по сигналу извне (Internet, радиосигнал и т.п.) она просыпается и творит с системой все, что захочет хозяин "закладки". Обнаружить такие "закладки" практически невозможно. Эта задача по силам только мощнейшим в мире микроэлектронным фирмам, стоимость такой операции соизмерима со стоимостью создания исследуемой микросхемы. При обилии номенклатуры таких микросхем задача становится непосильной

для экономики любой страны. В настоящее время ни кто не может дать гарантии, что в компьютерах Генштаба, Банка России, Правительства, Федерального собрания и других стратегически важных органов не "спят" диверсионные "закладки", и что они не проснуться в самый неподходящий для страны момент. Выход только один – в создании отечественных изделий микроэлектроники для стратегически важных систем. Только здесь процесс можно полностью контролировать и исключить появление "закладок". Но поскольку технологически мы отстаем от зарубежной микроэлектроники, необходимо привлекать другие средства повышения эффективности систем.

В этих условиях интерес к поиску системных методов повышения эффективности вычислительных средств пробуждается вновь. В печати заметно увеличилось количество соответствующих публикаций, в том числе и по модулярной арифметике. Ряд серьезных фирм начал, пока теоретические, задельные работы в этой области. В этой связи интересно и полезно знать историю и современное состояние отечественной модулярной арифметики.

Настоящая статья является попыткой комплексно отразить первую страницу истории зарождения и развития отечественной модулярной арифметики. Естественно, она далеко не полная и, наверное, в чем-то ошибочная. Но автор старался быть объективным, опираясь на сохранившиеся документы и воспоминания активных участников событий.

О троичных ЭВМ

В те же годы в нашей стране свершилась еще одна аналогичная трагедия. Было успешно начато и так же трагично оборвано еще одно прогрессивное направление вычислительной техники – троичные ЭВМ. Однако сначала намного истории.

Троичная бумажная логическая «машина» Лулия

Первое известное авторам упоминание о реальном применении троичной системы счисления относится к XIII веку. Тогда троичную логическую «машину» на бумаге в виде круговых диаграмм с секторами создал Раймунд Луллий (1235-1315 гг.). Прменялась ли эта бумажная «машина» реально – не известно. Известно, что Луллия забили камнями.

Троичная деревянная счетная машина Томаса Фулера

Первым упоминанием о технической реализации троичного счетного устройства является оп описание счетной машины Томаса Фулера сделанное Августусом ДеМорганом в 1840г. [].

Томас Фулер (Thomas Fowler, 1777 – 1843) был талантливым и известным изобретателем своего времени. Одним из самых значительных его изобретений была система центрального отопления, подобная современным. Но из-за ошибки при патентовании, его идея была украдена. Этот случай сделал его более скрытным в своих исследованиях. Так разработку троичной счетной машины Томас Фулер держал в секрете до самого конца, и лишь завершив её, пригласил известнейших людей своего времени (в том числе и ДеМоргана) на демонстрацию.

История создания счетной машины Томасом Фулером началась с написания «Таблиц для упрощения вычислений». Работая казначеем, Фулер постоянно имел дело с финансовыми вычислениями, осложненными не десятичной английской денежной системой. Фулер обнаружил, что подобные вычисления гораздо проще выполнять над числами с малыми основаниями (два и три). Углубившись в изучение этого вопроса, в 1838г. он составил и издал, таблицы для упрощения вычислений.

Книга Фулера содержала таблицы двоичных чисел в десятичном диапазоне от 1 до 130048 и таблицы троичных чисел в симметричном коде в десятичном диапазоне от 1 до 3985807. Кроме того, книга содержала подробные инструкции по выполнению ряда операций с помощью этих таблиц.

К 1840 году Фулер разработал и построил работающий макет счетной машины. Машина оперировала трехразрядными (в современной терминологии) троичными числами со знаком, и выполняла над ними операции умножения и деления. В совокупности с опубликованными таблицами, с помощью этой машины можно было выполнять вычисления над большими числами. Сам Фулер отмечал, что применение его машины эффективно лишь при большом количестве повторяющихся вычислений.

Макет, созданный Фулером, был изготовлен из дерева, и имел размеры 180 x 90 x 30 сантиметров. Такие большие размеры объяснялись низкой механической точночтью, которую обеспечивало де-

рево. По словам Фулера, металлический вариант машины был бы не больше настольной печатающей машинки.

В 1999 г. группа энтузиастов, наткнувшись на описание машины, сделанное ДеМорганом, и воссоздала машину Томаса Фулера. Работающий макет счетной машины Томаса Фулера хранится в Большом Торрингтонском музее в Аглии.
http://www.mortati.com/glusker/fowler/index.htm

Троичная ЭВМ «Сетунь»

В 1956 г. академик Сергей Львович Соболев, блестящий математик, широко эрудированный ученый, одним из первых в стране понявший значение вычислительной техники, в то время заведующий кафедрой вычислительной математики Московского государственного университета им. М.В. Ломоносова, выступил с инициативой разработки в МГУ ЭВМ. Предполагалось создать малогабаритную, недорогую, непритязательную в обслуживании и простую в использовании машину для учебных заведений, исследовательских лабораторий, конструкторских бюро для решения научно-технических задач средней сложности, а также для управления технологическими процессорами. С этой целью в механико-математическом факультете была создана специальная лаборатория, руководство разработкой ЭВМ Сергей Львович возложил на молодого инженера Николая Петровича Брусенцова.

Сергей Львович организовал семинар, в котором участвовали Михаил Романович Шура-Бура, Константин Адольфович Семендяев и

другие крупные математики. Как вспоминает Н.П. Брусенцов: «*Мы с моим коллегой Евгением Жоголевым изобретали, а старшие товарищи наводили на нас критику. Это, кстати, нам очень помогало двигаться вперед*». С.Л. Соболев устроил Н.П. Брусенцову стажировку в лаборатории у Льва Израилевича Гутенмахера в ИТМ и ВТ АН СССР, где как раз создавалась двоичная ЭВМ на магнитных элементах. «*Именно тогда у меня возникла мысль использовать троичную систему счисления. Она позволяла создать очень простые и надежные элементы, уменьшала их число в машине в семь раз по сравнению с элементами, используемыми Гутенмахером. Существенно сокращались требования к мощности источников питания, к отбраковке сердечников и диодов, и, главное, появилась возможность использовать натуральное кодирование чисел вместо применения прямого, обратного и дополнительного кодов чисел. После стажировки я разработал и собрал схему троичного сумматора, который оказался надежным и сразу же заработал. Соболев, узнав о моем намерении создать троичную ЭВМ, благословил меня и пообещал всю возможную помощь. Летом 57-года на пляже в Новом Афоне я прорисовал в тетрадке все детали будущей машины*» (Н.П. Брусенцов).

Так началась история первой в мире троичной ЭВМ «Сетунь», названной по имени речки, протекавшей около университета. Сетунь была машиной последовательного действия с блоком быстрого умножения. Ее главные особенности:

- троичная симметричная (с положительными и отрицательными значениями цифр) система представления чисел и команд,
- трехзначная логика,
- страничная двухуровневая организация памяти,
- пороговая реализация трехзначной логики на электромагнитных элементах с двухпроводной передачей трехзначных сигналов,
- длина операндов 9 и 18 тритов, троичный порядок числа с плавающей запятой – 5 тритов,
- система команд – 24 команды.

Приступая к созданию Сетуни ее разработчики изначально ставили перед собой задачу построения недорогой, простой в освоении, обслуживании и использовании, малогабаритной, надежной машины для вузов, КБ, лабораторий. В качестве основного запоминающего устройства использовали магнитный барабан ЭВМ "Урал, связанный с небольшой памятью (в современном понимании cash) по-

страничным обменом, что позволило, как минимум в 10-20 раз увеличить производительность машины. Важнейшим фактором обеспечения простоты и практичности машины явилось представление чисел и команд в троичном симметричном коде. При длине слова 9 тритов (1 трит соответствует 1,58 бита) и наборе всего лишь из 24 команд она была весьма эффективна при реализации широкого спектра алгоритмов, в т.ч. с плавающей запятой.

МЦВМ "Сетунь"

Малая цифровая вычислительная машина "Сетунь"

- Главный конструктор: Брусенцов Н.П.; основные разработчики: Жоголев Е.А., Маслов С.П., Веригин В.В.
- Организация-разработчик: ВЦ МГУ им. М. В. Ломоносова.
- Завод-изготовитель: Казанский завод математических машин ГКРЭ. Изготовитель логических элементов - Астраханский завод электронной аппаратуры и электронных приборов ГКРЭ.
- Год окончания разработки: 1959.
- Годы выпуска: 1961 – 1965, выпущено около 50 ЭВМ.
- Описание машины: одноадресная с одним аккумулятором, регистром множителя и индекс-регистром, значение которого используется с изменением либо без изменения знака. Длины операндов - 9 тритов и 18 тритов, троичный порядок

числа с плавающей запятой - 5 тритов. Два скоростных фотоввода с перфоленты, ленточный перфоратор, электроуправляемые пишущие машинки с русским и латинским алфавитами.

- Элементная база: электромагнитные пороговые с положительными и отрицательными весами входов элементы типа быстродействующих магнитных усилителей импульсов тока на ферритовых сердечниках и диодах. Тактовая частота - 200 кГц.
- Конструкция ЭВМ: модульная, шкаф-стойка с габаритами 2,9x1,85x0,5 м, съемные субблоки (конструктив ЭВМ М-20), вмещающие до 18 плат с логическими элементами.
- Технология: в условиях значительного разброса значений физических параметров, примененных в логических элементах диодов и ферритовых сердечников (которые поставщиками по существу не контролировались), введена сортировка тех и других на попарно соответствующие друг другу группы, благодаря чему производство элементов было практически безотходным, а их параметры жестко стандартизованными. При дальнейшей сборке из таких элементов логических узлов (субблоков) и блоков машины требовалась только правильность проводных соединений, проверяемая на стендах логического контроля.
- Программное обеспечение: основными средствами автоматизации программирования для машины "Сетунь" являются созданные под руководством Е. А. Жоголева так называемые "интерпретирующие системы":
- - ИП-2 для вычислений с 8-ю десятичными знаками в диапазоне 10^{-19} - 10^{+19}
- - ИП-3 для вычислений с 6-ю десятичными знаками в том же диапазоне,
- - ИП-4 для вычислений с комплексными числами (плавающая запятая, 8 десятичных знаков),
- - ИП-5 для вычислений с 12-ю десятичными знаками в диапазоне 10^{-19} - 10^{+19}.
- Кроме того, в ПО для "Сетуни" входили система ИП-Н СибНИИЭ, осуществляющая полную интерпретацию набора трехадресных команд машины М-20, другие ИП, создан-

ные в организациях - пользователях машины; автокод ПОЛИЗ с символьным языком программирования типа польской инверсной записи.

- Было выпущено более 30-ти брошюр в серии "Математическое обслуживание машины "Сетунь", в которых представлен широкий набор стандартных программ решения типовых математических, а также прикладных задач, автоматизированных систем статистической обработки, моделирования и т. п.
- Технико-эксплуатационные характеристики: потребляемая мощность - 2,5 кВА, площадь для размещения - 25-30 кв. м, функционирует при 15-30°С, заводская цена 27,5 тыс. руб.
- Машина последовательного действия с блоком быстрого умножения. Время выполнения операций: сложение-вычитание - 180 мкс, умножение, в частности с прибавлением третьего операнда либо с суммированием в аккумуляторе - 320 мкс, передача управления - 100 мкс. Оперативное ЗУ - 162 слова по 9 тритов. Память на МБ - 1944 либо 3888 слов по 9 тритов. Среднее время страничного (54 слова) обращения к МБ - 7500 мкс.
- Особенности ЭВМ: троичная симметричная (с положительными и отрицательными значениями цифр) система представления чисел и команд, трехзначная логика, страничная двухуровневая организация памяти, пороговая реализация трехзначной логики на электромагнитных элементах.
- Разработка "Сетуни" защищена 10-ю авторскими свидетельствами, удостоена Диплома первой степени и Большой золотой медали ВДНХ СССР.

Первый ее экземпляр, сделанный руками сотрудников лаборатории Н.П. Брусенцова: Е.А. Жоголевым, С.П. Масловым, В.В. Веригиным, В.С. Березиным, Б.Я. Фельдманом, Н.С. Карцевой, А.М. Тишулиной, В. П. Розиным и др., был готов к концу 1958 г. На десятый день комплексной наладки машина заработала. Такого в практике наладки разрабатываемых в те годы ЭВМ еще не было.

Но в практике тех времен было другое – директивное управление. Осенью 1959 года, когда ЭВМ уже работала, представителей МГУ

пригласили на Коллегию Государственного Комитета Радиоэлектроники - ГКРЭ. И там им сообщили, что с целью экономии государственных средств работы над ЭВМ «Сетунь» прекращаются. На замечание С.Л. Соболева: "*А вы хотя бы видели эту машину, ведь она уже существует?*", директор СКБ-245 В. В. Александров ответил: "*Нам не надо ни видеть, ни знать - должна быть авторитетная бумага с печатями и подписями*". После Коллегии Сергей Львович пошел в ЦК КПСС. Было принято решение провести межведомственные испытания ЭВМ, которые были проведены в апреле 1960 г. На них "Сетунь" показала 95% полезного времени (время, которое ЭВМ решает задачи по назначению, остальное время – ремонт в результате отказов: надежность ЭВМ тогда была низкая, 60% полезного времени считалось очень хорошим результатом).

Постановлением Совмина СССР серийное производство Сетуни было поручено Казанскому заводу математических машин. Элементы выпускались на Астраханском заводе и делали их там превосходно, стоил элемент 3,5 руб. Никаких высоких технологий там не было. Но руководство Казанского завода с самого начала относилось к Сетуни негативно. Она была слишком дешевой машиной, в действовавших тогда экономических условиях для завода весьма невыгодной. Выпускали по 12–20 машин в год, а вскоре и от этого отказались, всего было выпущено около 50 ЭВМ. Примерно 30 из них были поставлены в ВУЗы, остальные – в различные НИИ. Для сравнения сопоставим наиболее популярную в те годы и считавшуюся дешевой ЭВМ PDP-8 фирмы DEC США с Сетунью. Процессор PDP-8 – восьмибитный, у "Сетуни" процессор в пересчете на биты был 30-битным. PDP-8 стоила 20 тысяч долларов без всякой периферии, только один процессорный блок. Сетунь стоила 27,5 тысяч рублей со всей периферией.

Особый интерес к Сетуни проявили в Чехословакии. Они считали, что могли хорошо продавать Сетунь в соответствии с рыночными ценами и получать при этом высокую прибыль. По их приглашению Н.П. Брусенцов ездил в Чехословакию, где ему показали завод "Зброевка Яна Швермы", на котором планировалось выпускать Сетунь. Завод его просто восхитил. Чехи уже приготовили для Сетуни магнитные барабаны, печатающее устройство, устройство ввода. В общем, все было готово для производства ЭВМ, не было только документации на нее. Но в правительстве повели себя, как собака на сене. Когда Н.П. Брусенцов вернулся из Чехословакии, его вы-

звал референт председателя совета министров А.Н. Косыгина и попросил «*передать чешским товарищам, что документацию на Сетунь они получат сразу после освоения крупносерийного производства этой машины в Советском Союзе*». Т.е. никогда. Как вспоминает Николай Петрович: «*Позднее я узнал, что чехам говорили: все равно мы эту машину снимем с производства, так что вы ее не заказывайте. Вот так все и закончилось с Сетунью. В начале 70-х нас из главного корпуса ВЦ переселили на чердак общежития. А Сетунь, несмотря на то, что она была полностью исправной и загруженной задачами, через пару лет была уничтожена - ее разрезали и выкинули на свалку*».

Сработало лобби ГКРЭ: они не желали допускать в свои производственные мощности чужого главного конструктора, тем более со странной троичной ЭВМ, превосходящей по характеристикам ЭВМ того же класса их родных конструкторов. Т.е. произошло то же, что и с модулярными ЭВМ Юдицкого, только еще откровеннее.

Тот факт, что Сетунь надежно и эффективно работала во всех климатических зонах от Калининграда до Магадана и от Одессы до Якутска, причем без какого-либо сервиса и по существу без запасных частей, в расчет не принимался. Как вспоминает Н.П. Брусенцов: «*В Якутске "Сетунь" была в астрофизическом институте. У них была какая-то сложная задача, которую они в течение двух лет не могли поставить на большой машине "Урал-2". Потом кто-то сказал: "Давайте попробуем на "Сетуни". Все решили, что это шутка. Однако через полтора месяца задача была решена*».

И то, что на поставку Сетуни было очень много заявок не только внутри страны, но и из-за рубежа, и не только из соцстран, но также и из таких стран, как США и Англия, так же во внимание принято не было. Производство Сетуни было прекращено.

Закончим информацию о троичной ЭВМ «Сетунь» цитатой Н.П. Брусенцова из его статьи «Заметки о троичной цифровой технике» http://www.computer-museum.ru/histussr/12-2.htm:
«…*Экспериментальный образец машины "Сетунь", построенный в 1957 - 1958 гг., находился в эксплуатации 15 лет, причем из 4 тыс. использованных в нем пороговых элементов типа быстродействующих магнитных усилителей с питанием импульсами тока, были заменены вследствие отказов только 3 элемента (все 3 на первом году эксплуатации): 2 из-за пробоя диодов типа Д1, по-видимому,*

обладавших дефектами изготовления, и 1 из-за нарушения изоляции между обмотками импульсного трансформатора. Машина устойчиво работала при значительной нестабильности напряжения питающей электросети и в достаточно широком диапазоне температур окружающей среды (от +15 до +30° С).

Серийные экземпляры машин "Сетунь" успешно эксплуатировались в различных климатических зонах с холодным, с жарким, и с резко континентальным климатом (например, в Ашхабаде, Душанбе, Махачкале, Иркутске, Якутске, Одессе), причем без какого-либо сервисного обслуживания и практически без запасных частей. Едва ли это может свидетельствовать о плохой надежности аппаратуры.

… Принципиальные результаты разработки "Сетуни" настолько естественны и немудрены, что, с точки зрения формальных теоретиков многозначной логики, их попросту не видно. Похоже, что именно этим объясняется … заключение о слабости логических возможностей "Сетуни" ... Результаты состояли в том, что было экспериментально доказано, что троичная машина, по меньшей мере в условиях электромагнитной техники, оказывается существенно экономнее, быстрее, проще и математически совершенней функционально эквивалентной двоичной машины, выполненной на элементах того же типа.

Кроме того, было показано, что троичные устройства могут быть эффективно и просто реализованы на основе способа выполнения логических операций, названного впоследствии пороговой логикой, причем именно в трехзначном варианте с положительными и отрицательными весами логических входов данный способ становится практически приемлемым, благодаря значительному ослаблению требований к точности и стабильности параметров физических элементов и сигналов. Не менее важным было и то, что трехзначная логика с ее 33 одноместными и 39 двуместными операциями, трактуемая некоторыми философами как логика таинственного микромира, предстала перед инженером как давно известная ему логика положительного, отрицательного и равного нулю тока (или заряда), а перед программистом - как логика элементарных чисел: 0, 1, - 1 или логика значений, принимаемых алгебраическим знаком числа: +, -, 0. При этом выяснилось, что, хотя эта трехзначная логика сложнее двузначной, она вместе с тем удобнее для человека, легче осваивается и применяется.

… при внимательном рассмотрении выявляются следующие принципиальные и важные для практики особенности троичной ЭВМ:

1. *Числа всех типов представлены единым натуральным кодом. В двоичных машинах для представления чисел разных типов и даже разного назначения приходится применять различный код, например: числа с фиксированной запятой представляют обычно дополнительным или обратным кодом, мантиссы чисел с плавающей запятой - прямым кодом, порядки - кодом с избытком, адреса памяти - натуральным двоичным кодом.*

2. *Операции определены над операндами, длина которых варьируется и может быть неодинаковой у первого и второго операнда. При этом не требуется никаких усложнений операционного устройства, ни вспомогательных команд вроде двоичной команды "расширения знака".*

3. *При усечении слова, например в случае присваивания длинного значения короткому регистру, автоматически получается наилучшее при данной укороченной длине представление первоначального значения, и вместе с тем сохраняемая часть слова копируется неизменной. В двоичной машине соответствующие возможности с известными оговорками можно обеспечить, лишь предусмотрев два варианта операций - с округлением и без округления.*

4. *Единственная операция сдвига выполняет функции всех двоичных операций сдвига - логического, арифметического, с округлением и без округления, причем выполняет безукоризненно, чего нельзя сказать, например, об операции двоичного арифметического сдвига [6].*

5. *Знак числа в соответствии с общепринятой математической трактовкой этого понятия представлен трехзначной функцией, и разбиение чисел по знаку производится на положительные, отрицательные и равные нулю, в противоположность сбивающей с толку двоичной традиции двузначного знака и отнесения нуля к положительным числам.*

6. *Интервал значения мантиссы нормализованного числа - от 0,5 до 1,5 по абсолютной величине - характеризуется значительно лучшей устойчивостью по сравнению с используемыми в двоичных машинах интервалами 0,5 - 1,0 и 1,0 - 2,0.*

Эти и ряд других особенностей троичной архитектуры обусловили математическое совершенство, компактность и простоту реализации набора команд Сетуни, а главное, естественность, легкость понимания и применения машины пользователями».

Троичная ЭВМ «Сетунь-70»

В 1961–1968 гг. Н.П. Брусенцов вместе с Е.А. Жоголевым разработали архитектуру новой машины, названной затем "Сетунь-70". Было намечено к 1970 г. разработать действующий образец, и в апреле 1970 г. образец уже работал – к 100-летию со дня рождения В.И. Ленина все должны были делать производственные подарки, это был подарок от МГУ.

В техническом отношении "Сетунь 70" характеризуется рядом усовершенствований по сравнению с "Сетунью". Она имела двухстековый троичный процессор с послоговым кодированием программ и данных, идентификаторами операций и адресов служат трайты (шестерки тритов). Набор операций включает 81 операцию: 27 основных (тестирование и преобразование данных, управление ходом программы), 27 служебных (управление магнитным барабаном, внешними устройствами, системой прерываний), 27 макроопераций, микропрограммируемых пользователями. Элементная база: электромагнитные пороговые логические элементы с однопроводной передачей трехзначных сигналов. Это позволило почти в 2 раза уменьшить число электрических соединений, логические элементы стали проще, миниатюрней и потребляли в 2,5 раза меньше энергии, значительно улучшены параметры троичной памяти и магнитной записи троичного кода.

Сетунь-70 была задумана так, что обеспечивалась эффективная возможность ее программного развития. Команд в традиционном понимании не было – они виртуально складывались из слогов. Длина и адресность команд варьировалась по необходимости, начиная с нульадресной. На самом деле программист не думал о командах, а писал в постфиксной форме (ПОЛИЗ) выражения, задающие вычисления над стеком операндов. Для процессора эти алгебраические выражения являлись готовой программой, но алгебра дополнена операциями тестирования, управления, ввода-вывода. Пользователь мог пополнять набор слогов своими операциями и вводить (определять) постфиксные процедуры, использование ко-

торых практически не снижало быстродействия, но обеспечивало идеальные условия для структурированного программирования. Результат – трудоемкость программ уменьшалась в 5–10 раз при небывалой надежности, понятности, модифицируемости и т. п., а также компактности и скорости.

Минимальная непосредственно адресуемая единица главной памяти Сетуни 70 - 6-тритный трайт (~9,5 бита) - на редкость удобна. Трайт лишь немногим больше 8-битного байта, но уже достаточно велик, чтобы закодировать, например, алфавит, включающий русские и латинские заглавные и строчные буквы, цифры, математические и служебные знаки. В трайте целое число как 9-ричных, так и 27-ричных цифр. Два трайта - это 19 битов, три трайта - почти 29 битов и т. д.

Представление чисел симметричным троичным кодом позволило легко реализовать последовательное выполнение арифметических операций с варьируемой длиной операндов от 1 до 3 трайтов и длиной результата до 6 трайтов. Благодаря симметричности кода просто и естественно реализованы реверсивные счетчики и указатели стеков, играющих в архитектуре машины важную роль.

Существенной чертой Сетуни-70 является стековая организация процессора. Введение арифметического стека, т. е. магазина для автоматического запоминания промежуточных результатов, было обусловлено выбором в качестве языка машины польской инверсной записи (ПОЛИЗ), которую предпочли как удобный выходной язык трансляторов и вследствие ее компактности. Машинная программа представляется в ПОЛИЗ последовательностью слов (или слогов), в которой различаются операционные и адресные слова.

В 1975 г. машина подверглась модернизации, выразившейся в переделки небольшие по объему, но приведшие к принципиальному усовершенствованию архитектуры на основе идеи структурированного программирования Э. Дейкстры. Двухстековая организация процессора и ПОЛИЗ оказались исключительно благоприятными для реализации структурированного программирования на уровне языка машинных команд. При этом в условиях новой дисциплины программирования стали несущественными затруднения, возникавшие в связи с мелкостраничной структурой памяти.

Все, что потребовалось сделать - это ввести команды ветвления, цикла и вызова подпрограммы вместо практически не употребляв-

шихся команд приращения, убавления и установки нуля в регистре порядков. Новые команды, в отличие от обычных в ПОЛИЗ однословных команд, представлены словосочетаниями.

ЭВМ "Сетунь 70"

Малая цифровая вычислительная машина "Сетунь 70"

- Главный конструктор: Брусенцов Н. П.; основные разработчики: Жоголев Е. А., Маслов С. П., Рамиль Альварес Х.
- Организация-разработчик: Вычислительный центр Московского государственного университета им. М. В. Ломоносова. Ведомство: Министерство высшего образования СССР.
- Год окончания разработки: 1970.
- Год начала выпуска: машина серийно не выпускалась.
- Год прекращения производства: опытный образец машины "Сетунь 70" функционировал в составе автоматизированной системы обучения "Наставник" на факультете ВМиК МГУ до замещения его серийным микрокомпьютером "Электроника НЦ 80-20" (ДВК-2) в 1987 г.
- Область применения: решение научно-технических задач средней сложности; основание для микропрограммной реализации специализированных систем. На основе опытного образца машины созданы диалоговая система структуриро-

ванного программирования ДССП и автоматизированная система обучения "Наставник", эмулируемые в дальнейшем на серийных компьютерах.

- Число выпущенных машин: один опытный образец.
- Описание машины: двухстековый троичный процессор с послоговым кодированием программ и данных - идентификаторами операций и адресов служат трайты (шестерки тритов), последовательность которых представляет собой программу в польской инверсной (постфиксной) записи. Набор операций включает 81 операцию - 27 основных (тестирование и преобразование данных, управление ходом программы), 27 служебных (управление магнитным барабаном, внешними устройствами, системой прерываний), 27 макроопераций, микропрограммируемых пользователями. Память с непосредственным доступом состоит из девяти страниц по 81 трайту ОЗУ и 18-ти страниц ПЗУ. Магнитный барабан с постраничным обращением емкостью 972 страницы (в опытном образце машины задействовано 243). Каналов ввода-вывода три, до восьми устройств в каждом. На опытном образце ввод/вывод перфолентный и посредством электроуправляемой пишущей машинки "Консул 254". К машине был подключен также класс "Наставник" с 27-ю терминалами учащихся, оборудование для диагностики цветного зрения и устройство для оцифровывания графиков.

- Элементная база: электромагнитные пороговые логические элементы с однопроводной передачей трехзначных сигналов.
- Конструкция ЭВМ: модульная, шкаф-стойка 1,8x1,5x0,5 м, съемные платы с логическими элементами, до 40 элементов на плате.
- Технология: однопроводная передача трехзначных сигна-

лов сократила почти в два раза количество межэлементных и межблочных соединений.

- Программное обеспечение: операционная система, выполняющая функции загрузчика, отладчика и монитора, организацию обмена с магнитным барабаном и осуществление макроопераций, предоставляет пользователю макрорасширяемый редактор текстов, однопроходный ассемблер с входным языком структурированного программирования и библиотеку сервисных программ, призванных повысить эффективность разработки и облегчить использование программных систем. Наиболее широкое практическое применение получила автоматизированная система обучения "Наставник", которая явилась весьма действенным средством группового обучения теоретическим дисциплинам, проведения автоматизированных контрольных работ, коллоквиумов, экзаменов и различного рода тестов.
- Технико-эксплуатационные характеристики: потребляемая мощность - 1,5 кВА, площадь для размещения - 15-20 кв. м, производительность - 5-6 тыс. операций в се-кунду.
- Особенности ЭВМ: троичная симметричная система представления данных и программ, трехзначная логика в пороговой реализации на электромагнитных элементах с однопроводной передачей сигналов, страничная двухуровневая организация памяти, двухстековая архитектура, послоговое кодирование программ, управление ходом программы в духе структурированного процедурного программирования.

ЭВМ «Сетунь-70» была последней попыткой внедрения троичной системы в вычислительную технику, не понятой и не принятой современниками. И судьба у нее была хуже, чем у предшественницы, ее даже не пытались производить серийно. Мало того, пресекли всякую возможность дальнейших работ в этом направлении. Директивные указания были подкреплены практическими действиями: лаборатория Н. П. Брусенцова вскоре после создания машины "Сетунь-70" была выселена из помещения ВЦ МГУ на чердак студенческого общежития. Первое детище Николая Петровича – машина "Сетунь" (экспериментальный образец, проработавший безотказно 17 лет) была варварски уничтожена – ее разрезали на куски

и выбросили на свалку. Сетунь-70 сотрудники лаборатории забрали с собой на чердак и там на ее основе создали "Наставник" – уникальную систему обучения с помощью компьютера.

На этом история создания троичных ЭВМ в стране была директивно прервана.

В настоящее время интерес к троичной системе, по тем же причинам, что и к модулярной арифметике, в зарубежных средствах информации заметно возрос. Появилось множество публикаций и сообщений о проводимых исследованиях. В частности американские ученые пришли к выводу о троичности нейрона человеческого мозга.

Аналогичная ситуация и у нас: МГУ задумался о создании новой троичной ЭВМ, Росэлектроника пригласила на свой НТС Н.П. Брусенцова, ОАО «Ангстрем», совместно с Николаем Петровичем, ведет предварительную проработку путей построения троичной элементной базы, Санкт-Петербургский государственный политехнический университет проявляет интерес к созданию троичных ЭВМ для разрабатываемых им систем. Недавно Николай Петрович Брусенцов предложил объединить положительные стороны модулярности и троичности. Такой синтез заинтересовал академика В.М. Амербаева.

За истекшие после директивного прекращения работ над троичными ЭВМ годы Николай Петрович Брусенцов провел огромную работу по изучению и развитию троичной логики, начиная с логики Аристотеля. Ему удалось вскрыть многочисленные ошибки в толковании силлогистики Аристотеля, допущенные стоиками и их последователями, исказившими трехзначную логику Аристотеля. Введя хрисиппов принцип двухзначности, устраняющий диалектику путем “исключения третьего”, они превратили диалектическую трехзначную логику Аристотеля в схоластическую двоичную, в результате чего в его учении были «обнаружены» «ошибки» и «парадоксы». Николаю Петровичу удалось развить учение Аристотеля, построить строгую диалектическую трехзначную логику, соответствующую естественной человеческой логике, обеспечивающую возможность построения алгоритмов решения задач, ранее не находивших эффективных решений на двоичных компьютерах. Таким образом Н.П. Брусенцовым создана обновленная, лишенная парадоксов и ограничений трехзначная диалектическая логика –

мощная теоретическая база для практического построения троичной ЭВМ. Эту логику с полным основанием можно назвать трехзначной диалектической логикой Брусенцова

В МГУ. Обсуждение путей построения троичной ЭВМ.
Слева направо: Н.П. Брусенцов, П.Р. Машевич (зам. управляющего директора – директор по НИОКР и госзаказу ОАО»Ангстрем»),
академик В.М. Амербаев, аспирант Д.Б. Малашевич

.Показательны два факта, иллюстрирующие повышения интереса зарубежных специалистов к троичной системе в наши дни.

Первый приведен украинским виртуальным компьютерным музеем в экспозиции «Николай Петрович Брусенцов - творец первого и единственного в мире троичного компьютера "Сетунь» http://www.icfcst.kiev.ua/museum/Brusentsov_r.html: «*Прав или не прав Н.П. Брусенцов - покажет время. Со своей стороны приведу лишь один факт. В декабре 1993 г. я* (Б.Н. Малиновский) *встретился с известным специалистом в области компьютерной науки профессором С.В. Клименко, работающим в вычислительном центре Института физики высоких энергий (г. Протвино Московской области). Ученый только что возвратился из США, где по просьбе американской стороны прочитал небольшой курс лекций по истории развития компьютерной науки и техники в Советском Союзе.*

На мой вопрос – о чем и о ком спрашивали его американские слушатели, он ответил: "Почему-то только о Брусенцове и его машине "Сетунь"». Обращаем внимание – это в 1993 г., когда у нас и научные, и промышленные круги ни о Н.П. Брусенцове, ни о его Сетуни, ни о троичной системе и не вспоминали. Воистину, нет пророков в своем отечестве.

Второй рассказан автору В.С. Заборовским из Санкт-Петербургского государственного политехнического университета, об интересе которого к троичной системе мы уже упоминали. Вот рассказ Владимира Сергеевича о том, с чего этот интерес начался: «*Это было в марте 2004 году на конференции ICN'04 на острове Гваделупа (заморская территория Франции в Карибском море). Я выступал с докладом по сетевым процессорам - устройствам для обработки пакетного трафика в задачах маршрутизации и фильтрации. В ходе обсуждения возник вопрос о теории вычислительных систем в широком смысле и как можно повысить производительность вычислений. Один из участников, кажется аспирант из США, сказал, что среди известных решений есть одно, которое явно выделяется в аспекте теоретических преимуществ. Это решение было реализовано еще в 60-х годах в России при создании ЭВМ Сетунь, разработчиком которой был Н.П. Брусенцов. К этому времени я слышал о троичной систем счисления для ЭВМ, но не знал имени автора. Когда приехал в СПб сразу же нашел ссылки через поисковые системы и через своих коллег из МГУ вышел на контакт с Николаем Петровичем*».

Еще в XIII веке Раймунд Луллий (1235-1315 гг.) создал на бумаге в виде круговых диаграмм с секторами троичную логическую «машину». Этого Луллия забили камнями. Ну что ж, прогресс налицо. В начале 2005 г. Николай Петрович Брусенцов отметил свое 80-летие и не оставляет надежды сделать троичную ЭВМ на современном научно-техническом уровне.

УДК. 681.3

Д.И. Юдицкий

Высокопроизводительная модулярная ЭВМ «Алмаз»

(НИИ Физических проблем, Зеленоград, 1968 г.)

Справка о результатах разработки эскизного проекта высокопроизводительной ЭВМ «Алмаз» подготовлена 6 марта 1966 г. главным конструктором ЭВМ Юдицким Д.И. для генерального директора Центра микроэлектроники (Зеленоград) Ф.В. Лукина для доклада на конкурсной комиссии, рассматривающей эскизные проекты ЭВМ «М-9» (М.А. Карцев) и «Алмаз». Конкурс выиграл проект «Алмаз» (С.А. Лебедев снял свой проект с рассмотрения).

Справка представлена в факсимильном варианте, выполненном по подлиннику, сохранившемуся в семейном архиве Федора Викторовича Лукина и любезно предложенном его сыном, Владимиром Федоровичем Лукиным.

Юдицкий Д.И. и Лукин Ф.В

С П Р А В К А

Решением ЦК КПСС и Совета Министров СССР и решением ВПК на Научный Центр /Лукин Ф.В./, Институт точной механики и вычислительной техники МРП /Лебедев С.А./, Научно-исследовательский институт управляющих машин МПСА /Карцев М.А./ была возложена разработка вычислительных средств для территориальной системы ПРО страны /представление эскизных проектов в I кв.1967 г./

В точном соответствии с установленным сроком 30 марта 1967 г. Научный Центр представил Министерству Обороны эскизный проект /шифр "Алмаз"/ вычислительной машины, удовлетворяющей исходным данным Генерального Конструктора.

Анализ направлений развития вычислительной техники показывает, что применением одной только "грубой силы" - увеличением рабочих частот элементов - нельзя решить проблему достижения высокой производительности, тем более что повышение частотности элементов - это сложная физическая, техническая и технологическая проблема.

Поиски новых путей повышения эффективности выполнения арифметических операций в вычислительных машинах привели исследователей к заключению, что в рамках обычной позиционной системы значительного достижения в ускорении арифметических операций добиться почти невозможно. Те или иные отдельные приемы и усовершенствования алгоритмов выполнения операций, способствуя более рациональной организации арифметических устройств, оставляют все же производительность машины в пределах одного и того же порядка. Выход за эти пределы требует привлечения новых идей, новых методов организации, новой логики и новой арифметики.

Генеральной идеей повышения эффективной производительности цифровой вычислительной машины является идея параллелизма – распараллеливание процессов обработки, а также распределение работы для одновременного параллельного выполнения по отдельным устройствам и машинам, в связи с чем возникло направление создания многомашинных систем.

Весьма перспективным вариантом указанной идеи параллелизма является распараллеливание не на уровне алгоритмов обработки, а на уровне элементарных операций, т.е. разбиение обрабатываемых слов на малые части и параллельное выполнение элементарных операций над этими частями.

В этом плане необходимо было найти соответствующие теоретико-арифметические концепции, определяющие характер разбиения слова на части и способы их параллельной обработки. Такой концепцией является теория непозиционных систем счисления, созданная и разрабатываемая в Научном Центре.

Проведенные в последние годы исследования, позволили построить разнообразные системы счисления с той или иной степенью позиционности, частными их случаями являются известные в настоящее время в вычислительной технике системы.

Рассматривая целое число N, как значение полинома n-го порядка с целыми коэффициентами при некотором значении X, можно принять, что та или иная система счисления и характер выполнения в ней операций определяется способом задания этого полинома.

Задание полинома своими значениями в $n + 1$ точках приводит к системе счисления, в которой число задается совокупностью значений представляющего полинома в выбранных $n + 1$ точках /компонен

ты числа/ и выполнение рациональных операций над числами в этом представлении сводится к независимому выполнению этих операций над соответствующими компонентами.

В частном случае это представление приводит к системе остаточных классов.

Задание полинома своими значениями и значениями своих производных в некоторых точках приводит к системе счисления, в которой число задается совокупностью значений представляющего полинома и его производных /компоненты/ и выполнение рациональных операций над числами в этом представлении сводится к выполнению этих операций над соответствующими компонентами /сложение - независимо умножение - в соответствии с формулой Лейбница и т.п./

Частным случаем этого представления, когда заданы значение полинома и его n -производных в одной точке является обычная позиционная система счисления. Многочисленные исследования, на чале которым было положено К.Шенноном, проведенные за последнее десятилетие, убедительно показали, что возможно построение таких систем передачи информации, в которых за счет специального кодирования может быть создан иммунитет против самых разнообразных случайных искажений несущих информацию сигналов.

Развитие теории кодирования позволило совершенствовать практику конструирования более надежных вычислительных систем.

Разработанные ранее методы помехозащищенного кодирования оправдали себя при транспортировке и хранении информации, но в силу своей неарифметичности оказались не в состоянии обеспечить эффективное исправление ошибок, возникающих при выполнении арифметических операций. Попытки введения контроля и исправления оши-

бок при обработке информации в рамках позиционной системы приводит к существенному усложнению аппаратуры контроля, а неравноправность информационных и контрольных частей кодов не позволяет использовать одну и ту же аппаратуру для защиты контрольных частей кодов.

В результате проведенных исследований было установлено, что в непозиционных системах могут быть построены самокорректирующиеся коды, позволяющие восстанавливать истинные результаты вычислений по цепи элементарных операций, если во время этих вычислений имели место какие-либо искажения.

Была построена теория специального кодирования в непозиционных системах, позволяющая введением минимальной избыточности в представление слова, осуществлять исправление возникающих ошибок методами близкими к исправлению по смыслу на основе анализа последовательно получающихся слов в процессе обработки.

Применение методов специального кодирования значительно увеличивает функциональную надежность вычислительных машин и позволяет создавать "живучие" машины, сохраняющие работоспособность при выходе из строя значительной части оборудования.

Таким образом, требования Генерального Конструктора оказалось возможным удовлетворить I/ за счет использования разработанной в Научном Центре теории непозиционных систем исчисления, позволяющей добиваться высокой производительности на основе широкого распараллеливания выполнения элементарных операций и максимальной надежности в силу специфических самокорректирующих способностей непозиционных систем; 2/ за счет использования микроэлектронной технологии изготовления системы логических элемен-

тов и основных блоков и узлов вычислительной машины, удачно сочетающейся со спецификой непозиционных систем. Разработка машины проведена на основе системы логических элементов типа "Посол" со средним временем распространения порядка 25-30 наносекунд.

В эскизном проекте дан подробный анализ задач, решаемых многоканальными стрельбовыми комплексами /МКСК/ системы "Аврора", разработана структура вычислительного комплекса для управления МКСК на базе вычислительной машины "Алмаз", приведено теоретическое обоснование построения базовой машины со следующими тактико-техническими характеристиками:

1. Алгоритмическая производительность – 3,5÷4,0 млн. операций в секунду
2. Адресность – 2 адреса
3. Принцип действия – параллельный
4. Диапазон представления чисел – $\pm 2^{\pm 30}$

5. Предусмотрена возможность вычисления значения специальных функций в качестве элементарной команды.

6. Предусмотрена возможность работы со словами переменной длины.

7. Предусмотрена возможность параллельной обработки малоразрядной информации.

8. Предусмотрена возможность режима с плавающим диапазоном.

9. Введена система аппаратурного функционального контроля, автоматически обнаруживающего и исправляющего ошибки, возникающие не только при хранении и транспортировке информации, но и при переработке её в арифметическом устройстве при минимальной аппаратурной избыточности, благодаря чему обеспечена надежность работы и коэффициент боеготовности – 0,9999.

I0. Система памяти - двухступенчатая. Быстродействующая ступень - буферная память объемом 32 слова 55 разрядных.

II. Объем оборудования - II шкафов, каждый из которых I750x800x550, пульт управления, внешние устройства.

I2. Занимаемая площадь	- 80 + I00 м^2
I3. Потребляемая мощность	- 5 квт
I4. Система вентиляции	- автономная на каждый шкаф
I5. Стоимость серийной машины	- 2,6 млн.руб.
I6. Стоимость опытного образца	- 4,2 млн.руб.
I7. Стоимость разработки до I97I г. включая стоимость опытного образца	- 20 млн.руб.

В эскизном проекте был проведен анализ поступления информации от радиолокационных устройств. В результате для обеспечения ввода в машину огромного потока первичной информации и для предварительной ее обработки было создано оригинальное непрограммное устройство - преобразователь информации, осуществляющий в высоком темпе прием и обработку радиолокационной информации и свертывание ее в небольшой по объему массив информации. Выполняемая преобразователем информации обработка эквивалентна выполнению 4 млн. алгоритмических операций в секунду и экономит около 3 млн. бит емкости накопителя. Таким образом, суммарная производительность вычислительной машины "Алмаз" с приданным ей преобразователем информации составит 8 млн.алгоритмических операций в секунду на задачах обработки радиолокационной информации.

В разработке эскизного проекта ЭВМ "Алмаз" участвовали следующие организации Научного Центра: НИИ Физических Проблем, НИИ Точной Технологии,НИИ Точного Машиностроения. В этом проекте

нашли отражение результаты работы этих институтов в течение последних 3 лет.

Эскизные проекты, разработанные в соответствии с указанными выше постановлениями на протяжении II и III кв. 1967 г. рассматривались следующими организациями: 4-ое Главное Управление МО, НИИ-2 МО, СНИИ-45 МО, ОКБ "Вымпел" МРП.

Эскизный проект ЭВМ "Алмаз" был признан этими организациями выполненным на высоком научно-техническом уровне, удовлетворяющим требованиям Генерального Конструктора и принят Генеральным Конструктором в качестве базовой машины для системы управления многоканальным стрельбовым комплексом системы "Аврора".

В августе 1967 г. была создана Государственная комиссия по системе территориальной ПРО, на которой был успешно защищен эскизный проект ЭВМ "Алмаз".

Наряду с разработкой эскизного проекта и по настоящее время предприятия Научного Центра проводили макетирование основных элементов, узлов, блоков и несущих конструкций.

НИИ Точного Машиностроения разработал макет типовых несущих конструкций машины, в НИИ Точной Технологии разработана и изготовлена первая партия элементов "Посол" и единичные типовые ячейки ЭВМ "Алмаз", в НИИ Физических Проблем промоделированы некоторые методы организации арифметического устройства машины.

Однако в Научном Центре почти не проводилась разработка запоминающих устройств. Эскизный проект базируется на запоминающих устройствах, изготавливаемых в настоящее время отечественной промышленностью, мало удовлетворительных для высокопроизводительных вычислительных машин как с точки зрения требуемых параметров, так и с технологической точки зрения. Охарактеризуем вкратце состояние разработки высокопроизводительных вычислительных

средств за рубежом.

Ведущие фирмы США:

IBM, Control Data Corporation, Radio Corporation of America, Burroughs, Philco, Sparry Rand Corporation, Scientific Data Systems

и др. ведут большую работу по созданию высокопроизводительных вычислительных систем.

Характерной чертой этих фирм является сочетание разработок логики и структуры вычислительных машин с разработкой новых микроэлектронных элементов и запоминающих устройств иначе говоря полной разработкой всего комплекса от элементов до готовой аппаратуры.

По данным " Computers and Automation " /январь 1967гг/ на конец 1966 г. в США были завершены разработки вычислительных машин, наиболее производительные из которых характеризуются следующими параметрами:

Фирма	Модель	Быстродействие в сложениях/сек	Быстродействие элементов
IBM	360/75	1 млн	5 нсек
CDC	6600	2,5 млн	10 нсек
Philco	2000/212	1,5млн	5 нсек
Burroughs	B 5500	0,3 млн	20 нсек
Sparry Rond	1108	1,2 млн	5 нсек

Указанные цифры быстродействия ЭВМ подсчитываются по времени выполнений одной из самых коротких арифметических операций - операции сложения. Эффективная алгоритмическая производительность тех же ЭВМ оценивается, в среднем, с коэффициентом 0,6.

Таким образом, по алгоритмической производительности только ЭВМ "СДС-6600" переходит за I миллион алгоритмических операций в секунду.

Естественная тенденция развития вычислительной техники в США направлена на разработку все более высокопроизводительных машин, в первую очередь, для задач противоракетной обороны страны. Так, в Объединенной системе ПРО "Norad" используется комплекс вычислительных машин "Philco 2000/212", параметры которых приведены выше.

Сведений о составе вычислительных средств новой системы ПРО, на которую США предполагает затратить 5млрд. долларов /"Правда" от 23 сентября 1967 г/ не имеется.

Реализация эскизного проекта требует осуществления ряда организационных мероприятий, и , в первую очередь, создания специализованной организации в составе Научного Центра по проведению исследований в области системотехники.

Существующая форма организации /отдел в НИИФП/ не соответствует требованиям и задачам создания вычислительных комплексов для управления средствами ПРО и проведения системотехнических работ, необходимых для успешного развития микроэлектроники в нашей стране.

П Р И Л О Ж Е Н И Е

Создание высокопроизводительных отечественных вычислительных комплексов тормозится главным образом отсутствием быстродействующих запоминающих устройств ЗУ. В настоящее время в СССР сложилось крайне тяжелое положение в области разработок и, в особенности, в производстве ЗУ с малым временем обращения /менее I мксек/ и большой емкости /сотни тысяч слов/. Проблема повышения быстродействия ЗУ остро стоит перед советскими разработчиками и она несомненно будет разрешена. Но для её решения потребуется достаточно большое время и значительные средства.

Следует отметить одно важное обстоятельство: если I0 лет назад параметры отечественных ЗУ и ЗУ, выпускавшихся в США, были почти идентичны, то в настоящее время мы отстаем уже на порядок /лучшие отечественные ЗУ имеют Тобр = 2 мксек, а лучшие ЗУ в США имеют Тобр = 0,I мксек/. Аналогичное состояние наблюдается и в отношении емкостей ЗУ. Необходимо учесть также, что кривая темпов улучшения параметров ЗУ в США за последние 3-4 года резко пошла вверх.

В связи с этим представляется целесообразной закупка лицензий на технологию изготовления ЗУ с циклом обращения не более I мксек.

В качестве тех ЗУ, на которые целесообразно приобрести лицензии, можно указать следующие:

Фирма		Параметры: Время обращения в микросек.	Параметры: Объем /тыс. слов/	Литература
Philco	/США/	I,I5	32-2000	Журнал
Control Data Corporation	/США/	0,8÷I,0	4-262	-"-
Honewell	/США/	I,0	I6-262	-"-
IBM	/США/	0,5	8,I92	
IBM	/США/	0,75	I3I-I048	
Digital Equipment Corporation PDP	/США/	0,9	4-32	
Sperry Rand	/США/	0,375	I6-I3I	
Plessy	/Англия/	I,I-I,25	32-256	Проект на ВМ

отп. I экз.
исп.Юдицкий
печ.Батова
6.Ш.68г.

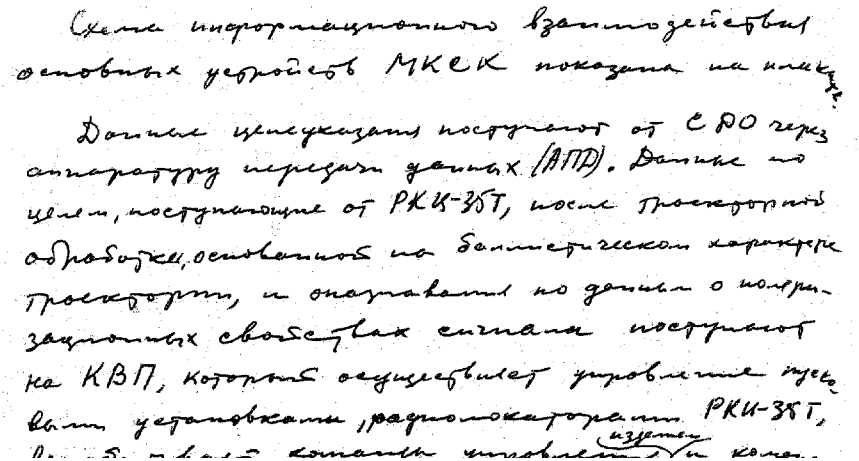
УДК. 681.3

Ф.В. Лукин

Доклады Ф.В. Лукина об ЭВМ «Алмаз» на конкурсной комиссии

(Центр микроэлектроники, Зеленоград, 1966 г.)

Сохранились тексты двух докладов генерального директора зеленоградского Центра микроэлектроники (позже Научный центр) о модулярной ЭВМ «Алмаз» на конкурсной комиссии эскизных проектов ЭВМ для систем ПРО. Тексты с личными правками Ф.В. Лукина любезно представлены его сыном, В.Ф. Лукиным.



Факсимиле черновика начала доклада 1, написанного Ф.В. Лукиным

Доклад 1

Схема информационного взаимодействия основных устройств МКСК (*многоканальный стрельбовой комплекс системы ПРО, прим. ред.*) показана на плакате (*не сохранился, прим. ред.)*).

Данные целеуказания поступают от СДО через аппаратуру передачи данных (АПД). Данные по целям поступают от РКЦ-35Т (*радиолокатор космической цели, т.е. ракеты противника, прим. ред.*), после траекторной обработки, основанной на баллистической характеристике траектории, и опознавания по данным о поляризационных свойствах сигнала поступают на КВП, который осуществляет управление пусковыми установками, радиолокаторами РКИ-35Т (*радиолокатор космического изделия, т.е. противоракеты, прим. ред.*), вырабатывает команды управления изделием и команду подрыва БЧ (*боевая часть, прим. ред.*).

При сопровождении изделия по ответчику, в отличие от сопровождения цели, исключается необходимость обнаружения и обработки сигналов, закрытых шумами. Однако, обработка траектории здесь сложнее, т.к. ускорение изделия изменяется в широких пределах.

В представленном эскизном проекте предлагается специализированный вычислительный комплекс «Алмаз», рассчитанный для наиболее эффективной реализации боевых алгоритмов МКСК.

Требования к параметрам вычислительных средств были уточнены в соответствии с исходными данными Генерального конструктора и результатами анализа алгоритмов, указанных на плакате.

Были оценены возможности создания высокопроизводительной наземной ЦВМ за счет увеличения рабочих частот элементов, а так же за счет выбора соответствующей структуры и логики.

В настоящее время подготовлены для запуска в серийное производство диодно-транзисторные логические элементы, работающие на тактовой частоте 6 МГц. Ожидается, что в 1968 г. будут разработаны элементы на тактовую частоту

13÷15 МГц, а в 1969 г. на 50 МГц. При традиционных методах построения сложной машины эти элементы позволяют соответственно получить быстродействие порядка 1,0 – 1,5 – 5,0 млн. операций в секунду. Однако быстродействие по реализации алгоритмов будет значительно ниже.

Дальнейшее увеличение частотности элементов и создание машины, работающей на высоких частотах является сложной физико-технологической проблемой, решение которой будет достигаться дорогой ценой и потребует много времени.

Более высокие возможности для повышения производительности ЦВМ создаются при выборе ее структуры и логики на основе методов распараллеливания обработки между отдельными вычислительными устройствами, связанными системой управления.

Исследования непозиционных систем счисления, проведенные в Научном Центре, показывают большую перспективность распараллеливания процессов обработки не на уровне алгоритмов, как это предусматривается в обычных вычислительных системах, а на уровне элементарных операций, т.е. параллельное выполнение операций над соответствующими частями слова.

Именно этот метод положен в основу машины «Алмаз».

В качестве основных логических элементов применены микросхемы «Посол» с тактовой частотой 6 ÷ 10 МГц.

Использование непозиционной арифметики позволило освободиться от необходимости учитывать при выполнении операций переносы из младших разрядов в старшие, которые усложняют аппаратурные решения и ограничивают возможность достижения высокого быстродействия.

В непозиционной системе оказалось возможным применить в арифметическом устройстве небольшие таблицы (матрицы), в которых закладываются результаты двухвходовых операций. При этом считывание производится за один такт. Более того, выполнение полиномов, функций Sin, Cos, e^x и др. так же является одно-тактовой операцией. Это позволило существенно повысить алгоритмическую производительность машины «Алмаз».

Важнейшим вопросом при построении ЦВМ является обеспе-

чение заданной надежности. Исследования непозиционных систем, проведенные в Научном Центре, позволили построить теорию кодовых представлений, обладающих способностью самокоррекции не только при передаче информации, но и при выполнении арифметических операций. Это позволило достигнуть заданную надежность при элементах, имеющих $\lambda=10^{-6}$.

При выборе структуры вычислительных средств задачи РКЦ и РКИ рассматривались раздельно.

Для РКЦ оказалось целесообразным использовать два типа вычислительных устройств: непрограммный преобразователь информации, предназначенный специально для формирования единичных замеров, и высокопроизводительная машина (с широким диапазоном представления чисел и большим набором команд) для решения остальных задач.

В пользу такого решения говорят высокие требования по скорости поступления информации (30 тысяч 100-разрядных слов в секунду), а также возможность получать при формировании замеров примерно десятикратное сжатие информации при преобразователе.

Задачи РКИ-35Т решаются высокопроизводительной ЦВМ, такой же, как применяется для РКЦ.

Алгоритмическая эффективность вычислительных средств считалась раздельно для преобразователя информации (ПИ), машины РКЦ и машины КВП, решающей задачи РКИ. Расчеты показали, что ПИ имеет производительность 3, 9 млн. алг. операций в секунду. При этом остается многократный резерв производительности, исчерпываемый полностью лишь при поступлении сигнальных пачек (по 5 слов) каждые 2 мксек.

Эффективность ВМ РКЦ исчисляется цифрой 3,5 млн. алгоритмических операций в секунду, а ВМ КВП – 3,9 млн. алгоритмических операций в секунду.

Пропускная способность ВС МКСК определяется как максимальное количество целей, при работе с которыми средства еще решают поставленные задачи. Расчеты, правда несколько упрощенные, показывают, что РКЦ-35Т могут работать по 700 целям, более осторожно можно считать 500 це-

лей. РКИ-35Т обеспечивает наведение 50 изделий.

На стадии эскизного проектирования проводилось и в настоящее время ведется моделирование реализации основных алгоритмов на универсальных ЦВМ. Полученные результаты подтверждают данные, полученные на основе предварительных расчетов.

Вычислительные средства МКСК, разрабатываемые в Научном Центре, используют в качестве базы технологию интегральных схем и другие методы микроэлектроники.

Это облегчает решения проблемы повышения надежности, уменьшения габаритов, весов, потребляемой энергии и стоимости.

Применяемые элементы «Посол» являются интегральными гибридными диодно-транзисторными логическими схемами, выполняющими функции И-НЕ/ИЛИ-НЕ, и характеризуются такими параметрами:

Время задержки распространения	– 20 ÷ 30 нсек,
Потребляемая мощность	– 10 ÷ 15 мВт,
Нагрузочная способность	– 4,
Число входов «И»	– 8.

Если принять во внимание эти параметры, то утверждение тов. Карцева на предыдущем заседании о том, что на элементах типа «Посол» может быть построена на традиционных принципах машина производительностью в 2 ÷ 4 млн. операций в секунду, вызывает некоторое недоумение. Особенно, если учесть, что в эскизном проекте ИТМ и ВТ, где речь идет о создании машин именно традиционного типа, предполагается для достижения такой производительности использовать элементы с более высокими (на порядок) параметрами по быстродействию.

Требования к надежности вычислительных средств определяются следующими цифрами: вероятность безотказной работы в течение 15 мин. Должна быть не менее 0,9999 и коэффициент готовности в установившемся режиме должен быть также не менее 0,9999.

Если бы машина была выполнена в непозиционной системе, то введением двух дополнительных оснований проблема на-

дежности была бы решена. Однако, в непозиционной системе может быть построена только половина оборудования машины. При этом вероятность безотказной работы незащищенной части будет меньше четырех девяток – 0,9875, а части, защищенной по непозиционной системе – 0,999936, т.е. подавляющая часть ненадежности приходится на долю оборудования, не допускающего построения в непозиционной системе.

Задача обеспечения надежности решается резервированием этой части оборудования на уровне блоков с применением контроля исправности. В этом случае надежность системы в целом составит 0,999912, т.е. удовлетворяет требованиям. Коэффициент готовности при этом будет обеспечен при времени восстановления 20 мин.

Машина «Алмаз» размещена в 10 шкафах на площади 50 ÷ 100 кв. м., потребляет 1 кВт электроэнергии.

Стоимость машины, понимая под этим затраты на производство машины, на комплектацию стендовой аппаратуры и ЗИП, была рассчитана по предложенному НИИ-2 МО методу и составляет для серийного образца 2,6 млн. руб., для опытного образца (может быть изготовлен в 1971 г.) – 4,2 млн. рублей.

В эскизном проекте показана возможность применения современной прогрессивной технологии – микроэлектроники для создания основных устройств высокопроизводительной машины, позволяющей изготавливать отдельные микросхемы в едином технологическом цикле.

В проекте отражено специальное назначение вычислительных средств МКСК. Для предварительной обработки радиолокационной информации создано специальное устройство преобразования, позволяющее проводить на проходе необходимое сжатие информации, поступающей в накопители вычислительных средств, что освобождает вычислительные средства РКЦ от необходимости выполнять 3,9 млн. операций в секунду и хранить более 100 тысяч полноразрядных слов.

Применение непозиционных систем (при относительно малом быстродействии элементов) дало возможность получить

алгоритмическую производительность порядка 4 млн. оп/сек и разработать специальные самокорректирующие коды, способные обнаруживать и исправлять ошибки не только при хранении и транспортировке информации, но и при ее обработке в устройствах вычислительной машины, что позволило при применении ограниченного резерва достигнуть необходимой надежности и боеготовности.

В целом в эскизном проекте показана возможность разработки вычислительных средств МКСК в соответствии с требованиями на этот комплекс.

Ряд вопросов, относящихся к организации вычислительных средств, проработаны в эскизном проекте недостаточно подробно и должны быть детализированы на этапе технического проектирования.

Доклад 2

В настоящее время области применения ЦВМ значительно расширились и объем информации, предъявляемой для обработки на ЦВМ различными реальными системами и вычислительными центрами, возрос настолько, что для реализации требуемой эффективной производительности необходимо привлечение существенно новых идей и новой технологической основы построения ЦВМ.

Генеральной идеей в повышении эффективной производительности вычислительных систем является распараллеливание и распределение процессов обработки для одновременного выполнения по отдельным машинам и устройствам. В соответствии с этой идеей создаются системы машин, связанных воедино сложной структурой иерархического управления. Реализация обработки информации в такой системе требует широкого распараллеливания алгоритмов обработки, что далеко не всегда возможно, т.к. значительное число практически важных алгоритмов имеют сугубо последовательный характер.

Весьма перспективным вариантом указанной генеральной идеи является распараллеливание не на уровне алгоритмов, а на уровне элементарных операций, т.е. разбиение обраба-

тываемого слова (числа) на малые части и параллельное выполнение элементарных операций над этими частями. В этом плане необходимо вести поиск соответствующих теоретико-арифметических концепций, определяющих характер частей разбиения слова, способов их параллельной обработки и путей восстановления полного значения слова по значениям отдельных его частей.

Такой концепцией явилась система остаточных классов, базирующаяся на классических разделах теории чисел (теория сравнений, теория первообразных корней и индексов и др.) За последние годы была создана машинная арифметика в системе остаточных классов и значительно продвинуто практическое внедрение системы.

В НИИ-37 коллективом разработчиков, часть которых в настоящее время работает в Центре микроэлектроники, построен действующий лабораторный образец, изготовлен и заканчивается отладкой заводской экземпляр и подготавливается серийное освоение ЦВМ, работающей в системе остаточных классов с весьма высокой эффективной производительностью.

Система остаточных классов не является единственной системой распараллеливания. В настоящее время в центре микроэлектроники разработана новая, более общая концепция, названная слабопозиционной системой, включающая, как частный случай, систему остаточных классов. Эта концепция позволяет улучшить, по сравнению с остаточными классами, реализацию операций, апеллирующих в той или иной форме ко всему слову в целом, и ввести числовые системы представления, принципиально немыслимые в остаточных классах (например, представления с участием производных). Для слабопозиционной системы теоретической базой является не только теория чисел, но и некоторые разделы классического анализа (теория интерполяции, теория полиномов и др.).

Именно указанные концепции собственно остаточных классов и слабопозиционных систем лежат в основе реализации арифметических устройств и методов выполнения операций в ЦВМ «Алмаз», разрабатываемой в Центре микроэлектро-

ники, которая создается на интегральных схемах и является первой отечественной машиной высокого класса, создаваемой на микроэлектронной базе.

Применение в ЦВМ высоких рабочих частот приводит к существенному усложнению аппаратурной части (задержки в передаче сигналов, необходимость монтажный провод рассматривать как двухпроводную линию связи с распределенными параметрами и невысоким волновым сопротивлением, трудности синхронизации, согласование элементов и т.д.).

Для упрощения аппаратурной части и увеличения устойчивости элементов в ЦВМ "Алмаз" принята рабочая частота 1 МГц, а необходимая эффективная производительность реализуется за счет новой системы счисления и оригинальной структуры арифметического устройства.

Для принятой в "Алмазе" слабопозиционной системы счисления все арифметические операции являются однотактными, следовательно, рабочее быстродействие машины – 1 млн. любых операций: в секунду. В позиционной машине простейшая операция типа сложения, с учетом межразрядных переносов, выполняется обычно за 5 машинных тактов. Для удобства последующих сопоставлений примем за единицу операцию сложения **С**:

$\mathbf{C_н}$ – операция сложения в слабопозиционной системе,

$\mathbf{C_п}$ – операция сложения в позиционной системе.

$\mathbf{C_п} = 5\mathbf{C_н}$.

Однако, основной характеристикой ЦВМ является не рабочее быстродействие, а эффективная производительность, т.е. производительность, достигаемая при реализации тех или иных классов алгоритмов. Эта характеристика зависит уже не только от рабочего быстродействия, а и от организации машины в целом, от ее структуры и логики, иначе говоря, от ее способности эффективно обрабатывать комплексы операций.

Эффективное быстродействие ЦВМ "Алмаз" значительно превышает ее рабочее быстродействие. Дело в том, что слабопозиционная система (включая ее частный случай – сис-

тему остаточных классов) обладает, помимо возможностей организации параллельной арифметической обработки частей слова, рядом существенных особенностей, которые могут быть использованы для повышения эффективной производительности, и которыми ни в коей мере не обладают обычные позиционные системы. Это:

• Возможность выполнения любой операции за 1 машинный такт, причем за машинный такт принимается такт частоты синхронизации.

• Возможность выполнения за один машинный такт любой сложности функции одной или двух переменных, если имеется однозначное соответствие между значениями операндов и результата. Выполнение таких функций может быть введено в систему команд ЭВМ в виде специальной операции.

• Возможность обнаружения и исправления ошибок в арифметическом устройстве.

Рассмотрим эти особенности подробнее.

1. Выполнения любой операции за 1 машинный такт.

Очевидно, что эффективная производительность позиционной машины существенно зависит от характера и состава операций, которые надлежит выполнить для реализации алгоритма. Так, например, для алгоритмов первичной и вторичной обработки радиолокационной информации в позиционной системе может быть принят следующий состав операций:

• Сложения-вычитания, логические и управленческие - 80%,

• Умножения - 15% (для умножения мы уже приняли оценку $4\mathbf{C_П}$),

• Деления - 5%. (Для деления можно принять оценку $15\mathbf{C_П}$).

Таким образом, в единицах **C** этот состав операций для позиционной машины может быть оценен:

$$80\mathbf{xC_П} + 4x15\mathbf{xC_П} + 5x15\mathbf{xC_П} = 215\mathbf{C_П} = 1075\mathbf{C_Н}.$$

А для ЦВМ "Алмаз", где $\mathbf{C_Н} = 1$, соответствующая оценка составляет 100 $\mathbf{C_Н}$.

Таким образом, только от однотактности всех операций ЦВМ

"Алмаз" (с системой команд, не включающей команды выполнения функций) по эффективной производительности выше позиционной более чем в 10 раз (1075 : 100 = 10,75).

2. Возможность выполнения функции за один машинный такт.

Реализация большого класса алгоритмов связана с вычислением значений функций одной или двух переменных, на которые в позиционных ЭВМ обычно затрачивается до 40 машинных операций типа сложения.

Среди операций, реализующих вычисление значений функций в позиционной машине, 25% составляют умножения, каждое из которых выполняется не менее чем за 4$\mathbf{C_П}$. Таким образом, на вычисление одного значения функции должно быть затрачено порядка:

$$40x0,75x\mathbf{C_П} + 40x0,25x4\mathbf{C_П} = 70\mathbf{C_П} = 350\mathbf{C_Н}.$$

Слабопозиционная система позволяет вычислять значения элементарных функций на основе их полиномиальных приближений одной выборкой из таблицы, т.е. также равно 1$\mathbf{C_Н}$.

Таким образом, значение элементарной функции в ЦВМ "Алмаз" (с системой команд, включающей команды выполнения функций) будет вычисляться в 350 раз быстрее, чем в позиционной машине с той же тактовой частотой.

Реально в алгоритмах решения задач ПРО используются и обычные операции, и элементарные функции. Из всей совокупности операций, реализующих алгоритмы ПРО, 15% отнимает вычисление значений элементарных функций. Тогда общая эффективная производительность ЦВМ "Алмаз» на этом классе алгоритмов будет больше эффективной производительности позиционной ЦВМ в:

$$0,15x350 + 0,85x10 = 61 \text{ раз}$$

3. Обнаружениу и исправлениу ошибок в арифметическом устройстве.

Эффективным способом обеспечения правильности обработки является применение самокорректирующихся кодовых систем. Такие системы были разработаны для передачи ин-

формации по каналам связи и давали возможность восстановить на приемном конце истинную переданную информацию, даже если она подвергалась по пути довольно существенным искажениям.

Оказывается, что в отличие от обычной позиционной системы, где самокоррекция принципиально возможна только при передаче информации, в слабопозиционной системе, включая и систему остаточных классов, могут быть построены самокорректирующиеся коды, позволяющие восстанавливать истинные результаты вычислений по цепи элементарных операций, если во время этих вычислений имели место какие-либо ошибки.

Такая теория специального кодирования построена. Она позволяет введением минимальной избыточности в представлении слова осуществлять исправление методами близкими к исправлению по смыслу на основе анализа получающихся искажений слов на основе последовательной обработки. Такая система специального кодирования предусматривается в ЦВМ "Алмаз". Она по своим результатам эквивалентна применению двойных просчетов для установления правильности проведенных вычислений и тройных для исправления возникшей ошибки.

Таким образом слабопозиционные машины всегда производительнее и надежнее позиционных и чем сложнее вычисляемые функции, чем их больше, тем они эффективнее.

УДК. 681.3

М.Д. Корнев

О структурных решениях в проекте ЭВМ 5Э53

(Научноисследовательский институт дальней радиосвязи)

В статье рассмотрены структурные особенности модулярной ЭВМ 5Э53, новые идеи и принципы построения ЭВМ и ее устройств, реализованные в проекте, которые позже получили широкое применение в вычислительной технике.

Шестидесятые и семидесятые годы прошлого века были, пожалуй, самыми плодотворными для вычислительной техники. Именно в это время были разработаны лучшие в архитектурном отношении ЭВМ и системы, обладающие рядом новшеств в области систем счисления, структурных методов построения, новшеств в области повышения надежности, обеспечения эффективных методов управления вычислениями и программирования.

Такой прогресс объяснялся тем, что в условиях “холодной войны” средства вычислительной техники требовались в первую очередь министерству обороны для реализации сложных автоматизированных локационных систем и систем управления различными видами вооружения.

Министерство обороны вынуждено было предъявлять по тем временам к разработчикам вычислительных средств предельные и за-

предельные требования по производительности, объемам памяти, надежности, средствам автоматизации программирования.

Коллектив, работающий под руководством Д.И. Юдицкого, должен был заниматься проектированием именно таких вычислительных средств в условиях отсутствия в стране развитой элементной базы (логических элементов, элементов памяти и т.д.), средств автоматизации проектирования, а самое главное – в отсутствии теоретически и практически обоснованных методов построения эффективных устройств.

К началу разработки одной из таких ЭВМ (5Э53) существовали ЭВМ с быстродействием до нескольких тысяч операций в секунду. Требовалась же производительность до 10 млн. алгоритмических операций в секунду, объем ОЗУ – не менее 10 Мбит, ППЗУ – до 3 Мбит, развитая аппаратура передачи данных на удаленные объекты. Эти требования были выполнены. Их удалось реализовать только благодаря коллективному подходу к разработке проекта, от систем счисления до элементной базы.

ЭВМ 5Э53 была прежде всего оригинальной по своей системе счисления – системе остаточных классов (СОК), благодаря которой ряд арифметических операций (типа сложения и умножение), а также процедур вычисления элементарных функций удавалось реализовать за один машинный такт (166нс), равный времени выборки данных из малоразрядной таблицы. Этот машинный такт определял предельное быстродействие ЭВМ. Чтобы достичь показателей требуемой производительности, нужно было обеспечить непрерывную работу арифметики. Это значит, что все операции выборки команд, их дешифрации, модификации адресов и чтения и записи операндов не должны были замедлять работу арифметики. Достичь этого удалось путем применения всевозможных методов совмещения времен выполнения различных операций.

В ЭВМ 5Э53 впервые появились:

- параллельно работающие арифметические блоки в АУ;
- раздельные тракты выборки команд и работы с данными;
- индексная арифметика, работающая параллельно с АУ;
- расслоение памяти;
- эффективные средства выполнения команд условных переходов;
- буферные устройства, у которых каждый такт работы эле-

мента буфера был равен машинному такту.

Все эти нововведения предварительно оценивались по объему оборудования и их влиянию на быстродействие ЭВМ. Вначале интуитивно выбранные способы потребовали своего теоретического обоснования и обобщений. Так был сделан вывод о том, что все виды совмещений в исполнении операций разделяются на два: совмещение в виде конвейера (трубопровода) и совмещение в виде параллелизма. В первом случае процесс исполнения сложной операции (или множества операций) осуществлялся на последовательно соединенных, а во втором – на несвязанных друг с другом и независимо работающих операционных блоках.

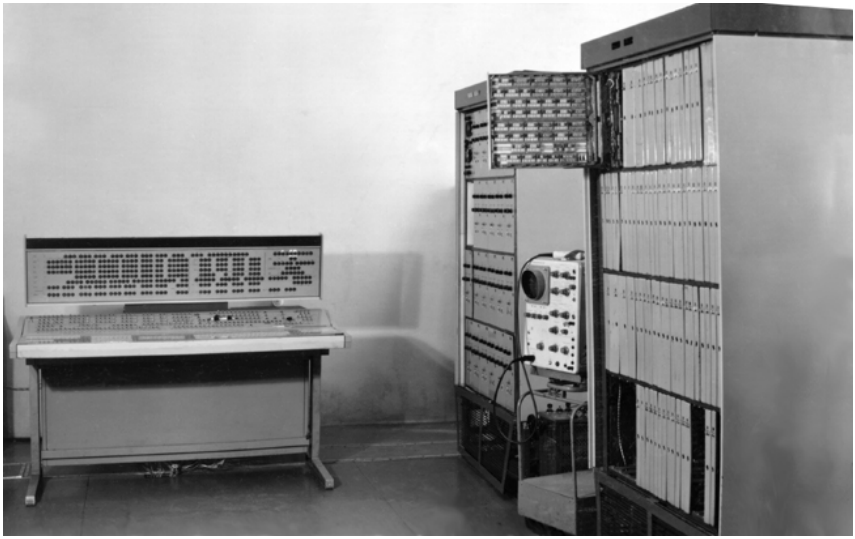
Очевидно, что в случае конвейера быстродействие устройства зависело от самого медленного операционного блока, а задержка одной операции в конвейере – от числа таких блоков. В случае параллелизма быстродействие зависело не столько от времени работы отдельного блока, сколько – от числа этих блоков в устройстве, а на задержку операции, кроме времени работы одного блока, существенно влияли средства коммутации операционных блоков.

Возникли задачи поиска методов эффективной загрузки таких устройств, вообще говоря, связанными друг с другом множествами операций.

Для решения задач удалось привлечь и развить существовавшие методы в экономике (транспортная задача, задача коммивояжера, задача загрузки двух станков) и ряд других методов исследования операций (теория расписаний, теория графов, теория массового обслуживания, математическое моделирование). В результате для каждого метода организации устройств (параллелизма и трубопровода) были найдены эффективные алгоритмы распараллеливания множества операций. С их помощью строились оптимальные, или близкие к оптимальным, расписания для выполнения множества несвязанных или связанных операций на заранее заданном количестве операционных блоков, а также определялось количество таких блоков, которое целесообразно иметь для реализации того или иного вида совмещения.

Применительно к ЭВМ 5Э53, в зависимости от решаемой задачи, в качестве операционных блоков рассматривались процессорные модули, предназначенные для выполнения так называемых модульных и немодульных операций СОК, блоки оперативной памяти,

БЗУ – регистровое быстродействующее запоминающее устройство и так далее. Множество операций определялось алгоритмами вычисления элементарных функций, пересчета координат, матричными задачами и так далее, то есть задачами, выбранными из алгоритмов специального программного обеспечения ЭВМ. Так для арифметического устройства было установлено, что для его эффективной работы достаточно иметь один модульный процессор и два немодульных. Было показано, что введение в состав ЭВМ БЗУ увеличивает ее быстродействие на 45%.



Фрагмент макетного образца ЭВМ "5Э53"

Появление в ЭВМ двух независимых трактов: тракта выборки команд и тракта операндов позволило осуществлять выборку команд из полупостоянной памяти, работающей с тактом 150нс, а работу с операндами организовать по принципу трубопровода. В его состав входило ОЗУ, выполненное по принципу аппаратного расслоения.

Идея такого расслоения впервые была предложена в неосуществленном проекте IBM-360/92. В соответствии с методом расслоения ОЗУ разбивается на некоторое число отдельных блоков (в 5Э53 их было восемь), а запись и чтение данных, расположенных в последовательных адресах ОЗУ, осуществляется в соседние блоки памяти. Благодаря такой организации ОЗУ времена чтения и записи операндов удалось существенно приблизить к машинному такту.

На стадии проектирования ЭВМ был проработан, а потом частично реализован способ повышения вычислений за счет команд условных переходов. Именно при исполнении этих команд, когда условие перехода выполнялось, возникали существенные ожидания в вычислениях, так как все буферы, в которых находились подготовленные команды и данные, необходимо было ликвидировать и начать их заполнение с новых адресов. Для реализации был выбран вариант анализа одновременно двух ветвей программы: одна в предположении, что условие перехода не выполнится, другая – на оборот. Этот подход оказался весьма эффективным. В дальнейшем на этом пути (уже не у нас) появился КЭШ команд, особенно эффективный при реализации циклов.

В заключение следует заметить, что многие структурные решения, найденные в далеком прошлом разработчиками различных ЭВМ, успешно развиваются и эксплуатируются в составах современных ЭВМ и процессорных чипах.

УДК 621.29

В.Г. Евстигнеев

Недвоичные компьютерные арифметики

(Московский научно-исследовательский телевизионный институт)

В статье с позиций сегодняшнего дня делается попытка оценить вклад Советских ученых прошлого столетия в развитие вычислительной техники Советского Союза и показать диапазон научных исследований в этой области и в частности в разработке новых компьютерных арифметик и их внедрении в создание принципиально новых вычислительных систем, намного опережавших достижения западных и в первую очередь американских специалистов в этой области. И только постепенное технологическое отставание Советской микроэлектронной промышленности не позволило сбыться смелым и много обещающим идеям Советских ученых того времени. Автор статьи сумел внести свой скромный вклад в общую копилку прогресса вычислительной техники того времени.

В 70 – 80 годы двадцатого столетия усилиями советских ученых С.А. Лебедева, В.М. Глушкова, М.А. Карцева, И.Я. Акушского, Д.И. Юдицкого, Г.Я. Гуськова, В.С. Семенихина, И.В. Прангишвили, Н.Я. Матюхина и многих других были созданы образцы вычислительной техники, превосходящие по своим параметрам, новизне идей и архитектуре все мировые достижения. В качестве примера назову лишь некоторые из вычислительных машин тех лет: серия ЭВМ класса БЭСМ И «Эльбрус», ЭВМ «Стрела», серии ЭВМ М-20, М-220, 5Э76, «Мир», ПС и др. и лишь после того, как СССР

стал воспроизводить ЭВМ класса IBM-360 (ЕС ЭВМ) и копировать, а не разрабатывать микроэлектронную элементную базу, судьба советской вычислительной техники была предрешена. Ответственность за это лежит не на ученых и разработчиках, а на руководстве страны.

В те же годы в СССР коллективы ученых исследовали и разрабатывали различные арифметики, позволявшие создавать ЭВМ с более высокой скоростью обработки данных по сравнению с широко распространенной двоичной системой счисления. Так, под руководством И.Я. Акушского и Д.И. Юдицкого была создана ЭВМ К-340 на основе системы счисления в остаточных классах, которая длительное время выпускалась нашей промышленностью и отличалась высокой производительностью и надежностью. Коллективом специалистов во главе с В.М. Глушковым были созданы и запущены в производство ЭВМ серии «Мир» с новой архитектурой и системой программирования, позволявшие производить вычисления с переменной (регулируемой) разрядностью. Тогда же под руководством И.В. Прангишвили разрабатываются и производятся ЭВМ серии ПС на основе ассоциативных процессоров с параллельной архитектурой, а под руководством М.А. Карцева – высокопроизводительные, высоконадежные вычислительные комплексы большой разрядности (до 512 двоичных разрядов) для специальных применений.

В начале 80-х годов появились первые публикации советских ученых о Фибоначчиевой системе счисления [1] и иерархической системе счисления, которые позволяли создавать более высокопроизводительные и надежные вычислительные средства на основе новых элементов микроэлектроники, в том числе и многозначных. Одновременно разрабатываются и алгоритмы выполнения арифметических операций в ЭВМ, основанные на новых арифметиках.

Кратко опишем наиболее интересные системы счисления для вычислительных устройств, быстродействие и надежность которых превосходят аналоги, основанные на двоичной арифметике.

Знако-разрядная система счисления.

Число в знако-разрядной системе счисления [2], как и в любой позиционной системе, можно записать в виде

$$X = \sum_{i=-m}^{n} x_i \cdot S^{-i},$$

где $x_i=\{-R, -R+1,...,-1, 0, 1,..., R\}$, $S=2R$ или $S=2R+1$.

Сложение двух чисел в знако-разрядной системе счисления выполняется в два такта. В первом такте формируются поцифровые промежуточные суммы ω_i и цифры поразрядных переносов t_i, которые могут принимать значения -1, 0 и +1, т.е. $x_i + y_i = \omega_i + t_i S$.

Во втором такте формируется окончательная сумма путем сложения цифр промежуточных разрядных сумм и соответствующим им цифр поразрядных переносов, т.е. $Z_i = \omega_i + t_{i+1}$.

Умножение чисел в знако-разрядной системе счисления выполняется последовательным сложением (вычитанием) и сдвигом вправо результатов умножения множимого на S-ичные цифры множителя, начиная с младшего S-ичного разряда. Деление чисел подчиняется общим правилам деления в S-ичной системе счисления.

Основным достоинством знако-разрядной системы счисления является то, что сигнал переноса при выполнении операции сложения распространяется не далее соседнего разряда, а время выполнения операции не зависит от разрядности операндов. То есть любая операция сложения выполняется за два такта (под тактом здесь понимается время вычисления разрядной суммы.

Фибоначчиева система счисления.

Среди позиционных весомозначных систем счисления есть системы, в которых веса разрядов выражаются не известным соотношением $\Delta_i=S_i$, а другими, например числами ряда Фибоначчи, т.е. $\Delta_t=\Delta_{t-2}+\Delta_{t-1}$. в этом случае система счисления называется Фибоначчиевой. Другой пример позиционной весомозначной системы счисления с нетрадиционным законом формирования весов разрядов – так называемая полиадическая система счисления. Веса разрядов в ней определяются выражением $\Delta_i=p_i\cdot\Delta_{i-1}$, где p_i – взаимно – простые числа.

Остановимся на Фибоначчиевой системе счисления. В работе [1] показано, что любое натуральное число N может быть представлено в двоичной p-системе счисления при $p\geq 0$, весами разрядов в которой являются числа Фибоначчи. При этом после каждой единицы слева направо следует не менее p нулей. Так, например, при $p=1$ число 75 в двоичной 1-системе счисления можно записать как

$$75=1001010100=1\cdot 55+0\cdot 34+0\cdot 21+1\cdot 13+0\cdot 8+1\cdot 5+0\cdot 3+1\cdot 2+0\cdot 1+0\cdot 1.$$

Отметим две особенности сложения значащих разрядов в двоичной 1-системе счисления. Во-первых, при суммировании единиц возникает перенос не одной единицы (как в классической двоичной системе счисления), а нескольких одновременно. Во-вторых, единицы можно складыватьдвумя способами. В первом способе при сложении *i*-х разрядов чисел в *i*-м разряде промежуточной суммы записывается 1 и возникают переносы двух единиц одновременно – в (*i-1*)-й и в (*i-p-1*)-й разряды. При втором способе сложения единиц в соответствующем (*i*-м) разряде промежуточной суммы записывается 0 (как и в классической двоичной арифметике) и возникает перенос *p+1* единиц (одна единица – в старший (*i*+1)-й и *p* единиц – в младшие (*i-p-1*), (*i-p-2*),…,(*i-2p*) разряды).

Наиболее рациональный способ умножения двоичных Фибоначчиевых чисел в 1-системе счисления аналогичен умножению в классической двоичной, хотя и обладает своей спецификой [1].

Основной способ деления чисел *(Z=X/Y)* в Фибоначчиевой системе счисления: накапливаются кратные числам Фибоначчи значения делителя, т.е. $N=Y\cdot K_j$ $(K_j=1,2,3,5,...)$. Кратные делителя сравниваются с делимым, начиная с максимального кратного. В зависимости от результата сравнения формируется частное, т.е.

$$Z=\sum_{j=1}^{I} K_j.$$

Несмотря на очевидную непрактичность Фибоначчиевой системы счисления для конструирования цифровых вычислительных устройств, работы создателя системы и его учеников представляют собой значительный научный результат, который показывает неисследованность разнообразия систем счисления и необходимость поиска систем с новыми качествами.

Система остаточных классов.

Система остаточных классов (СОК) – это непозиционная система счисления, числа в которой представляются остатками от деления на выбранную систему оснований $P_1, P_2,\ldots,P_n$ и являются взаимнопростыми числами. Операции сложения, вычитания и умножения над числами в СОК производятся независимо по каждому основанию без переносов между разрядами (основаниями). Диапазон представимых чисел $P=P_1\cdot P_2\cdot\ldots\cdot P_n$ [3].

Если задан ряд положительных взаимнопростых чисел P_1, P_2,…,P_n, то целое положительное число A на выбранные основания P_1, P_2,…,P_n, можно записать в виде $A=(\alpha_1, \alpha_2,\ldots,\alpha_n)$,

$$\alpha_i = A - \left[\frac{A}{P_i}\right]\cdot P_i, \qquad i = 1,2,...,n;$$

где [ ] – целочисленное деление. Это и есть запись числа в СОК.

Если исходные числа A, B, их сумма $A+B$ и их произведение $A\cdot B$ находятся в диапазоне $(0,P)$, то результаты операций сложения $A+B$ и умножения $A\cdot B$ могут быть однозначно представлены соответственно остатками γ_i и p_i по тем же основаниям P_i , т.е.

$A=(\alpha_1, \alpha_2,\ldots,\alpha_n)$, $B=(\beta_1, \beta_2,\ldots, \beta_n)$,

$A+B=(\gamma_1, \gamma_2,\ldots, \gamma_n)$, $AB=(p_1, p_2,\ldots, p_n)$,

$$\gamma_i = \alpha_i + \beta_i - \left[\frac{\alpha_i + \beta_i}{P_i}\right]\cdot P_i,$$

$$p_i = \alpha_i \cdot \beta_i - \left[\frac{\alpha_i \cdot \beta_i}{P_i}\right]\cdot P_i.$$

где [ ] – целочисленное деление. Это и есть запись числа в СОК.

Такие операции, как деление, сравнение и др., требующие информации о величине всего числа, в СОК выполняются по более сложным алгоритмам. И в этом заключается существенный недостаток данной системы счисления, сдерживающий ее широкое применение в качестве компьютерной арифметики. Однако сегодня даже в самых современных компьютерах при работе с большими и супербольшими числами используют СОК, ибо только эта арифметика позволяет получать результаты вычислений в реальном времени. В таких случаях в качестве оснований СОК применяют величины, близкие 2^m (m–двоичная разрядность компьютера), например 2^{m-1}-1, 2^{m-1}, $2^{m-1}+1$ и т.д. Компьютер вычисляет результат по одному из модулей за один проход. Другие области применения СОК – помехоустойчивое кодирование, криптография и т.п.

Начиная с 1952 года специалисты многих стран мира, включая и

СССР, занимались проблемой повышения скорости выполнения «неудобных» операций в СОК. Особую роль в решении данной проблемы сыграл И.Я. Акушский. Немалый вклад в эту область науки внесли также Д.И. Юдицкий,, В.М. Амербаев, А.А. Коляда.

Иерархические системы счисления.

В конце 80-х – начале 90-х годов родилась идея соединения позиционных и непозиционных систем счисления, т.е. конструирования иерархических систем, которые должны сочетать в себе положительные стороны включенных в них систем счисления и быть свободными от их недостатков [4, 5]. Принцип построения иерархических систем в целом прост. Выбирается некоторая внешняя система счисления $A=<\alpha,\Omega>$, где α - алфавит системы, а $\Omega=\Omega_0, \Omega_r$ – ее сигнатура. Сигнатура состоит из двух частей: операционной (Ω_0), содержащей символы операций системы, и реляционной (Ω_r), содержащей символы отношений. Цифры, т.е. элементы алфавита A этой системы, записываются в виде слов (кодов) другой (внутренней) системы счисления $B=<\beta,\Omega>$. Такую систему обозначают $A[B]$.

Рассмотрим пример. Пусть A – десятичная позиционная система, а B – двоичная система. Тогда $\alpha=\{0,1,2,..,9\}$, $\beta=\{0,1\}$. Двоичное кодирование цифр системы A (десятичной) производится, например, тетрадами:

$$0\rightarrow 0000,\ 1\rightarrow 0001,\ldots 9\rightarrow 1001$$

Тогда число 23 (десятичное) запишется в иерархической системе счисления в виде двух тетрад (0010, 0011). Система $A[B]$ в нашем примере – хорошо известная двоично-кодированная десятичная система, применяемая для представления десятичных чисел в современных ЭВМ.

Очевидно, что степень вложенности иерархической системы может быть и более двух. Иначе говоря, существуют иерархические системы счисления $A_0[A_1[A_2...[A_n]...]$ с основаниями $S_0, S_1,\ldots, S_n$, причем $S,> S_1>\ldots> S_n$. Система счисления (двоичная, восьмеричная, шестнадцатиричная), для которых

$$S_0 = S_1 \cdot 2_1^m, \qquad S_1 = S_2 \cdot 2_2^m, \ldots, \quad S_n = S_{n-1} \cdot 2_n^m$$

на уровне представления являются безизбыточными. Если данное

условие не выполняется, система избыточна (например, двоично-кодированная десятичная) [4].

Позиционно-остаточная система счисления.

При конструировании иерархических систем счисления большой интерес представляет сочетание систем различных типов. Рассмотрим систему вида *A[B]*, для которой *A* – позиционная система счисления с основанием *S*, а *B* – система счисления в остаточных классах с базовыми модулями $P_1, P_2,\ldots, P_r$, такими, что $P=P_1 \cdot P_2 \cdot \ldots \cdot P_r$. Такую систему называют позиционно-остаточной системой счисления [6].

Неравенство $P \geq S$ – необходимое и достаточное условие однозначного представления цифр *0, 1, … , S-1* позиционной системы наборами вычетов по модулям $P_1, P_2,\ldots, P_r$. Однако учитывая необходимость корректной реализации арифметических операций в системе *A[B]* (например, формирование переноса и т.п.), можно поставить более жесткое условие $P=P_1 \cdot P_2 \cdot \ldots \cdot P_r \geq 2S$.

Весьма важен выбор величины основания *S* позиционной системы счисления в статочных классах. Отдавая дань двоичной системе счисления, можно выбирать $S \geq 2^m$. В этом случае модули СОК и их произведение должны удовлетворять условию $P \geq 2^{m+1}$. Для человека же наиболее удобны основания, кратные *10 (100, 1000 и т.д.)*.

Двоично-кодированная десятичная система – в известной мере компромисс между человеком и компьютером. Но ее относительная избыточность – 26,5%. Чтобы преодолеть данный недостаток, ряд исследователей предлагают для арифметики с плавающей запятой вместо основания *10* использовать *100* [5]. Тогда для хранения двух десятичных цифр достаточно иметь семь двоичных разрядов вместо восьми (избыточность представления – 22,7%). Переход к основанию *1000* позволяет размещать три десятичные цифры в 10 двоичных разрядах вместо *12* (избыточность представления – 2,35%). Расплата за экономическое представление чисел при переходе к основаниям в вида 10^n – более сложные алгоритмы кодирования и декодирования таких чисел. Однако на уровне машинного представления арифметика все равно остается двоичной.

Арифметические операции в позиционно-остаточной системе счисления выполняются отдельно над цифрами внешней и внутренней системы. Такая ступенчатая реализация операций позволяет

практически без изменений переносить алгоритмы внешней системы счисления на операции в системе *A[B]*. При этом «цифровые» операции системы счисления *A* заменяются процедурами системы счисления *B*. [5]

Знако-разрядная позиционно-остаточная система счисления.

Еще один пример иерархической системы счисления – знако-разрядная система с основанием *S,* цифры которой представляются в системе остаточных классов с базовыми модулями $P=P_1$, $P_2,\ldots,P_r$. Достоинство данной системы счисления – высокая скорость выполнения арифметических операций над разрядными цифрами и минимальная длина пути распространения переноса между *S*-ичными разрядами (не далее соседнего разряда). Высокое быстродействие достигается за счет того, что при суммировании в каждом *S*-ичном разряде *(S>2)* одновременно формируются три величины:

$$x_i+y_i, \quad x_i+y_i-1, \quad x_i+y_i+1.$$

Затем одна из них выбирается в качестве результата в зависимости от значения сигнала переноса t_i, принимающего значения *-1, 0, +1*. [7,8,9,10]

Таким образом, появляется возможность параллельной обработки на нескольких компьютерах больших чисел с основаниями $S=2^m$. Обрабатывать большие числа в «реальном времени» способны даже двоичные персональные компьютеры, работающие по алгоритмам знако-разрядной позиционно-остаточной системы счисления.

Новейшие западные технологии, появляющиеся на российском рынке, в совокупности с отечественными разработками в области недвоичных компьютерных арифметик и синтеза новых способов и алгоритмов ускорения вычислений открывают перед разработчиками вычислительных систем новые возможности.

Литература

1. **Стахов А.П**. Введение в алгоритмическую теорию измерений. – М. Сов. Радио, 1997.
2. **Поспелов Д.А.** Арифметические основы вычислительных машин дискретного действия. – М. Высшая школа, 1970.
3. **Акушский И.Я., Юдицкий Д.И.** Машинная арифметика в остаточных классах. М. Сов. Радио, 1968.
4. **Schoichet S.R.** The LISP Machine. Mini-Micro System, 1978,

№11(5), p. 68-74.
5. **Евстигнеев В.Г.** S-ичный сумматор. – Электронная техника. Сер. 10, 1986, вып. 5(59), с. 17-19.
6. **Евстигнеев В.Г.** Недвоичная машинная арифметика и специализированные процессоры. – М. МИФИ СЕРВИС и АО «ИНСОФТ», 1992.
7. **Евстигнеев В.Г. Евстигнеева О.В.** Устройство для сложения многоразрядных q-ичных чисел. Авторское свидетельство № 1163321.
8. **Евстигнеев В.Г**. S-ичный сумматор. – Авторское свидетельство № 1273925.
9. **Евстигнеев В.Г. Евстигнеева О.В.** Устройство для сложения n-разрядных чисел в избыточной системе счисления. – Авторское свидетельство № 1188731.
10. **Евстигнеев В.Г.** Сумматор в знакоразрядной позиционно-остаточной системе счисления. – Авторское свидетельство № 1383349.

УДК. 681.3

В.М. Амербаев, В.Н. Дьячков

Модулярная арифметика как криптографический примитив

(ГУП «Научно-производственный центр «СПУРТ»», Россия)

В статье рассмотрены некоторые аспекты использования принципов модулярности (модульность и не модульность) в задачах информационной безопасности, базирующихся на симметричных шифр-системах.

Широкие возможности вычислительных средств выделили модульную арифметику или, в общем случае, арифметику кольца вычетов по модулю над кольцом главных идеалов [1] в особо важный класс управляемых примитивов криптографических преобразований. Это оказалось возможным благодаря тому, что упомянутые алгебраические структуры допускают эффективную адаптацию их операций к регистровым операциям арифметических процессоров современных компьютеров.

То, что модульная арифметика лежит в основе многих важных криптопреобразований является бесспорным фактом. Примером тому служат такие алгоритмы, как алгоритм RSA, алгоритм Эль-Гамаля, алгоритмы криптопреобразований над конечными полями.

Ниже пойдет речь об использовании в криптопреобразованиях модулярной арифметики [2,3], которая отличается от модульной тем,

что здесь существенное значение приобретает понятие «динамического диапазона». Дело в том, что в модулярной арифметике над кольцом целых чисел Z элементы кольца вычетов помимо кодовой интерпретации имеют и количественную трактовку, как целые числа. Поэтому здесь модульные вычисления над кодовыми словами в семантическом плане должны сопровождаться «мониторингом» возникновения возможных переполнений за основной диапазон, обычно называемый динамическим диапазоном, либо должны сопровождаться условиями, налагаемыми на операнды и операции, гарантирующими отсутствие подобных переполнений.

Несколько замечаний об используемых обозначениях и их смыслах.

• Символом $|x|_P$ обозначается вычет целого числа x по $\bmod P$.

По умолчанию этот же символ используется для обозначения наименьшего неотрицательного вычета.

• Z_P – кольцо вычетов по $\bmod P$.

$$Z_P := \{x \in Z \,|\, |x|_P = x\}.$$

Связь кольцевых операций кольца Z_P с арифметическими операциями $+$, $\times$ над целыми числами выражается формулами:

1) $\forall x, y \in Z$,

$$|x + y|_P = \left||x|_P + |y|_P\right|_P,$$

$$|x \times y|_P = \left||x|_P \times |y|_P\right|_P.$$

2) мультипликативная инверсия – если $(M, P) = 1$, то вычет $x := \left|M^{-1}\right|_P$ есть решение уравнения $|x \times M|_P = 1$.

• $|x|_P^-$ – абсолютно наименьший вычет целого числа x по $\bmod P$:

$$Z_P^- := \left\{x \in Z \,|\, -\frac{P}{2} \le x < \frac{P}{2}\right\}.$$

Связь абсолютно наименьшего вычета с наименьшим неотрица-

тельным вычетом по $\mathrm{mod}\, P$ выражается формулой:

$$\forall x \in \mathrm{Z} : \left|x\right|_P^{-} = \left|x + \left\lfloor \frac{P}{2} \right\rfloor\right|_P - \left\lfloor \frac{P}{2} \right\rfloor .$$

Здесь $\lfloor a \rfloor$ – стандартное обозначение для наибольшего целого числа, не превосходящего $a \in R:$.

$$\lfloor a \rfloor \le a < \lfloor a \rfloor + 1$$

Модулярная арифметика в симметричных шифрах

1. Мультипликативный примитив

Секретные параметры:

$$P_1, P_2, ..., P_m,\ (P_i, P_{i+1}) = 1,$$

$$M_1, M_2, ..., M_m,\ (M_i, P_i) = 1.$$

Условие согласования:

$$P_1 < P_2 < ... < P_m .$$

Уравнение шифрации:

$$S \text{ – сообщение; } S \in \mathrm{Z}_{P_1} :$$

$$\left|\left|SM_1\right|_{P_1} M_2\right|_{P_2} ... M_m\Big|_{P_m} = C,$$

Шифрация, описываемая последним уравнением, реализуется посредством m - раундов (итераций):

$$\left|SM_1\right|_{P_1} = x_1,$$

$$\left|x_1 M_2\right|_{P_2} = x_2,$$

$$\dots$$

$$\left|x_{m-2} M_{m-1}\right|_{P_{m-1}} = x_{m-1},$$

$$\left|x_{m-1} M_m\right|_{P_m} = C;$$

C -шифрограмма.

Процедура расшифрации (обоснование однозначности).

Так как $(M_m, P_m)=1$, то уравнение последнего раунда разрешимо в Z_{P_m}:

$$\left|x_{m-1}\right|_{P_m} = \left|CM_m^{-1}\right|_{P_m}.$$

Так как согласно $(m-1)$-му раунду шифрации $x_{m-1} \in Z_{P_{m-1}}$ и поскольку $Z_{P_{m-1}} \subset Z_{P_m}$, то

$$\left|x_{m-1}\right|_{P_m} = x_{m-1}.$$

Следовательно, корректно по вычету $\left|x_{m-1}\right|_{P_m}$ восстанавливается уравнение шифрации $(m-1)$-го раунда:

$$\left|x_{m-2}M_{m-1}\right|_{P_{m-1}} = x_{m-1}.$$

Продолжая этот процесс возврата к исходному сообщению, на m-ом шаге получим:

$$S = \left|x_1 M_1^{-1}\right|_{P_1}.$$

В соответствии с двумя интерпретациями вычета процессы шифрации и расшифрации могут разворачиваться двумя путями.

Если модули $P_1, P_2, ..., P_m$ выбираются, скажем, из диапазона

$$2^{t-1} < P_i < 2^t$$

где t-разрядность динамического диапазона компьютера, то вычеты по этим модулям удобно интерпретировать как числа представленные позиционным кодом процессора; поэтому процесс шифрации и расшифрации осуществляются в компьютерных командах над «компьютерными» числами: «выделить целую часть», «найти остаток..», «умножить», «сложить» и т.п.

Так организованную шифросистему будем называть мономодульной длинны m. Для ускорения расшифрации, естественно, предвы-

числить константы $\left|M_1^{-1}\right|_{P_1},...,\left|M_m^{-1}\right|_{P_m}$.

Разумеется, допустима организация прямого произведения n мономодульных схем шифрации фиксированной длинны m и вообще говоря, с различным ключевым материалом.

Мономодульные схемы хорошо укладываются в схемы поточных шифраторов с конвейерной структурой и управляемыми параметрами (M_i).

2. Модулярный примитив

Естественно возникает модулярный вариант, если модули $P_1, P_2,..., P_m$ имеют вид:

$$P_i = \prod_{j=1}^{r_i} p_j^{(i)},$$

где $p_1^{(i)},..., p_{r_i}^{(i)}$ – попарно взаимнопросты. В этом случае интерпретация вычетов для каждого i по $\bmod P_i$ имеет только кодовую трактовку (модулярный код). Теперь, скажем операция

$$x_i := \left|x_{i-1} M_i\right|_{P_i}$$

на i -ом раунде шифрации реализуется следующим образом.

Так как вычет x_{i-1} представлен модулярным кодом по основаниям модуля P_{i-1}, то первым шагом необходимо перевести его в модулярный код по основаниям модуля P_i. Соответствующая операция в модулярной арифметике называется операцией расширения кодового представления с динамического диапазона $Z_{P_{i-1}}$ на динамический диапазон Z_{P_i}. Известно, что эта операция равнозначна восстановлению величины числа. Вторым шагом выполняется модульное умножение результата операции расширения на M_i по основаниям модуля P_i.

Модулярная схема отличается существенно более глубоким перемешиванием на межраундовых шагах шифрации, в связи с перехо-

дом от одних секретных модулей к другим.

Интерес также представляет аддитивный вариант рассматриваемых схем, где вместо операции модульного умножения на M_i на каждом i-ом раунде используется операция модульного сложения.

Объединение этих подходов приводит к криптопримитиву с управляемой структурой алгоритма шифрации, которая управляется состоянием m-разрядного битового регистра: если на i-ом раунде значение переменной $r_i = 1$, то используется мультипликативное преобразование, если же $r_i = 0$, то аддитивное.

1.3. Криптопреобразование горнеровского типа.

Возможны и другие обобщения, например, использование схемы криптопреобразований горнеровского типа. Пусть задан (для простоты изложения) полином 4-ой степени:

$$a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \text{ над } Z_{P_1}.$$

Преобразование его по схеме Горнера имеет вид:

$$a_0 + x\left(a_1 + x\left(a_2 + x\left(a_3 + a_4 x\right)\right)\right),$$

которое порождает следующую итерационную процедуру вычислений значений полинома в точке x:

$$y_1 = a_3 + a_4 x,$$

$$y_2 = a_2 + y_1 x,$$

$$y_3 = a_1 + y_2 x,$$

$$y_4 = a_0 + y_3 x.$$

Криптопримитив горнеровского типа, получается из последней итерационной схемы, если ее переписать в виде:

$$y_1 = s_1 + a_4 k_1,$$

$$y_2 = s_2 + y_1 k_2,$$

$$y_3 = s_3 + y_2 k_3,$$

$$y_4 = s_4 + y_3 k_4,$$

где s_1, s_2, s_3, s_4 - компоненты кодового представления по $\mathrm{mod}\, P_1$ сообщения S; $k_1, k_2, k_3, k_4 \in Z_{P_1}$ – компоненты ключевого материала; a_4 – управляемый параметр.

Криптопреобразование горнеровского типа можно использовать на каждом раунде мультипликативного примитива, причем после каждого раунда целесообразно переставлять компоненты результата преобразования.

Подобные же конструкции синтезируются в кольце полиномов над конечными полями. Таким образом, возникает многообразие криптопримитивов.

Далее, вопросы упираются в комплексную оптимизацию по таким параметрам как стойкость, скорость, программная или аппаратная реализация криптопримитивов.

Стойкость здесь, как правило, обосновывается повышением качества рандомизированного управления, как ключевым материалом, так и структурой алгоритмов шифрации на расширенном пространстве выборок. При этом под рандомизацией понимается управление выборкой посредством того или иного алгоритма детерминированного хаоса или псевдослучайного генератора.

Литература

1. Ленг. С, Алгебра, М., Мир, 1968.
2. Акушский Н.Я, Юдицкий Д.И. Машинная арифметика в остаточных классах, М., "Сов. радио", 1968.
3. Свобода А., Развитие вычислительной техники в Чехословакии. Система счисления остаточных классов (СОК), Кибернетический сборник, т. 8, М., Мир, 1964.

УДК 519.7

О. А. Финько

Параллельные логические вычисления — прикладная область модулярной арифметики

(Краснодарское высшее военное училище (военный институт))

Предложено отображение классической алгебры логики на модулярную арифметику, которое открывает новые уникальные возможности по достижению высоких уровней производительности и отказоустойчивости средств гибких логических вычислений.

Сокращения:

АП — арифметический полином;
БФ — булева (ы) функция (и);
ЛДПФ — логическое дискретное преобразование Фурье (в заданном базисе);
ЛТЧП — логические теоретико-числовые преобразования (в заданном базисе);
ЦОС — цифровая обработка сигналов.

1. Введение

При решении задач синтеза и анализа дискретных устройств, построения систем логического управления сложными техническими объектами и процессами реального времени, реализации средств криптографической защиты информации

возникает необходимость в интенсивной обработке больших объемов логических типов данных. Однако традиционные методы описания логических функций, которые основаны на булевых формулах и полиномах Жегалкина, не обеспечивают требуемой эффективности при реализации их с помощью существующего парка информационно-вычислительных средств.

Ключ к решению этого противоречия дают методы реализации БФ с помощью АП, устанавливающие фундаментальную взаимосвязь между логическими и арифметическими типами данных [1—3]. В ряде работ предприняты усилия к расширению классов арифметико-логических форм БФ на основе представления БФ в спектральной области [4]. Это позволило установить связь разнообразных форм представления булевых функций и методов их синтеза, а также воспользоваться эффективным математическим аппаратом и средствами цифровой обработки сигналов для целей анализа и синтеза БФ.

Ряд важных преимуществ, связанных с ограничением числового диапазона представления результатов промежуточных вычислений, распараллеливанием вычислений, обеспечением контроля ошибок и отказоустойчивости вычислительных структур позволяют получить модулярные преобразования.

В докладе излагаются некоторые результаты автора [5—14], посвященные построению модулярных арифметических форм представления булевых функций, которые должны позволить распространить преимущества модулярной арифметики на ее новую прикладную область — параллельные логические вычисления.

2. Представление систем БФ посредством АП. Постановка задачи

Пусть дана система БФ от n переменных $X = x_1, x_2, \ldots, x_n$:

$$y_1 = f_1(X), \quad y_2 = f_2(X), \quad \ldots, \quad y_d = f_d(X), \qquad (1)$$

где y_j — значение, принимаемое j-й БФ $f_j(X)$; $x_i, y_j \in \{0,1\}$ $(i = 1, \ldots, n; \quad j = 1, \ldots, d)$. При этом кортеж значений БФ $y_d * y_{d-1} * \ldots * y_1$, где $*$ — разделительный знак, интерпретируется как код целого неотрицательного числа Y, представленного в

двоичной системе счисления: $(y_d y_{d-1} \dots y_1)_2 = Y = \sum_{j=1}^{d} y_j 2^{j-1}$,

где $(y_d y_{d-1} \dots y_1)_b = Y$ — представление Y в системе счисления по основанию b.

Пример 1.

Представление Y, соответствующее системе БФ

$$\begin{cases} f_1(X) = \overline{x_1 \oplus x_2}, \\ f_2(X) = \overline{x_1 \vee x_2}, \end{cases} \qquad (2)$$

приведено в табл. 1 (здесь и далее $\vee, \wedge, \oplus, \neg$ — символы операций логического сложения, умножения, сложения по модулю 2 и инверсии соответственно).

Таблица 1.

x_1	x_2	y_1	y_2	Y (десятичная запись)
0	0	1	1	3
0	1	0	0	0
1	0	0	0	0
1	1	0	1	1

2.1. Теорема о представлении системы БФ одним АП

Теорема 1 [1—3]

Произвольный кортеж БФ $f_d(X) * f_{d-1}(X) * \dots * f_1(X)$ может быть представлен АП (В.Д. Малюгин):

$$Y = D(X) = \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \qquad (3)$$

где здесь и далее по тексту статьи $i_1 i_2 \dots i_n = \sum_{u=1}^{n} i_u 2^{n-u}$, $i_u \in \{0, 1\}$;

$x_u^{i_u} = \begin{cases} x_u, & i_u = 1, \\ 1, & i_u = 0; \end{cases}$ $c_i \in Z$ $(i = 0, 1, \dots, 2^n - 1)$ и притом единственным

образом.

В качестве *доказательства* этой теоремы далее будут рассмотрены алгоритмы получения (3).

2.2. Алгебраический метод получения АП. Линейные АП

Алгебраический метод получения АП (3) заключается в реализации следующего алгоритма.

А Л Г О Р И Т М 1

Ш а г 1 . Получение АП $P_j(X)$ для каждой БФ $y_j = f_j(X)$, $j = 1, \ldots, d$:

$$f_j(X) = P_j(X) = \sum_{i=0}^{2^n-1} r_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}. \qquad (4)$$

Ш а г 2 . Получение АП, взвешенных весами 2^{j-1} $\left(j = 1, \ldots, d\right)$:

$$P'_j(X) = P_j(X) 2^{j-1} = \sum_{i=0}^{2^n-1} r'_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \qquad (5)$$

где $r'_{j,i} = r_{j,i} 2^{j-1}$ $\left(j = 1, \ldots, d; \quad i = 0, 1, \ldots, 2^n - 1\right)$.

Ш а г 3 . Получение искомого АП $D(X)$ (3) путем суммирования коэффициентов АП $P'_j(X)$ для всех $j = 1, \ldots, d$:

$$D(X) = \sum_{i=0}^{2^n-1} \sum_{j=1}^{d} r'_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} = \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \qquad (6)$$

где $c_i = \sum_{j=1}^{d} r'_{j,i}$ $\left(i = 0, 1, \ldots, 2^n - 1\right)$.

Пример 2.

Для системы БФ (2) реализация алгоритма 1 имеет вид.

Ш а г 1 . Используя соотношения

$$x_1 \wedge x_2 = x_1 x_2,$$
$$x_1 \vee x_2 = x_1 + x_2 - x_1 x_2,$$
$$x_1 \oplus x_2 = x_1 + x_2 - 2 x_1 x_2,$$

$$\bar{x} = 1 - x,$$

получим

$$f_1(X) = P_1(X) = \overline{x_1 \oplus x_2} = 1 - x_1 - x_2 + 2x_1x_2,$$
$$f_2(X) = P_2(X) = \overline{x_1 \vee x_2} = 1 - x_1 - x_2 + x_1x_2.$$

Ш а г 2 .

$$P_1'(X) = 2^0(1 - x_1 - x_2 + 2x_1x_2) = 1 - x_1 - x_2 + 2x_1x_2,$$
$$P_2'(X) = 2^1(1 - x_1 - x_2 + x_1x_2) = 2 - 2x_1 - 2x_2 + 2x_1x_2.$$

Ш а г 3 .

$$D(X) = 1 + 2 - (1+2)x_1 - (1+2)x_2 + (2+2)x_1x_2 = 3 - 3x_1 - 3x_2 + 4x_1x_2.$$

Из этого примера можно видеть, что числовой диапазон, требуемый для представления коэффициентов и результатов промежуточных вычислений АП, может значительно *превосходить* числовой диапазон, достаточный для представления Y .

Большое значение для представления d -выходных БФ $f(X)$ имеют *линейные* АП $L(X)$, которые определяются выражением

$$U = L(X) = d_0 + \sum_{i=1}^{n} d_i x_i = d_0 + d_1x_1 + \ldots + d_nx_n, \qquad (7)$$

где коэффициенты $d_0, d_1, \ldots, d_n$ — целые числа [3, 15]. При вычислении $f_j(X)$ используется оператор маскирования $\Xi^t\{U\}$ [15], служащий для определения t -го двоичного разряда (выхода) представления $U = a_r 2^{r-1} + \ldots + a_t 2^{t-1} + \ldots + a_2 2^1 + a_1 2^0$, т. е. $\Xi^t\{U\} = a_t$.

Пример 3.

Для линеаризации АП $P_j(X) = x_1 + x_2 - x_1x_2$, соответствующего БФ $f_j(X) = x_1 \vee x_2$ используется введение дополнительной (избыточной) БФ $f_j^{(1)}(X)$. При этом образуется система БФ:

$$f_j^{(1)}(X) = 1 \oplus x_1 \oplus x_2,$$
$$f_j^{(2)}(X) = x_1 \vee x_2.$$

Тогда $U = L(X) = 2^1 f_j^{(2)}(X) + 2^0 f_j^{(1)}(X) = 1 + x_1 + x_2$ и $f_j(X) = \Xi^2\{U\}$.

Таким образом, для представления систем БФ (1) с помощью линейных АП $L(X)$ используется тот же принцип взвешивания представлений БФ с помощью весов 2^i $(i = 0, 1, \ldots)$, что и при построении АП $D(X)$ (3). Однако значения i при этом выбираются с учетом введенных дополнительных БФ.

Пример 4.

Дана система БФ:

$$\begin{cases} f_A(X) = x_1 \wedge x_3, \\ f_B(X) = \bar{x}_1 \wedge x_2, \\ f_C(X) = \bar{x}_2 \wedge x_3. \end{cases} \qquad (8)$$

Для обеспечения линейности результирующего АП добавляют вспомогательные БФ $f_A^{(1)}(X)$, $f_B^{(3)}(X)$, $f_C^{(5)}(X)$ и получают систему БФ:

$$\begin{cases} f_A^{(1)}(X) = x_1 \oplus x_3, \\ f_A^{(2)}(X) = f_A(X), \end{cases} \begin{cases} f_B^{(3)}(X) = \bar{x}_1 \oplus x_2, \\ f_B^{(4)}(X) = f_B(X), \end{cases} \begin{cases} f_C^{(5)}(X) = \bar{x}_2 \oplus x_3, \\ f_C^{(6)}(X) = f_C(X). \end{cases}$$

Далее, в соответствии с (7) и примером 2 имеем:

$$U_A = L'_A(X) = 2^1 f_A^{(1)}(X) + 2^0 f_A^{(2)}(X) = x_1 + x_3,$$
$$U_B = L'_B(X) = 2^1 f_B^{(3)}(X) + 2^0 f_B^{(4)}(X) = 1 - x_1 + x_2,$$
$$U_C = L'_C(X) = 2^1 f_C^{(5)}(X) + 2^0 f_C^{(6)}(X) = 1 - x_2 + x_3.$$

Получаем линейный АП:

$$U = L(X) = 2^0 L'_A(X) + 2^2 L'_B(X) + 2^4 L'_C(X) = 20 - 4x_1 - 12x_2 + 16x_3. \qquad (9)$$

Для определения t-й БФ воспользуемся оператором маскирования $\Xi^t\{Y\}$:

$$f_A(X) = \Xi^2\{U\},$$
$$f_B(X) = \Xi^4\{U\},$$
$$f_C(X) = \Xi^6\{U\}.$$

Отметим, что линейная форма АП (9) достигнута за счет введения избыточных БФ и увеличения числового диапазона, необходимого для представления U в 2^3 раза.

2.3. Матричные преобразования

Под прямым и обратным матричным преобразованием (логическим дискретным преобразованием Фурье — ЛДПФ) понимают соответственно пару преобразований [4]:

$$\mathbf{C} = \mathbf{A}_{2^n}\mathbf{Y}, \qquad (10)$$

$$\mathbf{Y} = \mathbf{A}_{2^n}^{-1}\mathbf{C}, \qquad (11)$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования размерности $2^n \times 2^n$ (базис преобразования); $\mathbf{Y}$ — вектор истинности d-выходной БФ $f(X)$ такой, что $\mathbf{Y} = [\mathbf{Y}_d \mid \mathbf{Y}_{d-1} \mid \ldots \mid \mathbf{Y}_1]^T = \left[Y^{(0)} Y^{(1)} \ldots Y^{(2^n-1)} \right]^T$, где T — символ транспонирования; $Y^{(i)}$ — числовое значение, принимаемое d-выходной БФ $f(X)$ на i-м наборе булевых аргументов обычной таблицы истинности (см. пример 1); $\mathbf{C} = \left[c_0 \quad c_1 \quad \ldots \quad c_{2^n-1} \right]^T$ — вектор коэффициентов АП (3) (арифметический спектр БФ). Матрица $\mathbf{A}_{2^n} = \left[\begin{array}{c|c} \mathbf{A}_{2^{n-1}} & 0 \\ \hline -\mathbf{A}_{2^{n-1}} & \mathbf{A}_{2^{n-1}} \end{array} \right]$ является n-й кронекеровской степенью $\mathbf{A}_{2^n} = \bigotimes_{j=1}^{n} \mathbf{A}_1$ базовой матрицы $\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$; $\mathbf{A}_{2^n}^{-1} = \bigotimes_{j=1}^{n} \mathbf{A}_1^{-1}$, где $\mathbf{A}_1^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ — базовая матрица обратного преобразования. Матрица $-\mathbf{A}_{2^{n-1}}$ образуется из $\mathbf{A}_{2^{n-1}}$ заменой знаков единичных элементов на противоположные. Матричные преобразования хорошо алгоритмизируемы и удобны для практического применения.

Пример 5

Пусть задана трехвыходная БФ, векторы принимаемых значений, которой имеют вид:

$$\mathbf{Y}_1 = [01011011]^T,\ \ \mathbf{Y}_2 = [01100111]^T,\ \ \mathbf{Y}_3 = [01101001]^T.$$

Тогда

$$\mathbf{Y} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix}.$$

Выполняя прямое ЛДПФ (10), получим

$$\mathbf{C} = \mathbf{A}_{2^3} \cdot \mathbf{Y} = \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ \hdashline -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 0 & \\ 7 & x_3 \\ 6 & x_2 \\ -12 & x_2x_3 \\ 5 & x_1 \\ -10 & x_1x_3 \\ -8 & x_1x_2 \\ 19 & x_1x_2x_3 \end{bmatrix}.$$

Из анализа структуры матриц $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ следует, что максимальное количество единичных элементов находится в последней строке обеих матриц. Причем количество единичных элементов с одинаковыми знаками в нижней строке матрицы $\mathbf{A}_{2^n}$ равно 2^{n-1}. Учитывая, что максимальное значение, принимаемое элементами матрицы $\mathbf{Y}$, равно $2^d - 1$ (d — количество реализуемых одновыходных БФ), можно сделать вывод о том, что в результирующей матрице $\mathbf{C}$ максимальное абсолютное значение может иметь коэффициент $abs\left(c_{2^n-1}\right) = 2^{n-1}\left(2^d - 1\right)$, где $abs(a)$ — абсолютная величина a. Для его представления в двоичной системе счисления с учетом необходимости представления знака числа потребуется

$$N_{\mathbf{C}} = \left\lfloor \log_2 (2^{n-1}(2^d - 1)) \right\rfloor + 2 = n + d \qquad (12)$$

двоичных разрядов ($\lfloor x \rfloor$ — наибольшее целое число, не превосходящее x).

Для линейных АП проблема больших коэффициентов является еще более критичной. Однако в этом случае причиной большой величины коэффициентов является, прежде всего, большое количество реализуемых БФ, что в свою очередь вызвано необходимостью введения избыточных БФ, имеющих вспомогательный (служебный) характер.

3. Модулярные арифметико-логические формы

Одномодульной арифметикой будем называть арифметику кольца вычетов Z_m, где m — значение модуля. Наименьший неотрицательный вычет (в дальнейшем — вычет) целого числа N по модулю m будем обозначать как $\langle N \rangle_m^+$.

3.1. Полиномиальные модулярные арифметико-логические формы.

Теорема 2

Если $m > Y_{\max}$, где $Y_{\max}$ — максимальное значение, принимаемое Y, то произвольный кортеж БФ может быть представлен АП:

$$Y = \mu(X) = \left\langle \sum_{i=0}^{2^n - 1} \psi_i x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right\rangle_m^+, \qquad (13)$$

где $\psi_i = \langle c_i \rangle_m^+$, $\left(i = 0, 1, \dots, 2^n - 1\right)$.

Замечание 1. В общем случае $m \geq 2^d$.

Определение 1. Выражение (13) будем называть представлением БФ $f(X)$ на основе модулярной формы АП или обобщенным АП Жегалкина.

Сравнительный анализ АП $D(X)$ и $\mu(X)$ можно выполнить на примере некоторых элементарных БФ (табл. 2).

Принцип реализации БФ на основе одномодульной арифметики

поясняется с помощью блок-схемы, представленной на рис. 1.

Таблица 2.

$f(X)$	$D(X)$	$\mu(X)$
$\bar{x}_i$	$1-x_i$	$\langle 1+(m-1)x_i \rangle_m^+$
$x_1 \wedge x_2$	x_1x_2	x_1x_2
$x_1 \vee x_2$	$x_1+x_2-x_1x_2$	$\langle x_1+x_2+(m-1)x_1x_2 \rangle_m^+$
$x_1 \oplus x_2$	$x_1+x_2-2x_1x_2$	$\langle x_1+x_2+(m-2)x_1x_2 \rangle_m^+$
$\overline{x_1 \wedge x_2}$	$1-x_1x_2$	$\langle 1+(m-1)x_1x_2 \rangle_m^+$
$\overline{x_1 \vee x_2}$	$1-x_1-x_2+x_1x_2$	$\langle 1+(m-1)x_1+(m-1)x_2+x_1x_2 \rangle_m^+$

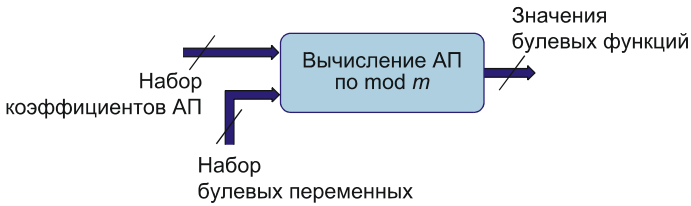


Рис. 1. Принцип реализации БФ
на основе одномодульной арифметики

Следствие 1.

Коэффициенты АП $\mu(X)$ (13) лежат в области целых неотрицательных чисел, а их числовой диапазон равен значению модуля m.

Следствие 2.

Если для одной и той же системы БФ заданы два арифметических полинома $D(X)$ (3) и $\mu(X)$ (13), а K_1 и K_2 — количество членов этих полиномов, то $K_2 \le K_1$.

Для пояснения следствия 2 рассмотрим следующий пример.

Пример 6.

Вернемся к рассмотрению системы БФ (2), которой согласно выражению (3) (пример 2) соответствует АП: $Y = D(X) = 3 - 3x_1 - 3x_2 + 4x_1x_2$. Применение теоремы 2 в общем случае дает: $Y = \mu(X) = \langle 3 + (m-3)x_1 + (m-3)x_2 + 4x_1x_2 \rangle_m^+$. При $m = 4$ получим $\mu(X) = \langle 3 + x_1 + x_2 \rangle_4^+$.

Таким образом, следствие 2 указывает на то, что модулярная форма АП (13) как минимум не усложняет полиномиальной формы представления систем БФ по показателям K_1 и K_2, а как максимум — позволяет уменьшить сложность АП за счет сокращения коэффициентов, кратных m. Следовательно, значение модуля m может выбираться не только по критерию собственной минимальности, но и по критерию минимальности K_2.

Лемма 1.

Если кортеж БФ (1) задан линейным АП (7), то при $m > U_{\max}$ справедлива модулярная форма линейного АП:

$$U = \lambda(X) = \left\langle \omega_0 + \sum_{i=1}^{n} \omega_i x_i \right\rangle_m^+ = \langle \omega_0 + \omega_1 x_1 + \ldots + \omega_n x_n \rangle_m^+, \qquad (14)$$

где $\omega_j = \langle d_j \rangle_m^+ \ (j = 0, 1, \ldots, n)$.

Замечание 2. Значения параметра t оператора $\Xi^t\{U\}$ при переходе от (7) к (14) не изменяются.

Определение 2. Выражение (14) будем называть представлением БФ на основе модулярной формы линейного АП.

Пример 7.

Для системы БФ (8), заданной линейным АП (9), параметр t оператора $\Xi^t\{U\}$ имеет максимальное значение $t_{\max} = 6$ и $U_{\max} = 36$. Выберем $m = 2^6 > 36$. Тогда $U = \langle 20 + 60x_1 + 52x_2 + 16x_3 \rangle_{64}^+$.

Пусть $x_1x_2x_3 = (011)_2$. Следовательно, $U = \langle 88 \rangle_{64}^{+} = (24)_{10} = (011000)_2$. Окончательно имеем:

$$f_A(X) = \Xi^2\{011000\} = 0,$$
$$f_B(X) = \Xi^4\{011000\} = 1,$$
$$f_C(X) = \Xi^6\{011000\} = 0.$$

Связь оператора $\Xi^t\{U\}$ с модулярной арифметикой устанавливается отношением: $\Xi^t\{U\} = \left\langle \left\lfloor \frac{U}{2^t} \right\rfloor \right\rangle_2^{+}$.

Замечание 3. Если для получения U используются избыточные БФ с номерами, превышающими $t_{\max}$ — максимальное значение параметра t оператора $\Xi^t\{U\}$, то модулю m можно присвоить значение $2^{t_{\max}}$. В этом случае вместо U в (14) следует писать $u = \langle U \rangle_{2^{t_{\max}}}^{+}$, при этом $u \le U$.

Таким образом основным свойством модулярной формы АП (13) является уменьшение числового диапазона, требуемого для его вычисления. Прежде чем сделать более точную оценку числового диапазона, рассмотрим принципы реализации *матричных* преобразований, основанных на модулярной арифметике.

3.2. Логические теоретико-числовые преобразования в базисе $\mathbf{A}_{2^n}$.

Теорема 3.

Если для d-выходной БФ $f(X)$ задана пара ЛДПФ (10) и (11) и $m > Y_{\max}$, где $Y_{\max}$ — максимальное значение, принимаемое Y, то справедлива следующая модулярная форма преобразований:

$$\mathbf{\Psi} = \left\langle \mathbf{A}_{2^n} \mathbf{Y} \right\rangle_m^{+}, \qquad (15)$$

$$\mathbf{Y} = \left\langle \mathbf{A}_{2^n}^{-1} \mathbf{\Psi} \right\rangle_m^{+}, \qquad (16)$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования; $\mathbf{Y}$ и $\mathbf{\Psi}$ — соответственно

вектор истинности БФ $f(X)$ и вектор коэффициентов модулярной формы АП $\mu(X)$ (13). Запись $\langle\cdot\rangle_m^+$, означает, что арифметические операции, используемые при произведении матриц $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ на вектор-столбец $\mathbf{Y}$ или $\mathbf{\Psi}$, выполняются по модулю m.

Для доказательства теоремы 3 необходимо учесть взаимооднозначность связи между матричной (10), (11) и полиномиальной (3) формами представления системы БФ. Тогда справедливость (15) и (16) вытекает из справедливости (13).

Полученная пара преобразований имеет много общего с теоретико-числовыми преобразованиями (number theoretic transforms) методов ЦОС [16].

Определение 3. Преобразования (15) и (16) будем соответственно называть модулярной формой прямого и обратного матричного арифметического преобразования или логическими теоретико-числовыми преобразованиями (ЛТЧП, logical number theoretic transforms).

Учитывая, что $\langle -1\rangle_m^+ = m-1$, выражение (15) можно переписать в другой форме:

$$\mathbf{\Psi} = \left\langle \mathbf{M}_{2^n}\,\mathbf{Y} \right\rangle_m^+, \qquad (17)$$

где $\mathbf{M}_{2^n} = \left\langle \mathbf{A}_{2^n} \right\rangle_m^+$. Запись $\left\langle \mathbf{A}_{2^n} \right\rangle_m^+$ означает, что отрицательные элементы (единицы) матрицы $\mathbf{A}_{2^n}$ заменяются на $m-1$.

Пример 8.

Продемонстрируем применение ЛТЧП (15) и (16) к двухвыходной БФ (2) с матрицей истинности, заданной табл. 1 (см. для сопоставления пример 2):

$$\text{ŕμ} = \left\langle \mathbf{A}_{2^2} \cdot \mathbf{Y} \right\rangle_{2^2}^+ = \left\langle \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \hdashline -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 \end{array}\right] \cdot \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\rangle_{2^2}^+ = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \\ x_2 \\ x_1 \\ x_1 x_2 \end{matrix},$$

$$\mathbf{Y} = \left\langle \mathbf{A}_{2^2}^{-1} \cdot \acute{r}\mu \right\rangle_{2^2}^{+} = \left\langle \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hdashline 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right] \cdot \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right\rangle_{2^2}^{+} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Пример 9.

Применение прямого ЛТЧП (17) при $m = 2^3$ к трехвыходной БФ из примера 5 дает результат:

$$\acute{r}\mu = \left\langle \mathbf{M}_{2^3} \mathbf{Y} \right\rangle_{2^3}^{+} = \left\langle \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 7 & 7 & 1 & 0 & 0 & 0 & 0 \\ \hdashline 7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 7 & 0 & 0 & 7 & 1 & 0 & 0 \\ 1 & 0 & 7 & 0 & 7 & 0 & 1 & 0 \\ 7 & 1 & 1 & 7 & 1 & 7 & 7 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix} \right\rangle_{2^3}^{+} = \begin{bmatrix} 0 \\ 7 \\ 6 \\ 4 \\ 5 \\ 6 \\ 0 \\ 3 \end{bmatrix} \begin{matrix} \\ x_3 \\ x_2 \\ x_2 x_3 \\ x_1 \\ x_1 x_3 \\ x_1 x_2 \\ x_1 x_2 x_3 \end{matrix}.$$

По аналогии с ЛДПФ в качестве оценки сложности ЛТЧП выберем размер матрицы-спектра. Для представления элементов матрицы $\mathbf{\Psi}$ потребуется $N_{\mathbf{\Psi}} = \lceil \log_2 m \rceil$ ($\lceil x \rceil$ — наименьшее целое число равное или превышающее x) двоичных разрядов или, при $m = 2^d$, $N_{\mathbf{\Psi}} = d$ двоичных разрядов, что в

$$\frac{N_{\mathbf{C}}}{N_{\mathbf{\Psi}}} = \frac{n}{d} + 1 \qquad (18)$$

раз меньше по сравнению с количеством разрядов $N_{\mathbf{C}}$, необходимых для представления элементов матрицы $\mathbf{C}$ (12).

Так как $N_{\mathbf{C}}$ и $N_{\mathbf{\Psi}}$ — это максимальные размерности (количество двоичных разрядов) коэффициентов АП (3) и (13) соответственно, то оценка (18) применима и к АП (13).

На рис. 2. представлена геометрическая интерпретация получаемого выигрыша в виде представления матриц $\mathbf{Y}$, $\mathbf{C}$ и $\mathbf{\Psi}$ (здесь ширина матриц означает количество двоичных символов, необходимых для представления элементов матриц-столбцов,

ПЛДПФ и ОЛДПФ — соответственно прямое и обратное ЛДПФ, а ПЛТЧП и ОЛТЧП — соответственно прямое и обратное ЛТЧП).

Однако этот выигрыш не удается сохранить для линейных АП, для которых числовой диапазон представления коэффициентов гарантированно можно уменьшить только в два раза — за счет переноса вычислений в область неотрицательных чисел. Препятствием для дальнейшего уменьшения, используемого числового диапазона является большая величина модуля m.

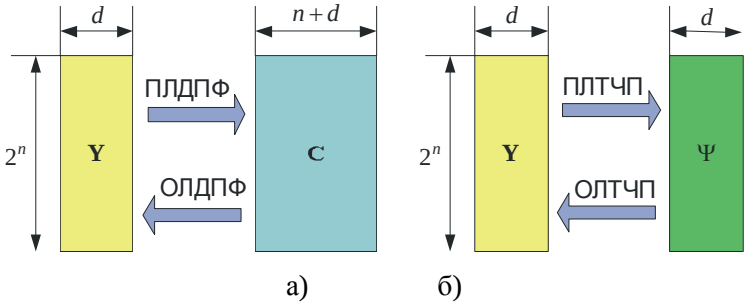


Рис. 2. Геометрическая интерпретация получаемого выигрыша

4. Модулярные арифметико-логические формы, основанные на Китайской теореме об остатках

При моделировании реальных цифровых устройств абсолютные значения коэффициентов линейных АП могут превышать величину 2^{100}. Поэтому требуется поиск более радикальных путей уменьшения используемых числовых диапазонов.

Пусть модуль m для (13) и (14) обладает свойством $m = \prod_{k=1}^{v} m_k$, причем $\gcd(m_i, m_j) = 1$; $i, j = 1, \ldots, v$; $i \neq j$ (здесь и далее $\gcd(a, b)$ — наибольший общий делитель a и b). Тогда в соответствии с Китайской теоремой об остатках Y можно взаимно однозначно отобразить в последовательность $\{Y\} = (\phi_1, \phi_2, \ldots, \phi_v)$, где $\phi_k = \langle Y \rangle_{m_k}^{+}$ $(k = 1, \ldots, v)$. При этом $Y \in Z_m$. Применение для каждого вычета ϕ_k $(k = 1, \ldots, v)$ рассмотренного выше подхода позволяет получить следующие положения.

4.1. Полиномиальные модулярные арифметико-логические формы, основанные на Китайской теореме об остатках

Теорема 4.

Если $m > Y_{\max}$, причем $m = \prod_{k=1}^{v} m_k$ и $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v;\ i \neq j)$, то произвольный кортеж БФ может быть однозначно представлен системой модулярных форм АП:

$$\begin{cases} \phi_1 = \mu_1(X) = \left\langle \sum\limits_{i=0}^{2^n-1} \psi_{i,1} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right\rangle_{m_1}^{+}, \\ \phi_2 = \mu_2(X) = \left\langle \sum\limits_{i=0}^{2^n-1} \psi_{i,2} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right\rangle_{m_2}^{+}, \\ \vdots \\ \phi_v = \mu_v(X) = \left\langle \sum\limits_{i=0}^{2^n-1} \psi_{i,v} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right\rangle_{m_v}^{+}, \end{cases} \qquad (19)$$

где $\psi_{i,k} = \langle c_i \rangle_{m_k}^{+}$ $(i = 0, 1, \ldots, 2^n - 1;\quad k = 1, \ldots, v)$.

Определение 4. Систему АП (19) будем называть полиномиальной формой представления БФ, основанной на Китайской теореме об остатках.

Замечание 4. Модулярные формы (19) и (13) связаны отношениями:

$$\{Y\} = (\phi_1, \phi_2, \ldots, \phi_v),$$

$$(\psi_{i,1}, \psi_{i,2}, \ldots, \psi_{i,v}) = \{\psi_i\} = |c_i|_m^{+} \quad (i = 0, 1, \ldots, 2^n - 1).$$

Для каждого АП системы (19) справедливы и следствия 1 и 2 (при этом вместо m необходимо рассматривать соответствующий модуль m_j $(j = 1, \ldots, v)$).

Лемма 2.

Если кортеж БФ (1) задан линейным АП $L(X)$ (7), то при $m > U_{\max}$, где $m = \prod_{k=1}^{v} m_k$, причем $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v;\ i \neq j)$,

справедлива следующая модулярная форма линейного АП:

$$\begin{cases} \phi_1 = \lambda_1(X) = \left\langle \omega_{0,1} + \omega_{1,1}x_1 + \ldots + \omega_{n,1}x_n \right\rangle_{m_1}^{+}, \\ \phi_2 = \lambda_2(X) = \left\langle \omega_{0,2} + \omega_{1,2}x_1 + \ldots + \omega_{n,2}x_n \right\rangle_{m_2}^{+}, \\ \vdots \\ \phi_v = \lambda_v(X) = \left\langle \omega_{0,v} + \omega_{1,v}x_1 + \ldots + \omega_{n,v}x_n \right\rangle_{m_v}^{+}, \end{cases} \qquad (20)$$

где $\omega_{j,k} = \left\langle d_j \right\rangle_{m_k}^{+} \quad (j = 0, 1, \ldots, n; \quad k = 1, 2, \ldots, v)$.

Справедливость (20) следует из применения доказательства справедливости (14) для каждого номера модуля (20) в отдельности и из Китайской теоремы об остатках.

Определение 5. Систему АП (20) будем называть линейной полиномиальной формой представления БФ, основанной на Китайской теореме об остатках.

Замечание 5. Модулярные формы (20) и (14) связаны отношениями:

$$\{U\} = (\phi_1, \phi_2, \ldots, \phi_v);$$

$$\left(\omega_{j,1}, \omega_{j,2}, \ldots, \omega_{j,v}\right) = \left\{\omega_i\right\} = \left\langle d_j \right\rangle_m^{+} \quad (j = 0, 1, \ldots, n).$$

Для упрощения изложения в дальнейшем не будем различать числа Y и U.

Решение системы равенств

$$\begin{cases} Y \equiv \phi_1 \pmod{m_1}, \\ Y \equiv \phi_2 \pmod{m_2}, \\ \vdots \\ Y \equiv \phi_v \pmod{m_v} \end{cases}$$

дает Китайская теорема об остатках. Для этого будем использовать запись

$$Y = \underset{k=1}{\overset{v}{\mathrm{CRT}}} \phi_k \pmod{m_k}. \qquad (21)$$

В современной трактовке Китайской теоремы об остатках для

вычисления (21) используется формула

$$Y = \operatorname{CRT}_{k=1}^{v} \phi_k \pmod{m_k} = \left\langle \phi_1 B_1 + \phi_2 B_2 + \ldots + \phi_v B_v \right\rangle_m^+, \quad (22)$$

где $B_k = q_k M m_k^{-1}$, q_k находится из сравнения $q_k M m_k^{-1} \equiv 1 \pmod{m_k}$ $(k = 1, \ldots, v)$ (здесь $a \equiv b \pmod{m_k}$ — a сравнимо с b по модулю m_k). Несмотря на классический вид формулы (22) она не всегда удобна для практического использования, в частности, из-за необходимости обеспечения большого числового диапазона. Более приемлемые для технической реализации формулы предложены в [16—19].

Примитивная блок-схема, поясняющая принцип реализации БФ посредством модулярных форм АП, основанных на Китайской теореме об остатках, представлена на рис. 3.

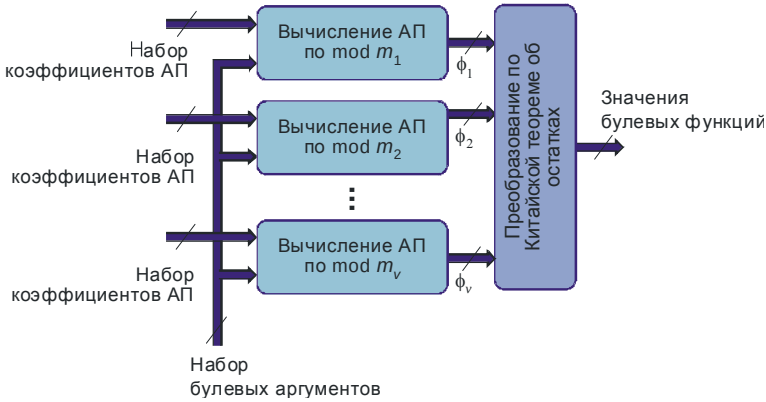


Рис. 3. Примитивная блок-схема принципа реализации БФ

4.2. Теоретико-числовые преобразования в базисе $\mathbf{A}_{2^n}$, основанные на Китайской теореме об остатках.

Лемма 3.

Если для d-выходной БФ $f(X)$ задана пара ЛТЧП (15) и (16) и $m > Y_{\max}$, причем $m = \prod_{k=1}^{v} m_k$ и $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v; \; i \neq j)$, то справедлива следующая модулярная арифметико-логическая форма преобразований:

$$\begin{cases} \boldsymbol{\Psi}_1 = \left\langle \mathbf{A}_{2^n} \boldsymbol{\Phi}_1 \right\rangle_{m_1}^{+}, \\ \boldsymbol{\Psi}_2 = \left\langle \mathbf{A}_{2^n} \boldsymbol{\Phi}_2 \right\rangle_{m_2}^{+}, \\ \vdots \\ \boldsymbol{\Psi}_v = \left\langle \mathbf{A}_{2^n} \boldsymbol{\Phi}_v \right\rangle_{m_v}^{+}; \end{cases} \qquad (23)$$

$$\begin{cases} \boldsymbol{\Phi}_1 = \left\langle \mathbf{A}_{2^n}^{-1} \boldsymbol{\Psi}_1 \right\rangle_{m_1}^{+}, \\ \boldsymbol{\Phi}_2 = \left\langle \mathbf{A}_{2^n}^{-1} \boldsymbol{\Psi}_2 \right\rangle_{m_2}^{+}, \\ \vdots \\ \boldsymbol{\Phi}_v = \left\langle \mathbf{A}_{2^n}^{-1} \boldsymbol{\Psi}_v \right\rangle_{m_v}^{+}, \end{cases} \qquad (24)$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования;

$$\boldsymbol{\Phi}_k = [\phi_k^{(0)}, \phi_k^{(1)}, \ldots, \phi_k^{(2^n-1)}]^T, \ \phi_k^{(r)} = \left\langle Y^{(i)} \right\rangle_{m_k}^{+}, \ k = 1, \ldots, v;$$

$$\boldsymbol{\Psi}_k = [\psi_{0,k}, \psi_{1,k}, \ldots, \psi_{2^n-1,k}]^T \ \ (k = 1, \ldots, v).$$

Доказательство справедливости (23) и (24) следует 1) из взаимооднозначности связи матричной (10) и (11) и полиномиальной (3) форм представления БФ и 2) из доказательства справедливости полиномиальной формы представления (19), основанной на Китайской теореме об остатках.

Определение 6. Пару систем матричных преобразований (23) и (24) будем называть ЛТЧП, основанными на Китайской теореме об остатках (ЛТЧП КТО).

Замечание 6. ЛТЧП КТО (23) и (24) связаны с ЛТЧП (15) и (16) следующими отношениями

$$\boldsymbol{\Psi} = \operatorname*{CRT}_{k=1}^{v} \boldsymbol{\Psi}_k \ (\mathrm{mod}\ m_k), \quad \mathbf{Y} = \operatorname*{CRT}_{k=1}^{v} \boldsymbol{\Phi}_k \ (\mathrm{mod}\ m_k).$$

На рис. 4 показана геометрическая интерпретация ЛТЧП КТО и его

взаимосвязь с ЛДПФ и ЛТЧП.

Согласно этой диаграмме смысл ЛТЧП КТО сводится к разложению каждой из матриц **Y** и **C** на v матриц меньшей «ширины» — $l_1, l_2, \ldots, l_v$, где $l_k = \lceil \log_2 m_k \rceil$, что позволяет упростить преобразование для каждой из полученных матриц $\mathbf{\Psi}_k$ или $\mathbf{\Phi}_k$ $(k = 1, 2, \ldots, v)$ в отдельности. Полученные результаты затем восстанавливаются с помощью Китайской теоремы об остатках. При этом спектр **Ψ** является матрицей ЛТЧП по модулю $m = \prod_{k=1}^{v} m_k$.

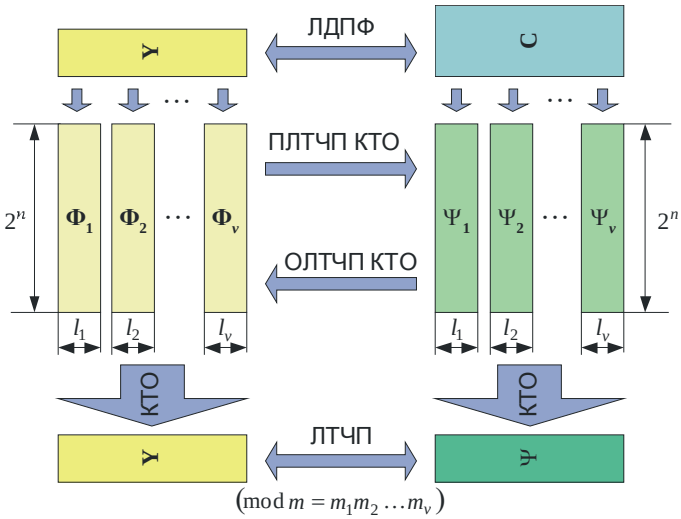


Рис. 4. Геометрическая интерпретация ЛТЧП КТО

5. Заключение

Основы вычислительных методов алгебры логики, используемые в настоящее время, были созданы в «докомпьютерную» эпоху и *плохо согласуются* с методами организации вычислений в современной компьютерной технике. Напротив, арифметическая логика полностью *соответствует* принципам построения современных и перспективных ЭВМ и позволяет раскрыть неиспользуемый в настоящее время потенциал вычислительной техники по реализации высокопроизводительных, гибких,

параллельных логических вычислений.

Модулярные арифметико-логические формы (общая классификация представлена на рис. 5) обладают рядом новых полезных свойств и ориентированы на *воплощение в современную и перспективную практику цифровой обработки информации идей арифметической логики* на основе высокоразвитого и прогрессивного научно-методического аппарата модулярной арифметики.

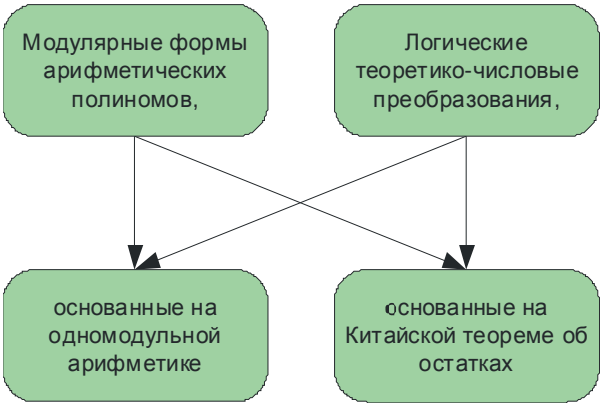


Рис. 5. Общая классификация
модулярных арифметико-логических форм

Достоинствами модулярных арифметико-логических форм являются:

- высокая степень *параллелизма* логических вычислений, которая может быть классифицирована как сверхпараллелизм;
- уникальные возможности по обеспечению *отказоустойчивости* и *живучести* средств логических вычислений;
- обеспечение *контроля* и *коррекции* ошибок на всех стадиях обработки, хранения, а также передачи информации;
- создание благоприятных условий для приоритетного использования быстродействующих *табличных* операционных устройств (в том числе на базе программируемой логики) за счет существенного уменьшения (по сравнению с двоичной системой счисления — на порядки) объема таблиц;
- уменьшение *сложности* представления логических функций на

основе АП;

- возможность *многоцелевого* использования средств логических вычислений, которая, в свою очередь, может быть использована для обеспечения отказоустойчивости и живучести вычислительной системы или для сокращения аппаратурных затрат за счет разделения решения задач во времени;
- реализация широкого класса логических алгоритмов, в том числе и нейросетевых алгоритмов (пороговый элемент представим одним линейным АП, система пороговых элементов (слой) — это система логический функций, поэтому также представима одним линейным АП).

В настоящее время модулярная арифметика широко применяется в методах и средствах ЦОС. Модулярные арифметико-логические формы позволяют *задействовать* высокоразвитый математический аппарат и совершенные технические средства ЦОС, базирующиеся на методах модулярной арифметики, для высококачественной реализации параллельных логических вычислений. В настоящей работе даны основы построения безызбыточных модулярных арифметико-логических форм. Принципы обеспечения контроля ошибок логических вычислений и построения отказоустойчивых вычислительных структур (в классе логических алгоритмов) даны в [5]. Обобщение модулярных арифметико-логических форм на многозначные логические функции дано в [5, 13, 14].

Таким образом, модулярные арифметико-логические формы, по-видимости, позволяют преодолеть важное *противоречие* двух основных способов реализации логических алгоритмов: *программного* (гибкого) и *аппаратного* (жесткого). Логические вычисления, обладающие достоинствами программной реализации, становится возможным реализовать специализированными вычислительными средствами, характеризующимися требуемым комплексом технических характеристик.

Литература

1. *Малюгин В. Д.* Параллельные логические вычисления посредством арифметических полиномов. — М.: Наука. Физматлит, 1997.
2. *Малюгин В. Д.* Реализация булевых функций арифметическими полиномами. — Автоматика и телемеханика. 1982. №4. С. 84–93.
3. *Малюгин В. Д.* Реализация кортежей булевых функций посредством линейных арифметических полиномов. — Автоматика и

телемеханика. 1984. № 2. С. 114–121.
4. *Шмерко В. П.* Синтез арифметических форм булевых функций посредством преобразования Фурье. — Автоматика и телемеханика. 1989. № 5. С. 134–142.
5. *Финько О. А.* Модулярная арифметика параллельных логических вычислений: Монография / Под. ред. В.Д. Малюгина. — М.: ИПУ РАН, 2003. 214 с.
6. *Финько О. А.* Реализация систем булевых функций большой размерности методами модулярной арифметики. — Автоматика и телемеханика. 2004. № 6. С. 37–60.
7. *Финько О. А.* Вариант классификации арифметических форм представления логических функций. — XIV Междунар. школа-семинар «Синтез и сложность управляющих систем». Н. Новгород, 27 октября — 1 ноября 2003. Сборник трудов / Под ред. академика РАН О.Б. Лупанова. — Н. Новгород: Изд-во Нижегородского педагогического ун-та, 2003. С. 83–84.
8. *Финько О. А.* Логические вычисления на основе теоретико-числовых преобразований. — Тр. II Междунар. конф. по проблемам управления (МКПУ II). — М.: Ин-т пробл. упр. им. Трапезникова РАН, Москва, 16–20 июня 2003.
9. *Финько О. А.* Модулярные формы арифметических полиномов для реализации систем булевых функций. — Тр. Междунар. конф. «Искусственные интеллектуальные системы» (IEEE AIS’03) и «Интеллектуальные САПР» (CAD-2003). — М.: Наука. Физматлит, 2003. С. 548–560.
10. *Финько О. А.* Параллельные логические вычисления методами модулярной арифметики. — II Междунар. конф. «Параллельные вычисления и задачи управления» (PACO–2004). Москва, 4–6 октября 2004. Сборник трудов. — М.: Ин-т проблем управ. им. В.А. Трапезникова РАН, 2004. 88 с.
11. *Финько О. А.* Сверхпараллельные логические вычисления методами модулярной арифметики. — Тр. Междунар. конф. «Искусственные интеллектуальные системы» (IEEE AIS’02) и «Интеллектуальные САПР» (CAD-2002). — М.: Наука. Физматлит, 2002. С. 448–455.
12. *Finko O. A.* Methods of problem-oriented representation and data processing in resources of the hardware support of intellectual systems. — Proc. IEEE Conf. Artificial Intelligence Syst. (AIS’02). September 5–10, 2002. P. 453–454. (in USA).
13. *Финько О. А.* Полиномиальная арифметика функций многозначной логики. — Известия вузов. Приборостроение. 2004. Т. 47, № 5. С. 41–46.
14. *Финько О. А.* Модулярные формы *k*-значных функций алгебры логики. — Автоматика и телемеханика. 2005. № 7.
15. *Шмерко В. П.* Теоремы Малюгина: новое понимание в логическом

управлении, проектировании СБИС и структурах данных для новых технологий. — Автоматика и телемеханика. 2004. № 6. С. 6–36.
16. *Норден П., Китте К.* Алгебраическая алгоритмика: Пер. с франц. — М.: Мир, 1999.
17. *Червяков Н. И., Коршунов 0. Е., Финько О. А.* Преобразователь кода системы остаточных классов в позиционный код. А.с. № 1343553. — Б.И. 1987. № 37. С. 288.
18. *Червяков Н.И., Коршунов О. Е., Финько О. А.* Преобразователь кода из системы остаточных классов в позиционный код. А.с. № 1388996. — Б.И. 1988. № 14. С. 167.
19. *Финько О. А.* Восстановление числа в системе остаточных классов с минимальным количеством оснований. — Электронное моделирование. 1998. Т. 20. № 3. С. 56–61. / *Finko O. A.* Number Restoration In the System of Residual Classes With a Minimum Number of Radices. — Engineering Simulation. 1999. V. 16. P. 329–334 (in

УДК 519.7

О. А. Финько

Параллельные логические вычисления — прикладная область модулярной арифметики

(Краснодарское высшее военное училище (военный институт))

Предложено отображение классической алгебры логики на модулярную арифметику, которое открывает новые уникальные возможности по достижению высоких уровней производительности и отказоустойчивости средств гибких логических вычислений.

Сокращения:

АП — арифметический полином;
БФ — булева (ы) функция (и);
ЛДПФ — логическое дискретное преобразование Фурье (в заданном базисе);
ЛТЧП — логические теоретико-числовые преобразования (в заданном базисе);
ЦОС — цифровая обработка сигналов.

1. Введение

При решении задач синтеза и анализа дискретных устройств, построения систем логического управления сложными техническими объектами и процессами реального времени, реализации средств криптографической защиты информации

возникает необходимость в интенсивной обработке больших объемов логических типов данных. Однако традиционные методы описания логических функций, которые основаны на булевых формулах и полиномах Жегалкина, не обеспечивают требуемой эффективности при реализации их с помощью существующего парка информационно-вычислительных средств.

Ключ к решению этого противоречия дают методы реализации БФ с помощью АП, устанавливающие фундаментальную взаимосвязь между логическими и арифметическими типами данных [1—3]. В ряде работ предприняты усилия к расширению классов арифметико-логических форм БФ на основе представления БФ в спектральной области [4]. Это позволило установить связь разнообразных форм представления булевых функций и методов их синтеза, а также воспользоваться эффективным математическим аппаратом и средствами цифровой обработки сигналов для целей анализа и синтеза БФ.

Ряд важных преимуществ, связанных с ограничением числового диапазона представления результатов промежуточных вычислений, распараллеливанием вычислений, обеспечением контроля ошибок и отказоустойчивости вычислительных структур позволяют получить модулярные преобразования.

В докладе излагаются некоторые результаты автора [5—14], посвященные построению модулярных арифметических форм представления булевых функций, которые должны позволить распространить преимущества модулярной арифметики на ее новую прикладную область — параллельные логические вычисления.

2. Представление систем БФ посредством АП. Постановка задачи

Пусть дана система БФ от n переменных $X = x_1, x_2, \ldots, x_n$:

$$y_1 = f_1(X), \quad y_2 = f_2(X), \quad \ldots, \quad y_d = f_d(X), \qquad (1)$$

где y_j — значение, принимаемое j-й БФ $f_j(X)$; $x_i, y_j \in \{0,1\}$ $(i = 1, \ldots, n; \quad j = 1, \ldots, d)$. При этом кортеж значений БФ $y_d * y_{d-1} * \ldots * y_1$, где $*$ — разделительный знак, интерпретируется как код целого неотрицательного числа Y, представленного в

двоичной системе счисления: $(y_d y_{d-1} \dots y_1)_2 = Y = \sum_{j=1}^{d} y_j 2^{j-1}$,

где $(y_d y_{d-1} \dots y_1)_b = Y$ — представление Y в системе счисления по основанию b.

Пример 1.

Представление Y, соответствующее системе БФ

$$\begin{cases} f_1(X) = \overline{x_1 \oplus x_2}, \\ f_2(X) = \overline{x_1 \vee x_2}, \end{cases} \qquad (2)$$

приведено в табл. 1 (здесь и далее $\vee, \wedge, \oplus, \neg$ — символы операций логического сложения, умножения, сложения по модулю 2 и инверсии соответственно).

Таблица 1.

x_1	x_2	y_1	y_2	Y (десятичная запись)
0	0	1	1	3
0	1	0	0	0
1	0	0	0	0
1	1	0	1	1

2.1. Теорема о представлении системы БФ одним АП

Теорема 1 [1—3]

Произвольный кортеж БФ $f_d(X) * f_{d-1}(X) * \dots * f_1(X)$ может быть представлен АП (В.Д. Малюгин):

$$Y = D(X) = \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \qquad (3)$$

где здесь и далее по тексту статьи $i_1 i_2 \dots i_n = \sum_{u=1}^{n} i_u 2^{n-u}$, $i_u \in \{0, 1\}$;

$x_u^{i_u} = \begin{cases} x_u, & i_u = 1, \\ 1, & i_u = 0; \end{cases}$ $c_i \in Z$ $(i = 0, 1, \dots, 2^n - 1)$ и притом единственным

образом.

В качестве *доказательства* этой теоремы далее будут рассмотрены алгоритмы получения (3).

2.2. Алгебраический метод получения АП. Линейные АП

Алгебраический метод получения АП (3) заключается в реализации следующего алгоритма.

А Л Г О Р И Т М 1

Ш а г 1 . Получение АП $P_j(X)$ для каждой БФ $y_j = f_j(X)$, $j = 1, \ldots, d$:

$$f_j(X) = P_j(X) = \sum_{i=0}^{2^n-1} r_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}. \qquad (4)$$

Ш а г 2 . Получение АП, взвешенных весами 2^{j-1} $\left(j = 1, \ldots, d\right)$:

$$P'_j(X) = P_j(X) 2^{j-1} = \sum_{i=0}^{2^n-1} r'_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \qquad (5)$$

где $r'_{j,i} = r_{j,i} 2^{j-1}$ $\left(j = 1, \ldots, d; \quad i = 0, 1, \ldots, 2^n - 1\right)$.

Ш а г 3 . Получение искомого АП $D(X)$ (3) путем суммирования коэффициентов АП $P'_j(X)$ для всех $j = 1, \ldots, d$:

$$D(X) = \sum_{i=0}^{2^n-1} \sum_{j=1}^{d} r'_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} = \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \qquad (6)$$

где $c_i = \sum_{j=1}^{d} r'_{j,i}$ $\left(i = 0, 1, \ldots, 2^n - 1\right)$.

Пример 2.

Для системы БФ (2) реализация алгоритма 1 имеет вид.

Ш а г 1 . Используя соотношения

$$x_1 \wedge x_2 = x_1 x_2,$$
$$x_1 \vee x_2 = x_1 + x_2 - x_1 x_2,$$
$$x_1 \oplus x_2 = x_1 + x_2 - 2 x_1 x_2,$$

$$\bar{x} = 1 - x,$$

получим

$$f_1(X) = P_1(X) = \overline{x_1 \oplus x_2} = 1 - x_1 - x_2 + 2x_1x_2,$$
$$f_2(X) = P_2(X) = \overline{x_1 \vee x_2} = 1 - x_1 - x_2 + x_1x_2.$$

Ш а г 2 .

$$P_1'(X) = 2^0(1 - x_1 - x_2 + 2x_1x_2) = 1 - x_1 - x_2 + 2x_1x_2,$$
$$P_2'(X) = 2^1(1 - x_1 - x_2 + x_1x_2) = 2 - 2x_1 - 2x_2 + 2x_1x_2.$$

Ш а г 3 .

$$D(X) = 1 + 2 - (1+2)x_1 - (1+2)x_2 + (2+2)x_1x_2 = 3 - 3x_1 - 3x_2 + 4x_1x_2.$$

Из этого примера можно видеть, что числовой диапазон, требуемый для представления коэффициентов и результатов промежуточных вычислений АП, может значительно *превосходить* числовой диапазон, достаточный для представления Y .

Большое значение для представления d -выходных БФ $f(X)$ имеют *линейные* АП $L(X)$, которые определяются выражением

$$U = L(X) = d_0 + \sum_{i=1}^{n} d_i x_i = d_0 + d_1x_1 + \ldots + d_nx_n, \qquad (7)$$

где коэффициенты $d_0, d_1, \ldots, d_n$ — целые числа [3, 15]. При вычислении $f_j(X)$ используется оператор маскирования $\Xi^t\{U\}$ [15], служащий для определения t -го двоичного разряда (выхода) представления $U = a_r 2^{r-1} + \ldots + a_t 2^{t-1} + \ldots + a_2 2^1 + a_1 2^0$, т. е. $\Xi^t\{U\} = a_t$.

Пример 3.

Для линеаризации АП $P_j(X) = x_1 + x_2 - x_1x_2$, соответствующего БФ $f_j(X) = x_1 \vee x_2$ используется введение дополнительной (избыточной) БФ $f_j^{(1)}(X)$. При этом образуется система БФ:

$$f_j^{(1)}(X) = 1 \oplus x_1 \oplus x_2,$$
$$f_j^{(2)}(X) = x_1 \vee x_2.$$

Тогда $U = L(X) = 2^1 f_j^{(2)}(X) + 2^0 f_j^{(1)}(X) = 1 + x_1 + x_2$ и $f_j(X) = \Xi^2\{U\}$.

Таким образом, для представления систем БФ (1) с помощью линейных АП $L(X)$ используется тот же принцип взвешивания представлений БФ с помощью весов 2^i $(i = 0, 1, \ldots)$, что и при построении АП $D(X)$ (3). Однако значения i при этом выбираются с учетом введенных дополнительных БФ.

Пример 4.

Дана система БФ:

$$\begin{cases} f_A(X) = x_1 \wedge x_3, \\ f_B(X) = \bar{x}_1 \wedge x_2, \\ f_C(X) = \bar{x}_2 \wedge x_3. \end{cases} \qquad (8)$$

Для обеспечения линейности результирующего АП добавляют вспомогательные БФ $f_A^{(1)}(X)$, $f_B^{(3)}(X)$, $f_C^{(5)}(X)$ и получают систему БФ:

$$\begin{cases} f_A^{(1)}(X) = x_1 \oplus x_3, \\ f_A^{(2)}(X) = f_A(X), \end{cases} \begin{cases} f_B^{(3)}(X) = \bar{x}_1 \oplus x_2, \\ f_B^{(4)}(X) = f_B(X), \end{cases} \begin{cases} f_C^{(5)}(X) = \bar{x}_2 \oplus x_3, \\ f_C^{(6)}(X) = f_C(X). \end{cases}$$

Далее, в соответствии с (7) и примером 2 имеем:

$$U_A = L'_A(X) = 2^1 f_A^{(1)}(X) + 2^0 f_A^{(2)}(X) = x_1 + x_3,$$
$$U_B = L'_B(X) = 2^1 f_B^{(3)}(X) + 2^0 f_B^{(4)}(X) = 1 - x_1 + x_2,$$
$$U_C = L'_C(X) = 2^1 f_C^{(5)}(X) + 2^0 f_C^{(6)}(X) = 1 - x_2 + x_3.$$

Получаем линейный АП:

$$U = L(X) = 2^0 L'_A(X) + 2^2 L'_B(X) + 2^4 L'_C(X) = 20 - 4x_1 - 12x_2 + 16x_3. \qquad (9)$$

Для определения t-й БФ воспользуемся оператором маскирования $\Xi^t\{Y\}$:

$$f_A(X) = \Xi^2\{U\},$$
$$f_B(X) = \Xi^4\{U\},$$
$$f_C(X) = \Xi^6\{U\}.$$

Отметим, что линейная форма АП (9) достигнута за счет введения избыточных БФ и увеличения числового диапазона, необходимого для представления U в 2^3 раза.

2.3. Матричные преобразования

Под прямым и обратным матричным преобразованием (логическим дискретным преобразованием Фурье — ЛДПФ) понимают соответственно пару преобразований [4]:

$$\mathbf{C} = \mathbf{A}_{2^n}\mathbf{Y}, \qquad (10)$$

$$\mathbf{Y} = \mathbf{A}_{2^n}^{-1}\mathbf{C}, \qquad (11)$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования размерности $2^n \times 2^n$ (базис преобразования); $\mathbf{Y}$ — вектор истинности d-выходной БФ $f(X)$ такой, что $\mathbf{Y} = [\mathbf{Y}_d \mid \mathbf{Y}_{d-1} \mid \ldots \mid \mathbf{Y}_1]^T = \left[Y^{(0)} Y^{(1)} \ldots Y^{(2^n-1)} \right]^T$, где T — символ транспонирования; $Y^{(i)}$ — числовое значение, принимаемое d-выходной БФ $f(X)$ на i-м наборе булевых аргументов обычной таблицы истинности (см. пример 1); $\mathbf{C} = \left[c_0 \quad c_1 \quad \ldots \quad c_{2^n-1} \right]^T$ — вектор коэффициентов АП (3) (арифметический спектр БФ). Матрица $\mathbf{A}_{2^n} = \left[\begin{array}{c|c} \mathbf{A}_{2^{n-1}} & 0 \\ \hline -\mathbf{A}_{2^{n-1}} & \mathbf{A}_{2^{n-1}} \end{array} \right]$ является n-й кронекеровской степенью $\mathbf{A}_{2^n} = \bigotimes_{j=1}^{n} \mathbf{A}_1$ базовой матрицы $\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$; $\mathbf{A}_{2^n}^{-1} = \bigotimes_{j=1}^{n} \mathbf{A}_1^{-1}$, где $\mathbf{A}_1^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ — базовая матрица обратного преобразования. Матрица $-\mathbf{A}_{2^{n-1}}$ образуется из $\mathbf{A}_{2^{n-1}}$ заменой знаков единичных элементов на противоположные. Матричные преобразования хорошо алгоритмизируемы и удобны для практического применения.

Пример 5

Пусть задана трехвыходная БФ, векторы принимаемых значений, которой имеют вид:

$$\mathbf{Y}_1 = [01011011]^T,\ \ \mathbf{Y}_2 = [01100111]^T,\ \ \mathbf{Y}_3 = [01101001]^T.$$

Тогда

$$\mathbf{Y} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix}.$$

Выполняя прямое ЛДПФ (10), получим

$$\mathbf{C} = \mathbf{A}_{2^3} \cdot \mathbf{Y} = \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ \hdashline -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 0 & \\ 7 & x_3 \\ 6 & x_2 \\ -12 & x_2x_3 \\ 5 & x_1 \\ -10 & x_1x_3 \\ -8 & x_1x_2 \\ 19 & x_1x_2x_3 \end{bmatrix}.$$

Из анализа структуры матриц $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ следует, что максимальное количество единичных элементов находится в последней строке обеих матриц. Причем количество единичных элементов с одинаковыми знаками в нижней строке матрицы $\mathbf{A}_{2^n}$ равно 2^{n-1}. Учитывая, что максимальное значение, принимаемое элементами матрицы $\mathbf{Y}$, равно $2^d - 1$ (d — количество реализуемых одновыходных БФ), можно сделать вывод о том, что в результирующей матрице $\mathbf{C}$ максимальное абсолютное значение может иметь коэффициент $abs\left(c_{2^n-1}\right) = 2^{n-1}\left(2^d - 1\right)$, где $abs(a)$ — абсолютная величина a. Для его представления в двоичной системе счисления с учетом необходимости представления знака числа потребуется

$$N_{\mathbf{C}} = \left\lfloor \log_2 (2^{n-1}(2^d - 1)) \right\rfloor + 2 = n + d \qquad (12)$$

двоичных разрядов ($\lfloor x \rfloor$ — наибольшее целое число, не превосходящее x).

Для линейных АП проблема больших коэффициентов является еще более критичной. Однако в этом случае причиной большой величины коэффициентов является, прежде всего, большое количество реализуемых БФ, что в свою очередь вызвано необходимостью введения избыточных БФ, имеющих вспомогательный (служебный) характер.

3. Модулярные арифметико-логические формы

Одномодульной арифметикой будем называть арифметику кольца вычетов Z_m, где m — значение модуля. Наименьший неотрицательный вычет (в дальнейшем — вычет) целого числа N по модулю m будем обозначать как $\langle N \rangle_m^+$.

3.1. Полиномиальные модулярные арифметико-логические формы.

Теорема 2

Если $m > Y_{\max}$, где $Y_{\max}$ — максимальное значение, принимаемое Y, то произвольный кортеж БФ может быть представлен АП:

$$Y = \mu(X) = \left\langle \sum_{i=0}^{2^n - 1} \psi_i x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right\rangle_m^+, \qquad (13)$$

где $\psi_i = \langle c_i \rangle_m^+$, $\left(i = 0, 1, \dots, 2^n - 1\right)$.

Замечание 1. В общем случае $m \geq 2^d$.

Определение 1. Выражение (13) будем называть представлением БФ $f(X)$ на основе модулярной формы АП или обобщенным АП Жегалкина.

Сравнительный анализ АП $D(X)$ и $\mu(X)$ можно выполнить на примере некоторых элементарных БФ (табл. 2).

Принцип реализации БФ на основе одномодульной арифметики

поясняется с помощью блок-схемы, представленной на рис. 1.

Таблица 2.

$f(X)$	$D(X)$	$\mu(X)$
$\bar{x}_i$	$1-x_i$	$\langle 1+(m-1)x_i \rangle_m^+$
$x_1 \wedge x_2$	x_1x_2	x_1x_2
$x_1 \vee x_2$	$x_1+x_2-x_1x_2$	$\langle x_1+x_2+(m-1)x_1x_2 \rangle_m^+$
$x_1 \oplus x_2$	$x_1+x_2-2x_1x_2$	$\langle x_1+x_2+(m-2)x_1x_2 \rangle_m^+$
$\overline{x_1 \wedge x_2}$	$1-x_1x_2$	$\langle 1+(m-1)x_1x_2 \rangle_m^+$
$\overline{x_1 \vee x_2}$	$1-x_1-x_2+x_1x_2$	$\langle 1+(m-1)x_1+(m-1)x_2+x_1x_2 \rangle_m^+$

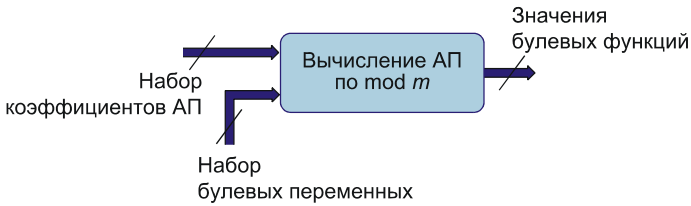


Рис. 1. Принцип реализации БФ
на основе одномодульной арифметики

Следствие 1.

Коэффициенты АП $\mu(X)$ (13) лежат в области целых неотрицательных чисел, а их числовой диапазон равен значению модуля m.

Следствие 2.

Если для одной и той же системы БФ заданы два арифметических полинома $D(X)$ (3) и $\mu(X)$ (13), а K_1 и K_2 — количество членов этих полиномов, то $K_2 \le K_1$.

Для пояснения следствия 2 рассмотрим следующий пример.

Пример 6.

Вернемся к рассмотрению системы БФ (2), которой согласно выражению (3) (пример 2) соответствует АП: $Y = D(X) = 3 - 3x_1 - 3x_2 + 4x_1x_2$. Применение теоремы 2 в общем случае дает: $Y = \mu(X) = \langle 3 + (m-3)x_1 + (m-3)x_2 + 4x_1x_2 \rangle_m^+$. При $m = 4$ получим $\mu(X) = \langle 3 + x_1 + x_2 \rangle_4^+$.

Таким образом, следствие 2 указывает на то, что модулярная форма АП (13) как минимум не усложняет полиномиальной формы представления систем БФ по показателям K_1 и K_2, а как максимум — позволяет уменьшить сложность АП за счет сокращения коэффициентов, кратных m. Следовательно, значение модуля m может выбираться не только по критерию собственной минимальности, но и по критерию минимальности K_2.

Лемма 1.

Если кортеж БФ (1) задан линейным АП (7), то при $m > U_{\max}$ справедлива модулярная форма линейного АП:

$$U = \lambda(X) = \left\langle \omega_0 + \sum_{i=1}^{n} \omega_i x_i \right\rangle_m^+ = \langle \omega_0 + \omega_1 x_1 + \ldots + \omega_n x_n \rangle_m^+, \qquad (14)$$

где $\omega_j = \langle d_j \rangle_m^+ \ (j = 0, 1, \ldots, n)$.

Замечание 2. Значения параметра t оператора $\Xi^t\{U\}$ при переходе от (7) к (14) не изменяются.

Определение 2. Выражение (14) будем называть представлением БФ на основе модулярной формы линейного АП.

Пример 7.

Для системы БФ (8), заданной линейным АП (9), параметр t оператора $\Xi^t\{U\}$ имеет максимальное значение $t_{\max} = 6$ и $U_{\max} = 36$. Выберем $m = 2^6 > 36$. Тогда $U = \langle 20 + 60x_1 + 52x_2 + 16x_3 \rangle_{64}^+$.

Пусть $x_1x_2x_3 = (011)_2$. Следовательно, $U = \langle 88 \rangle_{64}^{+} = (24)_{10} = (011000)_2$. Окончательно имеем:

$$f_A(X) = \Xi^2\{011000\} = 0,$$
$$f_B(X) = \Xi^4\{011000\} = 1,$$
$$f_C(X) = \Xi^6\{011000\} = 0.$$

Связь оператора $\Xi^t\{U\}$ с модулярной арифметикой устанавливается отношением: $\Xi^t\{U\} = \left\langle \left\lfloor \frac{U}{2^t} \right\rfloor \right\rangle_2^{+}$.

Замечание 3. Если для получения U используются избыточные БФ с номерами, превышающими $t_{\max}$ — максимальное значение параметра t оператора $\Xi^t\{U\}$, то модулю m можно присвоить значение $2^{t_{\max}}$. В этом случае вместо U в (14) следует писать $u = \langle U \rangle_{2^{t_{\max}}}^{+}$, при этом $u \le U$.

Таким образом основным свойством модулярной формы АП (13) является уменьшение числового диапазона, требуемого для его вычисления. Прежде чем сделать более точную оценку числового диапазона, рассмотрим принципы реализации *матричных* преобразований, основанных на модулярной арифметике.

3.2. Логические теоретико-числовые преобразования в базисе $\mathbf{A}_{2^n}$.

Теорема 3.

Если для d-выходной БФ $f(X)$ задана пара ЛДПФ (10) и (11) и $m > Y_{\max}$, где $Y_{\max}$ — максимальное значение, принимаемое Y, то справедлива следующая модулярная форма преобразований:

$$\mathbf{\Psi} = \left\langle \mathbf{A}_{2^n} \mathbf{Y} \right\rangle_m^{+}, \qquad (15)$$

$$\mathbf{Y} = \left\langle \mathbf{A}_{2^n}^{-1} \mathbf{\Psi} \right\rangle_m^{+}, \qquad (16)$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования; $\mathbf{Y}$ и $\mathbf{\Psi}$ — соответственно

вектор истинности БФ $f(X)$ и вектор коэффициентов модулярной формы АП $\mu(X)$ (13). Запись $\langle\cdot\rangle_m^+$, означает, что арифметические операции, используемые при произведении матриц $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ на вектор-столбец $\mathbf{Y}$ или $\mathbf{\Psi}$, выполняются по модулю m.

Для доказательства теоремы 3 необходимо учесть взаимооднозначность связи между матричной (10), (11) и полиномиальной (3) формами представления системы БФ. Тогда справедливость (15) и (16) вытекает из справедливости (13).

Полученная пара преобразований имеет много общего с теоретико-числовыми преобразованиями (number theoretic transforms) методов ЦОС [16].

Определение 3. Преобразования (15) и (16) будем соответственно называть модулярной формой прямого и обратного матричного арифметического преобразования или логическими теоретико-числовыми преобразованиями (ЛТЧП, logical number theoretic transforms).

Учитывая, что $\langle -1\rangle_m^+ = m-1$, выражение (15) можно переписать в другой форме:

$$\mathbf{\Psi} = \left\langle \mathbf{M}_{2^n}\,\mathbf{Y} \right\rangle_m^+, \qquad (17)$$

где $\mathbf{M}_{2^n} = \left\langle \mathbf{A}_{2^n} \right\rangle_m^+$. Запись $\left\langle \mathbf{A}_{2^n} \right\rangle_m^+$ означает, что отрицательные элементы (единицы) матрицы $\mathbf{A}_{2^n}$ заменяются на $m-1$.

Пример 8.

Продемонстрируем применение ЛТЧП (15) и (16) к двухвыходной БФ (2) с матрицей истинности, заданной табл. 1 (см. для сопоставления пример 2):

$$\text{ŕμ} = \left\langle \mathbf{A}_{2^2} \cdot \mathbf{Y} \right\rangle_{2^2}^+ = \left\langle \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \hdashline -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 \end{array}\right] \cdot \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\rangle_{2^2}^+ = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \\ x_2 \\ x_1 \\ x_1 x_2 \end{matrix},$$

$$\mathbf{Y} = \left\langle \mathbf{A}_{2^2}^{-1} \cdot \acute{r}\mu \right\rangle_{2^2}^{+} = \left\langle \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hdashline 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right] \cdot \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right\rangle_{2^2}^{+} = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Пример 9.

Применение прямого ЛТЧП (17) при $m = 2^3$ к трехвыходной БФ из примера 5 дает результат:

$$\acute{r}\mu = \left\langle \mathbf{M}_{2^3} \mathbf{Y} \right\rangle_{2^3}^{+} = \left\langle \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 7 & 7 & 1 & 0 & 0 & 0 & 0 \\ \hdashline 7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 7 & 0 & 0 & 7 & 1 & 0 & 0 \\ 1 & 0 & 7 & 0 & 7 & 0 & 1 & 0 \\ 7 & 1 & 1 & 7 & 1 & 7 & 7 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix} \right\rangle_{2^3}^{+} = \begin{bmatrix} 0 \\ 7 \\ 6 \\ 4 \\ 5 \\ 6 \\ 0 \\ 3 \end{bmatrix} \begin{matrix} \\ x_3 \\ x_2 \\ x_2 x_3 \\ x_1 \\ x_1 x_3 \\ x_1 x_2 \\ x_1 x_2 x_3 \end{matrix}.$$

По аналогии с ЛДПФ в качестве оценки сложности ЛТЧП выберем размер матрицы-спектра. Для представления элементов матрицы $\mathbf{\Psi}$ потребуется $N_{\mathbf{\Psi}} = \lceil \log_2 m \rceil$ ($\lceil x \rceil$ — наименьшее целое число равное или превышающее x) двоичных разрядов или, при $m = 2^d$, $N_{\mathbf{\Psi}} = d$ двоичных разрядов, что в

$$\frac{N_{\mathbf{C}}}{N_{\mathbf{\Psi}}} = \frac{n}{d} + 1 \qquad (18)$$

раз меньше по сравнению с количеством разрядов $N_{\mathbf{C}}$, необходимых для представления элементов матрицы $\mathbf{C}$ (12).

Так как $N_{\mathbf{C}}$ и $N_{\mathbf{\Psi}}$ — это максимальные размерности (количество двоичных разрядов) коэффициентов АП (3) и (13) соответственно, то оценка (18) применима и к АП (13).

На рис. 2. представлена геометрическая интерпретация получаемого выигрыша в виде представления матриц $\mathbf{Y}$, $\mathbf{C}$ и $\mathbf{\Psi}$ (здесь ширина матриц означает количество двоичных символов, необходимых для представления элементов матриц-столбцов,

ПЛДПФ и ОЛДПФ — соответственно прямое и обратное ЛДПФ, а ПЛТЧП и ОЛТЧП — соответственно прямое и обратное ЛТЧП).

Однако этот выигрыш не удается сохранить для линейных АП, для которых числовой диапазон представления коэффициентов гарантированно можно уменьшить только в два раза — за счет переноса вычислений в область неотрицательных чисел. Препятствием для дальнейшего уменьшения, используемого числового диапазона является большая величина модуля m.

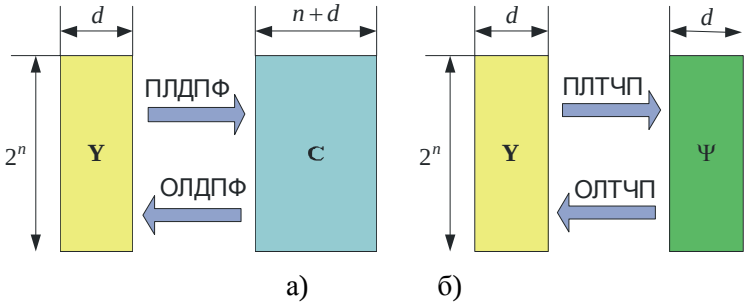


Рис. 2. Геометрическая интерпретация получаемого выигрыша

4. Модулярные арифметико-логические формы, основанные на Китайской теореме об остатках

При моделировании реальных цифровых устройств абсолютные значения коэффициентов линейных АП могут превышать величину 2^{100}. Поэтому требуется поиск более радикальных путей уменьшения используемых числовых диапазонов.

Пусть модуль m для (13) и (14) обладает свойством $m = \prod_{k=1}^{v} m_k$, причем $\gcd(m_i, m_j) = 1$; $i, j = 1, \ldots, v$; $i \neq j$ (здесь и далее $\gcd(a, b)$ — наибольший общий делитель a и b). Тогда в соответствии с Китайской теоремой об остатках Y можно взаимно однозначно отобразить в последовательность $\{Y\} = (\phi_1, \phi_2, \ldots, \phi_v)$, где $\phi_k = \langle Y \rangle_{m_k}^{+}$ $(k = 1, \ldots, v)$. При этом $Y \in Z_m$. Применение для каждого вычета ϕ_k $(k = 1, \ldots, v)$ рассмотренного выше подхода позволяет получить следующие положения.

4.1. Полиномиальные модулярные арифметико-логические формы, основанные на Китайской теореме об остатках

Теорема 4.

Если $m > Y_{\max}$, причем $m = \prod_{k=1}^{v} m_k$ и $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v;\ i \neq j)$, то произвольный кортеж БФ может быть однозначно представлен системой модулярных форм АП:

$$\begin{cases} \phi_1 = \mu_1(X) = \left\langle \sum\limits_{i=0}^{2^n-1} \psi_{i,1} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right\rangle_{m_1}^{+}, \\ \phi_2 = \mu_2(X) = \left\langle \sum\limits_{i=0}^{2^n-1} \psi_{i,2} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right\rangle_{m_2}^{+}, \\ \vdots \\ \phi_v = \mu_v(X) = \left\langle \sum\limits_{i=0}^{2^n-1} \psi_{i,v} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right\rangle_{m_v}^{+}, \end{cases} \qquad (19)$$

где $\psi_{i,k} = \langle c_i \rangle_{m_k}^{+}$ $(i = 0, 1, \ldots, 2^n - 1;\quad k = 1, \ldots, v)$.

Определение 4. Систему АП (19) будем называть полиномиальной формой представления БФ, основанной на Китайской теореме об остатках.

Замечание 4. Модулярные формы (19) и (13) связаны отношениями:

$$\{Y\} = (\phi_1, \phi_2, \ldots, \phi_v),$$

$$(\psi_{i,1}, \psi_{i,2}, \ldots, \psi_{i,v}) = \{\psi_i\} = |c_i|_m^{+} \quad (i = 0, 1, \ldots, 2^n - 1).$$

Для каждого АП системы (19) справедливы и следствия 1 и 2 (при этом вместо m необходимо рассматривать соответствующий модуль m_j $(j = 1, \ldots, v)$).

Лемма 2.

Если кортеж БФ (1) задан линейным АП $L(X)$ (7), то при $m > U_{\max}$, где $m = \prod_{k=1}^{v} m_k$, причем $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v;\ i \neq j)$,

справедлива следующая модулярная форма линейного АП:

$$\begin{cases} \phi_1 = \lambda_1(X) = \left\langle \omega_{0,1} + \omega_{1,1}x_1 + \ldots + \omega_{n,1}x_n \right\rangle_{m_1}^{+}, \\ \phi_2 = \lambda_2(X) = \left\langle \omega_{0,2} + \omega_{1,2}x_1 + \ldots + \omega_{n,2}x_n \right\rangle_{m_2}^{+}, \\ \vdots \\ \phi_v = \lambda_v(X) = \left\langle \omega_{0,v} + \omega_{1,v}x_1 + \ldots + \omega_{n,v}x_n \right\rangle_{m_v}^{+}, \end{cases} \qquad (20)$$

где $\omega_{j,k} = \left\langle d_j \right\rangle_{m_k}^{+} \quad (j = 0, 1, \ldots, n; \quad k = 1, 2, \ldots, v)$.

Справедливость (20) следует из применения доказательства справедливости (14) для каждого номера модуля (20) в отдельности и из Китайской теоремы об остатках.

Определение 5. Систему АП (20) будем называть линейной полиномиальной формой представления БФ, основанной на Китайской теореме об остатках.

Замечание 5. Модулярные формы (20) и (14) связаны отношениями:

$$\{U\} = (\phi_1, \phi_2, \ldots, \phi_v);$$

$$\left(\omega_{j,1}, \omega_{j,2}, \ldots, \omega_{j,v}\right) = \left\{\omega_i\right\} = \left\langle d_j \right\rangle_m^{+} \quad (j = 0, 1, \ldots, n).$$

Для упрощения изложения в дальнейшем не будем различать числа Y и U.

Решение системы равенств

$$\begin{cases} Y \equiv \phi_1 \pmod{m_1}, \\ Y \equiv \phi_2 \pmod{m_2}, \\ \vdots \\ Y \equiv \phi_v \pmod{m_v} \end{cases}$$

дает Китайская теорема об остатках. Для этого будем использовать запись

$$Y = \underset{k=1}{\overset{v}{\mathrm{CRT}}} \phi_k \pmod{m_k}. \qquad (21)$$

В современной трактовке Китайской теоремы об остатках для

вычисления (21) используется формула

$$Y = \operatorname{CRT}_{k=1}^{v} \phi_k \pmod{m_k} = \left\langle \phi_1 B_1 + \phi_2 B_2 + \ldots + \phi_v B_v \right\rangle_m^+, \quad (22)$$

где $B_k = q_k M m_k^{-1}$, q_k находится из сравнения $q_k M m_k^{-1} \equiv 1 \pmod{m_k}$ $(k = 1, \ldots, v)$ (здесь $a \equiv b \pmod{m_k}$ — a сравнимо с b по модулю m_k). Несмотря на классический вид формулы (22) она не всегда удобна для практического использования, в частности, из-за необходимости обеспечения большого числового диапазона. Более приемлемые для технической реализации формулы предложены в [16—19].

Примитивная блок-схема, поясняющая принцип реализации БФ посредством модулярных форм АП, основанных на Китайской теореме об остатках, представлена на рис. 3.

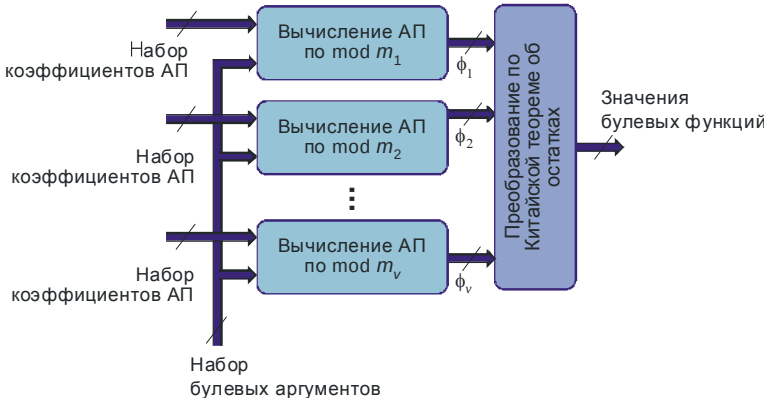


Рис. 3. Примитивная блок-схема принципа реализации БФ

4.2. Теоретико-числовые преобразования в базисе $\mathbf{A}_{2^n}$, основанные на Китайской теореме об остатках.

Лемма 3.

Если для d-выходной БФ $f(X)$ задана пара ЛТЧП (15) и (16) и $m > Y_{\max}$, причем $m = \prod_{k=1}^{v} m_k$ и $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v; \; i \neq j)$, то справедлива следующая модулярная арифметико-логическая форма преобразований:

$$\begin{cases} \boldsymbol{\Psi}_1 = \left\langle \mathbf{A}_{2^n} \boldsymbol{\Phi}_1 \right\rangle_{m_1}^{+}, \\ \boldsymbol{\Psi}_2 = \left\langle \mathbf{A}_{2^n} \boldsymbol{\Phi}_2 \right\rangle_{m_2}^{+}, \\ \vdots \\ \boldsymbol{\Psi}_v = \left\langle \mathbf{A}_{2^n} \boldsymbol{\Phi}_v \right\rangle_{m_v}^{+}; \end{cases} \qquad (23)$$

$$\begin{cases} \boldsymbol{\Phi}_1 = \left\langle \mathbf{A}_{2^n}^{-1} \boldsymbol{\Psi}_1 \right\rangle_{m_1}^{+}, \\ \boldsymbol{\Phi}_2 = \left\langle \mathbf{A}_{2^n}^{-1} \boldsymbol{\Psi}_2 \right\rangle_{m_2}^{+}, \\ \vdots \\ \boldsymbol{\Phi}_v = \left\langle \mathbf{A}_{2^n}^{-1} \boldsymbol{\Psi}_v \right\rangle_{m_v}^{+}, \end{cases} \qquad (24)$$

где $\mathbf{A}_{2^n}$ и $\mathbf{A}_{2^n}^{-1}$ — соответственно матрицы прямого и инверсного арифметического преобразования;

$$\boldsymbol{\Phi}_k = [\phi_k^{(0)}, \phi_k^{(1)}, \ldots, \phi_k^{(2^n-1)}]^T, \ \phi_k^{(r)} = \left\langle Y^{(i)} \right\rangle_{m_k}^{+}, \ k = 1, \ldots, v;$$

$$\boldsymbol{\Psi}_k = [\psi_{0,k}, \psi_{1,k}, \ldots, \psi_{2^n-1,k}]^T \ \ (k = 1, \ldots, v).$$

Доказательство справедливости (23) и (24) следует 1) из взаимооднозначности связи матричной (10) и (11) и полиномиальной (3) форм представления БФ и 2) из доказательства справедливости полиномиальной формы представления (19), основанной на Китайской теореме об остатках.

Определение 6. Пару систем матричных преобразований (23) и (24) будем называть ЛТЧП, основанными на Китайской теореме об остатках (ЛТЧП КТО).

Замечание 6. ЛТЧП КТО (23) и (24) связаны с ЛТЧП (15) и (16) следующими отношениями

$$\boldsymbol{\Psi} = \operatorname*{CRT}_{k=1}^{v} \boldsymbol{\Psi}_k \ (\mathrm{mod}\ m_k), \quad \mathbf{Y} = \operatorname*{CRT}_{k=1}^{v} \boldsymbol{\Phi}_k \ (\mathrm{mod}\ m_k).$$

На рис. 4 показана геометрическая интерпретация ЛТЧП КТО и его

взаимосвязь с ЛДПФ и ЛТЧП.

Согласно этой диаграмме смысл ЛТЧП КТО сводится к разложению каждой из матриц **Y** и **C** на v матриц меньшей «ширины» — $l_1, l_2, \ldots, l_v$, где $l_k = \lceil \log_2 m_k \rceil$, что позволяет упростить преобразование для каждой из полученных матриц $\mathbf{\Psi}_k$ или $\mathbf{\Phi}_k$ $(k = 1, 2, \ldots, v)$ в отдельности. Полученные результаты затем восстанавливаются с помощью Китайской теоремы об остатках. При этом спектр **Ψ** является матрицей ЛТЧП по модулю $m = \prod_{k=1}^{v} m_k$.

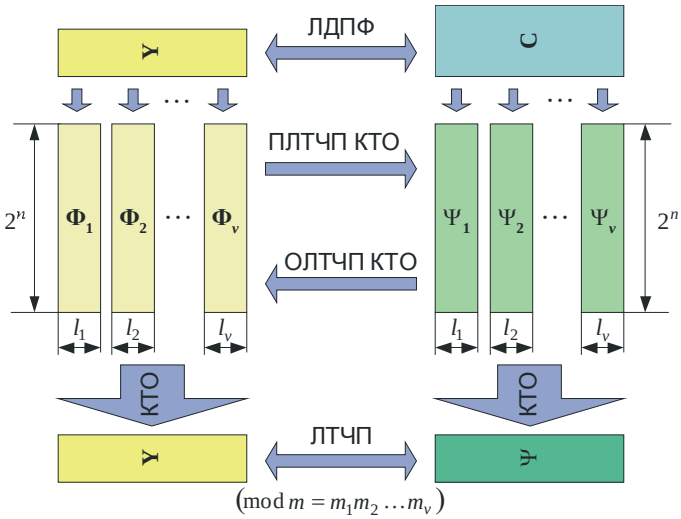


Рис. 4. Геометрическая интерпретация ЛТЧП КТО

5. Заключение

Основы вычислительных методов алгебры логики, используемые в настоящее время, были созданы в «докомпьютерную» эпоху и *плохо согласуются* с методами организации вычислений в современной компьютерной технике. Напротив, арифметическая логика полностью *соответствует* принципам построения современных и перспективных ЭВМ и позволяет раскрыть неиспользуемый в настоящее время потенциал вычислительной техники по реализации высокопроизводительных, гибких,

параллельных логических вычислений.

Модулярные арифметико-логические формы (общая классификация представлена на рис. 5) обладают рядом новых полезных свойств и ориентированы на *воплощение в современную и перспективную практику цифровой обработки информации идей арифметической логики* на основе высокоразвитого и прогрессивного научно-методического аппарата модулярной арифметики.

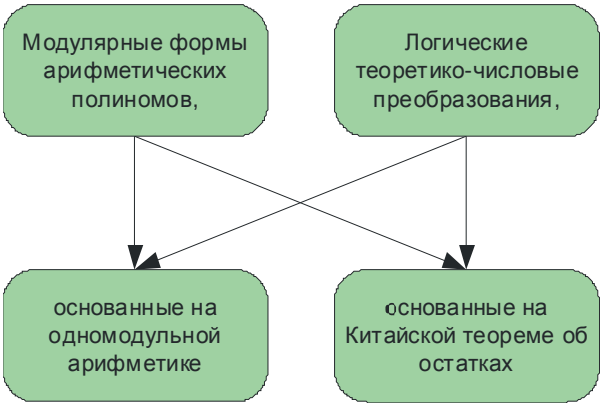


Рис. 5. Общая классификация
модулярных арифметико-логических форм

Достоинствами модулярных арифметико-логических форм являются:

- высокая степень *параллелизма* логических вычислений, которая может быть классифицирована как сверхпараллелизм;
- уникальные возможности по обеспечению *отказоустойчивости* и *живучести* средств логических вычислений;
- обеспечение *контроля* и *коррекции* ошибок на всех стадиях обработки, хранения, а также передачи информации;
- создание благоприятных условий для приоритетного использования быстродействующих *табличных* операционных устройств (в том числе на базе программируемой логики) за счет существенного уменьшения (по сравнению с двоичной системой счисления — на порядки) объема таблиц;
- уменьшение *сложности* представления логических функций на

основе АП;

- возможность *многоцелевого* использования средств логических вычислений, которая, в свою очередь, может быть использована для обеспечения отказоустойчивости и живучести вычислительной системы или для сокращения аппаратурных затрат за счет разделения решения задач во времени;
- реализация широкого класса логических алгоритмов, в том числе и нейросетевых алгоритмов (пороговый элемент представим одним линейным АП, система пороговых элементов (слой) — это система логический функций, поэтому также представима одним линейным АП).

В настоящее время модулярная арифметика широко применяется в методах и средствах ЦОС. Модулярные арифметико-логические формы позволяют *задействовать* высокоразвитый математический аппарат и совершенные технические средства ЦОС, базирующиеся на методах модулярной арифметики, для высококачественной реализации параллельных логических вычислений. В настоящей работе даны основы построения безызбыточных модулярных арифметико-логических форм. Принципы обеспечения контроля ошибок логических вычислений и построения отказоустойчивых вычислительных структур (в классе логических алгоритмов) даны в [5]. Обобщение модулярных арифметико-логических форм на многозначные логические функции дано в [5, 13, 14].

Таким образом, модулярные арифметико-логические формы, по-видимости, позволяют преодолеть важное *противоречие* двух основных способов реализации логических алгоритмов: *программного* (гибкого) и *аппаратного* (жесткого). Логические вычисления, обладающие достоинствами программной реализации, становится возможным реализовать специализированными вычислительными средствами, характеризующимися требуемым комплексом технических характеристик.

Литература

1. *Малюгин В. Д.* Параллельные логические вычисления посредством арифметических полиномов. — М.: Наука. Физматлит, 1997.
2. *Малюгин В. Д.* Реализация булевых функций арифметическими полиномами. — Автоматика и телемеханика. 1982. №4. С. 84–93.
3. *Малюгин В. Д.* Реализация кортежей булевых функций посредством линейных арифметических полиномов. — Автоматика и

телемеханика. 1984. № 2. С. 114–121.
4. *Шмерко В. П.* Синтез арифметических форм булевых функций посредством преобразования Фурье. — Автоматика и телемеханика. 1989. № 5. С. 134–142.
5. *Финько О. А.* Модулярная арифметика параллельных логических вычислений: Монография / Под. ред. В.Д. Малюгина. — М.: ИПУ РАН, 2003. 214 с.
6. *Финько О. А.* Реализация систем булевых функций большой размерности методами модулярной арифметики. — Автоматика и телемеханика. 2004. № 6. С. 37–60.
7. *Финько О. А.* Вариант классификации арифметических форм представления логических функций. — XIV Междунар. школа-семинар «Синтез и сложность управляющих систем». Н. Новгород, 27 октября — 1 ноября 2003. Сборник трудов / Под ред. академика РАН О.Б. Лупанова. — Н. Новгород: Изд-во Нижегородского педагогического ун-та, 2003. С. 83–84.
8. *Финько О. А.* Логические вычисления на основе теоретико-числовых преобразований. — Тр. II Междунар. конф. по проблемам управления (МКПУ II). — М.: Ин-т пробл. упр. им. Трапезникова РАН, Москва, 16–20 июня 2003.
9. *Финько О. А.* Модулярные формы арифметических полиномов для реализации систем булевых функций. — Тр. Междунар. конф. «Искусственные интеллектуальные системы» (IEEE AIS’03) и «Интеллектуальные САПР» (CAD-2003). — М.: Наука. Физматлит, 2003. С. 548–560.
10. *Финько О. А.* Параллельные логические вычисления методами модулярной арифметики. — II Междунар. конф. «Параллельные вычисления и задачи управления» (PACO–2004). Москва, 4–6 октября 2004. Сборник трудов. — М.: Ин-т проблем управ. им. В.А. Трапезникова РАН, 2004. 88 с.
11. *Финько О. А.* Сверхпараллельные логические вычисления методами модулярной арифметики. — Тр. Междунар. конф. «Искусственные интеллектуальные системы» (IEEE AIS’02) и «Интеллектуальные САПР» (CAD-2002). — М.: Наука. Физматлит, 2002. С. 448–455.
12. *Finko O. A.* Methods of problem-oriented representation and data processing in resources of the hardware support of intellectual systems. — Proc. IEEE Conf. Artificial Intelligence Syst. (AIS’02). September 5–10, 2002. P. 453–454. (in USA).
13. *Финько О. А.* Полиномиальная арифметика функций многозначной логики. — Известия вузов. Приборостроение. 2004. Т. 47, № 5. С. 41–46.
14. *Финько О. А.* Модулярные формы *k*-значных функций алгебры логики. — Автоматика и телемеханика. 2005. № 7.
15. *Шмерко В. П.* Теоремы Малюгина: новое понимание в логическом

управлении, проектировании СБИС и структурах данных для новых технологий. — Автоматика и телемеханика. 2004. № 6. С. 6–36.
16. *Норден П., Китте К.* Алгебраическая алгоритмика: Пер. с франц. — М.: Мир, 1999.
17. *Червяков Н. И., Коршунов 0. Е., Финько О. А.* Преобразователь кода системы остаточных классов в позиционный код. А.с. № 1343553. — Б.И. 1987. № 37. С. 288.
18. *Червяков Н.И., Коршунов О. Е., Финько О. А.* Преобразователь кода из системы остаточных классов в позиционный код. А.с. № 1388996. — Б.И. 1988. № 14. С. 167.
19. *Финько О. А.* Восстановление числа в системе остаточных классов с минимальным количеством оснований. — Электронное моделирование. 1998. Т. 20. № 3. С. 56–61. / *Finko O. A.* Number Restoration In the System of Residual Classes With a Minimum Number of Radices. — Engineering Simulation. 1999. V. 16. P. 329–334 (in

УДК 621.29

С.А. Инютин

Модулярные вычисления для задач большой алгоритмической сложности

(Сургутский государственный педагогический университет)

Определена область вычислений в сверхбольших компьютерных диапазонах с многоразрядными числами, в которой модулярная арифметика имеет преимущества перед традиционной -позиционной. На примере характеристических алгоритмов Pepen, Lucas-Lehmer введено понятие вычетной базисной операции. Введена классификация прямых и итерационных вычетных алгоритмов по алгоритмической сложности выполнения базовой операции.

The calculation problems in the super large computer diapason detect as the calculation branch with a advantage for a modular arithmetic. Given the conception of a basic deduction operation at the examples algorithms Pepen, Lucas-Lehmer. Given the classification direct and iteration deduction and modular algorithms with its complexity.

Вычисления с многоразрядными числами или вычисления с величинами, меняющимися в больших диапазонах, являются одной из областей, в которых модулярные вычислительные средства имеют преимущества перед иными, ориентированными на другие вычислительные базы. Важные для теории и практики математические задачи, требующие таких вычислений и больших

вычислительных ресурсов, лежат в областях прикладной и вычислительной теории чисел [1]. Большинство таких задач (или проблем) содержат целочисленные вычисления с числами или числовыми величинами, принимающими значения из больших и сверхбольших машинных диапазонов. В настоящее время интенсивно развивается прикладная теория чисел, отвечая потребности в надежной передаче, хранении и обработке коммерческой и иной цифровой информации. Возникает широкий спектр вычислительных задач [2,7], приводящих к вычислениям, при которых значения целочисленных переменных значительно, в 10^3,…10^6 и более раз превышают максимум типового компьютерного диапазона серийной вычислительной техники, определяемого длиной аппаратно-поддерживаемого машинного слова. Назовём такой диапазон большим целочисленным компьютерным диапазоном. Наличие эффективных методов вычислений в больших диапазонах позволяет ставить задачи вычислений в сверхбольших диапазонах, максимум которых достигает значения константы Виноградова - Гольбаха $3^{3^{15}}$.

Перечислим ряд задач, часто называемых из-за их сложности вычислительными проблемами [2].

1. Тестирования на простоту чисел специального вида.
2. Тестирования на простоту чисел произвольного вида.
3. Вычисление простых делителей и нахождение канонического мультипликативного разложения числа.
4. Поиск больших и сверхбольших простых чисел вида $4 \cdot k+1$.
5. Поиск псевдопростых чисел.
6. Поиск чисел близнецов.
7. Поиск нечетного совершенного числа.
8. Поиск цепочек простых чисел в арифметических прогрессиях.
9. Проверка местоположения нулей дзета-функции Римана.

Данный перечень не является исчерпывающим. Характерной особенностью указанных в нем задач является невозможность их

решения в настоящее время только аналитическими или алгебраическими методами, и по этой причине находят широкое использование вычислительные методы при поиске полного или частичного решения, контрпримеров. На некоторые из этих проблем современная точка зрения такова, что найти их удовлетворительные решения возможно только вычислительными методами. Решения перечисленных задач имеют большое теоретическое значение. На данный момент даже частные решения, полученные вычислительными методами, отдельных из указанных вычислительных проблем, находят, наряду с теоретическим, также широкое практическое приложение. Соотношение между их теоретической значимостью и практической ценностью постоянно меняется [3,10].

При классификации методов вычислений в больших и сверхбольших компьютерных диапазонах необходимо учитывать наряду с постановкой исходной задачи особенности модулярных вычислительных процессов.

Задачи вычислений в сверхбольших компьютерных диапазонах (СБД) определены ранее. Введём классификацию методов вычислений в СБД, взяв в качестве типовых методы, возникающие при решении задач тестирования на простоту чисел специального вида: Ферма и Мерсенна.

Кодонезависимые варианты алгоритмов тестирования имеют близкий вид [8].

Algorithm Pepen.

1. $m \leftarrow 0$, $A \leftarrow 3$
2. $A \leftarrow A^2 (mod\ F_n)$, $m \leftarrow m+1$
3. *if* $2^n - m = 0$ *then end else goto 2*
4. *end: if* $A = 2^{2^n}$ *then* F_n*- prime else* F_n*–not prime*

Algorithm Lucas-Lehmer.

1. $m \leftarrow 0$, $A \leftarrow 4$
2. $A \leftarrow A^2 - 2 (mod\ M_n)$, $m \leftarrow m+1$

3. *if m=n-2 then end else goto 2*
4. *end: if u≡0(mod M_n) then M_n- prime else M_n–not prime*

Базовой операцией этих методов является вычисление вычета от некоторой сверхбольшой величины по модулю, являющемуся большой величиной. Каноническое разложение большого модуля F_n или M_n , как правило, неизвестно. Возможно, что большой модуль является простым.

$$C = | A^{f(B)} |_B = A^{f(B)} - N \cdot B \ ,$$

где *A, B, f(B)* - большие числовые величины;

$A^{f(B)}$ - сверхбольшая числовая величина.

Так как $A^{f(B)}$ -сверхбольшая числовая величина, то прямые алгоритмы, базирующиеся на вычислении соотношения:

$$C = A^{f(B)} - [A^{f(B)} / B] \cdot B \ ,$$

невозможно или нецелесообразно реализовать на вычислительной технике из- за большой алгоритмической и временной сложности.

Модулярная арифметика МС(P^2) позволяет рассматривать это равенство в форме сравнения по модулю $P^2 > C$:

$$C(\mathrm{mod}\, P^2) \equiv$$
$$(A^{f(B)}(\mathrm{mod}\, P^2) - [A^{f(B)} / B](\mathrm{mod}\, P^2) \cdot B(\mathrm{mod}\, P^2))(\mathrm{mod}\, P^2)$$

При таком подходе возможны два способа итерационного вычисления большой величины $C(mod\ P^2)$.

Определение. Вычетным итерационным алгоритмом 1 -рода называется алгоритм, который вычисляют при $B<P^2$, в частности при $B<P$, большую величину $C(mod\ P^2)$:

$$C(\mathrm{mod}\, P^2) = | A^{f(B)} |_B \ (\mathrm{mod}\, P^2) \ ,$$

как последовательность:

$$C(\mathrm{mod}\, P^2) = | \ldots || A^{f_1(B)} |_B^{f_2(B)} |_B \ldots |_B \ (\mathrm{mod}\, P^2) \ ,$$

при этом предполагается выполнение соотношения, характеризующего мультипликативную структуру показателя степени, например заданного каноническим мультипликативным разложением:

$$f(B) = f_1(B) \cdot f_2(B)..... \quad .$$

Определение. Вычетным итерационным алгоритмом 2 -рода называется алгоритм, который вычисляет при $B<P^2$ сверхбольшую величину $N(mod\ P^2)$, а затем большую величину

$$C(mod\ P^2):$$

$$N(\mathrm{mod}\ P^2) =$$

$$[A^{f(B)} / B](\mathrm{mod}\ P^2) = ((A^{f(B)} - X_i(\mathrm{mod}\ P^2)) \cdot \backslash B \backslash^{-1})(\mathrm{mod}\ P^2)^{'}$$

где последовательность больших величин X_i , оцениваемая некоторым образом, стремится к $C(mod\ P^2)$.

Отличие между алгоритмами 1 ,2 -родов заключается в том, что для модулярного алгоритма 2- рода в МС(P^2) целая величина $N(mod\ P^2)$, вычисляемая методом формального деления в МС(P^2), должна совпасть с целой величиной N , полученной фактическим делением на B .

Среди итерационных вычетных алгоритмов 1,2 -родов целесообразно выделение подкласса алгоритмов, основанных на оценивании интервала, в котором может находиться сверхбольшая величина.

Например, для F_n n - числа Ферма выполняются неравенства:

$$2^{f(F_n)} < 3^{f(F_n)} < 4^{f(F_n)} \quad .$$

Если при таком оценивании границы вычисляются с меньшей алгоритмической сложностью и их разность меньше значения большой величины - P^2 , то возможно вычисление соответствующего вычета как алгебраической суммы вычета от значения границы и некоторой фиксированной разницы искомой сверхбольшой величины и границы интервала оценивания:

$$|3^{f(F_n)}|_B = \| 2^{f(F_n)}|_B + |\Delta|_B|_B \ (\mathrm{mod}\, P^2) \quad .$$

Это позволяет ввести следующие определения.

Определение. Алгоритм называется вычетным итерационным оценочным алгоритмом 1 –рода, если вычисляет большую величину $C(mod\ P^2)$ как алгебраическую сумму по модулю B в МС(P^2) некоторой сверхбольшой величины R и разности искомой сверхбольшой величины и R :

$$C = | R |_B + | R - A^{f(B)} |_B - 0 | B(mod\ P^2) \quad .$$

Определение. Алгоритм называется вычетным итерационным оценочным алгоритмом 2 –рода, если вычисляет сверхбольшую величину:

$$N = [\, A^{f(B)} / B \,]$$

как алгебраическую сумму в МС(P^2) некоторой сверхбольшой величины M и разности искомой сверхбольшой величины N и M по модулю P^2 , что позволяет большую величину C вычислять в соответствии с соотношением:

$$C(\mathrm{mod}\, P^2) =$$

$$(A^{f(B)}(\mathrm{mod}\, P^2) - (M(\mathrm{mod}\, P^2) - (N - M)(\mathrm{mod}\, P^2)) \cdot B)(\mathrm{mod}\, P^2)$$

При разработке способов преобразования проблемных алгоритмов в вычетную форму, ориентированную на модулярную арифметику и соответствующую вычислительную базу, необходимо учесть следующие моменты [4,5,6].

Пусть вычислительная задача заключается в получении для содержательной, математически - корректно постановленной проблемы из некоторой области результатов численных расчетов для переменных, изменяющихся в сверхбольших диапазонах. При множестве преобразований вычислительного алгоритма должна сохраняться или заменяться на эквивалентную исходная постановка проблемы. Конечные численные или на их основе качественные результаты должны быть инвариантны к способу получения. На промежуточных этапах вычислительного процесса инвариантность результатов вычислений не требуется. Следовательно, алгоритм вычислительного процесса может быть преобразован в другую, например, «вычетную форму». Даже, если при преобразованиях на этом уровне теряется эквивалентность постановок более высокого уровня, всё равно такие

преобразования возможны на этом уровне, если при этом сохраняется эквивалентность в постановке проблемы на ещё более высоком уровне абстракции. Процесс преобразований может быть продолжен, его естественным пределом является необходимость сохранения инвариантности конечных результатов к методу их получения.

Литература

1. *Рибенбойм П.* Рекорды простых чисел. // Успехи математических наук. Т. 42, вып. 5, 1987. -С.119-176.
2. *Lenstra H.W. , Tijdeman R..J.* Computational Methods in Number Theory. –Amsterdam: Math. Cent., 1982. –198p.
3. *Ноден П. и др*. Алгебраическая алгоритмика. –М.: Мир, 1999. – 720с.
4. *Амербаев В.М.* Теоретические основы машинной арифметики. - Алма-Ата: Наука, 1976. -320 с.
5. *Инютин С.А*. Основы многоразрядной алгоритмики. -Сургут: РИО, 2002. -137с.
6. *Inyutin S.A.* Parallel Square Modular Computer Algebra // Lecture Notes in Computer Science: Parallel Processing and Applied Mathematics (PPAM). – German -Poland: Springer, 2003, -LNCS № 3019. –p. 993-997.
7. *Инютин С.А*. Модулярные вычисления в сверхбольших компьютерных диапазонах // Известия вузов. Электроника. - 2001, -№ 6. –с. 34-39.
8. *Инютин С.А*. Помехозащитные модулярные кодовые конструкции квадратичного диапазона // Вестник Тюменского госуниверситета. –Тюмень: -2003, - № 5. –с. 173-180.
9. *Инютин С.А*. Компьютерная модулярная алгебра квадратичного диапазона и область ее приложения // Вестник Тюменского госуниверситета. –Тюмень: -2001, - № 2. –с. 141-148.
10. *Инютин С.А*. Вычислительные задачи большой алгоритмической сложности и модулярная арифметика // Вестник Тюменского госуниверситета. –Тюмень: -2002, - № 3. – с. 3-9.

УДК. 621.29

А.А. Коляда, Н.А. Коляда, А.Ф. Чернявский

Мультипроцессорная технология модулярных вычислений

(НИИ Прикладных физических проблем им. А.Н. Севченко БГУ, г. Минск)

Изложены принципиальные основы мультипроцессорной технологии модулярной обработки информации. В частности, дана математическая формализация базовой аддитивно-мультипликативной модели процедур алгоритмического ядра, полностью реализуемых в режиме модульных вычислений с применением минимально избыточной модулярной арифметики. Предложен новый метод (метод доминирующего модуля) для преобразования минимально избыточного модулярного кода в позиционный код. Количество модульных операций для выполнения данной процедуры уменьшается, как минимум, в $\log_2 k$ раз (k – число модулей). В сравнении с известными аналогами разработанный метод обладает гораздо большей экономичностью.

Principal foundation of multiprocessor technology modular processings of the information are stated. In particular, mathematical formalization of base additive-multiplicate model of procedures of the algorithmic nucleus completely sold in a mode of modular calculations

with application minimally redundant modular of arithmetics is given. The new method (a method of the dominating module) for transformation a minimally redundant modular code in a positional code is offered. The amount of modular operations for performance of the given procedure decreases, at the minimum, in $\log_2 k$ time (k - number of modules). In comparison with known analogues the developed method possesses much more affectivity.

Как известно, модулярные вычислительные структуры (МВС) в настоящее время широко применяются для решения обширных классов трудоёмких задач в самых разных прикладных областях науки и техники [1-7]. При этом модулярная арифметика (МА), благодаря своему естественному внутреннему параллелизму, в последние годы выдвигается в разряд наиболее приоритетных базовых средств для передовых высокопроизводительных компьютерных технологий таких, в частности, как мультипроцессорная, суперкомпьютерная, нейронносетевая и другие [7-9]. Наряду с крупными успехами в сфере создания и внедрения в практику специализированных высокоскоростных отказоустойчивых БИС- и СБИС-архитектур на основе МВС значительное развитие модулярное направление получило и в части разработки новых классов модулярных систем счисления (МСС), которые отличаются от традиционных аналогов более совершенными процедурами выполнения немодульных операций. К таким МСС, прежде всего, следует отнести минимально избыточные МСС (МИМСС) [1].

Несмотря на то, что высокоскоростные системы модулярной обработки информации (СМОИ) на основе параллельно-конвейерных БИС- и СБИС-архитектур, как правило, обладают большим объёмом оборудования, применение их в соответствующих областях, безусловно, оказывается оправданным. Это относится, например, к таким сферам приложений как цифровая обработка сигналов, высокоточные вычисления, защита информации и т.п. Вместе с тем, в настоящее время интенсивно ведутся многообещающие исследования и конкретные проектные разработки, которые нацелены на реализацию (как можно более полно) фундаментальных преимуществ МА на вычислительных системах позиционного типа. Речь, в частности, идёт о внедрении, так называемых, многомашинной и особенно мультипроцессорной технологий модулярной обработки информации (ТМОИ), причём преимущественно на программном уровне. Обозначенный подход

позволяет синтезировать принципиально новые варианты МА, которые характеризуются несоизмеримо большей свободой выбора значений модулей МСС и объёмов рабочих таблиц, чем в случае применения чисто аппаратного подхода. Это открывает исключительно широкие возможности для расширения пределов действия, так называемого, режима модульных вычислений (РМВ) и, что особенно важно, без использования дорогостоящих специализированных средств.

В настоящей статье излагаются принципиальные основы мультипроцессорной ТМОИ, компьютерно-арифметическую базу которой составляет минимально избыточная МА (МИМА).

Семейство вычислительных процессов, на выполнение которых (полностью в РМВ) ориентирована предлагаемая ТМОИ укладывается в аддитивно-мультипликативную (АМ) модель вида

$$X_{r,\,t}(l) = \sum_{n=0}^{N_{r,\,t,\,l}-1} C_{r,\,t,\,l,\,n}\, x_{r,\,t}(n)$$

$$(r = \overline{0,\, R-1};\; t = \overline{0,\, T_r-1};\; l = \overline{0, L_{r,\,t}-1}), \qquad (1)$$

где $\{X_{r,\,t}(l)\}_{l=\overline{0,L_{s,\,t}-1}}$ и $\{x_{r,\,t}(n)\}_{n=\overline{0,N_{r,\,t,\,l}-1}}$ – соответственно выходная и входная последовательности t-й элементарной (базовой) процедуры r-й стадии (r-го шага) описываемого вычислительного процесса; $C_{r,\,t,\,l,\,n}$ – некоторые константы; R, T_r, $L_{r,\,t}$ и $N_{r,t,l}$ – натуральные числа. Входные последовательности

$$\boldsymbol{x}_{r,t} = \{x_{r,\,t}(n)\}_{n=\overline{0,N_{r,\,t,\,l}-1}}$$

формируются из элементов выходных последовательностей

$$\boldsymbol{X}_{s,\,t} = \{X_{s,\,t}(l)\}_{l=\overline{0,L_{s,\,t}-1}} \quad (s \in \mathbf{Z}_r;\; t \in \mathbf{Z}_{T_s};$$

через $\mathbf{Z}_m$ обозначается кольцо наименьших неотрицательных вычетов по натуральному модулю m, т.е. $\mathbf{Z}_m = \{0,\, 1,\, m-1\}$) определяемых согласно тому или иному правилу элементарных процедур, которые относятся к шагам с нулевого по $(r-1)$-й при $r \neq$

0 и из элементов входной последовательности $\boldsymbol{x}=\{x(n)\}_{n=\overline{0,\,N-1}}$ (N – длина последовательности) рассматриваемой вычислительной процедуры в случае $r = 0$. Аналитически структуру и принцип формирования входных последовательностей элементарных процедур (1), в совокупности составляющих R-шаговый рекурсивный модульный процесс выделенного семейства, в общих чертах можно описать формулой

$$x_{r,\,t}(n) = \begin{cases} x(n_{t,\,l,\,n}), & \text{если } r = 0, \\ X_{s,\,t_r,\,s}(l_{s,\,t_{r,\,s},\,l,\,n}), & \text{если } r \neq 0, \end{cases} \qquad (2)$$

где $\{n_{t,\,l,n}\}_{n=\overline{0,\,N_{0,\,t,\,l}-1}}$ – подмножество элементов кольца $\mathbf{Z}_N$; $s \in \mathbf{Z}_r$; $t_{r,\,s} \in \mathbf{Z}_{T_s}$; конкретный вид отображения $s \to t_{r,\,s}$ зависит от значений параметров r, t и l;

$$\{\forall\, l_{s,\,t_{r,\,s},\,l,\,n} \in Z_{L_{s,\,t_{r,\,s}}} \mid n \in Z_{N_{s,\,t_{r,\,s},\,l}};$$

$$t_{r,\,s} \in \mathbf{Z}_{T_s};\ s \in \mathbf{Z}_r\} = \mathbf{Z}_{N_{r,\,t,\,l}}.$$

Выходная последовательность $\boldsymbol{X}=\{X(l)\}_{l=\overline{0,\,L-1}}$ (L – натуральное число) исходной вычислительной процедуры составляется из элементов выходных последовательностей

$$\boldsymbol{X}_{R-1,t} = \qquad\qquad = \{X_{R-1,\,t\,(l)}\}_{l=\overline{0,\,L_{R-1,\,t}-1}}$$

базовых процедур заключительного (R–1)-го шага реализуемого процесса; при этом, естественно, имеет место равенство

$$\sum_{t=0}^{T_{R-1}-1} L_{R-1,\,t} = L.$$

Элементы подлежащей преобразованию последовательности $\boldsymbol{x}$ =

$\{x(n)\}_{n=\overline{0,N-1}}$, а также фигурирующие в (1) константы $C_{r,\,t,\,l,\,n}$ в общем случае будем считать целыми комплексными числами (ЦКЧ): $x(n) = x'(n) + jx''(n)$; $C_{r,\,t,\,l,\,n} = C'_{r,\,t,\,l,\,n} + j\,C''_{r,\,t,\,l,\,n}$ ($j = \sqrt{-1}$ – мнимая единица). Предполагается, что действительные $x'(n)$ и мнимые $x''(n)$ части элементов $x(n)$ принадлежат диапазону

$\hat{\mathbf{D}} = \mathbf{Z}_{2P}^{-} = \{-P, -P+1, \ldots, P-1\}$ исходных данных используемой МИМСС (P – натуральное число; через $\mathbf{Z}_m^-$ обозначается кольцо абсолютно наименьших вычетов по модулю m: $\mathbf{Z}_m^- = \{-\lfloor m/2 \rfloor, -\lfloor m/2 \rfloor + 1, \ldots, \lceil m/2 \rceil - 1\}$; символические записи $\lfloor x \rfloor$ и $\lceil x \rceil$ употребляются для обозначения ближайших к вещественной величине x соответственно слева и справа целых чисел (ЦЧ)), а действительные $C'_{r,\,t,\,l,\,n}$ и мнимые $C''_{r,\,t,\,l,\,n}$ части коэффициентов $C_{r,\,t,\,l,\,n}$ являются элементами множества $\{-Q, -Q+1, \ldots, Q\}$ (Q – натуральное число). Обычно $C'_{r,\,t,\,l,\,n}$ и $C''_{r,\,t,\,l,\,n}$ представляют собой числители дробей $C'_{r,\,t,\,l,\,n}/Q$ и $C''_{r,\,t,\,l,\,n}/Q$, которые аппроксимируют соответствующие вещественные аналоги $c'_{r,\,t,\,l,\,n}$ и $c''_{r,\,t,\,l,\,n}$ согласно формулам $C'_{r,\,t,\,l,\,n} = \lfloor Q\,c'_{r,\,t,\,l,\,n} \rfloor$ и $C''_{r,\,t,\,l,\,n} = \lceil Q\,c''_{r,\,t,\,l,\,n} \rceil$. Через $\lfloor x \rfloor$ обозначается ближайшее к x ЦЧ:

$$\lfloor x \rfloor = \begin{cases} \lfloor x \rfloor, & \text{если } x < \lfloor x \rfloor + 0{,}5\,, \\ \lceil x \rceil, & \text{если } x \geq \lfloor x \rfloor + 0{,}5. \end{cases}$$

Поскольку при комплексных последовательности **x** и коэффициентах $C_{r,\,t,\,l,\,n}$ входные и выходные последовательности элементарных процедур также комплексные: $\boldsymbol{x}_{r,\,t} = \{x_{r,\,t}(n) = x'_{r,\,t}(n) + x''_{r,\,t}(n)\}_{n=\overline{0,N_{r,\,t,\,l}-1}}$; $\boldsymbol{X}_{r,\,t} = \{X_{r,\,t}(l) = X'_{r,\,t}(l) + j\,X''_{r,\,t}(l)\}_{l=\overline{0,L_{r,\,t}-1}}$, то (1) целесообразно переписать в виде

$$\begin{cases} X'_{r,t}(l) = \sum\limits_{n=0}^{N_{r,t,l}-1} \left(C'_{r,t,l,n}\, x'_{r,t}(n) - C''_{r,t,l,n}\, x''_{r,t}(n) \right) \\ X''_{r,t}(l) = \sum\limits_{n=0}^{N_{r,t,l}-1} \left(C''_{r,t,l,n}\, x'_{r,t}(n) + C'_{r,t,l,n}\, x''_{r,t}(n) \right) \end{cases}, \qquad (3)$$

$$(r = \overline{0,\ R-1};\ t = \overline{0,\ T_r-1};\ l = \overline{0,\ L_{r,t}-1}).$$

Компьютерная реализация описанного семейства вычислительных процессов с применением мультипроцессорной ТМОИ предполагает декомпозицию АМ формы (3) на k независимых модульных субпроцессов:

$$\begin{cases} X'_{r,t,l\,|\,i} = \left| \sum\limits_{n=0}^{N_{r,t,l}-1} \left(\chi'_{r,t,l,n,0\,|\,i} - \chi''_{r,t,l,n,1\,|\,i} \right) \right|_{m_i}, \\ X''_{r,t,l\,|\,i} = \left| \sum\limits_{n=0}^{N_{r,t,l}-1} \left(\chi'_{r,t,l,n,1\,|\,i} + \chi''_{r,t,l,n,0\,|\,i} \right) \right|_{m_i} \end{cases} \qquad (4)$$

$$(r = \overline{0,\ R-1};\ t = \overline{0,\ T_r-1};\ l = \overline{0,\ L_{r,t}-1};\ i = \overline{1,k}),$$

где $X'_{r,t,l|i} = \left| X'_{r,t}(l) \right|_{m_i}$; $X''_{r,t,l|i} = \left| X''_{r,t}(l) \right|_{m_i}$;

$$\chi'_{r,t,l,n,0\,|\,i} = \left| C'_{r,t,l,n}\, \chi'_{r,t,n\,|\,i} \right|_{m_i}; \quad \chi''_{r,t,l,n,1\,|\,i} = \left| C''_{r,t,l,n}\, \chi''_{r,t,n\,|\,i} \right|_{m_i};$$

$$\chi'_{r,t,l,n,1\,|\,i} = \left| C''_{r,t,l,n}\, \chi'_{r,t,n\,|\,i} \right|_{m_i}; \quad \chi''_{r,t,l,n,0\,|\,i} = \left| C'_{r,t,l,n}\, \chi''_{r,t,n\,|\,i} \right|_{m_i};$$

$$\chi'_{r,t,n\,|\,i} = \left| x'_{r,t}(n) \right|_{m_i}; \quad \chi''_{r,t,n\,|\,i} = \left| x''_{r,t}(n) \right|_{m_i};$$

через $|a|_m$ обозначается элемент кольца $\mathbf{Z}_m$, сравнимый с числом a (в общем случае рациональным) по модулю m; m_1, m_2,…, m_k – основания базовой МИМСС; k – число оснований.

Субпроцесс (4) по модулю m_i выполняется независимо от других субпроцессов на отдельном процессоре СМОИ, причём в РМВ, т.е. с использованием только модульных операций. Именно в этом и заключается фундаментальное преимущество мультипроцессорной ТМОИ.

Из вышеприведенного описания АМ модели вычислительных процедур алгоритмического ядра мультипроцессорной ТМОИ следует, что корректность применения РМВ обеспечивают МИМСС, удовлетворяющие теореме.

Теорема 1. Пусть $N_{R-r} = \max\limits_{t,\,l}\{N_{r,\,t,\,l}\}$ ($r = \overline{0, R-1}$). Тогда динамический диапазон $\mathbf{D} = \mathbf{Z}^{-}_{2M} = \{-M, -M+1, \ldots, M-1\}$ ($M = m_0 M_{k-1} = \prod\limits_{i=1}^{k-1} m_i$) МИМСС с основными модулями $m_1, m_2, \ldots, m_k$ и вспомогательным модулем m_0, выбираемым из условий $m_0 \geq \rho$ и $m_k \geq 2m_0 + \rho$, где $\rho = \max\{\rho_{k-1}(X) | X \in \mathbf{Z}_{M_{k-1}}\}$; $\rho_{k-1}(X)$ – ранговая характеристика (k–1)-го порядка, определяемая равенством

$$\left| X \right|_{M_{k-1}} = \sum_{i=1}^{k-1} M_{i,k-1} \left| M^{-1}_{i,\,k-1}\, \chi_i \right|_{m_i} - M_{k-1}\, \rho_{k-1}(X); \qquad (5)$$

$M_{i,\,k-1} = M_{k-1}/m_i$; $\chi_i = \left| X \right|_{m_i}$; включает любые возможные значения величин $X'_{R-1,\,t}(l)$ и $X''_{R-1,\,t}(l)$ ($l = \overline{0, L_{R-1,\,t} - 1}$; $t = \overline{0, T_{R-1} - 1}$), что обеспечивает корректность РМВ (см. (3)), если

$$M > P(2Q)^R \prod_{r=1}^{R} N_r, \qquad (6)$$

где P и Q – полумощности соответственно диапазона $\hat{\mathbf{D}} = \mathbf{Z}^{-}_{2P}$ исходных данных и диапазона $\{-Q, -Q+1, \ldots, Q\}$ используемых констант.

Необходимо отметить, что для многих конкретных вычислительных процедур нижнее пороговое значение полумощности M динамического диапазона **D** МИМСС, определяемое неравенством (6) оказывается существенно завышенным, и оно может быть заменено на более точную оценку.

Согласно представляемой ТМОИ модель (5) вычислительных процессов требует последовательного выполнения лишь трёх принципиально различных с точки зрения МА этапов:

- преобразования элементов входной последовательности ***x*** из позиционного кода (ПК) в минимально избыточный модулярный код (МИМК);
- реализации помодульных компонентов вычислительного процесса;
- перевода элементов выходной последовательности ***X*** из МИМСС в позиционную систему счисления.

Так как в рамках развиваемого мультипроцессорного принципа построения СМОИ существуют исключительно широкие возможности для использования таблиц большой и сверхбольшой ёмкости, то в этих условиях входное кодовое преобразование для $X \in \hat{\mathbf{D}}$ легко осуществляется за время одного обращения к памяти согласно правилу

$$(\chi_1, \chi_2, \ldots, \chi_k) = (TBMOD1[X], TBMOD2[X], \ldots, TBMODk[X]), \qquad (7)$$

где $\chi_i = \left| X \right|_{m_i}$; $TBMODi$ – идентификатор таблицы преобразования $X \to \chi_i$, которая состоит из $2P$ слов разрядностью $b_i = \lceil \log_2 m_i \rceil$ бит; $i = \overline{1,k}$.

Отметим, что если некоторый из модулей МИМСС, например m_k, является двоичной экспонентой: $m_k = 2^{b_k}$, то необходимость в таблице $TBMODk$ отпадает.

Вообще говоря, основания базовой МИМСС для мультипроцессорной ТМОИ целесообразно выбирать как можно большими с тем, что бы их число k было минимальным. Реализация такой стратегии в рамках мультипроцессорного принципа модулярной обработки информации (МОИ) на программном

уровне открывает качественно новые возможности.

Из неравенства $m_k \geq 2m_0 + \rho \geq 2m_0 + k - 2$, которое имеет место для МИМСС [1], а также из (6) следует, что модули m_1, m_2, …, m_k должны удовлетворять условию

$$M_{k-1}(m_k - k + 2) \geq 2M > 2P(2Q)^R \prod_{r=1}^{R} N_r . \quad (8)$$

Исходя из фундаментальных критериев эффективности базовой МИМА, на основании (5) – (8) осуществлен компьютерный анализ наиболее приемлемых вариантов конфигурационных параметров мультипроцессорной ТМОИ. Проведенный анализ позволяет, в частности, заключить:

- один из модулей МИМСС целесообразно выбрать равным степени числа 2, например $m_k = 2^{32}$;
- операции сложения, вычитания и умножения по модулю $m_k = 2^{b_k}$ следует осуществлять с помощью системных аппаратных средств;
- для реализации модульных сумм типа (5) выгодно применять так называемый таблично-аккумулятивный метод, предполагающий суммирование вычетов на процессоре с получением результирующего остатка при помощи таблицы;
- умножение цифр МИМК на константы по модулям целесообразно выполнять посредством двумерных таблиц с использованием на их входах вместо значений констант их порядковых номеров.

Из теоремы 1 видно, что с увеличением объёмов N_r базовых процедур и порядка R рекурсивности АМ модели рассматриваемого семейства вычислительных процессов динамический диапазон **D** применяемой МИМСС неуклонно расширяется. Поэтому выходное кодовое преобразование, т.е. получение по МИМК (χ_1, χ_2, …, χ_k) произвольного $X \in \mathbf{D}$ его ПК, чаще всего, приходится осуществлять с масштабированием. Таким образом, задача заключается в формировании ПК некоторой целочисленной оценки $\hat{X}$ дроби $x = X/S$, где S – выбранный масштаб.

При использовании сверхбольшого модуля интервального индекса (ИИ) вида $m_k = 2^{b_k}$ для выполнения указанной операции выходного кодового преобразования удаётся разработать исключительно простой и эффективный метод, который назван *методом доминирующего модуля*. Его основу составляет нижеследующая теорема.

Теорема 2. Пусть масштабирующий множитель S для элементов выходных последовательностей АМ модели (4) процедур алгоритмического ядра мультипроцессорной ТМОИ имеет вид

$$S = sM_{k-1} \ (s \geq 2k - 2). \qquad (9)$$

Тогда в МИМСС с основаниями $m_1, m_2, \ldots, m_k$ и динамическим диапазоном **D**, удовлетворяющим теореме 1, дробь $x = X/S$ $(X \in \mathbf{D})$ может быть аппроксимирована целочисленной величиной

$$\hat{X} = \lfloor I(X)/s \rfloor, \qquad (10)$$

где I(X) – ИИ числа X, определяемый равенством

$$X = \sum_{i=1}^{k-1} M_{i,\,k-1} \left| M_{i,\,k-1}^{-1} \ \chi_i \right|_{m_i} + M_{i,\,k-1} I(X). \qquad (11)$$

При этом аппроксимация (10) имеет абсолютную погрешность $\left| x - \hat{X} \right| < 1$.

Доказательство. Вычитая и добавляя в правой части интервально-модулярной формы (11) величину $M_{k-1}\rho_{k-1}(X)$ и применяя затем (5), получим

$$X = \sum_{i=1}^{k-1} M_{i,\,k-1} \left| M_{i,\,k-1}^{-1} \ \chi_i \right|_{m_i} - M_{k-1}\rho_{k-1}(X) +$$

$$+ M_{k-1}\rho_{k-1}(X) + M_{k-1} I(X) =$$

$$= \left| X \right|_{M_{k-1}} + M_{k-1}\rho_{k-1}(X) + M_{k-1} I(X). \qquad (12)$$

Применяя симметрическую версию леммы Евклида [1], представим

$I(X)$ в виде $I(X) = \left| I(X) \right|_s^- + \lfloor I(X)/s \rfloor s$ (через $\left| a \right|_m^-$ обозначается элемент множества $\mathbf{Z}_m^-$, сравнимый с a по модулю m). С учётом приведенного равенства, а также (9) из (12) после деления на S имеем

$$X / S = \left(\left| X \right|_{M_{k-1}} + M_{k-1} \rho_{k-1}(X) \right) /$$

$$/ \left(s M_{k-1} \right) + \left| I(X) \right|_s^- / s + \mathrm{I(X)/s}. \qquad (13)$$

Так как

$$0 \le \left| X \right|_{M_{k-1}} < M_{k-1}\ ;\ \rho_{k-1}(X) \le \rho \le k-2 \quad [1];$$

$$s \ge 2(k-1) \text{ (см. (9))}; -0{,}5\, s \le \left| I(X) \right|_s^- < 0{,}5\, s,$$

то

$$0 \le \left(\left| X \right|_{M_{k-1}} + M_{k-1} \rho_{k-1}(X) \right) /$$

$$/(s\, M_{k-1}) < (k-1)\, M_{k-1} / 2(k-1)\, M_{k-1}) = 0{,}5$$

и

$$-0{,}5 \le \left| I(X) \right|_s^- / s < 0{,}5.$$

Поэтому из (13) следует, что для целочисленного приближения (10) к дроби X/S выполняется условие $(X/S - \lfloor I(X)/s \rfloor) \in [-0{,}5;\ 1)$.

Таким образом, теорема 2 доказана.

Как показывает (10), преобразование МИМК в ПК по методу доминирующего модуля практически сводится к вычислению в МИМСС ИИ $I(X)$ исходного числа $X = (\chi_1, \chi_2, \ldots, \chi_k)$.

Как известно [1], интервально-индексная характеристика $I(X)$ восстанавливается по компьютерному ИИ $\hat{I}_k(X) = \left| I(X) \right|_{m_k}$ с помощью формулы

$$I(X)=\begin{cases}\hat{I}_k(X), & \text{если } \hat{I}_k(X)<m_0,\\ \hat{I}_k(X)-m_k, & \text{если } \hat{I}_k(X)\ge m_k-m_0-\rho.\end{cases} \qquad (14)$$

Необходимое расчётное соотношение для $\hat{I}_k(X)$ вытекает из интервально-модулярной формы (11):

$$\hat{I}_k(X)=\left|\sum_{i=1}^{k-1}\mathrm{TII}i[\chi_i]+R_{k,k}(\chi_k)\right|_{m_k}, \qquad (15)$$

где TIIi – идентификатор таблицы ИИ по модулю m_i, формируемой по правилу $\mathrm{TII}\,i[\chi_i] = R_{i,k}(\chi_i) = \left|-m_i^{-1}\left|M_{i,k-1}^{-1}\chi_i\right|_{m_i}\right|_{m_k}$;

$R_{k,k}(\chi_i) = \left|M_{k-1}^{-1}\chi_k\right|_{m_k}$.

Согласно условию $m_k \ge 2m_0+k-2$ для фигурирующего в (15) вспомогательного модуля m_0 верна оценка $m_0 \le \lfloor (m_k-k+2)/2 \rfloor = 2^{b_k-1}+1+\lfloor -k/2 \rfloor$. Заметим, что компьютерная реализация (14), а, значит, и (10) упрощается, если выполняется неравенство

$$m_k-m_0-\rho \ge m_k-m_0-k+2 \ge m_k/2. \qquad (16)$$

В этом случае ввиду $m_k = 2^{b_k}$ вычет $\hat{I}_k(X)$ вычисляемый по формуле (15), представляет собой не что иное как b_k – битовый дополнительный код ИИ $I(X)$ и, следовательно, он непосредственно может использоваться в (10) (вместо прямого кода числа $I(X)$). В соответствии с (16) корректность обозначенного режима вычислений обеспечивается при

$$m_0 \le 0{,}5\,m_k-k+2 = 2^{b_k-1}-k+2.$$

Дальнейшее упрощение операции (10) достигается при использовании множителя $s \ge 2(k-1)$ (см. (9)), который является двоичной экспонентой. Это позволяет заменить операцию деления на ($\log_2 s$) – битовый сдвиг вправо двоичного кода компьютерного ИИ $\hat{I}_k(X)$.

Таким образом, *метод доминирующего модуля* позволяет осуществить преобразование с масштабированием МИМК в ПК за время $(k-1)$ $t_{сум} + t_{ум} + t_{сд}$, где $t_{сум}$, $t_{ум}$ и $t_{сд}$ – времена выполнения соответственно операций сложения, умножения и сдвига. При этом необходимые затраты практически сводятся лишь к одномерным таблицам ИИ: ТПi ($i = \overline{1, k-1}$).

Представленные в настоящей статье результаты исследований по проблематике создания новой, так называемой, мультипроцессорной ТМОИ заключаются в нижеследующем.

1. Проведенная математическая формализация принципа мультипроцессорной МОИ с применением МИМА на программном уровне показывает, что семейство вычислительных процессов, обладающих модульным или квазимодульным операционным спектром, в том числе и рекурсивного типа, адекватно укладывается в АМ модель.
2. Благодаря использованию больших и сверхбольших модулей, а также таблиц большой ёмкости, в рамках мультипроцессорной ТМОИ АМ модель полностью согласуется с РМВ, и в этих условиях фундаментальные преимущества МИМА реализуются в максимальной мере.
3. Предложенный новый метод – *метод доминирующего модуля* для преобразования с масштабированием МИМК в ПК впервые обеспечивает выполнение данной процедуры за число модульных операций порядка $O(k)$. Для наиболее быстродействующих из известных методов [10] соответствующее количество таких операций составляет величину порядка $O(k \log_2 k)$. При этом по сложности реализации разработанный метод обладает гораздо большей экономичностью.

Литература

1. Коляда А.А., Пак И.Т. Модулярные структуры конвейерной обработки цифровой информации. Мн.: Университетское, 1992. 256 с.
2. Чернявский А.Ф., Данилевич В.В., Коляда А.А., Селянинов М.Ю. Высокоскоростные методы и системы цифровой обработки информации. Мн.: Белгосуниверситет, 1996. 376 с.
3. Василевич Л.Н., Коляда А.А., Ревинский В.В. Высокоскоростная модулярная реализация адаптивных цифровых фильтров с конечной

импульсной характеристикой // Весці АН Беларусі. Сер. фіз.-мат. навук. 1997. № 1. С. 126–131.
4. Модулярные принципы построения процессоров для дискретного преобразования Фурье / Л.Н. Василевич, А.А. Коляда, М.Ю. Селянинов, А.Ф. Чернявский // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. 2001. № 4. С. 108–117.
5. Коляда А.А., Селянинов М.Ю. Чернявский А.Ф. Применение модулярной вычислительной технологии в системах защиты информации // Управление защитой информации. 2000. Т.4, №1. С. 27–30.
6. RNS-FPT nerget architectures for orthogonal DWT / J. Ramirez, A. Garsia, P. Fernandez and other // Electron. Lett. 2000. 36, N4. P. 1198–1199.
7. A scalable parallel algorithm for training a hierarchical mixture of neural experts / P.F. Estevez, M.E. Paugam, D. Puzenat, M Ugarte // Parallel Comput. 2002. 28, N6. P. 861-891.
8. Plessmann K. A parallel highly modular object-oriented computer architecture // 10 юбил. Международн. Симп. по пробл. модулярных инф.-выч. сист. и сетей.- Санкт-Петербург, Россия, 13-18 сент., 1993. Пленар. докл. М., 1996. С. 97–109.
9. A modular multi-PC system for real-time applications / K. Plessmann, J. Wollert and others // там же. С. 110–119.
10. Инютин С.А. Модулярные вычисления в сверхбольших компьютерных диапазонах // Изв. вузов. Электрон. 2001. №6. С. 65–73.

УДК. 681.3

Н.И. Червяков

Методы и принципы построения модулярных нейрокомпьютеров

(Ставропольский военный институт связи ракетных войск)

В статье рассмотрены вопросы модулярной (остаточной) арифметики и ее применение в разработке и организации модулярных нейрокомпьютерных вычислительных машин. Особое внимание уделено алгоритмам вычислений позиционных характеристик, время определения которых влияет на быстродействие модулярного нейрокомпьютера.

Противоречие между вычислительной сложностью и скорость определения позиционных характеристик разрешено путем совместного применения китайской теоремы об остатках (КТО) и модифицированного алгоритма Х. Гарнера, что позволило найти универсальную позиционную характеристику, представленную в виде коэффициентов обобщенной позиционной системы счисления.

Проведена классификация основных базисных операций и их реализация в нейросетевом базисе. Введено понятие мультинейропроцессора. Предложено для коррекции ошибок использовать сеть Хопфилда и Хэмминга. Разработана адаптивная параллельно-конвейерная нейронная сеть для обнаружения, локализации и исправления ошибок. Рассмотрен принцип построения отказоустойчивых модулярных нейрокомпьютеров. Показано, что разработка нового класса модулярных

нейрокомпьютеров связана с успехами развития программируемых логических интегральных схем.

Ресурсы вычислительной техники, функционирующей в позиционной системе счисления, постоянно совершенствуются и увеличиваются, но они не могут быть безграничными в принципе. Для решения многих научных, технических и промышленных задач не хватает мощности современных компьютеров. Поиски новых путей повышения эффективности вычислений привели исследователей к объективному выводу, что в рамках обычной позиционной системы счисления скачкообразного ускорения выполнения операций добиться невозможно. Те или иные отдельные приемы совершенствования алгоритмов выполнения операций, развития архитектурных особенностей и др. способствуют увеличению производительности, но оставляют ее в рамках фон-неймановской возможности. Выход из этой ситуации найден в привлечении новых идей в области организации и функционирования ЭВМ на основе остаточной арифметики, которая обладает специфическими характеристиками, являющимися необычайными и полезными. Для представления и обработки цифровой информации остаточная арифметика предлагает ценный взгляд и более широкую перспективу, так как система остаточных классов представляет решительное отклонение от хорошо известных, на вид неизменных законов, которые управляют системами весовых (позиционных) чисел с фиксированными основаниями, т.е. десятичные и двоичные системы счисления. Стимулирующим фактором развития остаточной арифметики является нейроматематика – новый, перспективный раздел вычислительной математики, связанный с разработкой методов и алгоритмов решения задач в нейросетевом базисе, обладающего свойствами параллельного представления и обработки информации. Предпосылкой к созданию нейрокомпьютерных вычислительных средств с модулярным представлением и обработкой данных является семантическое сходство математических моделей нейронных сетей и китайской теоремы остатков [1, 2]. Оказалось, что новая организация ЭВМ, построенная на основе согласованных свойств остаточной (модулярной) арифметики и нейронных сетей позволяет выйти на следующий более прогрессивный этап развития параллельных принципов обработки информации, который обеспечивает решение более сложных задач.

Это утверждение основано на принципе адекватности математических моделей системы остаточных классов и нейронных сетей, обладающих свойствами массово-параллельных вычислений, нейроны которых выполняют роль арифметических элементов, имеющих характеристики по модулю. Если количество синапсов нейронной сети согласовано с количеством оснований системы остаточных классов, то нейронная сеть становится естественным представлением системы остаточных классов, что и послужило началом исследования совместного использования искусственных нейронных сетей и системы остаточных классов с целью создания нового класса ЭВМ – модулярных нейрокомпьютеров [1, 2].

Рассматривая вычисления в пределах модулярного кода все операции, выполняемые нейрокомпьютером, можно разбить на две группы: модульные и немодульные. Первая группа, составляющая арифметические операции (сложение, вычитание и умножение) в остаточной арифметике включает в себя быстрые операции, в то время операции второй группы являются медленными. Вторая группа включает нахождение вычета числа, его преобразование, определение переполнения диапазона, расширение базы оснований, масштабирование и другие и требует иногда частичного преобразования в позиционную систему. Следовательно, для того, чтобы решить проблемы, для которых остаточная арифметика хорошо подходит или наоборот, чтобы определить использовать ли остаточную арифметику для решения заданной проблемы, необходимо рассмотреть такие арифметические операции, которые будут необходимы. Существенным является сокращение медленных операций в общей системе используемых операций, либо разработать методы и алгоритмы быстрого выполнения медленных операций, тогда остаточная арифметика будет иметь более широкую применимость. Решение проблемы можно достичь, если решать их методами, подходящими для позиционных систем счисления.

Центральной операцией для ряда вычислительных проблем является операция вычисления вычетов по модулям, входящим в выбранную систему модулей модулярной арифметики. Эта операция должна быть базисной в конструктируемой компьютерной арифметики, которая накладывает требования на выбор и алгоритмы модульных и немодульных операций,

обеспечивающих максимальную простоту операции вычисления вычетов.

Отображение вычислительного алгоритма в систему остаточных классов для выполнения на модулярной вычислительной базе является настройкой вычисляемых алгоритмов на вычислительную базу. Взаимная настройка компьютерной арифметики на вычислительную проблему позволяет получить оптимальные по быстродействию и аппаратным затратам вычислительные алгоритмы [3, 4].

Разработка теоретических основ вычислительных средств на базе модулярной арифметики требует решения проблемы, заключающейся в замене числового значения модулярной величиной, которая определенным образом связана со значениями компонент модулярного представления. Эти операции являются медленными, например, к таким операциям можно отнести сравнение чисел, определение знака числа, переполнение машинного диапазона и другие [1-6].

Именно медленность их выполнения сдерживает широкое применение модулярной машинной арифметики.

Для решения этой проблемы целесообразно использовать два пути:

- Первый путь – поиск простых алгоритмов выполнения этих операций.
- Второй путь – реализация вычислительных алгоритмов с помощью быстродействующих модульных операций.

Для реализации первого подхода предлагаются специальные функционалы [1-6] – количественные характеристики отношения порядка над множеством модулярных векторов, элементы декартового произведения колец-классов вычетов по простым модулям. Одно из устоявшихся названий для функционалов – позиционные характеристики (ПХ) модулярной величины или числовой величины в модулярном коде.

К позиционным характеристикам предъявляются противоречивые требования, ПХ должны [3-6]:

- быть универсальной основой большинства (или всех) немодульных операций, включая операции кодирования и

декодирования помехозащищенного кода;

- обладать меньшей сложностью выполнения немодульных операций по сравнению с другими методами;
- эффективно вычисляться, желательно, модульным способом.

Поиск эффективных и универсальных ПХ важен для теоретических основ модулярных вычислительных структур и вычислительных средств на их основе.

В основе алгоритмов выполнения немодульных операций лежат методы вычислений ПХ, сложность которых непосредственно влияет на скорость выполнения немодульных операций в модулярной арифметике. Поиск некоторого компромисса в удовлетворении требований, предъявляемых к ПХ, привел к введению характеристик: ранг, след, нормированный ранг, исходный ранг, ядро числа, модульное ядро, нормированное ядро, квазислед [3-6].

Анализ рассмотренных ПХ показал, что значения модулярной величины по ним не всегда определяются однозначно. Множество введенных ПХ обуславливают необходимость выявления связей между ними, нахождении способов и условий их активного использования, конструктирования алгоритмов выполнения немодульных операций на их основе и проведения сравнительной оценки алгоритмической и временной сложности их вычисления.

Сложность вычисления ПХ связана с разрядностью модулярного представления, поэтому, если система многомодульная, то, желательно, найти одну такую ПХ, которая удовлетворяла бы требованиям скорости вычисления, универсальности и удобства использования в машинной арифметике. Наиболее полно этим требованиям отвечает позиционная характеристика, связанная с коэффициентами обобщенной позиционной системы счисления [6-7], которая позволяет организовать эффективное выполнение немодульных операций.

Позиционные характеристики на основе различных форм китайской теоремы остатков и на основе системы со смешанными основаниями разработаны в [1-7]. Область применения первых лежит в маломодульных системах остаточных классов, а вторых – многомодульных.

Противоречие между вычислительной сложностью определения позиционных характеристик и их быстродействием разрешено путем совместного применения китайской теоремы остатков и смешанной системы счисления [1, 2]. При этом методе определения позиционных характеристик ортогональные базисы представляются в смешанной системе счисления, что позволило модифицировать алгоритм Х.Л. Гарнера. Этот подход позволяет вычислять коэффициенты обобщенной позиционной системы счисления за один цикл синхронизации, в то время, как метод Х.Л. Гарнера, эту операцию выполняет итеративно за 2(n-1) циклов синхронизации, где n – число оснований системы остаточных классов [1, 2, 7].

Теорема кодирования Н. Сабо [7] показывает, что нет лучших методов определения позиционных характеристик, при которых не нарушается их однозначность, чем переход чисел из СОК в ОПСС, поскольку величина числа в модулярном представлении существенным образом зависит от всех остатков числа.

Использование китайской теоремы остатков обеспечивает метод перевода чисел из остаточного представления в позиционное. Хотя этот метод в принципе прост, применение его данной форме не эффективно, так как, желательно, чтобы модулярный нейрокомпьютер выполнял операции по модулю p_i, где i = 1, 2, …, n, а не по модулю $P = \prod_{i=1}^{n} p_i$ как этого требуется по китайской теореме остатков.

В противоположность метод преобразования со смешанными основаниями эффективно можно реализовать в модулярном нейрокомпьютере, так как в нем используются только операции по модулю p_i. По этой причине представление чисел со смешанными основаниями представляют особую важность в остаточных вычислениях по следующим причинам:

- система со смешанными основаниями – это позиционная система счисления и, следовательно, в ней легко выполняются операции позиционного типа;
- преобразование из остаточной системы счисления в систему со смешанными основаниями и наоборот в силу их связности осуществляются наиболее быстро в модулярных

нейрокомпьютерах.

Класс немодульных процедур модулярной компьютерной арифметики, являющихся базисными при разработке вычислительных остаточных алгоритмов, содержит следующие основные операции [1-7]:

1. Вычисление вычета (остатка) от числа, представленного в позиционной системе счисления, с получением результата по модулю p_i, где $i = 1, 2, \ldots, n$.
2. Определение знака модулярной величины – результата операции.
3. Сравнение модулярных величин по числовому значению между собой.
4. Определение переполнения за аддитивный или мультипликативный диапазон.
5. Операция деления.
6. Округление числа или результата вычислений.
7. Масштабирование числа или результата вычислений.
8. Расширение модулярной величины на дополнительные основания.
9. Определение ранга числа (r – ранг, это число раз, превышающий машинный диапазон).
10. Операции декодирования модулярного кода, вычисление синдрома, обнаружение, локализация и коррекция ошибок при кодировании помехозащищенным кодом.

Перечисленные операции являются важнейшими для машинной модулярной арифметики, и поэтому возникает необходимость разработки базовых операций и однотипных блоков для их реализации.

Базисные операции модулярной системы – это минимальное множество операций, на которых возможна реализация вычислительных остаточных или вычетных алгоритмов на имеющейся вычислительной базе, а набор операционных блоков, реализующих эти операции называется функциональным элементным базисом. Для эффективности набор операций и блоков может быть избыточным.

Для эффективной настройки модулярной арифметики на проблемную область необходимо выбрать оптимальный алгоритм базисной операции.

Рассмотрим проблему конструктирования алгоритмов выполнения базисных операций и их реализацию в нейросетевом базисе.

Операция вычисления вычета по произвольному модулю является основной базовой операцией в модулярных вычислительных алгоритмах и она дает наибольший вклад в алгоритмическую сложность, поэтому ее алгоритм и нейросетевая реализация должны быть оптимальной на выбранной вычислительной базе. В [1, 2] разработан метод определения вычета и его реализация на основе нейронной сети конечного кольца (НСКК).

Нейроны НСКК выполняют роль арифметических элементов, которые имеют характеристику оператора по модулю, а не типовых нелинейных функций активации, применяемых в обычных нейронных сетях. НСКК является базовой моделью при вычислении вычета и выполнения арифметических операций.

В работах [1, 2] исследованы вопросы ускоренного нахождения модульных и немодульных процедур на основе НСКК, которые выполняются за единицы циклов синхронизации. Благодаря этому, применение модулярной арифметики может дать значительные преимущества не только в таких приложениях, в которых основная доля вычислений приходится на точное умножение, возведение в степень больших чисел в сочетании со сложением и вычитанием, но и в которых довольно часто появляется необходимость в делении либо сравнении и определении знака числа, или при проверке, не “выходит” ли результат за пределы области допустимых значений.

Для повышения эффективности обработки данных в модулярных нейрокомпьютерах разработано операционное устройство в виде мультинейропроцессора [1, 2], состоящее из n-нейропроцессоров, обеспечивающих выполнение операций по модулю p_i $(i = 1, 2, ..., n)$. В качестве нейропроцессоров используются НСКК.

На основе модифицированного алгоритма Х.Л. Гарнера и НСКК разработаны эффективные нейросетевые вычислительные средства реализации основных немодульных базисных функций, которые

подтверждены авторскими свидетельствами и патентами на изобретение и реализованы на программируемых логических интегральных схемах фирмы Xilinx типа FPGA [1, 2, 9, 10, 11, 15, 16].

В [1, 2] показано развитие общей теории помехоустойчивого кодирования в избыточной системе остаточных классов, которое является теоретической базой для разработки и проектирования высоконадежных непозиционных нейропроцессоров и комплексов вычислительных систем.

На основе использования разработанных методов и алгоритмов коррекции ошибок предложено для коррекции ошибок в избыточной системе вычетов использовать сети Хопфилда и Хэмминга [10]. Для повышения эффективности коррекции ошибок разработана архитектура адаптивной параллельно-конвейерной нейронной сети для обнаружения, локализации и исправления ошибок в модулярных нейрокомпьютерных системах, что открывает новые перспективы для разработки отказоустойчивых модулярных вычислительных систем [12].

В [13] исследована структура высокопроизводительного модулярного нейрокомпьютера, функциональные возможности которого определяются нейросетевыми алгоритмами арифметики системы остаточных классов и доказано, что применение модулярного кодирования в нейронных сетях позволяет разрабатывать информационные системы с естественным параллелизмом, которые являются идеальной основой для проектирования нового класса вычислительных структур, позволяющих ставить и решать новые задачи, недоступные для позиционных кодов, такие как, например, построение “живучих”, близких к биологическим, систем. При этом в процессе функционирования информационные системы способны сохранять работоспособность за счет снижения в допустимых пределах каких-либо показателей качества при возникновении отказов и сбоев, а также перераспределять исходные данные между сохранившимися вычислительными ресурсами при деградации системы.

Для решения этой проблемы разработана архитектура нейронной сети оптимизации плана перераспределения нейропроцессоров мультинейропроцессорных систем в реальном масштабе времени.

Предложенная идея совмещения организации системы остаточных классов с организацией и функционированием нейронных сетей находит все большее применение [14]. Такой подход обеспечивает реорганизацию самоорганизуемых вычислительных средств в динамике вычислительного процесса, благодаря замечательной особенности системы остаточных классов – наличие обменных операций между надежностью, точностью и быстродействием сети.

Новым и интересным направление является организация самоустранения ошибок и сбоев вычислительных нейроподобных средств, функционирующих в системе остаточных классов, которые найдут широкое применение в специализированных системах.

Показано, что известные традиционные структуры параллельных информационных систем не приспособлены для решения задач с не распараллеливаемыми алгоритмами, поэтому в полной мере не могут быть использованы их преимущества по быстродействию. Весьма перспективным и плодотворным в практическом плане является распараллеливание на уровне машинных операций при использовании модулярного кодирования.

Широкий диапазон исследований, охвативших вопросы построения модулярных нейрокомпьютеров, привлек внимание многих исследователей к научным разработкам в этом направлении.

Теоретический уровень полученных научных результатов сопоставим с мировым, а по ряду позиций опережает аналогичные отечественные и зарубежные разработки в данной области, например, решении основной проблемы модулярного кодирования, высоко эффективном вычислении позиционных характеристик, разработке модулярных нейронных сетей функционального базиса для выполнения модульных и немодульных операций и другие.

Объединение модулярного принципа представления и обработки данных и нейронных структур в единую вычислительную архитектуру зачастую приводит к свойствам, которых нет у них по отдельности.

В заключении отметим, что отказоустойчивые нейрокомпьютеры, функционирующие в системе остаточных классов, найдут широкое применение в специализированных системах, а также в различного рода расширителях, спецпроцессорах и мультипроцессорных

системах со специальной архитектурой. Разработка нового класса модулярных нейрокомпьютеров связана с успехами развития наиболее перспективного направления элементной базы – программируемых логических интегральных схем.

Литература

1. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А. – М.: Физматлит, 2003. – 288 с.
2. Нейрокомпьютеры в остаточных классах / Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н. Учебное пособие для вузов– М.: Радиотехника, 2003. – 272 с.
3. Амербаев В.М. Теоретические основы машинной арифметики. – Алма-Ата: Наука, 1976. – 320 с.
4. Инютин С.А. Теория и методы моделирования вычислительных структур с параллелизмом машинных операций. – М.: Докторская диссертация, 2002. – 264 с.
5. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. – М.: Советское радио, 1968. – 440 с.
6. Торгашев В.А. Система остаточных классов и надежность ЦВМ. – М.: Сов. радио, 1973. – 120 с.
7. Кнут Д. Искусство программирования. Т. 2. Получисленные алгоритмы. – М.: Вильянс, 2001. – 832 с.
8. Галушкин А.И. Теория нейронных сетей. Кн. 1: Учеб. пособие для вузов / Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2000. – 416 с.
9. Червяков Н.И. Нейронная сеть для расширения кортежа числовой системы вычетов. Решение о выдаче патента на изобретение по заявке № 2003124641/09, 27.01.05.
10. Червяков Н.И., Шапошников А.В. Нейронная сеть для коррекции ошибок в модулярных нейрокомпьютерах. Решение о выдаче патента на изобретение по заявке № 2003127818/09, 21.01.05.
11. Червяков Н.И., Горденко Д.В. Нейронная сеть для округления и масштабирования чисел, представленных в системе остаточных классов. Решение о выдаче патента на изобретение по заявке № 2003115586/09, 24.11.04.
12. Архитектура адаптивной параллельно-конвейерной нейронной сети для коррекции ошибок в модулярных нейрокомпьютерных системах / Червяков Н.И., Галкина В.А, Стрекалов Ю.А., Лавриненко С.В. // Нейрокомпьютеры: разработка, применение. – 2003. – № 6. – С. 47-60.
13. Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н. Структура нового специализированного процессора // Нейрокомпьютеры: разработка, применение. – 2003. – № 6. – С. 3-22.

14. Червяков Н.И., Сахнюк П.А., Шапошников А.В., Галкина В.А. Нейронный цифровой фильтр с модулярной обработкой данных // Нейрокомпьютеры: разработка, применение. – 2002. – № 11. – С. 20-28.
15. Червяков Н.И., Сивоплясов Д.В. Нейронная сеть для преобразования полиадического кода в код системы остаточных классов. Решение о выдаче патента на изобретение по заявке № 2003115683/09, 10.03.05.
16. Червяков Н.И., Малофей А.О., Рыбальченко М.С., Щелкунова Ю.О. Нейронная сеть для вычисления позиционных характеристик непозиционного кода. Решение о выдаче патента на изобретение по заявке № 2003119016/09, 17.02.05.

УДК. 681.3 В.М. Амербаев, А.Л. Стемпковский, Г.Э. Широ

Модулярный быстродействующий согласованный фильтр

(Институт проблем проектирования в микроэлектронике РАН)

Излагаются принципы построения, приводятся архитектура и оценки сложности аппаратной реализации цифрового согласованного фильтра, работающего в модулярной арифметике. Показано, что при аппаратных затратах, не превышающих таковые в фильтрах, построенных в обычной позиционной арифметике, предлагаемый фильтр обладает более высоким быстродействием - в два и более раза. Приводятся примеры.

General description, architecture, and estimations of apparatus complexity of the digital FIR Filter based on modular arithmetic are described. It is shown that the proposed filter is two - three times faster compare to one having same apparatus expenses but based on traditional position arithmetic. Examples are presented.

Постановка задачи

В радио- и звуколокации (radar, sonar) широкое применение находят согласованные фильтры, как наиболее эффективное средство распознавания и временного сжатия сложных фазо- и частотно-модулированных сигналов. Рисунок 1 иллюстрирует в качестве примера результат обработки линейно-частотно-модулированного (ЛЧМ) сигнала согласованным фильтром.

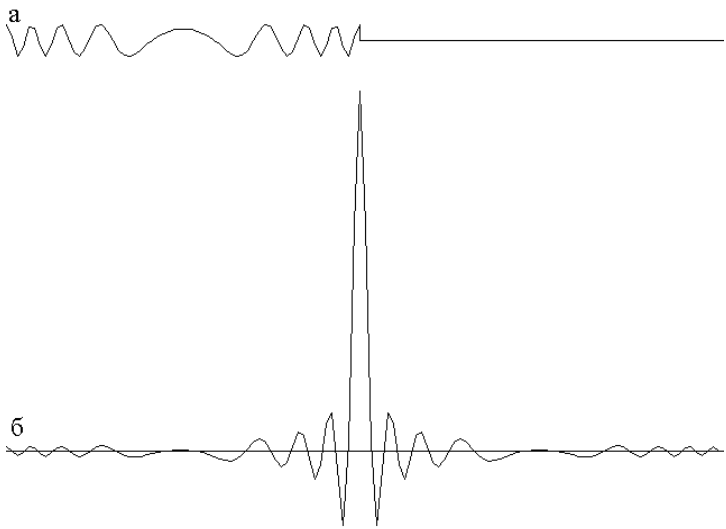


Рис. 1. Обработка ЛЧМ-сигнала цифровым согласованным фильтром:

а – исходный сигнал; б – результат обработки.

В соответствии с классической теорией цифровых линейных систем, согласованный фильтр реализует формулу свертки:

$$\dot{sv}(n+z) = \sum_{k=1}^{K} \dot{s}(n-k) * \dot{h}(k) \qquad (1)$$

Здесь $\dot{s}$, $\dot{h}$ комплексные представления оцифрованных последовательностей сигнала и импульсной характеристики фильтра, n, k – соответственно номера отсчетов сигнала и импульсной характеристики, К – длина импульсной характеристики фильтра, $\dot{sv}$ – комплексное представление результата свертки, z – конвейерная задержка фильтра.

Для обеспечения условия согласования импульсной характеристики последняя выбирается зеркальной и комплексно-сопряженной относительно фазовой структуры сигнала: $\dot{h}(k) = \bar{\dot{s}}(-n)$

Аппаратная реализация цифрового согласованного фильтра

Непосредственная аппаратная реализация формулы (1),

соответствующей цифровой согласованной фильтрации выполняется схемой рис. 2.

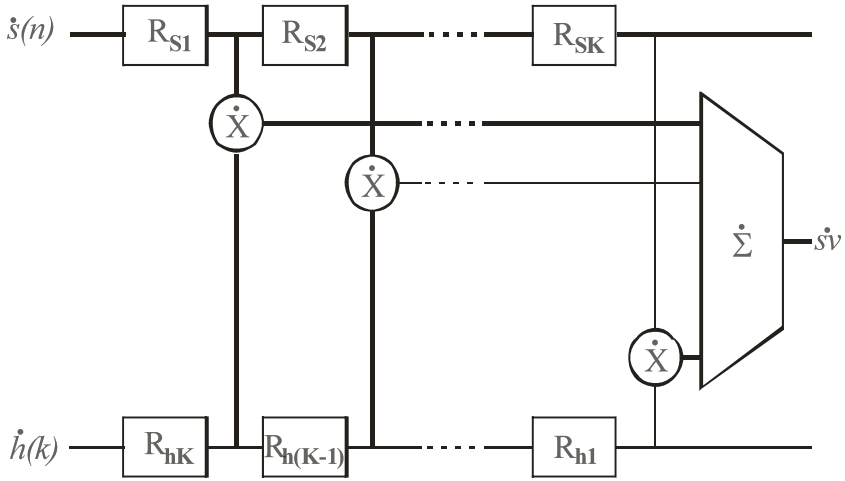


Рис. 2. Аппаратная реализация цифрового согласованного фильтра

Схема состоит из двух сдвиговых регистров Rs и Rh, длиной K звеньев каждый, предназначенных для загрузки соответственно комплексных отсчетов сигнала и опорной функции. Заметим, что нумерация звеньев в регистрах взаимно-обратная – в соответствии с требованиями зеркальной импульсной характеристики согласованного фильтра и в соответствии со знаком индекса k в формуле (1) при нумерации отсчетов сигнала $\dot{s}(n-k)$ импульсной характеристикой фильтра $\dot{h}(k)$.

Отсчеты сигнала и импульсной характеристики, хранящиеся в регистрах Rs и Rh, попарно перемножаются на комплексных умножителях, полученное произведение суммируется на комплексном пирамидальном сумматоре. Все упомянутые арифметические операции производятся в каждом (i-ом) такте работы схемы, работающей с периодом T, равным периоду дискретизации сигнала.

Оценка аппаратной сложности цифрового согласованного фильтра, работающего в позиционной системе счисления

Основная вычислительная сложность схемы (рис. 2) приходится на

комплексные умножители, число которых определяется длиной К фильтра, а разрядность – разрядностью операндов $r(\dot{s})$, $r(\dot{h})$. В качестве примера, имеющего практическое значение для реализации радиолокационных систем, здесь и далее будем рассматривать фильтр со следующими параметрами:

- Длина К = 512 звеньев,
- Разрядность $r(\dot{s}) = r(\dot{h})$ = (12 + 12) разрядов, т. е. разрядность представления реальной (синфазной) и мнимой (квадратурной) составляющих равны по 12 разрядов.
- Период дискретизации Т = 10 нс.

Для простоты изложения числа $\dot{s}$ и $\dot{h}$ будем далее считать реальными: s, h.

Оценка аппаратной сложности $V^{П}$ схемы (рис. 2), построенной в позиционной системе счисления, может быть выражена формулой:

$$V^{П} = O_V \; K \; r^2, \qquad (2).$$

где O_V – оценочный коэффициент, зависящий от применяемой элементной базы и, как будет показано ниже – от метода вычислений.

Применительно к обычно применяемой позиционной системе представления операндов, r-разрядный матричный умножитель должен содержать r^2 полных трехвходовых сумматоров [1], каждый из которых может быть построен на 12 парах К-МОП транзисторов.

Таким образом оценку $V^{П}$ в парах К-МОП транзисторов получаем равной $V^{П} = 12 * K * r^2$. Для рассматриваемого примера $V = 12 * 512 * 12^2 \cong 900.000$ пар транзисторов.

Формула свертки (1) может быть вычислена с помощью схемы, обладающей меньшей сложностью, чем по оценке (2), при переходе в частотную область и использовании процедур Быстрого Преобразования Фурье (БПФ). В ряде практических применений так и делают. Однако неизбежные при БПФ циклы вычислений, число которых по крайней мере 2 $\log_2$ К, приводит к соответствующей конвейерной задержке длиной:

$$\Delta t = 2K \log_2 K \text{ тактов},$$

которая часто неприемлема.

Далее будем рассматривать вопрос аппаратной реализации только схемы (рис. 2), работающей во временной области, основное достоинство которой – возможность получения результата фильтрации с минимально возможной конвейерной задержкой, равной К тактов, т. е. равной длине импульсной характеристики фильтра.

Оценка быстродействия

Как было показано, основная аппаратная сложность реализации фильтра в виде вычислителя формулы свертки (1) – (рис. 2) определяется сложностью построения r x r – разрядных умножителей. Именно эти умножители определяют и быстродействие фильтра – минимальное значение T^P – периода дискретизации сигнала $\dot{s}(n)$, обрабатываемого фильтром в реальном масштабе времени.

Известно [1] , что время Т срабатывания матричного r x r – разрядного умножителя, построенного с использованием «дерева Уоллеса» и сумматора с предвычислением переносов имеет «оптимистичную» оценку быстродействия:

$$\dot{O}^P = O * 2\log_2(2r) * t_1,$$

где t_1 – время предвычисления одного каскада переноса.

Для рассматриваемого примера разрядности r представления операндов r, равного 12 получим:

$$T^P = O * 2 * \log_2 (2 * 12) * t_1 = O * 10 * t_1.$$

По-видимому, полученная оценка является теоретическим пределом повышения быстродействия умножителя, работающего в классической позиционной арифметике. Дальнейшее повышение быстродействия может быть получено за счет использования «модулярной» арифметики.

Принципы модулярной арифметики. Преимущества, недостатки

Воспользуемся следующими основными принципами

представления чисел в «модулярной» арифметике и арифметических действий над ними [1].

Принцип 1: Любое целое число a, находящееся в диапазоне 0, …, N, может быть восстановлено по множеству вычетов – остатков от деления этого числа на множество модулей $\{ m_1, m_2, \ldots, m_n \}$.

$$a \Leftrightarrow \{ \alpha_1, \alpha_2, \ldots, \alpha_n \} = \{ | a |_{m_1}, | a |_{m_2}, \ldots, | a |_{m_n} \}$$

при этом модули m суть попарно взаимно-простые числа и их произведение $M = m_1 m_2 \ldots m_n \leq N$ – перекрывает диапазон представления числа a.

Пример 1. Число a = 478, находящееся в диапазоне 0 … 1309, может быть представлено в модулярном виде:

$$478 = \{ | 478 |_7, | 478 |_{11}, | 478 |_{17} \} = \{ 2, 5, 2 \}.$$

Здесь в качестве модулей выбраны попарно взаимно-простые числа $\{m_1,m_2,m_3\} = \{ 7, 11, 17 \}$, произведение которых M=7 * 11 * 17 = 1309.

Для восстановления числа a по его модулярному представлению воспользуемся алгоритмом, основанном на «Китайской теореме об остатках» [1].

Вычислим для выбранного набора модулей набор структурных чисел $\{k_1,k_2,k_3\}$:

$$k_1 = \left| \frac{M}{m_1} \right|_{m_1} = \left| \frac{1309}{7} \right|_7 = 5;$$

$$k_2 = \left| \frac{M}{m_2} \right|_{m_2} = \left| \frac{1309}{11} \right|_{11} = 9;$$

$$k_3 = \left| \frac{M}{m_3} \right|_{m_3} = \left| \frac{1309}{17} \right|_{17} = 9$$

Для полученных структурных чисел вычислим обратные им:

$\{ | k_1^{-1} |_{m_1}, | k_2^{-1} |_{m_2}, | k_3^{-1} |_{m_3} \}$:

$$| k_1 \; k_1^{-1} |_{m_1} = 1 \quad \Rightarrow | 5 * 3 |_7 = 1, \text{ следовательно } k_1^{-1} = 3.$$

$| k_2 \ k_2^{-1} |_{m_2} = 1 \quad \Rightarrow | 9 * 5 |_{11} = 1$, следовательно $k_2^{-1} = 5$.

$| k_3 \ k_3^{-1} |_{m_3} = 1 \quad \Rightarrow | 9 * 2 |_{17} = 1$, следовательно $k_3^{-1} = 2$.

Теперь число a может быть восстановлено по формуле:

$$a = \left| \alpha_1 k_1^{-1} \frac{M}{m_1} + \alpha_2 k_2^{-1} \frac{M}{m_2} + \alpha_3 k_3^{-1} \frac{M}{m_3} \right|_M ;$$

$$a = \left| 2 \cdot 3 \frac{1309}{7} + 5 \cdot 5 \frac{1309}{11} + 2 \cdot 2 \frac{1309}{17} \right|_{1309} = 478$$

Пример 2. Число b = – 478 , находящееся в диапазоне $-\frac{1309}{2} ... + \frac{1309}{2}$, при сохранении предыдущего набора модулей {7, 11, 17} может быть представлено в модулярном виде следующим образом.

Для отрицательного числа вычислим его дополнение до M:

$$\bar{b} = M - 478 = 1309 - 478 = 831$$

Теперь представим $\bar{b}$ в модулярном виде :

$$b \Rightarrow \bar{b} \Leftrightarrow \{ | 813 |_7; \ | 831 |_{11}; \ | 831 |_{17} \} = \{ 5, 6, 15 \}.$$

Принцип 2: Сумма (произведение) по модулю m двух чисел a, b равна сумме (произведению) по этому модулю вычетов этих чисел:

$$| a + b |_m = | a |_m (+_m) | b |_m$$

$$| a * b |_m = | a |_m (* _m) | b |_m$$

Здесь $(+_m)$, $(* _m)$ – операции сложения и умножения по модулю m, алгоритмы которых аналогичны алгоритмам представления чисел по модулю m:

$$a (+_m) b = | a + b |_m \qquad a (* _m) b = | a * b |_m$$

Пример 3. $| 46 + 19 |_7 = | 46 |_7 (+_7) | 19 |_7$

$2 = | 4 + 5 |_7$

$2 = 2.$

Пример 4. $| 46 * 19 |_7 = | 46 |_7 (*_7) | 19 |_7$

$$6 = |4 * 5|_7$$
$$6 = 6.$$

Рассмотренные свойства модулярной арифметики позволяют сделать важный для практики вывод: операции сложения и умножения (положительных и отрицательных чисел) могут выполняться не с исходными числами, а с вычетами, для каждого модуля отдельно.

Поскольку величины модулей, а, следовательно, и вычетов, существенно меньше, чем исходные числа, операции сложения и умножения могут выполняться на малоразрядных быстродействующих сумматорах и умножителях.

Данное преимущество приходится «оплачивать» процедурами преобразования операндов из позиционного представления в модулярное и обратно. Применительно к рассматриваемой задаче построения цифрового согласованного фильтра эта «плата» оказывается приемлемой: прямое и обратное преобразования здесь достаточно выполнять на входе и на выходе длинного вычислительного конвейера. При этом основной объем вычислений выполняется в модулярной арифметике.

Умножение в системе «дискретных логарифмов»

При построении умножителей, работающих в модулярной арифметике, можно воспользоваться индексным или «дискретно-логарифмическим» представлением операндов и заменить операцию модулярного умножения операцией модулярного сложения индексов или дискретных логарифмов [1].

Дискретно-логарифмическое представление модулярного числа $\log_{\omega}^{D}|a|_m$ основывается на понятии первообразного «корня по простому модулю m». Таким корнем ω(m) является целое число, возведение которого в степень 1, 2, … , (m-1) дает неповторяющиеся вычеты по модулю m.

Например, для m=7 первообразным корнем является ω(7) = 3.

Действительно:

$$|3^1|_7 = |3|_7 = 3.$$
$$|3^2|_7 = |9|_7 = 3.$$
$$|3^3|_7 = |27|_7 = 3.$$

$| 3^4 |_7 = | 81 |_7 = 3.$
$| 3^5 |_7 = | 243 |_7 = 3.$
$| 3^6 |_7 = | 729 |_7 = 3.$

Таким образом, получаем таблицу дискретных логарифмов (табл.1).

Таблица 1.

Дискретные логарифмы для m=7.

$\| a \|_7$	$\log_3^D \|a\|_7$
0	$-\infty$
1	6
2	2
3	1
4	4
5	5
6	3

Строка $0 \Leftrightarrow -\infty$ здесь введена искусственно, однако обработка нулевых вычетов и, соответственно, значений $\log^D = (-\infty)$, реализуема. Смотри пример 6.

Если столбцы в таблице 1 поменять местами и рассортировать строки по первому столбцу, получим таблицу антилогарифмов (табл. 2).

Таблица 2.

Дискретные антилогарифмы для m = 7.

$\log_3^D \|a\|_7$	$\overline{\log_3^D} \|a\|_7 = \|a\|_7$
$-\infty$	0
1	3
2	2
3	6
4	4
5	5
6	1

Теперь процедуру умножения вычетов можно представить следующим образом:

$$|a|_m (*_m) |b|_m = \overline{\log}_\omega^D \left[\log_\omega^D (a) (*_{m-1}) \log_\omega^D (b) \right]$$

Рассмотренный выше пример 4 можно решить следующим образом.

Пример 5.

$$6 = 4 (*_7) 5 = \overline{\log}_3^D \left[\log_3^D (4) (*_6) \log_3^D (5) \right] = \overline{\log}_3^D \left[4 (+_6) 5 \right] = \overline{\log}_3^D [3] = 6.$$

При вычислениях мы воспользовались таблицами 1, 2 дискретных логарифмов и антилогарифмов.

Следующий пример иллюстрирует применимость процедуры логарифмирования к нулевым операндам.

Пример 6.

$$| 46 * 0 |_7 = 0$$

$$0 = 4 (*7) 0 = \overline{\log}_3^D \left[\log_3^D (4) (*_6) \log_3^D (0) \right] = \overline{\log}_3^D \left[4 (+_6) - \infty \right] = 0.$$

При этом применимо следующее правило суммирования дискретных логарифмов при условии, что один из них равен $-\infty$:

$$x + (-\infty) = -\infty$$

$$(-\infty) + x = -\infty$$

$$(-\infty) + (-\infty) = -\infty$$

Таким образом, операция умножения выполнена с помощью более экономной операции модулярного сложения, причем без увеличения разрядности операндов. «Платой» за полученную экономию являются операции логарифмирования и антилогарифмирования, выполнение которых требует соответствующих таблиц.

Пример 7. Решение задачи свертки в модулярной дискретно-логарифмированной арифметике.

Пусть в цифровом согласованном фильтре с длиной импульсной характеристики К=4 в некоторый n-й момент времени, т. е. в некотором n-ом такте, операнды приняли значения, соответствующие рис. 3. К окончанию этого такта на выходе фильтра должно быть получено число 71 – результат свертки в данном такте.

Пусть результаты свертки на любом такте не выходят за пределы $-\frac{1309}{2}...+\frac{1309}{2}$.

В качестве множества модулей выберем такие, произведение М которых не менее 1309:

$$1309 \le M = \{m_1 \cdot m_2 \cdot m_3\} = \{7,11,17\}$$

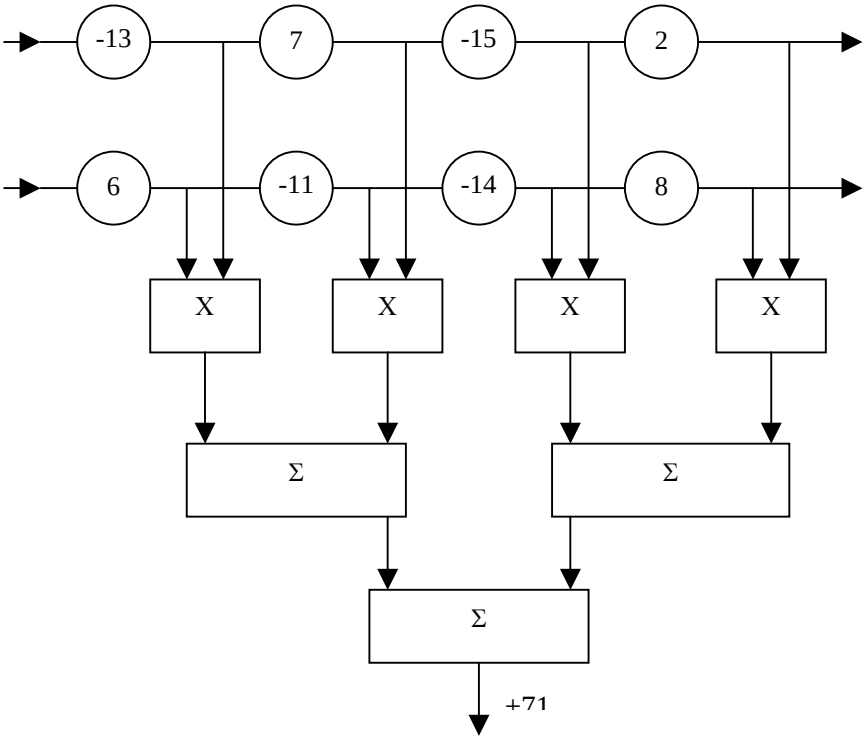


Рис. 3. Пример вычисления свертки
для произвольного (n-го) такта работы фильтра

Представим исходные операнды в модулярном виде, а затем в дискретно-логарифмированном виде (таблица 3). При этом отрицательные числа взяты в виде дополнений до М, например:

$$-13 \Rightarrow 1309 - 13 = 1296$$

$$|-13|_7 = |1296|_7 = 1$$

Попарное умножение, или попарное сложение дискретных

логарифмов и затем вычисление антилогарифмов дают результаты, представленные в виде таблицы 8.

Таблица 3

Преобразование операндов для примера 7:
«позиционная система» → «модулярное представление» → «дискретно-логарифмическое представление»

Опе= ранды	$\lvert x \rvert_{17}$	$\lvert x \rvert_{17}$	$\lvert x \rvert_{17}$	$\log_3^D \lvert x \rvert_7$	$\log_2^D \lvert x \rvert_{11}$	$\log_3^D \lvert x \rvert_{17}$
–13	1	9	4	6	6	12
6	6	6	6	3	9	15
7	0	7	7	$-\infty$	7	11
–11	3	0	6	1	$-\infty$	15
–15	6	7	2	3	7	14
–14	0	8	3	$-\infty$	3	1
2	2	2	2	2	1	14
8	1	8	8	6	3	10

Кроме таблиц 1, 2 логарифмов и антилогарифмов для m=7 потребуются аналогичные таблицы 4, 5, 6, 7 для m = 11, m = 17. Первообразные корни здесь ω(11)=2, ω(17)=3.

Таблица 4
Дискретные логарифмы для m = 11

$\lvert a \rvert_{11}$	$\log_2^D \lvert a \rvert_{11}$
0	$-\infty$
1	10
2	1
3	8
4	2
5	4
6	9

Таблица 5
Дискретные антилогарифмы для m=11

$\log_2^D \lvert a \rvert_{11}$	$\overline{\log_2^D} \lvert a \rvert_{11} = \lvert a \rvert_{11}$
$-\infty$	0
1	2
2	4
3	8
4	5
5	10
6	9

7	7
8	3
9	6
10	5

7	7
8	3
9	6
10	1

Таблица 6
Дискретные логарифмы
для m = 17

$\lvert a \rvert_{17}$	$\log_3^D \lvert a \rvert_{17}$
0	$-\infty$
1	16
2	14
3	1
4	12
5	5
6	15
7	11
8	10
9	2
10	3
11	7
12	13
13	4
14	9
15	6
16	8

Таблица 7
Дискретные антилогарифмы для
m=17

$\log_3^D \lvert a \rvert_{17}$	$\overline{\log}_3^D \lvert a \rvert_{17} = \lvert a \rvert_{17}$
$-\infty$	0
1	3
2	9
3	10
4	13
5	5
6	15
7	11
8	16
9	14
10	8
11	7
12	4
13	12
14	2
15	6
16	1

Таблица 8.

Результаты попарных перемножений операндов для примера 5
в дискретно-логарифмическом виде

$a \cdot b$	$\log_3^D \left[\lvert a \rvert_7 (*_7) \lvert b \rvert_7 \right] =$ $\log_3^D \lvert a \rvert_7 (+_6) \log_3^D \lvert b \rvert_7$	$\log_2^D \left[\lvert a \rvert_{11} (*_{11}) \lvert b \rvert_{11} \right] =$ $\log_2^D \lvert a \rvert_{11} (+_{10}) \log_2^D \lvert b \rvert_{11}$	$\log_3^D \left[\lvert a \rvert_{17} (*_{17}) \lvert b \rvert_{17} \right] =$ $\log_3^D \lvert a \rvert_{17} (+_{16}) \log_3^D \lvert b \rvert_{17}$
(–13)*6	$6(+_6)3 = 3$	$6(+_{10})9 = 3$	$12(+_{16})15 = 11$

7*(–11)	(–∞) ($+_6$)3 = (–∞)	7($+_{10}$)(–∞) = (–∞)	11($+_{16}$)15 = 10
(-15)*(-14)	3($+_6$) (–∞) = (–∞)	7($+_{10}$)3 = 10	14($+_{16}$)1 = 15
2*8	2($+_6$)6 = 2	1($+_{10}$)3 = 4	14($+_{16}$)10 = 8

Воспользовавшись таблицами 2, 5, 7 антилогарифмов и просуммировав результаты попарного перемножения в модулярном виде, получим результат свертки в модулярном виде (табл. 9).

Таблица 9.

Окончательный результат для примера 5

$a \cdot b$	$\lvert a\rvert_7 (*_7) \lvert b\rvert_7$	$\lvert a\rvert_{11} (*_{11}) \lvert b\rvert_{11}$	$\lvert a\rvert_{17} (*_{17}) \lvert b\rvert_{17}$
(–13)*6	6	10	7
7*(–11)	0	0	8
(–15)*(–14)	0	1	6
2*8	2	5	16
Σ = 71	$\lvert \Sigma_7 \rvert = 1$	$\lvert \Sigma_{11} \rvert = 5$	$\lvert \Sigma_{17} \rvert = 3$

Модулярное представление числа 71:

$$71 \Rightarrow \{ |71|_7, |71|_{11}, |71|_{17} \} = \{ 1, 5, 3 \}.$$

Функциональная схема фильтра, работающего в «модулярной арифметике».

Модулярный принцип представления данных позволяет распараллелить вычислительный процесс и, соответственно, схему фильтра на m параллельных функционально-идентичных каналов –

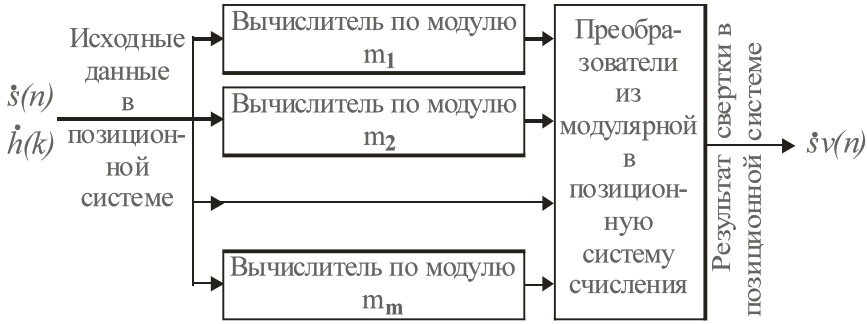


рис. 4.

Рис. 4. Функциональная схема фильтра

Рассмотрим функциональную схему одного канала модулярного вычислителя, обрабатывающего данные по одному (i-му) модулю m_i – рис. 5. Для простоты изложения отсчеты сигнала s(n) и импульсной характеристики h(k) принимаем не комплексными, а реальными.

Подобно схеме фильтра, работающего в позиционной системе (рис. 2), схема рис. 5 содержит два K-звенных сдвиговых регистра для хранения и сдвига отсчетов сигнала и опорной функции. Отличие здесь в представлении чисел s(n), h(k) – позиционная система с одной стороны, $\log^D|s|_{m_i}$, $\log^D|h|_{m_i}$ – модулярная дискретно-логарифмическая система с другой. Для получения чисел в модулярной дискретно-логарифмической форме на входе сдвиговых регистров стоят функциональные преобразователи $a \rightarrow \log^D|a|_{m_i}$.

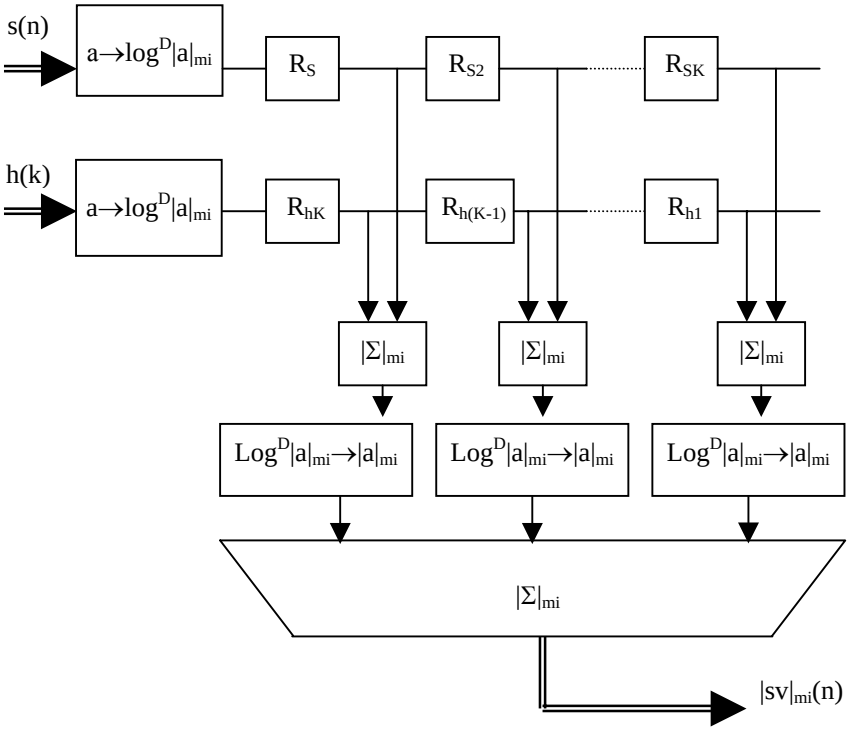


Рис. 5. Функциональная схема модулярного вычислителя

Роль умножителей схемы рис. 2 в схеме рис. 5 выполняют модулярные сумматоры $|\Sigma|_{m_i-1}$ – и это главное преимущество. Для выполнения последующего суммирования полученных результатов последние должны быть преобразованы из дискретно-логарифмической в модулярную форму $\log^D|a|_{m_i} \rightarrow |a|_{m_i}$. Эти антилогарифмические преобразователи – «плата» за отсутствие в схеме рис. 5 умножителей. Число преобразователей К равно числу звеньев фильтра. Как будет показано далее, именно они определяют в основном уровень интеграции микросхемы, реализующей функциональную схему рис. 5.

Пирамидальный сумматор $|\Sigma|_{m_i}$ строится из сумматоров по модулю m_i, аналогичных обыкновенным «позиционным» сумматорам. Выходной преобразователь модулярных кодов в позиционные (рис. 4) реализует алгоритм, основанный на «китайской теореме об остатках». Пример работы этого алгоритма был рассмотрен выше.

Оценка аппаратной сложности цифрового согласованного фильтра, работающего в «модулярной арифметике».

Оценку аппаратной сложности схемы рис. 4, 5 вычислим применительно к рассмотренному выше примеру 12-разрядного фильтра с длиной импульсной характеристики К=512. Набор модулей должен быть выбран таким, чтобы их произведение М перекрывало диапазон представления результата свертки sv:

$$M \geq 2^{12} * 2^{12} * 512 = 2^{33}$$

Данному условию удовлетворяет, например, следующий набор простых чисел:

$$2^{33} \leq M = m_1 * m_2 * \ldots * m_n = 3, 5, 7, 13, 17, 19, 23, 29, 31.$$

Для кодировки данного набора потребуется 37 разрядов:

$$\log_2 m_1 + \log_2 m_2 + \ldots + \log_2 m_n = 2 + 3 + 3 + 4 + 5 + 5 + 5 + 5 + 5 = 37.$$

Ограничимся, как ранее, оценкой (2) аппаратной сложности умножителей. В схеме рис. 5 основная сложность умножителей приходится на таблицу дискретных логарифмов. При реализации этих таблиц в виде ПЗУ суммарная емкость ПЗУ для одного умножителя будет равна:

$$V_{пзу} = (m_1 * \log_2 m_1 + m_2 * \log_2 m_2 + \ldots + m_n * \log_2 m_n) \text{ бит} =$$

$$= (3*2 + 5*3 + 7*3 + 13*4 + 17*5 + 19*5 + 23*5 + 29*5 + 31*5) \text{ бит} = 689 \text{ бит}.$$

Таблицы антилогарифмов обладают свойством симметрии, обусловленной коммутативностью умножения и знаковой симметрией абсолютно наименьших вычетов, с учетом которой:

$$V_{пзу} = 689 * 0.25 \approx 173 \text{ бит}.$$

При реализации ПЗУ на К-МОП транзисторах при условии кодировки одного бита одной парой К-МОП транзисторов получим $V_{пзу} = 173$ пар транзисторов.

Для рассматриваемой в примере длины импульсной характеристики К=512, получим оценку аппаратной сложности фильтра в целом:

$$V^M = 512 * V_{ПЗУ} = 512 * 173 = 88.000 \text{ пар транзисторов}.$$

Данная оценка значительно лучше полученной ранее для фильтра, работающего в позиционной арифметике: $V^П \approx 900.000$ пар транзисторов. Соотношение $\frac{V^П}{V^M} \approx 10$ в пользу «модулярной арифметики».

Преимущества «модулярного» варианта сохраняются и при уточнении оценок. В случае «позиционного» варианта оценку необходимо увеличить с учетом того, что в схеме должны быть применены не простые матричные умножители, а быстродействующие [1]. С другой стороны, для «модулярного» варианта необходимо учесть увеличение по сравнению с «позиционным» разрядности входных регистров (для рассматриваемого примера – в 3 раза).

Оценка быстродействия цифрового согласованного фильтра, работающего в «модулярной» арифметике

Быстродействие T^M схемы рис. 5, работающей в «модулярной» арифметике, определяется быстродействием сумматоров, работающих по максимальному модулю m_n. В рассматриваемом примере $m_n = 31$. Разрядность соответствующего сумматора равна 5. Следовательно,

$T^M = O * 5 * t_{пер}$, где $t_{пер}$ – время переноса сумматора.

Данная оценка лучше полученной ранее для фильтра, работающего в позиционной арифметике: $T^P = O * 10 * t_{пер}$. Соотношение $\frac{T^P}{T^M} = 2$ в пользу «модулярной» арифметики.

Преимущества «модулярного» варианта сохраняются и даже увеличиваются при уточнении оценки T^P. Для быстродействующего умножителя, применяемого в схеме рис. 2, время предвычисления переноса в 1,5 – 2 раза больше времени вычисления переноса в трехвходовых сумматорах, на которых строится схема рис. 5.

Выводы

1. Эффективным методом построения быстродействующих цифровых согласованных фильтров, работающих в реальном времени, является использование «модулярной арифметики», при которой оцифрованные исходные данные на входе фильтра преобразуются из позиционной системы счисления в модулярную систему, все промежуточные вычисления производятся в модулярной системе, затем на выходе фильтра результирующие данные преобразуются в позиционную систему.
2. Аппаратная сложность фильтра, рассмотренного в качестве примера, с длиной импульсной характеристики К = 512 и разрядностью операндов, равной 12, построенного по «модулярному» принципу, оказывается существенно меньшей, чем для аналогичного фильтра, работающего в обычной позиционной арифметике. При возрастании разрядности операндов преимущества «модулярного» принципа увеличиваются.
3. Главным преимуществом построения фильтров по модулярному принципу является повышение быстродействия, что для устройств, работающих в реальном времени, является решающим требованием. Максимальная величина задержки в таких фильтрах определяется временем срабатывания малоразрядных сумматоров (в примере – 5 разрядов). По сравнению с аналогичным фильтром, работающим в позиционной системе, достигается выигрыш в быстродействии в

два и более раза.

4. Так как на промежуточных этапах фильтрации по модулярному принципу не производится округление (как это имеет место в позиционном варианте), то результат фильтрации по модулярному принципу имеет максимально достижимую точность для данного типа фильтрации.

Литература

1. Т. Кормен, Ч. Лейзерсон, Р. Ривест. Алгоритмы. Построение и анализ. Перевод с английского А. Шеня. МЦНМО. М., 1999. – 960 с., 263 ил.

УДК 681.142

В.П. Ирхин

Табличная реализация операций модулярной арифметики

(Международный институт компьютерных технологий, г. Воронеж)

В статье проведен анализ алгоритмов проведения модульных операций в табличном исполнении. Получены граничные оценки аппаратурных затрат

Одним из важных достоинств модулярной арифметики является малоразрядность операндов и результата операции. Это обстоятельство позволяет применять табличные методы, при которых бинарные операции превращаются в однотактовые, осуществляемые простой выборкой из таблиц.

Недостатком данного подхода, препятствующему внедрению его в практику, является существенный рост аппаратурных затрат при увеличении разрядности операндов, ввиду того, что сами операции выполняются в однопозиционном коде. Рассмотрим ряд алгоритмов, позволяющих оптимизировать структуру таблиц путем преобразования исходных данных.

Применительно к операции модульного умножения $\left|A \cdot B\right|_m$ (A, B –

операнды; m модуль операции) известно [1], что ее таблица Кэли обладает симметрией диагоналей, а также вертикали и горизонтали, проходящих между величинами $[m/2]$ и $]m/2[$. Обозначив индекс операнда $\gamma_A(\gamma_B)$, получим

$$\gamma_A(\gamma_B)=\begin{cases}0, \text{åñëè } 0 \le A(\hat{A}) <]m/2[\\ 1, \text{åñëè }]m/2[\le A(B) < m\end{cases}.$$

Определим значения $A'(B')=m-A(B)$, если операнды принадлежат диапазону $[(m+1)/2, m-1]$. При $\gamma_A=\gamma_B$ $|\grave{A}\cdot\hat{A}|_m=|A'\cdot B'|m$, а в противном случае $(\gamma_A\neq\gamma_B)$ $|\grave{A}\cdot\hat{A}|_m=m-|A'\cdot B'|_m$ [1].

Аналогичным образом выведем подобные соотношения для операций модульного сложения и вычитания, сохраняя принятые обозначения. Полученные результаты сведены в таблицу.

$\gamma_{\grave{A}},\gamma_{\hat{A}}$	$\lvert\grave{A}+\hat{A}\rvert_m$	$\lvert\grave{A}-\hat{A}\rvert_m$
$\gamma_{\grave{A}}=\gamma_{\hat{A}}=0$	$\lvert\grave{A}+\hat{A}\rvert_m$	$\lvert\grave{A}-\hat{A}\rvert_m$
$\gamma_{\grave{A}}=0,\gamma_{\hat{A}}=1$	$\lvert\grave{A}-\hat{A}'\rvert_m$	$\lvert\grave{A}+\hat{A}'\rvert_m$
$\gamma_A=1,\gamma_B=0$	$m-\lvert\grave{A}'-\hat{A}\rvert_m$	$m-\lvert\grave{A}'+\hat{A}\rvert_m$
$\gamma_A=\gamma_B=1$	$m-\lvert\grave{A}'+\hat{A}'\rvert_m$	$m-\lvert\grave{A}'-\hat{A}'\rvert_m$

Следовательно, при реализации базового набора операций модульной арифметики можно использовать 1/8 часть полной таблицы, инвертируя по модулю полученный результат в зависимости от соотношений $\gamma_{\grave{A}}$ и $\gamma_{\hat{A}}$. Возможность достижения этого объясняется тем, что паре входных операндов $\grave{A}$ и $\hat{A}$ соответствует определенный элемент таблицы, который можно интерпретировать как результат произвольной модульной операции. Данное обстоятельство позволяет проводить одновременно несколько бинарных операций на одной таблице с использованием симметричного матричного вычислителя [2] и половинными значениями входных операндов.

Таким образом, реализация всех осей симметрии таблицы Кэли для

модульных операций предполагает построение таблицы с общим числом элементов (узлов) N равным $m^2/8$, что примерно на порядок уменьшает аппаратурные затраты.

Дальнейшее сокращение числа элементов таблицы возможно путем применения двухуровневых табличных методов.

Рассмотрим алгоритм сокращения таблицы, обеспечивающий ее деление на функционально законченные части с коэффициентом деления d. Приняв следующие обозначения: $\Delta =]m/d[$ и $\alpha(\beta) = |\grave{A}(\hat{A})|_\Delta$, имеем

$$|\grave{A} \pm \hat{A}|_m = |(\alpha \pm \beta) + (\gamma'_A \pm \gamma_B) \cdot \Delta|_m,$$

где $\gamma_{\grave{A}(B)} = \overline{0, d}$ в отличие от (1) является обобщенным индексом операнда. Для реализации соотношения (2) с последующим восстановлением результата операции по модулю используется схема, представленная на рис. 1.

$$N = \frac{1}{2}(m/d)^2 + \frac{1}{2}d^2 + (m/d) \cdot d\ .$$

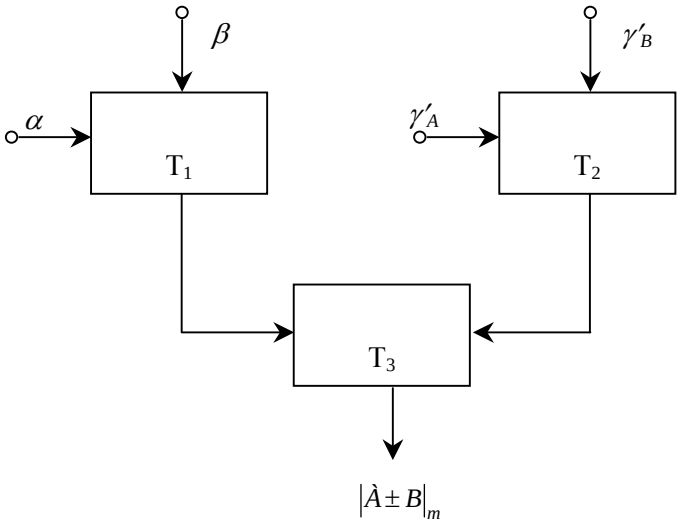


Рис. 1. Двухуровневая табличная структура

В этой схеме имеются три таблицы (T_1 и T_2 для получения промежуточных результатов) и T_3 – для заключительного, однако общее число узлов их меньше, чем в одноуровневой структуре. Умножение на величину Δ согласно (2) в однопозиционном коде выполняется простой перекоммутацией выходных шин таблицы T_2. Определим минимальное число всех табличных элементов

Тогда $\frac{\partial N}{\partial d} = -m^2 d^{-3} + d = 0$,

откуда $d_{опт} = \sqrt{m}$ и $N_{min} = 2m$.

Следовательно, при выборе $d = \sqrt{m}$ число табличных узлов обладает не квадратичной, а линейной зависимостью от величины модуля m, что позволяет использовать табличные методы выполнения операций модулярной арифметики с операндами $A(B)$ более значительными по величине.

Другой алгоритм, соответствующий реализации двухуровневой структуры (рис. 1), использует понятие внутреннего модуля устройства K. В этом случае

$$\left|A \pm \hat{A}\right|_m = \left|K(\alpha \pm \beta) + (\Delta_A \pm \Delta_B)\right|_m,$$

где $\alpha(\beta) = [A(B)/K]$, $\Delta_{A(B)} = \left|A(B)\right|_K$.

Число табличных узлов N при этом будет равно $N = \frac{1}{2}(m/K)^2 + \frac{1}{2}K^2 + (m/K)\cdot K$, откуда следует $K_{опт} = \sqrt{m}$ и $N_{min} = 2m$.

Отметим, что выражения (2) и (4) инвариантны, вследствие чего оптимальные параметры и величины N_{min} одинаковы. При использовании схемы на рис.1 отличие заключается в формировании операндов для таблиц T_1 и T_2, а также в восстановлении результата операции по модулю m.

Следующим этапом сокращения табличного оборудования является таблично-групповой метод [3]. Сущность его состоит в последовательном использовании группы таблиц при проведении модульной операции. Отличие его от известных в позиционных

системах счисления алгоритмов заключается в том, что величина разряда $t >> 2$ и разрядная операция выполняется табличным путем. Модульная операция производится таблично-групповым методом в t- ичной системе счисления. Соответствующая этому методу структура для числа разрядов, равных трем представлена на рис. 2.

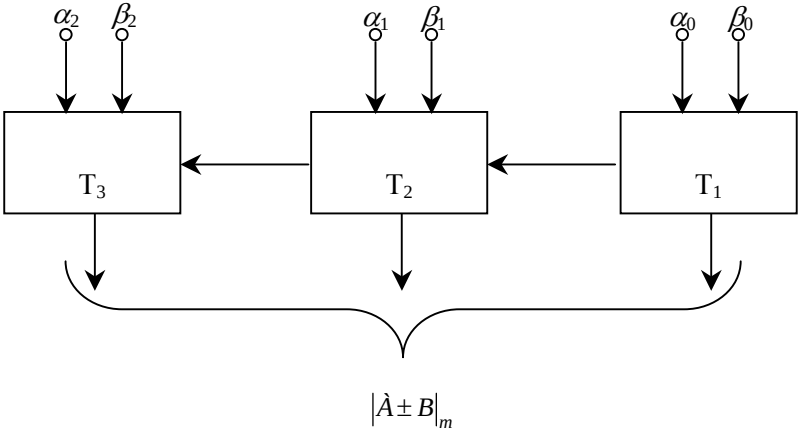


Рис. 2. Структурная схема реализации таблично-группового метода

При этом $A = (\alpha_2, \alpha_1, \alpha_0)$ и $\hat{A} = (\beta_2, \beta_1, \beta_0)$ – представление операндов в t-ичной системе счисления. Число табличных узлов N при исполнении трехразрядной схемы с учетом диагональной симметрии составляет

$$N = \frac{1}{2}\left(m/t^2\right)^2 + \frac{1}{2}t^2 + \frac{1}{2}t^2 .$$

Тогда $t_{\text{ПТ}} = \sqrt[3]{m}$, а $N_{\text{инф}} = \frac{3}{2}\sqrt{m^3}$. При величине t равной степени числа два результат модульной операции представляется в двоичной системе счисления. Следует отметить, что $m \le t^3 - 1$, а выбором соответствующего значения t можно обеспечить условие $t^3 - m < t$, при этом получение результата операции реализуется коррекцией только в нулевом разряде.

Ввиду того, что данный метод занимает промежуточное положение

между сумматорным и табличным вариантами, то при его использовании можно регулировать соотношение между аппаратными затратами и быстродействием проведения модульной операции.

Дальнейшее развитие этого метода, связанное с увеличением разрядности операндов $A(B)$ позволяет довести величину N до значения $N = \frac{n}{2} m^{\frac{2}{n}}$, где n – разрядность операндов $A(B)$, откуда получаем $n_{\text{îïò}} = 2\ln m$, $N_{\text{èëí}} = e\ln m$.

Полученная логарифмическая зависимость числа узлов таблицы от величины модуля определяет нижнюю границу, так как анализ проводился без учета восстановления результата операции по модулю m, когда t не равно степени числа 2. Расчеты, проведенные для предельных оценок N при многоуровневой табличной реализации, показывают, что в этом случае $N_{\text{èëí}} = 2 \cdot e \cdot \ln m$ [3].

Выбор конкретного метода сокращения табличного оборудования должен производиться исходя из требований к элементной базе операционного устройства модулярной арифметики. Предлагаемые алгоритмы могут быть использованы в вычислительных трактатах непозиционных спецпроцессов, основанных на других принципах выполнения модульных операций для повышения их быстродействия.

Полученные предельные оценки оборудования табличных вычислительных структур могут найти применение при анализе и сравнительной оценке перспективных алгоритмов для проведения операций модулярной арифметики.

Литература

1 Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. – М.: Сов. радио, 1968.

2 А.с. 1775721 СССР. Арифметическое устройство по модулю / В.П. Ирхин и др. – Опубл. в Б.И. №42, 1992.

3 Ирхин В.П. Проектирование непозиционных специализированных процессов. – Воронеж: Изд-во Воронежского госуниверситета, 1999.

УДК 681.04

Ю.Д.Полисский

Сравнение чисел в системе остаточных классов

(*НИИ автоматизации черной металлургии, г. Днепропетровск*)

В статье выполнен обзор основных работ по сравнению чисел, представленных в системе статочных классов.

In the article are realized the survey of principial works from number comparing represented in residue class system.

В настоящее время в связи с развитием современной техники, информационных и управляющих систем все большее применение находят новые принципы на основе представления данных в системе остаточных классов [1].

Системой остаточных классов (СОК) называется система счисления [2,3], в которой произвольное число А представляется в виде набора наименьших неотрицательных остатков по модулям m_1, m_2, …, m_n, т.е. $A=[A(\mathrm{mod}\ m_1), A(\mathrm{mod}\ m_2), \ldots, A(\mathrm{mod}\ m_n)]$ или

$$A=(\alpha_1,\alpha_2,\ldots,\alpha_n). \qquad (1)$$

Здесь $\alpha_i=A(\mathrm{mod}\ m_i)$. При этом, если числа m_i взаимно простые, то представление числа А в виде (1) является единственным, а объем диапазона [0,M) представимых чисел в этом случае равен

$$M=m_1m_2\ldots m_n. \qquad (2)$$

Для чисел диапазона [0,M), представленных в виде (1), арифметические операции сложения, вычитания и умножения выполняются с остатками α_i независимо друг от друга, причем по простым правилам.

К достоинствам такого представления чисел относится также малоразрядность остатков, высокая точность и надежность, способность системы к самокоррекции. Однако возникают серьезные трудности при реализации так называемых немодульных операций, для выполнения которых необходимо знание цифр операндов по всем разрядам [4]. Одной из таких немодульных операций является операция сравнения чисел.

Решение любой задачи управления всегда включает сравнение в каждый момент времени состояния управляемых объектов с заданным состоянием, соответствующим алгоритму функционирования системы. Это сравнение может быть достаточно простым, но зачастую требует решения весьма сложных многомерных задач. Целью сравнения в наиболее простом случае является обнаружение факта совпадения или несовпадения значений величин. В более сложных случаях операции сравнения включают оценку степени несовпадения состояний сравниваемых объектов, выделение объекта с доминирующим в некотором смысле состоянием и оценку степени доминирования и т.п. В дальнейшем будем применять термин «сравнение чисел», подразумевая под ним целенаправленный процесс обработки информации для получения указанных результатов.

Операция сравнения чисел [5] в одних случаях представляет собой сопоставление значений двух чисел A и B (попарное сравнение) и проверку наличия того или иного признака у результата, в других случаях при сравнении анализируется группа чисел (групповое сравнение), а результат представляет собой наибольшее или наименьшее их значение, признак положения чисел по отношению к граничным значениям либо к некоторому фиксированному значению или длину диапазона значений анализируемых чисел. Конкретное выполнение операции сравнения при решении прикладных задач может осуществляться программно либо специальными устройствами сравнения (или компараторами),

реализующими выбранные алгоритмы.

Устройство сравнения - устройство, предназначенное для попарного или группового сравнения данных с выработкой результата сравнения в виде сигналов и чисел, представляющих собой качественные и количественные оценки значений сравниваемых данных.

КЛАССИФИКАЦИЯ СРАВНЕНИЯ ЧИСЕЛ

Решение задач сравнения чисел, в результате которого достигается та или иная цель, осуществляется применением широкого класса различных операций по обработке сравниваемых чисел. При описании процесса сравнения указание цели и применяемых операций дает необходимое и достаточное представление о сравнении, поэтому целесообразно оба эти момента отразить в его определении.

Будем под сравнением чисел понимать совокупность операций над сравниваемыми числами, имеющих целью установить количественные и качественные оценки соотношения сравниваемых чисел. Такими оценками являются признаки равенства или неравенства сравниваемых чисел, значения большего, меньшего из них, наибольшей, наименьшей разности, признак выхода какого-либо из сравниваемых чисел за установленные границы или значение отклонения исследуемых чисел от одного или нескольких фиксированных чисел. Наконец, может предусматриваться совместное получение указанных результатов.

В основу классификации сравнения чисел положены признаки, позволяющие выделить существенные для анализа и синтеза алгоритмов и устройств сравнения характеристики процесса сравнения. Эти признаки включают указание вида сравнения и достигаемой цели, используемых алгоритмов и этапности процесса сравнения.

В зависимости от вида сравнения алгоритмы могут быть разделены на две группы: алгоритмы попарного и группового сравнения данных. Это соответствует двум отмеченным ранее типам решаемых задач: сравнения состояния объекта с заданным состоянием и выделения объекта с доминирующим в некотором смысле состоянием. Целесообразность выделения алгоритма

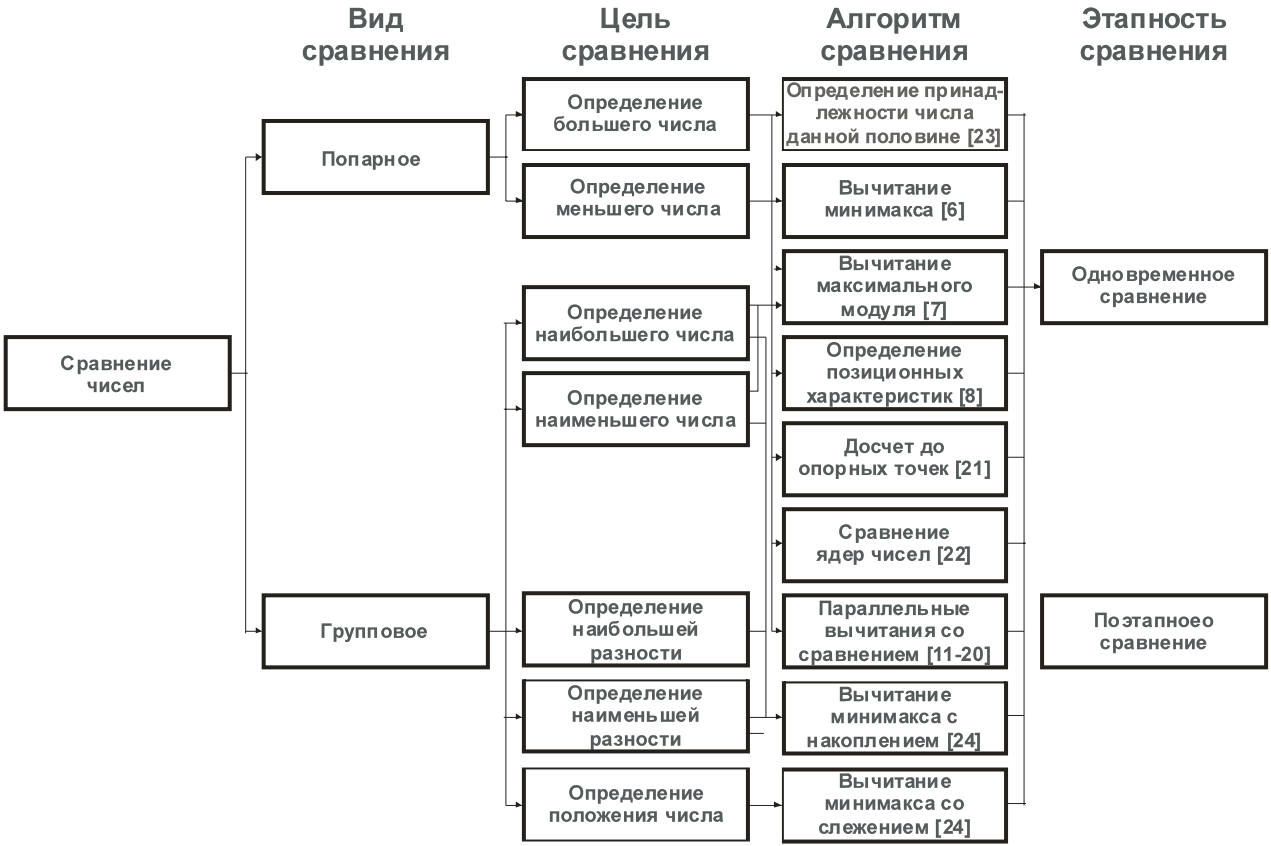
попарного сравнения в качестве самостоятельного подтверждается не только количеством работ, посвященных этому виду сравнения, но и тем, что во многих случаях групповое сравнение осуществляется поэтапно с использованием попарного сравнения. Кроме того, для попарного сравнения разработаны специальные методы и схемы, позволяющие осуществлять ускоренное решение задач сравнения. Эта принадлежность к попарному или групповому сравнению данных и выбрана в качестве наиболее общего признака классификации.

Следующий признак классификации включает указание цели, которая должна быть достигнута в результате сравнения. Как отмечалось ранее, при попарном сравнении - это определение равенства или неравенства чисел, большего или меньшего из них и их разности. При групповом сравнении количество решаемых задач увеличивается. Помимо выбора наибольшего или наименьшего из группы чисел дополнительно возникают задачи определения наибольшей и наименьшей разности, а также положения исследуемых чисел по отношению к одному или нескольким фиксированным числам.

Решение каждой из указанных задач может быть осуществлено применением тех или иных алгоритмов. Поэтому указание используемого алгоритма является следующим классификационным признаком сравнения.

Наконец, при групповом сравнении в одних случаях осуществляется одновременное сравнение всех данных, в других случаях - это поэтапный процесс, каждый этап которого основан на попарном сравнении. От этапности процедуры сравнения зависит быстродействие, которое является критерием эффективности алгоритма сравнения. Кроме того, этапность влияет на схемное решение устройства сравнения. Поэтому указание этапности процесса сравнения также принято в качестве классификационного признака.

Ниже приведена иллюстрация классификации сравнения чисел, выполненной в соответствии с описанными признаками, библиография основных работ, в которых приведены теоретические и практические результаты – в списке литературы [1-25].


Вид сравнения
Цель сравнения
Алгоритм сравнения
Этапность сравнения
Сравнение чисел
Попарное
Групповое
Определение большего числа
Определение меньшего числа
Определение наибольшего числа
Определение наименьшего числа
Определение наибольшей разности
Определение наименьшей разности
Определение положения числа
Определение принадлежности числа данной половине [23]
Вычитание минимакса [6]
Вычитание максимального модуля [7]
Определение позиционных характеристик [8]
Досчет до опорных точек [21]
Сравнение ядер чисел [22]
Параллельные вычитания со сравнением [11-20]
Вычитание минимакса с накоплением [24]
Вычитание минимакса со слежением [24]
Одновременное сравнение
Поэтапноео сравнение

ОСНОВНЫЕ РЕШЕНИЯ ПО СРАВНЕНИЮ ЧИСЕЛ

Рассмотрение будем производить согласно принятой классификации, при этом вначале описывается попарное, затем групповое сравнение чисел.

Попарное сравнение чисел

Пусть A и B – сравниваемые числа:

$$A=(\alpha_1,\alpha_2,\ldots,\alpha_n),$$
$$B=(\beta_1,\beta_2,\ldots,\beta_n). \qquad (3)$$

Необходимо определить результат A > B, A < B или A=B. Сравнение будем проводить для ненулевых чисел, в противном случае результат очевиден.

Если системой управления не предъявляются высокие требования к быстродействию, в основу сравнения может быть положен следующий алгоритм [6]. На каждом такте сравнения для первого и второго из сравниваемых чисел одновременно определяется максимальный остаток по всем модулям

$$r^1_{\mathbf{x}}=\max\{\alpha_1,\alpha_2,\ldots,\alpha_n\},$$
$$r^2_{\mathbf{x}}=\max\{\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\ldots,\boldsymbol{\beta}_n\}, \qquad (4)$$

выбирается

$$r^{к1}_{\mathbf{m}}=\min\{r^1_x,r^2_x,\ldots,r^к_x\}=(r^{к1}_{\mathbf{1}},r^{к1}_{\mathbf{2}},\ldots,r^{к1}_{\mathbf{i}},\ldots,r^{к1}_{\mathbf{n}})$$

и одновременно для обоих чисел находятся

$$A^1=A-r^{к1}_{\mathbf{m}}=(\alpha^1_1,\alpha^1_2,\ldots,\alpha^1_n)$$

и (6)

$$B^1=B-r^{к1}_{\mathbf{m}}=(\beta^1_1,\beta^1_2,\ldots,\beta^1_n),$$

Где:
$\alpha^1_1=(\alpha_1-r^{к1}_{\mathbf{1}})\pmod{m_1}$,
$\alpha^1_2=(\alpha_2-r^{к1}_{\mathbf{2}})\pmod{m_2}$,
……,
……,
$\alpha^1_i=(\alpha_i-r^{к1}_{\mathbf{i}})\pmod{m_i}$,
……,
……,
$\alpha^1_n=(\alpha_n-r^{к1}_{\mathbf{n}})\pmod{m_n}$

и

$$\beta^1_1 = (\beta_1 - r^{к1}_{\mathbf{1}}) \pmod{m_1},$$
$$\beta^1_2 = (\beta_2 - r^{к1}_{\mathbf{2}}) \pmod{m_2},$$
……,
……,
$$\beta^1_i = (\beta_i - r^{к1}_{\mathbf{i}}) \pmod{m_i},$$
……,
……,
$$\beta^1_n = (\beta_n - r^{к1}_{\mathbf{n}}) \pmod{m_n}.$$

Если одно из чисел (6) обратилось в нуль, процесс сравнения завершается. В противном случае процесс определения (4), (5), (6) продолжается для чисел A^1 и B^1, A^2 и B^2, …, A^s и B^s до тех пор, пока одно из чисел не обратится в нуль. Если $A^s = A^{s-1} - \mathbf{r}^{кs}_{\mathbf{m}} = 0$, а $B^s = B^{s-1} - \mathbf{r}^{кs}_{\mathbf{m}} \neq 0$, то A<B.

Если распределение чисел равновероятно, то среднее значение числа из диапазона (2) $A_{ср}$= M/2, и количество К тактов сравнения можно приближенно оценить следующим образом. Пусть $m_1<m_2<\ldots<m_n$. Тогда $\alpha_{max} = \beta_{max} = (m_n/2)$ и $r = \min\{\alpha_{max}, \beta_{max}\} = (m_n/2)$. Тогда $К = (A_{ср}/r) = = m_1 m_2 \ldots m_{n-1}$.

Почти такое же быстродействие дает сравнение по алгоритму [7]. Пусть $m_1<m_2<\ldots<m_n$. На первом такте одновременно выполняются операции

$$A_1 = A - \alpha_n = (\alpha^1_1, \alpha^1_2, \ldots, \alpha^1_i, \ldots, \alpha^1_{n-1}, 0)$$

и

$$B_1 = B - \beta_n = (\boldsymbol{\beta^1_1, \beta^1_2, \ldots, \beta^1_i, \ldots, \beta^1_{n-1}}, 0), \qquad (7)$$

где

$$\alpha^1_1 = (\alpha_1 - \alpha_n) \pmod{m_1},$$
$$\alpha^1_2 = (\alpha_2 - \alpha_n) \pmod{m_2},$$
……,
……,
$$\alpha^1_i = (\alpha_i - \alpha_n) \pmod{m_i},$$
……,
……,
$$\alpha^1_n = 0 \pmod{m_n}.$$

и

$\beta^1_1 = (\beta_1 - \beta_n) \pmod{m_1}$,
$\beta^1_2 = (\beta_2 - \beta_n) \pmod{m_2}$,
……,
……,
$\beta^1_i = (\beta_i - \beta_n) \pmod{m_i}$,
……,
……,
$\beta^1_n = 0 \pmod{m_n}$.

Если $A_1 = B_1$, то $A>B$ при $\alpha_n > \beta_n$, и процесс сравнения на этом завершается. Если же $A_1 \neq B_1$, то A_1 и B_1 кратны m_n и могут оказаться кратными одному или нескольким из остальных модулей системы. Поэтому на последующих тактах из обоих чисел A_1 и B_1, A_2 и B_2, …,A_s и B_s одновременно вычитается некоторое число $r=(m_n+\pi)$ до тех пор, пока одно из чисел не обратится в нуль. Если $A_s = A_{s-1} - r = 0$, а $B_s=B_{s-1}-r\neq 0$, то $A<B$. При этом для равновероятного распределения чисел $\pi \approx (1/ m_1)+ + (1/ m_2) +… + (1/ m_{n-1})$.

Гораздо большее быстродействие достигается при сравнении по алгоритму [8]. Пусть $m_1<m_2<…<m_n$. На первом такте одновременно выполняются операции $A_1=A- \alpha_1$ и $B_1=B- \beta_1$. Если $A_1= B_1$, то $A>B$ при $\alpha_1 > \beta_1$, и процесс сравнения на этом завершается. Если же $A_1\neq B_1$, то соотношение, например, $\alpha_1 > \beta_1$, запоминается. Числа A_1 и B_1 кратны m_1. На втором такте одновременно выполняются операции $A_2=(A_1/m_1)$, где $A_2=(\alpha^1_2, …,\alpha^1_n)$, и $B_2 = (B_1/m_1)$, где
$B_2=(\beta^1_2, …,\beta^1_n)$, с сокращением объема диапазона $(0, M_1]$ чисел до $M_1= m_2…m_n$. На следующем такте одновременно выполняются операции $A_3=A_2- \alpha^1_2$ и $B_3=B_2- \beta^1_2$. Если при этом $A_3=B_3$ и $\alpha^1_2<\beta^1_2$, то предпочтение отдается последнему соотношению, и $A<B$. Если же $A_3=B_3$ и $\alpha^1_2=\beta^1_2$, то о результате сравнения судят по запомненному соотношению $\alpha_1 > \beta_1$, и процесс сравнения на этом завершается. Если же $A_3\neq B_3$, то вместо $\alpha_1>\beta_1$ запоминается соотношение $\alpha^1_2<\beta^1_2$. Числа A_3 и B_3 кратны m_2. В том случае, если ни на одном из тактов не обнаруживается $A_i=B_i$, описанные действия повторяются до получения $A_{2(n-1)}=(\alpha^{n-1}_n)$ и $B_{2(n-1)}=(\beta^{n-1}_n)$, а о результате сравнения судят по соотношению между α^{n-1}_n и β^{n-1}_n. По существу здесь выполняется потактовый переход к полиадическому коду. В [9] показано, что не существует наилучшего метода определения значений позиционных

характеристик, при котором бы не нарушалась их однозначность, чем переход числа от системы остаточных классов к полиадическому коду, поскольку величина числа в СОК зависит от всех остатков числа. В [10] доказана невозможность сравнения чисел на основе анализа лишь соответствующих компонент этих чисел ни, пользуясь терминологией авторов, в логической, ни в алгебраической, ни в смешанной постановке задачи.

Верхняя оценка количества тактов сравнения по данному алгоритму К=2(n-1).

Алгоритм, описанный в работах [11-20], основан на идее [7]. Здесь также на первом такте одновременно выполняются операции

$$A_1=A-\alpha_n=(\alpha^1_1,\alpha^1_2,\ldots,\alpha^1_i,\ldots,\alpha^1_{n-1},0)$$

и

$$B_1=B-\beta_n=(\boldsymbol{\beta}^1_1,\boldsymbol{\beta}^1_2,\ldots,\boldsymbol{\beta}^1_i,\ldots,\boldsymbol{\beta}^1_{n-1},0).$$

Если $A_1= B_1$, то A>B при $\alpha_n > \beta_n$, и процесс сравнения на этом завершается. Если же $A_1\neq B_1$, то A_1 и B_1 кратны m_n. Однако здесь с целью повышения быстродействия потактное вычитание m_n производится не в процессе сравнения, а выполнено предварительно с сохранением в виде констант полученных значений. Количество таких констант $K= m_1m_2\ldots m_{n-1}$. Например, для системы модулей $m_1 =2$, $m_2 =3$, $m_3 =5$ таблица констант имеет вид:

$(0 \div (N-1))* m_n$ ($\Delta *5$)	$m_1 =2$	$m_2 =3$	Позиция нуля
	α^1_1	$\boldsymbol{\beta}^1_1$	
$0* m_3 = 0$	0	00	1
$1* m_3 = 5$	1	10	2
$2* m_3 = 10$	0	01	3
$3* m_3 = 15$	1	00	4
$4* m_3 = 20$	0	10	5
$5* m_3 = 25$	1	01	6

Далее из этой таблицы по значениям α^1_1 и $\boldsymbol{\beta}^1_1$ выбирается соответствующая запись и для каждого числа строится однорядный унитарный код длины $m_1m_2\ldots m_{n-1}$ двоичных разрядов, в каждом из которых только на r_1 – ом и r_2 – ом ($A_1- r_1* m_n =0$ и $B_1- r_2* m_n =0$) местах будут нули, а на остальных единицы. При этом номер позиции нуля и характеризует величину числа. Затем выполняется

сравнение номеров позиций нуля. Если $r_1 < r_2$, то A<B, если же $r_1 > r_2$, то A>B.

Приведенный алгоритм повышает быстродействие, однако требует существенных аппаратурных затрат для хранения количества констант, кратного $m_1m_2\ldots m_{n-1}$, и хранения унитарных кодов длины $m_1m_2\ldots m_{n-1}$.

Алгоритм [21] также основан на идее [7]. Среди чисел диапазона (0,M] есть числа $A_t=(\alpha^t_1,\alpha^t_2,\ldots,\alpha^t_i,\ldots,\alpha^t_j,..,\alpha^t_n)$, кратные одновременно нескольким модулям, например, m_i и m_j. Тогда $\alpha^t_i=0$, $\alpha^t_j=0$. Исходя из этого, в [21] в зависимости от требований к быстродействию системы и аппаратурным затратам выбираются определенные модули и составляется таблица констант – опорных точек. Для системы модулей $m_1=2$, $m_2=3$, $m_3=5$, $m_4=7$ ($M=m_1*m_2*m_3*m_4=210$) таблица опорных чисел, кратных, например, $m_2*m_4=21$ состоит из $(M/m_2*m_4)=10$ строк и имеет вид:

Опорная точка A	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$
0	(0, 0, 0, 0)
21	(1, 0, 1, 0)
42	(0, 0, 2, 0)
63	(1, 0, 3, 0)
84	(0, 0, 4, 0)
105	(1, 0, 0, 0)
126	0, 0, 1, 0)
147	(1, 0, 2, 0)
158	(0, 0, 3, 0)

Далее к обоим числам потактно добавляются единицы с одновременным сравнением образующихся значений $A_\tau=(\alpha^\tau_1, \alpha^\tau_2,\ldots, \alpha^\tau_i,.., \alpha^\tau_n)$ и $B_\tau=(\boldsymbol{\beta}^\tau_1, \boldsymbol{\beta}^\tau_2,\ldots, \boldsymbol{\beta}^\tau_i,.., \boldsymbol{\beta}^\tau_n)$ с константами таблицы. По достижении опорных точек их номера N_A и N_B запоминаются и затем сравниваются. Если $N_A > N_B$, то A>B, если же $N_A<N_B$, то A<B. При $N_A=N_B$ сравниваются количества подсуммированных единиц, которое не превышает в данном случае $K=m_2* m_4=21$. Количество тактов, необходимое для достижения опорных точек, и определяет быстродействие алгоритма.

Если нет ограничений на аппаратурные затраты, то для достижения максимального быстродействия целесообразно, по данному алгоритму, выбрать в качестве опорных точек числа диапазона, кратные наименьшему модулю. В этом случае приходим к таблице

констант предыдущего алгоритма, а количество тактов, необходимое для достижения опорных точек, не превышает величины наименьшего модуля.

Представим

$$A=R_{Ai}*m_i + \alpha_i,$$

$$B=R_{Bi}*m_i + \boldsymbol{\beta}_i,$$

где $R_{Ai}=[A/m_i]$, $R_{Bi}=[B/m_i]$. Пусть $m_1<m_2<\ldots<m_n$. Работа алгоритма [22] основана на вычислении R_{Ai} и R_{Bi} и их последующем сравнении. Если $R_{Ai} > R_{Bi}$ A>B. Если A=B, то сравниваются α_i и $\boldsymbol{\beta}_i$. При $\alpha_i > \boldsymbol{\beta}_i$ A>B. Для принятой системы модулей по ортогональным базисам H_1, H_2, …, H_i…, H_n находятся коэффициенты $r_1=(H_1/m_n)$, $r_2=(H_2/m_n)$, …, $r_i=(H_i/m_n)$, …, $r_n=((H_n-1)/m_n)$ и $r_0=(M/m_n)$, а R_A и R_B находятся, как

$$R_A=((r_{1*}\alpha_1)(\mathrm{mod}\ r_0)+ (r_{2*}\ \alpha_2)\ (\mathrm{mod}\ r_0) +\ldots+ +(r_{i*}\ \alpha_i)\ (\mathrm{mod}\ r_0)+\ldots+(r_{n*}\ \alpha_n)\ (\mathrm{mod}\ r_0))\ (\mathrm{mod}\ r_0),$$

$$R_B=((r_{1*}\boldsymbol{\beta}_1)(\mathrm{mod}\ r_0)+ (r_{2*}\ \boldsymbol{\beta}_2)\ (\mathrm{mod}\ r_0) +\ldots+ +(r_{i*}\ \boldsymbol{\beta}_i)\ (\mathrm{mod}\ r_0)+\ldots+(r_{n*}\ \boldsymbol{\beta}_n)\ (\mathrm{mod}\ r_0))\ (\mathrm{mod}\ r_0).$$

Оценка количества тактов сравнения по данному алгоритму K=n.

Будем отличать числа первой и второй половины диапазона. Если $2=m_1<m_2<\ldots<m_n$, то при $0\le A<(M/2)$ A- число первой половины, а при $(M/2)\le A<M$ A - число второй половины. Если же все модули нечетные, то при $0\le A\le((M-1)/2)$ A - число первой половины, а при $((M-1)/2)<A<M$ A - число второй половины.

Алгоритм сравнения, приведенный в работе [23], которая является первой из известных работ по сравнению чисел в системе остаточных классов, основан на определении принадлежности чисел и их разности данной половине.

В соответствии с [23] модули системы упорядочиваются, как $2=m_1<m_2<\ldots<m_n$.

Испытуемое число $A=(\alpha_1, \alpha_2, \ldots, \alpha_n)$.

Тогда число, обладающее остатками $\alpha_2, \alpha_3, \ldots, \alpha_n$, является числом первой половины. Метод [23] основан на определении четности числа первой половины с остатками $\alpha_2, \alpha_3, \ldots, \alpha_n$ и сравнении ее с четностью испытуемого числа, которая определяется остатком α_1.

В случае совпадения четностей данное число относится к первой половине, в противном случае – ко второй. При этом четность числа с остатками α_2, α_3, …, α_n находится итеративным путем с переходом к полиадическому коду, выполняемым аналогично [8], и фиксацией смены четностей на каждом шаге итерации.

Если сравниваемые числа разных половин, например, A – число первой половины, а B - число второй половины, то результат A<B очевиден. Если же A и B числа одной половины, то составляется разность Δ=A-B и определяется принадлежность Δ данной половине. В том случае, если Δ принадлежит первой половине, A≥B. Верхняя оценка количества тактов сравнения, приведенная в [23], K=1+6(n-2).

Групповое сравнение чисел

Пусть

$$\begin{aligned} A_1&=(\alpha^1_1, \alpha^1_2, \ldots, \alpha^1_n)\ ,\\ A_2&=(\alpha^2_1, \alpha^2_2, \ldots, \alpha^2_n)\ ,\\ &\ldots\ldots\ldots\ldots\ldots\ldots,\\ &\ldots\ldots\ldots\ldots\ldots\ldots,\\ A_к&=(\alpha^к_1, \alpha^к_2, \ldots, \alpha^к_n)\ . \end{aligned} \qquad (8)$$

сравниваемые числа.

Определение максимального и минимального из группы чисел при невысоких требованиях к быстродействию достигается на основе алгоритма, приведенного в работе [7], которая является, по-видимому, первой из известных работ по групповому сравнению чисел в системе остаточных классов. Пусть $m_1<m_2<\ldots<m_n$. Сравниваемые числа (8) представим в виде

$$\begin{aligned} A_1&=\theta_1*m_n+\alpha^1_n\ ,\\ A_2&=\theta_2*m_n+\alpha^2_n\ ,\\ &\ldots\ldots\ldots\ldots\ldots,\\ &\ldots\ldots\ldots\ldots\ldots,\\ A_k&=\theta_k*m_n+\alpha^k_n \end{aligned} \qquad (9)$$

На первом такте одновременно для всех чисел (5) выполняется операция

$$\begin{aligned} A^1_1&=A_1-\alpha^1_n=\theta_1*m_n\ ,\\ A^1_2&=A_2-\alpha^2_n=\theta_2*m_n\ , \end{aligned}$$

$$\begin{aligned} &\ldots\ldots\ldots\ldots\ldots\ldots, \\ &\ldots\ldots\ldots\ldots\ldots\ldots, \\ &A^1_к = A_к - \alpha^к_n = \theta_к * m_n . \end{aligned} \qquad (10)$$

На втором и последующих тактах сравнения одновременно из всех чисел (9) вычитается m_n. При этом в нуль раньше других обратится число с $\theta_{min} = \min\{\theta_1, \theta_2, \ldots, \theta_k\}$, которое сохраняется как минимальное. Соотношение величин $\alpha^1_n, \alpha^2_n, \ldots, \alpha^к_n$ в случае неодновременного обращения всех чисел в нуль не имеет значения, так как $\alpha^i_n < m_n$. При одновременном обращении в нуль нескольких чисел в качестве минимального сохраняется число из этой совокупности с $\alpha_{min} = \min\{\alpha^1_n, \alpha^2_n, \ldots, \alpha^к_n\}$. Процесс одновременного вычитания из всех чисел (9) величины m_n продолжается до обращения в нуль всех чисел. Последнее обратившееся в нуль число сохраняется как максимальное. При одновременном обращении в нуль нескольких чисел в качестве максимального сохраняется число из этой совокупности с наибольшим α^i_n. Количество К тактов сравнения $K = 1 + m_1 m_2 \ldots m_{n-1}$.

Расширение совокупности результатов – получение в результате сравнения наибольшего, наименьшего чисел, их максимальной и минимальной разности достигается с помощью алгоритма [24], использующего идею [6]. Здесь также на каждом такте сравнения одновременно для всех сравниваемых чисел определяются

$$\begin{aligned} &r^1_{\mathbf{x}} = \max\{\alpha^1_1, \alpha^1_2, \ldots, \alpha^1_n\}, \\ &r^2_{\mathbf{x}} = \max\{\alpha^2_1, \alpha^2_2, \ldots, \alpha^2_n\}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ &r^к_{\mathbf{x}} = \max\{\alpha^к_1, \alpha^к_2, \ldots, \alpha^к_n\}, \end{aligned} \qquad (11)$$

и выбирается

$$r^к_{\mathbf{m}} = \min\{r^1_x, r^2_x, \ldots, r^к_x\} = (r^к_{\mathbf{1}}, r^к_{\mathbf{2}}, \ldots, r^к_{\mathbf{i}}, \ldots, r^к_{\mathbf{n}}), \qquad (12)$$

и одновременно для всех чисел находятся

$$\begin{aligned} &A^1_{\mathbf{1}} = A_{\mathbf{1}} - r^к_{\mathbf{m}} = (\alpha^1_{11}, \alpha^1_{21}, \ldots, \alpha^1_{n1}), \\ &A^1_{\mathbf{2}} = A_{\mathbf{2}} - r^к_{\mathbf{m}} = (\alpha^2_{12}, \alpha^2_{22}, \ldots, \alpha^2_{n2}), \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ &A^1_{\mathbf{j}} = A_{\mathbf{i}} - r^к_{\mathbf{m}} = (\alpha^j_{1j}, \alpha^j_{2j}, \ldots, \alpha^j_{nj}), \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \end{aligned} \qquad (13)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$A^1_{\mathbf{k}}=A_{\mathbf{k}} - r^{\kappa}_{\mathbf{m}} =(\alpha^k_{1k},\alpha^k_{2k},\ldots, \alpha^k_{nk}),$$

где

$$\alpha^j_{ij}= (\alpha^j_i - r^k_{\mathbf{i}}) \,(\mathrm{mod}\, m_i),$$

и одновременно значения $r^{\kappa}_{\mathbf{m}}$ накапливаются на регистрах наибольшего и наименьшего чисел. Процесс продолжается до тех пор, пока одно из чисел (13) не обратится в нуль. Это число и является наименьшим, оно записано на своем регистре.

Далее процесс определения (11), (12), (13) продолжается для всех ненулевых чисел с накоплением дополнительно и на регистре максимальной разности до обращения в нуль последнего из чисел, которое является наибольшим. Это число записано на своем регистре. При каждом очередном обращении в нуль на регистрах оставшихся ненулевыми чисел записаны значения разности между исходным и очередным обратившимся в нуль числом, из которых в процессе вычитания выбирается минимальное значение. При равновероятном распределении чисел приближенная временная оценка такая же, как и при попарном сравнении, использующем идею [6].

Определение ближайшего к заданному числу большего и меньшего из группы чисел с помощью алгоритма [25] также использует идею алгоритма [6]. Здесь также на каждом такте сравнения одновременно для всех сравниваемых чисел определяются

$$\begin{aligned}
&r^1_{\mathbf{x}}= \max\{\alpha^1_1, \alpha^1_2, \ldots, \alpha^1_n\},\\
&r^2_{\mathbf{x}}= \max\{\alpha^2_1, \alpha^2_2, \ldots, \alpha^2_n\},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots,\\
&r^{\kappa}_{\mathbf{x}}= \max\{\alpha^{\kappa}_1, \alpha^{\kappa}_2, \ldots, \alpha^{\kappa}_n\},
\end{aligned} \qquad (14)$$

и выбирается

$$r^{\kappa}_{\mathbf{m}}=\min\{r^1_x, r^2_x, \ldots, r^{\kappa}_x\}=(r^{\kappa}_{\mathbf{1}}, r^{\kappa}_{\mathbf{2}}, \ldots, r^{\kappa}_{\mathbf{i}}, \ldots, r^{\kappa}_{\mathbf{n}}), \qquad (15)$$

и одновременно для всех чисел находятся

$$\begin{aligned}
&A^1_{\mathbf{1}}=A_{\mathbf{1}} - r^{\kappa}_{\mathbf{m}} =(\alpha^1_{11},\alpha^1_{21},\ldots, \alpha^1_{n1}),\\
&A^1_{\mathbf{2}}=A_{\mathbf{2}} - r^{\kappa}_{\mathbf{m}} =(\alpha^2_{12},\alpha^2_{22},\ldots, \alpha^2_{n2}),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,
\end{aligned}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \quad (16)$$
$$A^1_{\mathbf{j}}=A_{\mathbf{i}} - r^{\kappa}_{\mathbf{m}} =(\alpha^j_{1j},\alpha^j_{2j},\ldots, \alpha^j_{nj}),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots,$$
$$A^1_{\mathbf{k}}=A_{\mathbf{k}} - r^{\kappa}_{\mathbf{m}} =(\alpha^k_{1k},\alpha^k_{2k},\ldots, \alpha^k_{nk}),$$

где

$$\alpha^j_{ij}= (\alpha^j_i - r^k_{\mathbf{i}}) \ (\mathrm{mod}\ m_i).$$

Одновременно значения $r^{\kappa}_{\mathbf{m}}$ накапливаются на регистрах. Процесс продолжается до тех пор, пока одно из чисел (16) не обратится в нуль.

Далее процесс определения (14), (15), (16) продолжается для всех ненулевых чисел до обращения в нуль последнего из чисел с накоплением на регистрах очередного и предыдущего из обращенных в нуль чисел. Такое определение основано на фиксировании последовательных моментов обращения в нуль заданного и последующего числа из группы чисел. При равновероятном распределении чисел приближенная временная оценка такая же, как и для предыдущего алгоритма.

ЗАКЛЮЧЕНИЕ

Проведенный обзор основных решений по сравнению чисел в системе остаточных классов позволяет высказать следующие соображения.

При всем разнообразии алгоритмов полученных решений все они основаны на двух принципах. Первый принцип заключается в предварительном преобразовании непозиционного представления в некоторое позиционное и использовании полученного позиционного представления либо непосредственно для решения основной задачи, либо для решения подзадач основной задачи. Второй принцип заключается в вычитании некоторых констант из сравниваемых чисел, выполняемом либо итерационно в процессе сравнения, либо выполненным предварительно. При этом быстродействие алгоритмов, использующих первый принцип, значительно выше, чем алгоритмов, использующих второй принцип на итерационной основе. Однако достигается это за счет увеличения аппаратурных затрат. Использование алгоритмов, использующих второй принцип с предварительно выполненным вычитанием, дает некоторое увеличение быстродействия по

сравнению с алгоритмами, использующими первый принцип, но требует существенных аппаратурных затрат.

На основе выполненного обзора можно сделать вывод об отсутствии к настоящему времени эффективных решений по сравнению чисел в системе остаточных классов. Поскольку различными алгоритмами решение достигается с разными техническими и экономическими показателями, дальнейшие работы по сравнению чисел представляется целесообразным проводить в направлении достижения оптимального соотношения указанных показателей для заданных условий применения.

Литература

1.Н.И.Червяков. Методы и принципы построения модулярных нейрокомпьютеров. Сайт http://www.computer-museum.ru/, 2005

2.И.Я.Акушский, Д.И.Юдицкий. Машинная арифметика в остаточных классах. М.: Сов. Радио, 1968. 440 с.

3.А.Ахо, Дж.Хопкрофт, Дж.Ульман. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.

4.Л.Б.Копыткова, Н.И.Червяков. Реализация деления чисел в системе остаточных классов на модули системы. Вестник Ставропольского государственного университета, 34/2003, с.7-11

5.Ю.Д.Полисский. Цифровое сравнение данных а АСУ ТП и схемах автоматики. М.: Энергия, 1979. 136 с.

6.Ю.Д.Полисский, М.Г.Факторович. Устройство для сравнения чисел в системе остаточных классов. Авт. свид. СССР №618739 М. Кл^2 G 06 F 7/04, 1978.

7.М.Г.Факторович, Ю.Д.Полисский. Устройство для определения максимального и минимального из “n” чисел, представленных в системе остаточных классов. Авт. свид. СССР №603987 М. Кл^2 G 06 F 7/04, 1978.

8.М.Г.Факторович, Ю.Д.Полисский. Устройство для сравнения чисел, выраженных в системе остаточных классов. Авт. свид. СССР №608155 М. Кл^2 G 06 F 7/04, 1978.

9.Сабо Н. Определение знака в неизбыточных системах счисления остаточных классов. Кибернетический сборник, №8. М., «Мир», 1964.

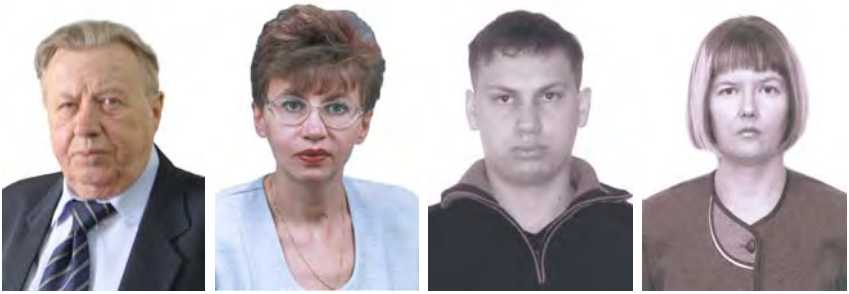
10.Амербаев В.М., Касимов Ю.Ф. О сравнении чисел в непозиционных системах счисления. // Теория кодирования и оптимизации сложных систем. Алма-Ата: Наука, 1977. С. 47 — 54.

11.В.А.Краснобаев, Е.И.Бороденко, А.И.Бецков и др. Устройство для сравнения чисел в системе остаточных классов. Авт. свид. СССР №1037244 G 06 F 7/04, 1983.

12.В.А.Краснобаев, Л.Г.Трусей. Устройство для сравнения чисел в системе остаточных классов. Авт. свид. СССР №1121670 G 06 F 7/04, 1984.
13.В.А.Краснобаев. Устройство для сравнения чисел в системе остаточных классов. Авт. свид. СССР №1145338 G 06 F 7/04, 1985.
14.В.А.Краснобаев. Устройство для сравнения чисел в системе остаточных классов. Авт. свид. СССР №1160394 G 06 F 7/04, 1985.
15.В.И.Долгов, В.А.Краснобаев, А.В.Брезгунов. Устройство для сравнения чисел в системе остаточных классов. Авт. свид. СССР №1224803 G 06 F 7/04, 1986.
16.В.А.Краснобаев, И.Д.Горбенко, М.А.Гальцев и др. Устройство для сравнения чисел в системе остаточных классов. Авт. свид. СССР №1427358 G 06 F 7/04, 1987.
17.В.А.Краснобаев, О.Н.Фоменко, В.П.Ирхин и др. Устройство для сравнения в системе остаточных классов. Авт. свид. СССР №1552171 G 06 F 7/04, 1990.
18.В.А.Краснобаев. Методы арифметического сравнения чисел, представленных кодом системы остаточных классов // АСУ и приборы автоматики. 1987. Вып. 84. С. 74 -76.
19.В.А.Краснобаев. Методы сравнения чисел, представленных кодом системы остаточных классов // Электрон.моделирование. 1988. № 2. С. 84 – 87.
20.В.А.Краснобаев. Методы сравнения операндов в системе остаточных классов // Радиотехника. 2003. Вып. 134. С. 223 – 228 .
21.О.В.Ревинский. Устройство для сравнения чисел. Авт. свид. СССР №1439574 G O6 F 7/04, 1988
22.Хлевной С.Н., Сагдеев К.М. Устройство для сравнения чисел в модулярном коде. Авт. свид. СССР, G06 F 7/04, 1986
23.В.Н.Тейтельбаум. Сравнение чисел в чешской системе счисления. ДАН СССР, 1958, т.121, №5.
24.Ю.Д.Полисский. Устройство для сравнения чисел в системе остаточных классов. Патент Украины, 42649 A, G 06 F 7/04, 2001.
25.Ю.Д.Полисский. Устройство для определения в системе остаточных классов числа, ближайшего к заданному. Патент Украины, 47630 A, G 06 F 7/04,2002.

УДК 681.3

Н.И. Червяков, И.Н. Лавриненко,
С.В. Лавриненко, О.С. Мезенцева



Методы и алгоритмы округления, масштабирования и деления чисел в модулярной арифметике

(Ставропольский военный институт связи ракетных войск)

Рассмотрены методы и алгоритмы деления числа в модулярном коде на одно из оснований или их произведение. Разработан метод деления числа в модулярном коде при произвольных значениях делимого и делителя.

В связи с тем, что модулярная арифметика целочисленная, то при вычислениях промежуточные значения операндов могут переполнять динамический диапазон. Подобная проблема может возникнуть и в традиционных компьютерах, если они оперируют с целыми числами. Во избежание переполнения надо промасштабировать (уменьшить) значения операндов. Промасштабированные величины затем используются в следующих итерациях. Это означает, что операция масштабирования должна применяться к данным с использованием заранее заданной константы, которая округляется до ближайшего целого. Все эти операции связаны с операцией деления.

В данной работе рассмотрим проблемы округления, масштабирования и деления в СОК. Поскольку СОК не является взвешенной системой счисления, операция деления, которая включает сравнение по величине двух операндов, не может считаться простой.

Деление в модулярной арифметике относится к немодульным операциям и является одной из важнейших операций в модулярной компьютерной арифметике, так как лежит в основе многих других операций и входит в состав операций вычислительных алгоритмов.

Операцию деления в СОК можно отнести к одной из трех различных форм [1, 2]:

1. Деление с нулевым остатком.
2. Округление и масштабирование.
3. Основное деление.

Рассмотрим все основные формы модулярного деления.

Деление с нулевым остатком

Для этой формы деления известно, что делимое представляет собой целое число, кратное делителю, а также известно, что делитель и P являются взаимно простыми. Эта категория имеет ограниченную область использования, поскольку должно быть известно априори, удовлетворены ли условия, необходимые для осуществления операции. Для этого алгоритма используется следующая теорема 1.

Теорема 1

Если a делится на b без остатка и наибольший общий делитель (НОД) величин a и b равен 1, то

$$\left|\frac{a}{b}\right|_{p_i} = \left|\frac{1}{b}a\right|_{p_i} \tag{1}$$

для всех p_i, где $\left|\frac{1}{b}\right|_{p_i}$ – мультипликативная обратная к b величина, взятая по модулю p_i.

Доказательство.

Предположим, что необходимые два условия удовлетворены, тогда a/b это целое число, представленное в остатках имеет вид

$$\left(\left|\frac{a}{b}\right|_{p_1}, \left|\frac{a}{b}\right|_{p_2}, \ldots, \left|\frac{a}{b}\right|_{p_n}\right). \tag{2}$$

Выполним вычисление $\frac{a}{b}\cdot a$ в остаточном коде:

$$\left|\frac{a}{b}\right|_P \leftrightarrow \left\{\left|\frac{a}{b}\right|_{p_1}, \left|\frac{a}{b}\right|_{p_2}, \ldots, \left|\frac{a}{b}\right|_{p_n}\right\} \tag{3}$$

или

$$\frac{|b|_P}{\left|\frac{a}{b}\cdot b\right|_P} \leftrightarrow \frac{\{|b|_{p_1}, |b|_{p_2}, \ldots, |b|_{p_n}\}}{\left\{\left|\left|\frac{a}{b}\right|_{p_1} |b|_{p_1}\right|_{p_1}, \left|\left|\frac{a}{b}\right|_{p_2} |b|_{p_2}\right|_{p_2}, \ldots, \left|\left|\frac{a}{b}\right|_{p_n} |b|_{p_n}\right|_{p_n}\right\}}. \tag{4}$$

Так как $\left|\frac{a}{b}\cdot b\right|_P = |a|_P$, следовательно

$$\left|\left|\frac{a}{b}\right|_{p_i} |b|_{p_i}\right| = |a|_{p_i}. \tag{5}$$

По уникальности мультипликативной инверсии следует, что

$$\left|\frac{a}{b}\right|_{p_i} = \left|\frac{1}{b}a\right|_{p_i}.$$

Если b не делит a, то величина $\frac{a}{b}$ не является целой и выражение $\left|\frac{a}{b}\right|_{p_i}$ не определено. Следовательно, (1) не имеет смысла.

Пример 1. Деление с нулевым остатком.

Для модулей p_1=29, p_2=32 и p_3=31 разделим число 1872 на 9.

Решение. Остаточное представление 1872 – это (16, 16, 12). Остаточное представление 9 это – (9, 9, 9), тогда для 1872/9=208 остаточный код

$$\left\{16\cdot\left|\frac{1}{9}\right|_{29}, 16\cdot\left|\frac{1}{9}\right|_{32}, 12\cdot\left|\frac{1}{9}\right|_{31}\right\} = (5,16,22) \leftrightarrow 208.$$

С другой стороны, если мы делим 1873 на 9 (1873 не делится на 9 без остатка), то получим

$1873 \leftrightarrow (17,17,13)\cdot(13,25,7) = (18,9,29) \leftrightarrow 6601$, что абсолютно неправильно.

Масштабирование целых положительных чисел

При делении этой формы делимое является произвольным, а делителем может быть любой сомножитель P, представляющий собой произведение первых степеней некоторых модулей. Это деление аналогично делению на степень числа 2 в двоичной арифметике в том смысле, что деление на числа, принадлежащие определенному ограниченному множеству, выполняется быстрее, чем деление на произвольный делитель. Деление в любой целочисленной системе счисления определяется формулой $a = \left[\frac{a}{b}\right]\cdot b + |a|_b$, где a представляет собой делимое, b – делитель, $\left[\frac{a}{b}\right]$ – целая часть отношения a к b (частное), а $|a|_b$ – остаток (наименьший целый положительный остаток). Целью алгоритма масштабирования является нахождение $\left[\frac{a}{b}\right]$ для значений b из ограниченной области. Заметим, что $\left[\frac{a}{b}\right] = \frac{a-|a|_b}{b}$. Следовательно, в системе вычетов $\left[\frac{a}{b}\right]$ представляется величинами $\left(\left|\frac{a-|a|_b}{b}\right|_{p_1}, \left|\frac{a-|a|_b}{b}\right|_{p_2}, \ldots, \left|\frac{a-|a|_b}{b}\right|_{p_n}\right)$, где $\left|\frac{a-|a|_b}{b}\right|_{p_i}$ принимают целые значения. Если b совпадает с одним из p_i или является произведением первых степеней некоторых модулей p_i, то $|a|_b$ можно найти. Тогда по теореме, используемой в форме деления с нулевым остатком, для всех i, для которых НОД величин p_i и b равен 1, можно получить

$$\left|\frac{a-|a|_b}{b}\right|_{p_i} = \left[\frac{a}{b}\right]_{p_i} = \left|\frac{1}{b}\cdot\left(a-|a|_b\right)\right|_{p_i} . \qquad (6)$$

Это уравнение задает цифры системы вычетов для $\left[\frac{a}{b}\right]$ для всех таких цифр, что НОД величин p_i и b равен 1. Остальные цифры

могут быть найдены с помощью метода расширения базы. Таким образом, алгоритм масштабирования состоит из двух этапов:

1. Деление с нулевым остатком.

2. Расширение базы.

Процесс масштабирования покажем числовым примером.

Пример 2. Масштабирование положительного числа единичным модулем.

Для модулей p_1=2, p_2=3, p_3=5 и p_4=7 определим остаточное представление значения целого числа $\left[\frac{a}{5}\right]$. Пусть a имеет остаточный код $(1, 2, 4, 3) \leftrightarrow 59$. В качестве делителя используется модуль p_3.

Решение. Сначала определим остаточное представление числа, которое делится на 5 и является ближайшим целым к a, не превышающим a, то есть $a - |a|_5$. Это можно найти путем вычитания остатка a по модулю 5.

Модули	2	3	5	7	
	$59 \leftrightarrow (1,$	2,	4,	3)	
Вычитаем $\|a\|_5 = 4 \leftrightarrow$	(0	1,	4,	4)	
	(1, 1, 0, 6)				$\leftrightarrow a - \|a\|_5$

Результат делится на 5 кроме модуля p_3, который сам является делителем. Все модули простые по отношению к делителю. Применяем метод деления с нулевым остатком, при этом остаточную цифру по модулю 5 временно игнорируем.

Умножаем на $\left\|\frac{1}{5}\right\|_{p_i}$	$a - \|a\|_5$	$\leftrightarrow$	(1, 1, –, 6)		
		$\leftrightarrow$	(1, 2, –, 3)	2,3,7 $\leftrightarrow$	$\frac{a - \|a\|_5}{5}$
			(1, 2, –, 4)		

Исходный интервал определения для всего набора модулей был равен [0-209], а $\frac{a - |a|_5}{5}$ оказался в интервале [0-41], поэтому оста-

точное представление (1, 2, –, 4) не ясно. Остаток по модулю 5 может быть найден путем расширения базы. Это можно сделать по методу Гарнера или предложенному методу в работе [3]. Для этого остаток по модулю 5 примем за 0 в первом случае и за $|a|_5$ – во втором.

Расширение базы:

на основе известного метода Гарнера

Номер операции	Модули	2,	3,	5,	7
	a -$\|a\|_5$ $\leftrightarrow$	(1,	2,	0,	4)
1.	Вычитаем 1	(1,	1,	1,	1)
		(0,	1,	4,	3)
2.	Умножаем на $\left\|\frac{1}{2}\right\|_{p_i}$	(-,	2,	3,	4)
		(-,	2,	2,	5)
3.	Вычитаем 2	(-,	2,	2,	2)
		(-,	0,	0,	3)
4.	Умножаем на $\left\|\frac{1}{3}\right\|_{p_i}$	(-,	-,	2,	5)
		(-,	-,	0,	1)
5.	Вычитаем 1	(-,	-,	1,	1)
		(-,	-,	4,	0)

Если $\left[\frac{a}{5}\right]$ обозначить как z_5, тогда получим $|z_5|+4=0$, отсюда

6. $|z_5| = -4 \bmod 5 = 1 \bmod 5$.

Итак $\left[\frac{a}{5}\right] = (1,\ 2,\ 1,\ 4) \leftrightarrow 11.$

на основе предложенного метода

Матрица констант для набора с измененным порядком модулей

2,	3,	5,	7
1	1	3	2
0	2	4	2
0	0	6	2
0	0	0	3

1. Умножение

$$\begin{array}{l|cccc|c|cccc} 1\cdot & 1 & 1 & 3 & 2 & \to & 1 & 1 & 3 & 2 \\ 2\cdot & 0 & 2 & 4 & 2 & \to & 0 & 2 & 4 & 2 \\ 4\cdot & 0 & 0 & 6 & 2 & \to & 0 & 0 & 6 & 2 \\ |a|_5\cdot & 0 & 0 & 0 & 3 & \to & 0 & 0 & 0 & 3|a|_5 \end{array}$$

2. Сложение $\quad (1,\ 2,\ 1,\ 17+3|a|_5)$

3. Вычисление остатка по mod 5

$17+3|a|_5 = 0$

Или $|a|_5 = -17\left|\frac{1}{3}\right|_5 \bmod 5 \equiv$

$\equiv -34 \bmod 5 \equiv -4 \bmod 5 \equiv 1 \bmod 5.$

Итак $\left[\frac{a}{5}\right] = (1,\ 2,\ 1,\ 4) \leftrightarrow 11.$

В методе Гарнера для замены вычитания сложением необходимо использовать дополнительный код, при этом для вычитания необ-

ходимо две операции. Выигрыш предложенного метода оценивается как $\frac{3(n-1)}{3} = n-1$.

Пример 3. Масштабирование положительного числа несколькими модулями.

В примере 2 коэффициентом масштабирования был только один модуль. В этом примере коэффициентом масштабирования будет произведение двух модулей, а именно $3 \times 5 = 15$. Вначале делим на 3 и полученное частное является новым делимым для делителя, равного 5, деление на 5 выдает значения целого числа частного. Для завершения операции масштабирования необходимо выполнить операцию расширения базы. Изменение последовательности деления сначала выполнить деление на 5, а затем на 3 не меняет результата.

Для модулей p_1=2, p_2=3, p_3=5 и p_4=7 число a=89 $\leftrightarrow$ (1, 2, 4, 5) масштабируем коэффициентом 15. Обозначим результат $\left[\frac{a}{15}\right]$ как z.

Решение.

Модули	2, 3, 5, 7	
Остаточное представление для a	(1, 2, 4, 5)	
Вычитаем $\lvert a\rvert_3 = 2$	(0, 2, 2, 2)	
	(1, 0, 2, 3)	$\overset{2,3,5,7}{\leftrightarrow} \quad a - \lvert a\rvert_3$
Умножаем на $\left\lvert\frac{1}{3}\right\rvert_{P_i}$	(1, –, 2, 5)	
	(1, –, 4, 1)	$\overset{2,\ 5,7}{\leftrightarrow} \frac{a - \lvert a\rvert_3}{3}$
Вычитаем $\lvert a\rvert_5 = 4$	(0, –, 4, 4)	
	(1, –, 0, 4)	$\overset{2,\ 3,7}{\leftrightarrow} \frac{a - \lvert a\rvert_3}{3} - \lvert a\rvert_5$
Умножаем на $\left\lvert\frac{1}{5}\right\rvert_{P_i}$	(1, –, –, 3)	
	(1, –, –, 5)	

Для расширения базы внесем 0 в пропущенные колонки для метода Гарнера и обозначим как $|a|_3$ и $|a|_5$ – для предложенного метода.

Метод Гарнера	Предложенный метод
1. Вычитаем 1 $\begin{array}{r}(1,\ 0,\ 0,\ 5)\\ (1,\ 1,\ 1,\ 1)\\ \hline (0,\ 2,\ 4,\ 4)\end{array}$ 2. Умножаем на $\left\|\frac{1}{2}\right\|_{p_i}$ $\begin{array}{r}(-,\ 2,\ 3,\ 4)\\ \hline (-,\ 1,\ 2,\ 2)\end{array}$ 3. Вычитаем 2 $\begin{array}{r}(-,\ 2,\ 2,\ 2)\\ \hline (-,\ 2,\ 0,\ 0)\end{array}$ 4. Тогда $\left\|\frac{1}{2}\|z|_3+2\right\|=0$ и $\left\|\frac{1}{2}\|z|_5+2\right\|=0$. Следовательно, $\|z|_3=2$ и $\|z|_5=0$. Отсюда, масштабируемое число $\left[\frac{89}{15}\right]$ это $(1, 2, 0, 5)\leftrightarrow 5$.	Разобьем матрицу констант для измененной последовательности модулей на 2 матрицы Модули $\begin{array}{r}2,\ 7,\ 3\\ \hline\end{array}$ $\begin{array}{r}2,\ 7,\ 5\\ \hline\end{array}$ $\begin{vmatrix}1, & 3, & 1\\ 0, & 4, & 2\\ 0, & 0, & 2\end{vmatrix}$ $\begin{vmatrix}1, & 3, & 2\\ 0, & 4, & 1\\ 0, & 0, & 3\end{vmatrix}$ 1. Умножаем $\begin{array}{l}1\cdot\\ 5\cdot\\ \|x\|_3\cdot\end{array}\begin{vmatrix}1, & 3, & 1\\ 0, & 20, & 10\\ 0, & 0, & 2\cdot\|x\|_3\end{vmatrix}$; $\begin{array}{l}1\cdot\\ 5\cdot\\ \|x\|_5\cdot\end{array}\begin{vmatrix}1, & 3, & 2\\ 0, & 20, & 5\\ 0, & 0, & 9\cdot\|x\|_5\end{vmatrix}$ $\begin{array}{r}1,\ 2,\ 14+2\cdot\|x\|_3\\ \hline\end{array}$ $\begin{array}{r}1,\ 2,\ 10+3\cdot\|x\|_5\\ \hline\end{array}$ Отсюда
$14+2\cdot\|x\|_3=0,$ $\|x\|_3=-7\bmod 3\equiv 2\bmod 3$	$10+3\cdot\|x\|_5=0,$ $\|x\|_5=-10\left\|\frac{1}{3}\right\|_5\equiv 20\bmod 5\equiv$ $\equiv 0\bmod 5$
	$\left[\frac{89}{15}\right]$=(1, 2, 0, 5)$\leftrightarrow$5

Итак, для масштабирования числа большим коэффициентом масштаба используется последовательное деление на простые числа и расширение базы модулей СОК.

Математические модели масштабируемых чисел другого знака

Отрицательные числа в СОК можно записать как $P-a$. Если известно, что число отрицательное, то легко можно его определить из $P-a$, затем проводится масштабирование на b, получаем $\left[\frac{a}{b}\right]$ и затем представляем результаты как P+$\left[\frac{a}{b}\right]$. Если же знак неизвестен, то возникает определенная сложность. При масштабировании отрицательного числа как будто бы число положительное, резуль-

татом будет $\frac{P}{b}+\left[\frac{a}{b}\right]$ вместо необходимого P+$\left[\frac{a}{b}\right]$. Поэтому перед масштабированием необходимо определить знак a, этого процесса можно избежать, если принять во внимание следующее обстоятельство: деление на b отображает все числа в интервале $\left[0,\frac{P}{2}-1\right]$ в интервал $\left[0,\frac{P}{2b}-1\right]$ и все числа в интервале $\left[\frac{P}{2},P-1\right]$ в интервал $\left[\frac{P}{2b},\frac{P-1}{2}\right]$. Отсюда можно выполнить вначале деление числа на b, а затем, принимая во внимание интервал, в котором находится $\left\|\left[\frac{a}{b}\right]\right\|_P$, определяется знак a. Если a – отрицательное число $\left|-\frac{P}{b}\right|_P$, то к нему прибавляется по модулю P для получения правильного ответа P+$\left[\frac{a}{b}\right]$.

Определение интервала, в котором находится $\left[\frac{a}{b}\right]$ требует такого же количества времени, что и для определения знака числа. Однако, как было рассмотрено в предыдущем примере, процесс масштабирования требует операции расширения базы модулей СОК на основе цифр ОПСС, поэтому можно определить местонахождение числа $\left[\frac{a}{b}\right]$ путем использования цифр ОПСС.

Рассмотрим метод одновременного масштабирования и распознавания знака.

Пример 4. Одновременное масштабирование и определение знака.

Для модулей p_1=13, p_2=9, p_3=11, p_4=7 и p_5=2 масштабируем число $a=-979 \leftrightarrow (9,\ 2,\ 0,\ 1,\ 1)$ на число $b=7\cdot 11$ с округлением до ближайшего целого числа.

Модули	13, 9, 11, 7, 2	
Остаточное представление числа a	(9, 2, 0, 1, 1)	
Вычитаем 1	(1, 1, 1, 1, 1)	
	(8, 1, 10, 0, 0)	
Умножаем на $\left\|\frac{1}{7}\right\|_{P_I}$	(2, 4, 8, –, 1)	
	(3, 4, 3, –, 0)	
Вычитаем 3	(3, 3, 3, –, 1)	
	(0, 1, 0, –, 1)	
Умножаем на $\left\|\frac{1}{11}\right\|_{P_I}$	(6, 5, –, –, 1)	
	(0, 5, –, –, 1)	$13, 9, 2 \leftrightarrow \left[\frac{P-979}{7\cdot 11}\right]$

Внесем 0 в пропущенные колонки для расширения базы

	(0, 5, 0, 0, 1)
Вычитаем 0	(0, 0, 0, 0, 0)
	(0, 5, 0, 0, 1)
Умножаем на $\left\|\frac{1}{13}\right\|_{P_I}$	(–, 7, 6, 6, 1)
	(–, $\boxed{8}$, 0, 0, 1)
Вычитаем 8	(–, 8, 8, 1, 0)
	(–, 0, 3, 6, 1)
Умножаем на $\left\|\frac{1}{9}\right\|_{P_I}$	(–, –, 5, 4, 1)
	(–, –, 4, 3, $\boxed{1}$)
Вычитаем 1	(–, –, 1, 1, 1)
	(–, –, 3, 2, 0)

Пусть z будет результатом этой операции масштабирования, то есть $z = \left[\frac{P-979}{7\cdot 11}\right]$, тогда:

$$\frac{1}{13}\cdot\frac{1}{9}|z|_{11} + 3 = 0,\ |z|_{11} = 1, \qquad \frac{1}{13}\cdot\frac{1}{9}|z|_{7} + 2 = 0,\ |z|_{7} = 4.$$

В строку (0, 5, –, –, 1) $\overset{13,9,2}{\leftrightarrow}\left[\frac{P-979}{7\cdot 11}\right]$ добавляем $|z|_{11}, |z|_7$, тогда остаточное представление z будет равно (0, 5, 1, 4, 1). В зависимости от знака *a*, остаточное представление z будет либо $\left[\frac{a}{b}\right]$, либо $P+\left[\frac{a}{b}\right]$. Цифры ОПСС для z по модулям 13, 4, 2 были вычислены на протяжении процесса масштабирования и обозначились $\square$ в преобразованиях. Отсюда *z* можно выразить как $z=1(9\cdot 13)+8(13)+0(1)$.

Если a – положительное число, то $|a|_P$ должно быть в интервале $\left[0,\frac{P/2-1}{77}\right]$ или $[0,(9\cdot 13)-1]$. Так как наиболее значимой цифрой ОПСС $|z|_P$ является 1, то $|z|_P$ не может входить в этот интервал. Отсюда *a* должно быть отрицательным. Следовательно, для получения правильного результата необходимо $\left[-\frac{P}{b}\right]_P$ сложить с $|z|_P$. Для завершения примера необходимо число (0, 0, 8, 4, 0), которое является величиной $\left|-\frac{P}{b}\right|_P$ сложить с числом (0, 5, 1, 4, 1). Результатом является число $z=\left[\frac{-979}{77}\right]=-13$.

Разработка метода и алгоритма основного модулярного деления

Рассмотренные модели связаны со специальными случаями и неприменимы в ситуации, когда и делимое, и делитель представляют собой произвольные целые числа. Последний случай представлен формой 3.

Различные алгоритмы деления целых чисел $\frac{a}{b}$ можно описать итеративной схемой, используемой так называемый метод спуска Ферма [4]. Конструируется некоторое правило φ, которое каждой

паре целых положительных чисел *a* и *b* ставит в соответствие некоторое целое положительное *q* такое, что *a-bq=r>0*. Тогда деление *a* на *b* осуществляется по следующему правилу: согласно операции φ паре чисел *a* и *b* ставится в соответствие число q_1, такое, что $a - bq_1 = r_1 \geq 0$. Если r_1<b, то деление закончено, если же $r_1 \geq b$, то, согласно φ, паре чисел (r_1, b) ставится в соответствие q_2, такое, что $r_1 - bq_1 = r_1 \geq 0$.

Если ($r_2 < b$), то деление завершается, если же ($r_2 \geq b$), то, согласно φ_1, паре (r_2, *b*) ставится в соответствие q_3 такое, что $r_2 - bq_3 = r_3 \geq 0$ и так далее. Так как последовательное применение операции φ приводит к строго убывающей последовательности положительных целых чисел $a \geq r_1 > r_2 > r_3 > \ldots \geq 0$, то процесс является конечным и алгоритм реализуется за конечное число шагов.

В общем случае *b* может быть и не равным модулю или их произведению. Здесь встает проблема выбора *b* таким, чтобы оно было равным либо модулю, либо их произведению. Если эта проблема будет решена, тогда итерации могут быть сведены к процессу масштабирования, которые рассмотрены выше. Для решения этой проблемы вначале докажем теорему о границах изменения *b*.

Теорема. 2. Если на К-шаге зафиксирован случай $0 \leq r_{k-1} - bq_k = r_k < b$, тогда частное *q* от деления целых чисел *a* на *b* будет равно $\sum_{i=1}^{k} q_1 + r_k'$. Если $r_k < \frac{b}{2}$, то $r_k' = 0$, а если $r_k > \frac{b}{2}$, то $r_k' = 1$.

Доказательство. Докажем для случая, если $\bar{b} = \lambda \cdot b$ при $\lambda = 1, 2, \ldots$. Для доказательства выполним следующую последовательность действий:

$$q_1 = \left[\frac{a}{b}\right] = \frac{1}{\lambda}\left[\frac{a}{b}\right] \text{ при } a = a_0$$

$$a_1 = a_0 - bq_1 = a - b\frac{1}{\lambda}\left[\frac{a}{b}\right] = a - a\frac{1}{\lambda} = a\left(1 - \frac{1}{\lambda}\right);$$

$$q_2 = \left[\frac{a_1}{\bar{b}}\right] = \frac{a_1}{\lambda \cdot b} = \frac{a(1 - \frac{1}{\lambda})}{\lambda \cdot b} = \left[\frac{a}{b}\right]\frac{1}{\lambda}(1 - \frac{1}{\lambda});$$

$$a_2 = a_1 - bq_2 = a\left(1-\frac{1}{\lambda}\right) - b\left[\frac{a}{b}\right]\frac{1}{\lambda}\left(1-\frac{1}{\lambda}\right) = a\left(1-\frac{1}{\lambda}\right) - a\frac{1}{\lambda}\left(1-\frac{1}{\lambda}\right) =$$

$$= a\left(1-\frac{1}{\lambda}\right)\left(a-\frac{1}{\lambda}\right) = a\left(1-\frac{1}{\lambda}\right)^2;$$

$$q_3 = \left[\frac{a_2}{\overline{b}}\right] = \left[\frac{a_{12}}{\lambda \cdot y}\right] = \frac{a(1-\frac{1}{\lambda})^2}{\lambda \cdot b} = \left[\frac{a}{b}\right]\frac{1}{\lambda}(1-\frac{1}{\lambda})^2;$$

$$a_3 = a_2 - bq_3 = a\left(1-\frac{1}{\lambda}\right)^2 - a\left[\frac{a}{b}\right]\frac{1}{\lambda}\left(1-\frac{1}{\lambda}\right)^2 = a\left(1-\frac{1}{\lambda}\right)^2 \cdot \left(1-\frac{1}{\lambda}\right) = a\left(1-\frac{1}{\lambda}\right)^3;$$

$$q_4 = \left[\frac{a_3}{\overline{b}}\right] = \frac{a-(1-\frac{1}{\lambda})^3}{\lambda \cdot b} = \frac{a}{b}\frac{1}{\lambda}(1-\frac{1}{\lambda})^3$$

……

$$q_k = \left[\frac{a}{\overline{b}}\right] \cdot \frac{a}{b}\frac{1}{\lambda}(1-\frac{1}{\lambda})^{k-1};\ q_k = \frac{a_{k-1}}{\overline{b}},\ \overline{b} \le \lambda_{k-1}$$

$$q_1 + q_2 + \ldots + q_k = \left[\frac{a}{b}\right] \cdot \frac{1}{\lambda} + \left[\frac{a}{b}\right] \cdot \frac{1}{\lambda} \cdot \left(1-\frac{1}{\lambda}\right) + \left[\frac{a}{b}\right] \cdot \frac{1}{\lambda} \cdot \left(1-\frac{1}{\lambda}\right)^2 +$$

$$+ \left[\frac{a}{b}\right] \cdot \frac{1}{\lambda} \cdot \left(1-\frac{1}{\lambda}\right)^3 + \ldots + \left[\frac{a}{b}\right] \cdot \frac{1}{\lambda} \cdot \left(1-\frac{1}{\lambda}\right)^{k-1} =$$

$$= \left[\frac{a}{b}\right] \cdot \frac{1}{\lambda} \cdot \left[1 + \left(1-\frac{1}{\lambda}\right) + \left(1-\frac{1}{\lambda}\right)^2 + \ldots + \left(1-\frac{1}{\lambda}\right)^{k-1}\right] =$$

$$= \left[\frac{a}{b}\right] \cdot \left(1 - \left(1-\frac{1}{\lambda}\right)^k\right).$$

Итак, $\sum_{i=1}^{k} q_i = \left[\frac{a}{b}\right] \cdot \left(1 - \left(1-\frac{1}{\lambda}\right)^k\right)$.

Проведенные расчеты на ЭВМ приведены на графике рисунком 1.

Из рисунка 1 видно, что в качестве делителя лучшие характеристики получаются при $\lambda = 1;2$. При $\lambda = 1$ частное представляет собой точное значение, а при $\lambda = 2$ частное при малом числе итераций приближается к точному ее значению. Таким образом, в качестве делителя выбирается величина $b \le \ddot{\bar{b}} < 2b$.

Заметим, что при $\lambda = 1$ сумма $\sum_{i=1}^{k} q_i = \left[\frac{a}{b}\right] = \frac{a}{b}$. Для вычисления частного с точностью 0.9 и выше значение λ целесообразно выбрать равное двум, то есть $b \le \bar{b} < 2b$.

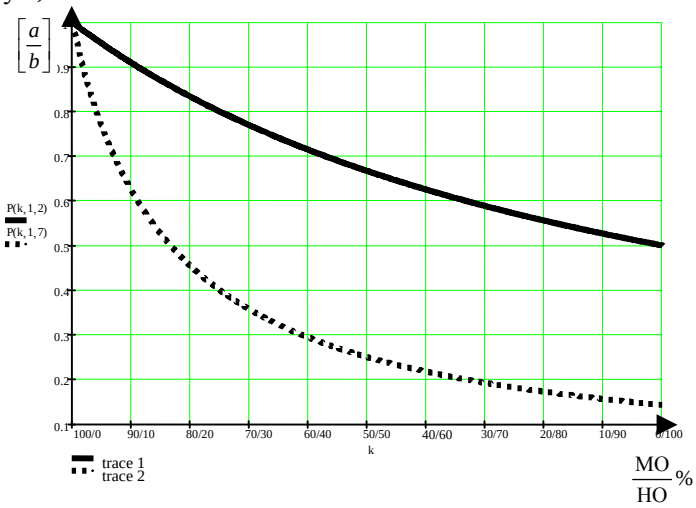


Рисунок 1 – График зависимости значения величины частного от значения величины делителя и числа итераций

Проблема разработки оптимальных вычислительных алгоритмов деления побуждает к разработке таких операций φ_1, которые бы минимизировали число шагов спуска Ферма и вместе с тем достаточно просто реализовывались на заданной вычислительной базе. Кроме того, на способ формирования операции φ существенно влияет также принятая система кодирования числовой информации. Теперь возникает еще одна проблема, каким образом полученный приблизительный делитель $\bar{b}$ свести либо к величине одного модуля или их произведению?

Предлагается модифицированный модулярный алгоритм деления целых чисел на основе метода спуска Ферма, который направлен на использование деления на приблизительный делитель $\overline{b}$, в предположении, что $\overline{b}$ либо целое положительное число попарно простое с p_1, p_2, $\ldots p_n$, либо целое положительное число, представляющее собой произведение чисел, попарно простых с p_1, p_2, $\ldots p_n$. Этот приблизительный делитель выберем из значения делителя, используемого в применении алгоритма масштабирования. Так как в этом случае b не равно $\overline{b}$ ошибка деления будет представлена в частном, которое при выполнении итерации будет уменьшаться до нуля.

Допустим, что и делимое a и делитель b являются положительными числами, и что значение для $\overline{b}$ найдено в соответствии с условием $b \le \overline{b} < 2b$, где $\overline{b}$ – это допустимый делитель для алгоритма масштабирования. Метод нахождения $\overline{b}$, удовлетворяющий этому условию, рассмотрен выше.

В алгоритме деления первым этапом является этап вычисления частного по алгоритму масштабирования, при котором $q_1 = \left[\frac{a}{b}\right]$. Найденный таким образом q_1 далее используется в рекурсивных соотношениях $a_i = a_{i-1} - bq_i, a_0 = a$ и $q_i = \left[\frac{a_{i-1}}{\overline{b}}\right]$ для получения q_2, q_3 и так далее.

Эта повторяющаяся процедура продолжается до тех пор, пока q_i=0, либо до a_i =0.

Если это возникает на r-ом повторении, то $q = \left[\frac{a}{b}\right] = \sum q_i + q'_r$,

где $q' = \begin{cases} q_r, \text{если } q_r \neq 0 \text{ и } a_r = 0; \\ 1, \text{если } q_r = 0 \, u \, a_{r-1} \geq b \text{ для любых } \overline{b} \neq b; . \\ 0, \text{иначе.} \end{cases}$

Действительность этого алгоритма зависит от трех предпосылок:

1. Или q_i, или a_i становится нулевым после последнего числа повторений.

2. Ряд $\prod_{i=0}^{r-1} q_i + q'_r$ должен быть равен $\left[\frac{a}{b}\right]$.

3. Для любого b существует подходящий $\bar{b}$. Причем $\bar{b}$ определяется из условия $b \le \bar{b} < 2b$ и удовлетворяющий условию алгоритма масштабирования.

Таблица 1 – Цифры приблизительного делителя

если $b_i = 0$ для $i \neq k$		если $b_i \neq 0$ для $i \neq k$	
b_p	Q	b_p	Q
1	1	1	2
2	2	2	3
3	3	3	5
4	5	4	5
5	5	5	3·2
6	3·2	6	7
7	5·2	7	5·2
8	5·2	8	5·2
9	5·2	9	5·2
10	5·2	10	11
11	11	11	13
12	13	12	13
13	13	13	7·2
14	7·2	14	5·3
15	5·3	15	17
16	17	16	17
17	17	17	19
18	19	18	19
19	19	19	7·3
20	7·3	20	7·3
21	7·3	21	7·3
22	11·2	22	23

Приблизительный делитель $\bar{b}$ можно найти путем использования наиболее значимой ненулевой цифры, представленного $\bar{b}$ в полиадической системе счисления. Эту ненулевую цифру заменим ближайшим простым числом или произведением простых чисел. Тогда делитель $\bar{b}$ можно представить в виде простого числа или произ-

ведения простых чисел, что позволит использовать для вычисления частного алгоритм масштабирования.

Для определения $\bar{b}$ можно составить таблицу 1 приблизительного делителя.

В таблице 1 приведен список допустимых значений $\bar{b}$ для системы модулей 23, 19, 17, 13, 11, 7, 5, 3, и 2.

Если система модулей СОК выбрана иной, то таблицу 1 можно аппроксимировать.

Пример 5. В остаточной системе, состоящей из модулей 23, 19, 17, 13, 11, 7, 5, 3, и 2 (P=223092870) делим *a*=10304312 на *b*=1401. Округленное частное $q = \left[\frac{a}{b}\right]$.

Решение. Вначале представим *b* в обобщенной позиционной системе счисления в порядке уменьшаемой значимости b_9=0, b_8=0, b_7=0, b_6=0, b_5=0, b_4=0, b_3=3, b_2=3, b_1=21, где b_i определяем из уравнения

$$b = b_9(23\cdot19\cdot17\cdot13\cdot11\cdot7\cdot5\cdot3) + b_8(23\cdot19\cdot17\cdot13\cdot11\cdot7\cdot5) +$$
$$+ b_7(23\cdot19\cdot17\cdot13\cdot11\cdot7) + +b_6(23\cdot19\cdot17\cdot13\cdot11) + b_5(23\cdot19\cdot17\cdot13) +$$
$$+ b_4(23\cdot19\cdot17) + b_3(23\cdot19) + b_2 23 + b_1$$

Используя таблицу 1 с $b_l = b_3$, получаем $\bar{b} = 5\cdot19\cdot23 = 2185$, так как b_l является наиболее значимой ненулевой цифрой обобщенной позиционной системы и определяется выражением $\bar{b} = Q\prod_{i=1}^{k-1} p_i$, где Q дано в таблице 1.

Отсюда: $q_1 = \left[\frac{a}{b}\right] = \left[\frac{10304312}{2185}\right] = 4715$;

$a_1 = a_0 - bq_1 = 10304312 - (1401)\cdot(4715) = 3698597$;

$q_2 = \left[\frac{3698597}{2185}\right] = 1692$;

$a_2 = 3698597 - (1401)\cdot(1692) = 1328105$.

Далее получаем остальные значения a_i и q_i

q_3 =607,	a_3 =477698;
q_4 =218,	a_4 =172280;
q_5 =78,	a_5 =63002;
q_6 =28,	a_6 =23774;
q_7 =10,	a_7 =9764;
q_8 =4,	a_8 =4160;
q_9 =1,	a_9 =2759;
q_{10}=1,	a_{10} =1358;
$q_{11}=\left[\frac{1358}{2185}\right]=0$.	

Так как $q_r = 0$ (то есть $q_{11} = 0$), но $a_{r-1} \geq b, \text{то} q'_r = 0$. Следовательно,

$$q = \sum_{i=1}^{10} q_i = 4715 + 1692 + 607 + 218 + 78 + 28 + 10 + 4 + 1 = 7354 .$$

Полученный результат можно легко проверить обычным делением a =10304312 на b =1401. для вычисления округленного частного потребовалось десять итераций, так как числа были выбраны обдуманно, чтобы получилось много операций. Это происходит в тех случаях, если a – большое число, а b – относительно малое число, а $\bar{b}$ – аппроксимация b.

Модифицируем полученный алгоритм на язык кольцевых операций системы остаточных классов. Для этого рассмотрим следующий пример.

Пример 6. В остаточной системе, состоящей из модулей 7, 5, 3, 2 необходимо разделить число a =201 $\rightarrow$ (5, 1, 0, 1) на число b =8 $\rightarrow$ (1, 3, 2, 0). Округленное частное обозначим как $q = [a/b]$.

Решение. Вначале преобразуем делитель b в ОПСС в порядке уменьшаемой значимости:

$b = b_4(7 \cdot 5 \cdot 3) + b_3(7 \cdot 5) + b_2 7 + b_1$, тогда $b = 0 \cdot (7 \cdot 5 \cdot 3) + 0 \cdot (7 \cdot 5) + 1 \cdot 7 + 1$, где $b_2 = 1, b_1 = 1$.

Используя таблицу 1 с $b_p = b_2$ и $b_i \neq 0$ для $i \neq p$, получим $\bar{b} = Q\prod_{i=1}^{p-1} p_i$, где $Q = 2$ или $\bar{b} = 2 \cdot 7$.

Далее по алгоритму масштабирования, изложенному выше находим $q_1 = \left[\frac{a}{b}\right]$, где $\bar{b}$ – это произведение двух модулей $7 \cdot 2$.

$$q_1 = (0,4,2,0) \to 14.$$

Используя q_1 найдем

$$a_1 = a_0 - bq_1 = (5,1,0,1) - (1,3,2,0 \cdot 0,4,2,0) = (5,4,2,1) \to 89.$$

Далее получаем остальные значения a_i и q_i:

$$q_2 = \left[\frac{a_1}{\bar{b}}\right] = (6,1,0,0) \to 6,\ a_2 = a_1 - bq_1 = (5,4,2,1) - (1,3,2,0 \cdot 6,1,0,0) =$$
$$= (6,1,2,1) \to 41;$$

$$q_3 = \left[\frac{a_2}{\bar{b}}\right] = (2,2,2,0) \to 2,\ a_3 = a_2 - bq_2 = (6,1,2,1) - (1,3,2,0 \cdot 2,2,2,0) =$$
$$= (4,0,1,1) \to 25;$$

$$q_4 = \left[\frac{a_3}{\bar{b}}\right] = (1,1,1,1) \to 1,\ a_4 = (4,0,1,1) - (1,3,2,0 \cdot 1,1,1,1) = (3,2,2,1) \to 17;$$

$$q_5 = \left[\frac{a_4}{\bar{b}}\right] = (1,1,1,1),\ a_5 = (3,2,2,1) - (1,3,2,0 \cdot 1,1,1,1) = (2,4,0,1) \to 9;$$

$$q_6 = \left[\frac{a_5}{\bar{b}}\right] = (1,1,1,1) \to 1,\ a_6 = (2,4,0,1) - (1,3,2,0 \cdot 1,1,1,1) = (1,1,1,1).$$

Так как $a_5 > \frac{b}{2}$, то $q_6 = 1$. Следовательно,

$$q = \sum_{i=1}^{b} q_i = (0,4,2,0) + (6,1,0,0) + (2,2,2,0) + (1,1,1,1) + (1,1,1,1) + (1,1,1,1) =$$
$$= (4,0,1,1) \to 25.$$

Действительно $[a/b] = [201/8] = 25$.

Выводы

1. Показано, что деление в модулярной арифметике имеет три формы: деление с нулевым остатком, округление, масштабирование и основное деление.

2. Разработаны алгоритмы масштабирования чисел с одинаковыми и разными знаками, которые состоят из операции деления и расширения базы и реализуются с помощью модульных вычислений. Показано, что при использовании разработанного метода при расширении чисел выигрыш достигает $(n-1)$ раза.

3. Доказана теорема о модульном выполнении значения частного в случае, если делимое и делитель являются произвольными числами.

4. Разработан итерационный модулярный метод общего деления на основе модификации метода спуска Ферма, исходными данными которого являются произвольные значения делимого и делителя. При этом приблизительный делитель выбирается равным простому числу или их произведению, который в дальнейшем используется в итерациях получения промежуточных и окончательного значений частного. Представленная в частном ошибка при выполнении итераций уменьшается до нуля. Если допустимая ошибка задана не выше 0.1, то достаточно провести всего четыре итерации.

5. Определено правило выбора приблизительного делителя на основе свойств обобщенной позиционной системы счисления, которое позволяет делитель представить в виде простого числа или произведений простых чисел, на основе которых происходит округление делимого с целью выполнения алгоритма деления с нулевым остатком.

Литература

1. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. – М.: Советское радио, 1968. – 440 с.
2. Srabo N., Tanaka R. Residue arithmetic and its applications to computer technology. – New-York, 1967.
3. Червяков Н.И., Мезенцева О.С., Лавриненко И.Н., Сивоплясов Д.В. Метод расширения динамического диапазона модулярного нейрокомпьютера // Нейрокомпьютеры: разработка, применение. – 2005. – № 7. – С. 64-69.
4. Амербаев В.М. Теоретические основы машинной арифметики. – Алма-Ата: Наука, 1976. – 324 с.

УДК. 681.3

Осепянц О.А., Исмаилов Ш-М.А

Методика генерации оптимального основания для представления чисел в системе остаточных классов

(*Дагестанский государственный технический университет, кафедра вычислительной техники*)

При решении практических задач на ЭВМ часто необходимо делать выбор между быстродействием и затратами памяти. В статье рассматривается разработанная авторами методика выбора модулей системы остаточных классов, обеспечивающих оптимальное соотношение быстродействие/затраты памяти для некоторых алгоритмов машинной арифметики системы остаточных классов.

Одной из важнейших задач при создании программного обеспечения (*далее – ПО*) является нахождение компромисса между быстродействием и затратами аппаратных ресурсов, причём решение этой задачи существенно зависит от конкретных условий применения. Замена позиционных систем счисления (*далее – СС*) на систему остаточных классов (*СОК*) позволяет оперативно оптимизировать эти параметры в каждом конкретном случае путём подбора векторного основания СС. То есть при создании ПО программист может вводить в систему процедуру оптимизации модулей, которая

позволит автоматически (незаметно для пользователя) либо через системные настройки выбирать оптимальное для каждого конкретного случая соотношение между быстродействием и затратами ресурсов. Более того, такой алгоритм может быть встроен в операционную систему и функционировать незаметно как для пользователя, так и для программиста.

1. Анализ алгоритмов модульной арифметики

Рассмотрим следующий алгоритм сложения, наиболее простой при аппаратной реализации.

1) *В счётчик A_m (имеется в виду: счётчик A по модулю m, то есть производящий подсчёт в диапазоне от 0 до m-1) заносится число a;*
2) *В двоичный счётчик B заносится b;*
3) *Пока $B > 0$ делать:*
   a) *Увеличить значение A_m на 1,*
   b) *Уменьшить значение B на 1,*
4) *Конец (в A_m – сумма чисел).*

Данный алгоритм предельно детализирован (1 команда = 1 такт процессора) с целью упростить дальнейший анализ.

Проведём анализ этого алгоритма согласно [2][1].

Прежде всего определим параметр, по которому будем оценивать сложности алгоритма – размер исходной задачи. Как уже упоминалось, длительность вычислений определяется значениями исходных чисел. Поэтому в качестве размера задачи следует использовать значение модуля m как меры максимального представимого числа. В общем случае при применении векторного основания $\beta = [m_1, m_2, ..., m_n]$ и параллельном вычислении по всем модулям в качестве размера задачи берётся $m_{\max}$ – максимальный из модулей.

[1] Следует учитывать, что все последующие рассуждения и выкладки приведены применительно к описанным здесь алгоритмам. Для других, отличных от них алгоритмов анализ проводится аналогично (методика описана в [2]).

Итак, при нахождении временной сложности $g_t^+(m)$ в приведённом алгоритме будет доминировать цикл *3)*, при чём, поскольку при практической реализации шаг *a)* может выполняться параллельно с шагом *b)*, то каждый этап цикла потребует один такт процессора. Всего же цикл повторится *b* раз (в предельном случае – m_{max} раз). Таким образом, временную сложность этого алгоритма можно записать в виде:

$$g_t^+(m) = O(m), \qquad (1.1)$$

(читается: временная сложность порядка *m*).

Поскольку при выполнении этого и всех последующих алгоритмов память расходуется только на хранение исходных чисел и результата, оценка емкостной сложности будет приведена ниже для всех алгоритмов сразу.

Вычитание. Рассмотрим алгоритм вычитания.

1) В счётчик A_m заносится b;

2) В счётчик B заносится m;

3) Пока A_m>0 делать:

a) Уменьшить значение B на 1,

b) Уменьшить значение A на 1,

4) В счётчик A_m заносится a; //На этом этапе в счётчике B находится значение <u>b</u>

5) Пока $B > 0$ делать:

a) Увеличить значение A_m на 1,

b) Уменьшить значение B на 1,

6) Конец (в A_m – разность чисел).

Вторая часть алгоритма полностью соответствует вышеприведён-

ному алгоритму сложения.

В данном алгоритме присутствует два цикла. Один этап каждого из них также выполняется за 1 такт процессора. Однако следует заметить, что цикл *3)* выполнится *b* раз, а цикл *5) – (m-b)* раз. Таким образом, при любом *b* циклы *3)* и *5)* в сумме выполнятся *m* раз и, следовательно, временная сложность $g_t^-(m)$ этого алгоритма также определяется выражением (1.1)

Умножение $a \cdot b$ сводится к вычислению $|ab|_m$. При реализации умножения на счётчиках, оно также сводится к сложению: $|ab|_m = \left|\sum_{i=1}^{b} |a|_m\right|_m$. При b=0 соответственно и $|ab|_m = 0$.

Сформулируем этот алгоритм:

1) *В счётчик A_m заносится значение* 0*;*
2) *В счётчик B заносится b;*
3) *Пока* $B > 0$ *делать:*
   a) *В счётчик C заносится число a,*
   b) *Пока* $C > 0$ *делать:*
      i) *Увеличить значение A_m на 1,*
      ii) *Уменьшить значение C на 1,*
   c) *Уменьшить значение B на 1,*
4) *Конец (в A_m – произведение чисел).*

Как видно из алгоритма, для операции умножения потребуется третий счётчик. Дополнительное удобство заключается в том, что результат всех трёх операций (сложение, вычитание и умножение) получается в одном и том же счётчике A_m, что также значительно упрощает аппаратную реализацию.

В данном алгоритме имеет место вложенный цикл. Цикл *3)* состоит из 3-х этапов, первый и третий из которых выполняются за 1 такт процессора, а второй (вложенный цикл) – за *a* тактов. Сам же цикл 3) выполнится *b* раз. Значит, общее время выполнения цикла 3) составит $b(a+2)$, а в предельном случае - $m \times (m+2) \approx m^2$. Отсю-

да временная сложность алгоритма умножения:

$$g_t^{\times}(m) = O(m^2) \qquad (1.2)$$

Деления. Операция деления сводится к нахождению произведения делимого на мультипликативный обратный делителя по модулю системы остаточных классов Для такого деления наиболее эффективным является матричных метод. Алгоритм такого деления практически совпадёт с вышеприведённым алгоритмом умножения за исключением строки 2, где в счётчик *B* вместо числа *b* заносится полученное с помощью ПЗУ значение b^{-1}.

Таким образом, временная сложность $g_t^{:}(m_{\max})$ деления также определяется из (1.2)

Из всего вышесказанного следует, что при уменьшении m_{max} *(а добиться этого можно увеличивая количество модулей)* происходит увеличение быстродействия.

Найдём теперь емкостную сложность вышеприведённых алгоритмов.

Поскольку, как отмечалось ранее, при выполнении этих алгоритмов память расходуется только на хранение исходных данных и результата, следовательно, емкостная сложность определяется исключительно формой представления чисел, то есть векторным основанием $\beta = [m_1, m_2, ..., m_n]$. Предположим, что для хранения всех разрядов представления числа используются ячейки памяти одинакового размера. В этом случае целесообразно использовать равномерный весовой критерий. Тогда емкостная сложность алгоритмов окажется пропорциональной числу модулей и составит:

$$g_{mem}(m_{\max}) = O(n) \qquad (1.3)$$

2. Основные требования к векторному основанию

Пусть $\{x_1, x_2, ..., x_l\}$ – множество входных данных; $\{y_1, y_2, ..., y_k\}$ – множество выходных данных вычислительного процесса. Тогда сам вычислительный процесс можно представить в следующем виде:

$$\begin{cases} y_1 = f_1(x_1, x_2, ..., x_l) \\ y_2 = f_2(x_1, x_2, ..., x_l) \\ \quad .... \\ y_k = f_k(x_1, x_2, ..., x_l) \end{cases}, \text{ или } \vec{Y} = F(\vec{X}) \qquad (2.1)$$

Векторное основание $\beta = [m_1, m_2, ..., m_n]$ для проведения вычислений (2.1) должно обеспечивать однозначное представление всех входных и выходных данных. Как показано в [1], промежуточные данные на выполнение этого условия контролировать не обязательно.

Пусть $W : x_i \in W, y_j \in W, i = \overline{1..l}, j = \overline{1..k}$ – область значений входных и выходных данных, w_{max} – максимальный по модулю элемент из W. Перечислим требования, выполнение которых позволит однозначно представить все элементы W по модулю $\beta = [m_1, m_2, ..., m_n]$:

1. Для обеспечения однозначного соответствия любого $w_i \in W$ и его представления $|w_i|_\beta$, согласно [1], необходимо выполнение следующего условия:

$$(m_i, m_j) = 1, \quad i \neq j \qquad (2.2)$$

Здесь (m_i,m_j) – наибольший общий делитель чисел m_i и m_j

2. Для выполнения операции деления a/b необходимо существование элемента, обратного делителю по векторному основанию β, т.е. $b^{-1}(\beta)$. Согласно [1], необходимым и достаточным условием этого является:

$$b \neq 0, \quad (b, m_i) = 1, \ i = \overline{1..n} \qquad (2.3)$$

Следовательно, выбор в качестве $m_1, m_2, \ldots, m_n$ простых чисел гарантирует выполнение условий 1 и 2.

3. Для исключения псевдопереполнения при представлении исходных и конечных данных, согласно [3], должно быть выполнено следующее условие:

$$\begin{cases} M > w_{\max} & \text{при } W \subset I_+ \\ M > 2w_{\max} & \text{при } W \subset I \\ \left.\begin{array}{l} M \geq 2w_{\max}^2 + 1 \\ M \geq 2/\varepsilon^2 + 1 \end{array}\right\} & \text{при } W \subset Q \end{cases} \qquad (2.4)$$

Здесь:

$$M = \prod_{i=1}^{n} m_i \ ;$$

ε – точность представления рациональных чисел

Q – множество рациональных чисел

I – множество целых чисел

I_+ – множество целых неотрицательных чисел

Для дальнейшего удобства перепишем (2.4) в следующем виде:

$$M \geq M_0, \text{ где } M_0 = \begin{cases} w_{\max} & \text{при } W \subset I_+ \\ 2w_{\max} & \text{при } W \subset I \\ \max\left(2w_{\max}^2 + 1,\ 2/\varepsilon^2 + 1\right) & \text{при } W \subset Q \end{cases} \qquad (2.5)$$

Запись max(…) *означает максимальное из перечисленных значений.*

Таким образом алгоритм нахождения векторного основания β сводится к генерации последовательности простых чисел $m_1 < m_2 < ... < m_n$, произведение которых больше или равно M_0.

3. Задача оптимизации векторного основания

Оптимизация основания производится по максимуму быстродействия и по минимуму расхода памяти. Но поскольку алгоритмы сложения, вычитания, умножения и деления имеют разные временные сложности, а при решении различных задач могут доминировать любые из них, следовательно целевая функция должна учитывать их реальное соотношение в решаемой задаче. Сведя задачу максимума быстродействия к задаче минимума временных затрат, полу-

чим целевую функцию следующего вида:

$$Z = k_+ a_+ + k_- a_- + k_\times a_\times + k_\div a_\div + k_{mem} a_{mem} \qquad (3.1)$$

Здесь: k_+, k_-, $k_\times$, $k_\div$ *– весовые коэффициенты операций сложения, вычитания, умножения и деления, зависящие от среднего числа соответствующих операций, выполняемых в ходе решения задачи;*

k_{mem} *– весовой коэффициент затрат памяти;*

a_+, a_-, $a_\times$, $a_\div$ *– составляющие, учитывающие временные затраты на операции сложения, вычитания, умножения и деления;*

a_{mem} *– составляющая, учитывающая затраты памяти;*

Таким образом, варьируя значения весовых коэффициентов в зависимости от конкретных условий и желаемого результата, а затем решая задачу минимума *Z*, можно получить искомое оптимальное основание.

Подставив в (3.1) выражения (1.1), (1.2) и (1.3) для временной и емкостной сложностей алгоритмов, получим:

$$Z = k_+ m_{\max} + k_- m_{\max} + k_\times m_{\max}^2 + k_\div m_{\max}^2 + k_{mem} n \qquad (3.2)$$

где $m_{\max}$ *– максимальный модуль основания* $\beta = [m_1, m_2, ..., m_n]$ *(в случае одномодульного основания вместо* $m_{\max}$ *следует использовать m.)*

4. Генерация векторного основания

С учётом вышесказанного, генерация векторного основания при известном M_0 сводится к нахождению такого $M \geq M_0$, при разложении которого на простые множители $m_1 < m_2 < ... < m_n$ выражение (3.2) принимает минимальное значение. Сделать это можно методом пассивного поиска, последовательно разлагая на множители числа M_0, M_0+1, M_0+2,…, найти такое *M*, которое удовлетворяет вышеназванному условию.

Следствие: составное *M* не разделится на составное *b*, если до этого *M* было максимально возможное число раз разделено на все простые делители числа *b*.

Таким образом, если последовательно максимально возможное число раз делить M на натуральные числа в порядке возрастания, начиная с числа 2, то на момент деления на составное b, M уже окажется разделённым на все простые делители числа b. То есть, согласно теореме 4.1, b не войдёт в число делителей M, так как деление произойдёт с остатком, причём такое разложение будет состоять не более, чем из $r + M - 1$ операций деления (r – *количество простых делителей числа* M).

Этот метод можно ускорить, уменьшив число операций деления следующим образом: согласно [4], если M не имеет делителей в диапазоне $2...\sqrt{M}$, то M – простое число и, следовательно, не имеет нетривиальных делителей. В этом случае число операций по разложению M на простые делители не превысит $r + \sqrt{M} - 1$.

Поскольку решено использовать метод пассивного поиска, следует определить границы поиска для M: нижняя граница – это, как уже упоминалось, значение M_0 из (2.5), а верхняя граница должна определяться аппаратными возможностями, то есть максимально возможным значением модулей m_i и их максимальным числом. Тогда верхняя граница поиска будет определена по формуле:

$$M_{\max} = (m_{i\max})^{n_{\max}} \qquad (4.1)$$

где: $n_{\max}$ – *максимально допустимое в системе число модулей;*

$m_{i\max}$ – *максимально допустимое в системе значение модуля.*

Однако в этом случае число итераций неоправданно возрастёт (например, для $n_{\max}$=16 и $m_{i\max}$=255, число итераций составит: 3×10^{38}), что потребует нереально больших временных затрат. Практика же показала, что в большинстве случаев оказывается достаточным гораздо меньшее число итераций. Проводя эксперименты над описанным здесь алгоритмом в системе программирования Delphi 6 (в диапазоне значений M_0 от 10 до 10^9), я предпочёл выбрать более гибкий вариант нахождения $M_{\max}$:

$$M_{\max} = 300 \times 2^{\lg M} \qquad (4.2)$$

Программа с использованием (4.2) вполне удовлетворяла требованиям, предъявляемым к экспериментальным программам в указанном диапазоне значений, однако эта формула не является опти-

мальной. Оптимизировав значение $M_{\max}$, мы получим существенное увеличение быстродействия.

В процессе поиска каждое найденное разложение следует подвергать проверке на выполнение условия (4.3), вытекающего из сформулированных выше требований к основанию СОК:

$$\begin{cases} n \le n_{\max} \\ m_i \le m_{i\max}, i = 1..n, \\ m_1 < m_2 < ... < m_n \end{cases} \qquad (4.3)$$

Невыполнение этого условия означает, что данное разложение нельзя использовать в качестве основания.

Итак, теперь мы готовы сформулировать окончательный алгоритм генерации оснований.

5. Алгоритм генерации оптимального основания СОК

Входные данные:

$w_{\max}$ – максимальное по модулю значение из области определения входных и выходных данных решаемой задачи;

ε – точность (для рациональных чисел). Условимся, что при ε=0 мы имеем дело с множеством целых неотрицательных чисел, при ε=-1 – с множеством целых чисел, а при $0<\varepsilon<1$ – с множеством рациональных чисел;

k_+, k_-, $k_\times$, $k_\div$, k_{mem} – весовые коэффициенты операций сложения, вычитания, умножения и деления в решаемой задаче, а также весовой коэффициент затрат памяти – задаются исходя из специфики решаемой задачи и имеющихся аппаратных ресурсов;

$n_{\max}$ – максимально возможное в системе число модулей;

$m_{i\max}$ – максимально возможное в системе значение каждого модуля.

Выходные данные: $\beta = [m_1, m_2, ..., m_n]$ – векторное основание СОК.

Алгоритм:

1) Ввод значений $w_{\max}$, ε, k_+, k_-, $k_\times$, $k_\div$, k_{mem}, $n_{\max}$, $m_{i\max}$.

2) $$M_0 = \begin{cases} w_{\max} & \text{при } \varepsilon = 0 \\ 2w_{\max} & \text{при } \varepsilon = -1 \\ \max\left(2w_{\max}^2 + 1,\ {}^{2}\!/_{\varepsilon^2} + 1\right) & \text{при } 0 < \varepsilon < 1 \end{cases}$$

//Нижняя граница поиска

3) $M_{\max} = 300 \times 2^{\lg M_0}$ *//Верхняя граница поиска*

4) $M=M_0$, $M_X=M$, $Z=$ любое достаточно большое число

5) $x = M_x$, $\beta = [\ ]$, $n = 0$ *//Инициализация переменных*

6) Для $i = \overline{2..M_x}$ делать:*//Разложение M_x на простые множители*

   a) Если $i > \sqrt{M_x}$ и $n = 0$ то перейти к п. 7

   //Если M_x – простое

   b) Если $x = 1$ то перейти к п.8

   //Исключение лишних итераций

   c) $y = {}^{x}\!/_{i}$ *//Пробное деление*

   d) Если y не имеет дробной части, то:

      i) $x = y$ *//Запоминание результата деление*

      ii) $n = n + 1$

      iii) $\beta_n = i$ *//Запоминание найденного делителя*

      iv) Перейти к п. 6b

7) Если $n = 0$ то: //Если M_x – простое…

   a) $n = n + 1$

   b) $\beta_n = M_x$

8) $m_{\max} = \max(\beta)$ *//Выделение наибольшего делителя*

9) Если $n \le n_{\max}$, $m_{\max} \le m_{i\max}$ и $\beta_j \ne \beta_l$ $(j \ne l)$ то:

a) Если $Z > k_+ m_{\max} + k\_ m_{\max} + k_\times m_{\max}^2 + k_\div m_{\max}^2 + k_{mem} n$ то: *//Сравнение значений целевой функции*

i) $Z = k_+ m_{\max} + k\_ m_{\max} + k_\times m_{\max}^2 + k_\div m_{\max}^2 + k_{mem} n$

//Запоминание значения целевой ф-и

ii) $M = M_x$ *//Запоминание промежуточн. значения M*

10) $M_x = M_x + 1$

11) Если $M_x \le M_{\max}$ то перейти к п. 5

//Проверка следующего M

12) $x = M$, $\beta = [\ ]$, $n = 0$ *//Инициализация переменных*

13) Для $i = \overline{2..M}$ делать: *//Разложение M на простые множители*

a) Если $i > \sqrt{M}$ и $n = 0$ то перейти к п. 14

//Если M_x – простое

b) Если $x = 1$ то перейти к п.8

//Исключение лишних вычислений

c) $y = x / i$ *//Пробное деление*

d) Если *y* не имеет дробной части, то:

i) $x = y$ *//Запоминание результата деления*

ii) $n = n + 1$

iii) $\beta_n = i$ *//Запоминание найденного модуля*

iv) Перейти к п. 13b

14) Если $n = 0$ то: //Если *M* – простое…

a) $n = n + 1$

b) $\beta_n = M$

15) Вывод $\beta = [m_1, m_2, ..., m_n]$

16) Конец

***Примечание**: в случае применения алгоритмов модульной арифметики, отличных от описанных здесь (см. «Анализ алгоритмов модульной арифметики»), целевую функцию (3.2) следует находить исходя из их анализа аналогично описанной здесь методике. В остальном же приведённый алгоритм пригоден для любых вариантов реализации операций модульной арифметики.*

6. Анализ алгоритма и исключение избыточных итераций

Главным недостатком полученного алгоритма следует считать резкое увеличение числа итераций при больших значениях M_0, согласно (4.2), что в совокупности с достаточно медленной процедурой разложения на простые множители даёт резкое снижение быстродействия. Частично справиться с этим недостатком можно уменьшив число итераций до минимально необходимого, но обеспечивающего нахождение минимума целевой функции.

Итак, проанализируем зависимость $Z(M)$.

На рисунке 6.1 представлен график функции $Z(M)$ (для $k_+ = 1$, $k_- = 1$, $k_\times = 1$, $k_\div = 1$, $k_{mem} = 400$, $n_{\max} = 16$, $m_{i\max} = 255$), из которого видно, что данная зависимость имеет хаотический характер, определяемый формой разложений M на простые множители, то есть параметрами $m_{\max}$ и n. Не смотря на это, в области максимумов наблюдается чёткая зависимость, определяемая формой целевой функции и значениями её коэффициентов. Однако, поскольку в нашем случае решается задача минимума, проанализируем область минимумов данной функции, для чего программным путём построим для её графика огибающую по минимальным значениям. Для этого для каждого M найдём минимальное значение целевой функции на интервале $[M - \Delta M; M + \Delta M]$, а полученные минимумы соединим на графике отрезками.

С увеличением M значения функции в минимумах также возраста-

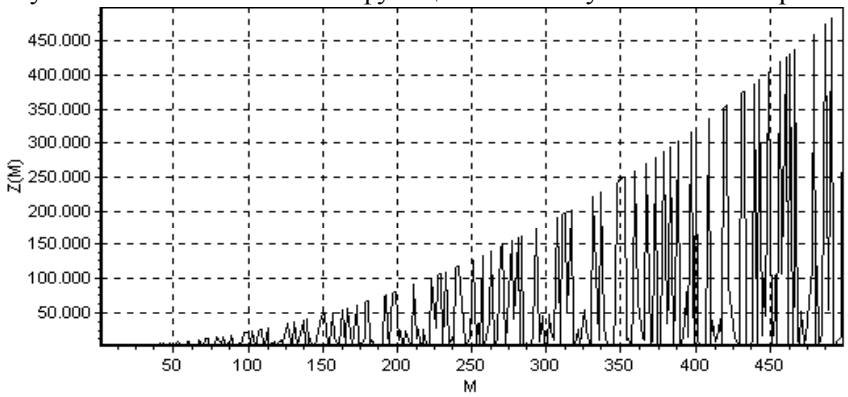


Рисунок 6.1.

ют. Аналогичный результат был получен и для других значений коэффициентов. Следовательно, на любом интервале [M_0; M_{max}] минимум функции $Z(M)$, вероятнее всего, будет находиться ближе к нижней границе интервала. Однако полученный алгоритм проходит весь интервал [M_0; M_{max}] от начала до конца и, следовательно, выполняет излишний объём вычислений.

Наиболее очевидным решением здесь является выбор значения M_{max}, исключающего лишние этапы вычислений. Однако с уменьшением значения M_{max} повышается вероятность выхода минимума целевой функции за границы интервала поиска. Существует и другой вариант, более гибкий: ввести в основной цикл алгоритма проверку условия, был ли уже найден минимум. Если минимум найден → выйти из цикла. Но здесь возникает другая проблема: как определить момент нахождения минимума?

Как видно из рис. 6.1, значение функции $Z(M)$ изменяется скачкообразно. Так, согласно графику, на интервале [100; 200] её производная меняет знак 39 раз, причём на разных участках графика для интервалов одинаковой ширины это значение в среднем остаётся постоянным. Таким образом, учитывая вышесказанное относительно огибающей ф-и $Z(M)$, можно утверждать, что если за предыдущие ΔM итераций новый минимум не найден, то он не будет найден и при последующих итерациях и, следовательно, процесс поиска можно завершить. В этом случае произойдёт ровно столько итераций, сколько требуется для отыскания минимума. То-

гда роль параметра M_{max} сведётся к ограничению максимального числа итераций и, следовательно, его значение следует выбирать с запасом, чтобы исключить выход минимума функции за пределы интервала. На быстродействие это не повлияет, так как реальное число итераций теперь не будет зависеть от M_{max}.

Для реализации этого метода в Алгоритм генерации оптимального основания СОК следует внести следующие изменения:

1) Вводятся две дополнительные переменные: *C* (счётчик нерезультативных итераций), *ΔM* (максимально допустимое число следующих подряд нерезультативных итераций)

2) В п. 5 вводится инициализация переменных *C* и *ΔM*:

C=0; *ΔM =заданное по умолчанию или введённое пользователем значение;*

3) Добавить П. 9.a.iii:

C=0; *//если итерация результативна, то сбросить счётчик*

4) Между пп. 8 и 9 добавить п. 8.1:

C=*C*+1; *//если итерация нерезультативна – инкремент счётчика*

5) Между пп. 10 и 11 добавить п. 10.1:

Если *C*> *ΔM*, перейти к п. 12

6) В п. 3 задать для M_{max} достаточно большое значение, гарантирующее попадание минимума целевой функции в интервал поиска.

В предельном случае можно совсем исключить из алгоритма операции с M_{max}, заменив в п. 11 проверку условия $M_x \le M_{max}$ на безусловный переход. В этом случае алгоритм упростится, однако максимальное число итераций не будет иметь ограничений, что может повлечь за собой ошибки переполнения разрядной сетки ЭВМ.

Для испытания полученного алгоритма в среде Borland Delphi 6 была создана программа, реализующая его. Испытания проводились на ЭВМ со следующими параметрами:

Таблица 6.1

Параметр	**Значение**
Центральный процессор	Intel Celeron 600
Материнская плата	Zida T810
Тактовая частота	600 МГц
Объём оперативной памяти	64 МБ
Операционная система	Microsoft Windows 98

При входных данных: $\varepsilon = 0$, $k_{+} = 1$, $k_{-} = 1$, $k_{\times} = 1$, $k_{\div} = 1$, $k_{mem} = 400$, $n_{\max} = 16$, $m_{i\max} = 255$, были получены следующие значения длительностей вычислительного процесса (таблица 6.2):

Как видно из таблицы 6.2, в результате принятых мер быстродействие существенно возросло. Однако при $w_{\max} > 10^5$ в обоих случаях быстродействие алгоритма резко падает, что является ограничивающим фактором в практическом применении алгоритма.

В случае встраивания данного алгоритма в операционную систему возможны случаи, когда процесс оптимизации основания потребует гораздо больше машинного времени, чем сами вычисления. Выходом из положения является хранение в памяти ЭВМ заранее сгенерированного с помощью данного алгоритма массива оптимизированных оснований для различных фиксированных значений весовых коэффициентов. Работа генератора оснований в этом случае сводится к выбору наиболее подходящего основания из списка. Также в систему может быть включена реализация данного алгоритма, ориентированная на решение задач малого размера.

Таблица 6.2

$w_{\max}$	**t, мс**	
	без исключения избыточных итераций (M_{max} согласно (4.2))	*с исключением избыточных итераций (ΔM=100)*
10	0	0
10^2	0	0
10^3	0	0
10^4	50	30
10^5	2.360	1.590

10^6	26.920	19.440
10^7	373.430	316.540

Литература:

1. Грегори Р., Кришнамурти Е. Безошибочные вычисления. Методы и приложения. – М.: «Мир», 1988
2. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979
3. Осепянц О.А. Проблема выбора модулей при проведении вычислений в системе остаточных классов. Сборник «Современные информационные технологии в управлении. Всероссийская научно-техническая конференция». – Махачкала, 2003 г. С. 87-90.
4. Оре О. Приглашение в теорию чисел. – М.: Наука, 1980
5. Виноградов И.М. Основы теории чисел. – М.: Наука, 1981

УДК 621.29

Ш.А. Оцоков

Об ускорении операции сложения чисел с плавающей точкой на основе модулярной арифметики

(Московский энергетический институт)

Представлены разработанные методы для сдвига мантиссы и определения ошибки переполнения при сложении чисел с плавающей точкой в модулярной системе счисления. Разработана структурная схема устройства для сдвига мантиссы в модулярной системе счисления и получены оценки его временных затрат. Получены оценки эффективности применения модулярной арифметики для сложения чисел с плавающей точкой.

В модулярной системе счисления (МСС) любое целое число представляется в виде набора остатков от деления на выбранные модули. Арифметические операции распараллеливаются по каждому модулю и высокое быстродействие достигается тем, что они проводятся с операндами, малыми по величине, меньшими выбранных модулей. Диапазон представления целых чисел в модулярной системе счисления определяется произведением ее модулей [1].

Переход к модулярной арифметике порождает и множество проблем, среди которых можно выделить: сложность округления чисел деление, реализация арифметических операций с плавающей

точкой и др.

Данная работа посвящена проблеме ускорения операции суммирования чисел, представленных в формате с плавающей точкой в модулярной арифметике и опирается на проведенные ранее исследования в этой области [1].

Пусть представление числа $A = (E, F)$ с плавающей точкой имеет вид:

Порядок F	Мантисса E

где E - мантисса, F - порядок числа.

Пусть задана модулярная система счисления с взаимно простыми модулями: $m_1, m_2, ..., m_n$.

Тогда

$$E = (\alpha_1, \alpha_2, ..., \alpha_n),$$

где $\alpha_1 = E \bmod m_1, ..., \alpha_n = E \bmod m_n$

Соответственно представление числа A в модулярной системе счисления имеет вид:

Порядок F	E mod m_1	E mod m_2	...	E mod m_n

Правило сложения чисел с плавающей точкой в позиционной системе счисления, в соответствии со стандартом «IEEE Standard 754 Floating Point Numbers» включает в себя следующую последовательность действий [2]:

1. Выбрать число с меньшим порядком и сдвинуть его мантиссу вправо на количество разрядов, равное разности порядков.
2. Установить порядок результата равным большему порядку операндов.
3. Выполнить сложение/вычитание мантисс и определить знак результата.

4. Нормализовать результат в случае необходимости.

В данной статье рассматривается сложение ненормализованных чисел, поэтому выполняются только первые три действия.

Первое действие в модулярной системе счисления выполняется так же как и в позиционной системе счисления. Рассмотрим второе и третье действие.

2. Операция сдвига вправо на s разрядов.

Обозначим число, определяемое последними s - разрядами числа A через Q . Тогда после сдвига на s - разрядов получим число As , определяемое в десятичной системе счисления следующим образом:

$$As = \frac{A - Q}{10^{s}}$$

или в модулярной системе счисления по модулям

$\beta = (m_1, m_2, \ldots, m_n)$

$D = (d_1, \ldots, d_n)$, где

$$d_1 = (A - Q \bmod m_1) \bmod m_1, \ldots, d_n = (A - Q \bmod m_n) \bmod m_n$$

$$As \bmod \beta = \left(\left(10^{-s} \cdot d_1\right) \bmod m_1, \ldots, \left(10^{-s} \cdot d_n\right) \bmod m_n \right) \qquad (1)$$

Рассмотрим способ определения значения Q .

Формула для преобразования числа из модулярной системы в позиционную имеет следующий вид [1]:

$$A \equiv \sum_{i=1}^{n} \alpha_i \cdot B_i \pmod{M},$$

где

$$M = \left(\prod_{j=1}^{n} m_j \right),$$

$B_i = \left(\left(\frac{M}{m_i} \right)^{-1} \bmod m_i \right) \cdot \left(\frac{M}{m_i} \right)$ - ортогональные базисы МСС

или, что то же самое,

$$A = \sum_{i=1}^{n} \alpha_i \cdot B_i - r_A \cdot M \ , \qquad (2)$$

где r_A – ранг, значение которого варьируется в диапазоне от нуля до n.

Известен метод последовательных сложений для определения ранга числа за n-итераций без перехода к позиционному представлению [1].

Тогда из (2) следует, что число Q , определяемое последними s-разрядами числа A, может быть найдено следующим образом:

$$Q = \sum_{i=1}^{n} \alpha_i \cdot \left(B_i \bmod 10^{s+1} \right) - r_A \cdot (M \bmod 10^s) \qquad (3)$$

В формуле (3) используется $\bmod \ 10^{s+1}$ для того, чтобы получить значение Q , большее нуля.

Переходя к определению способа реализации сдвига мантиссы на s-разрядов, рассмотрим возможные принципы организации устройства сдвига и оценим его временные затраты.

Устройство для сдвига мантиссы состоит из n блоков (одномодульных устройств сдвига), каждый из которых выполняет одни и те же функции для своего модуля модулярной системы.

В целях повышения быстродействия используется постоянное запоминающее устройство для хранения значений $d_{ij} = \left(\alpha_i \cdot \left(B_i \cdot \bmod 10^{s+1} \right) \right) \bmod m_j, \ 0 < i \le n, \ 0 < j \le n$.

В каждом j - м блоке устройства каждая пара d_{ij} параллельно суммируются с помощью $n - 1$ сумматоров, одновременно вычисляется ранг числа, параллельно от результата суммирования вычитаются значения кратные модулям, конечным результатом Q является значение разности по номеру соответствующее значению

ранга. Затем вычисляется значение As в соответствии с формулой (1).

Оценки временных затрат в условных тактах гипотетического устройства определяются выражением $n + 1$, где n -число модулей.

Условный такт - это время считывания данных из ПЗУ или сложения 2-х одноразрядных чисел.

Пусть p -разрядность мантиссы, тогда $s \le p$.

Формула для оценки временных затрат в условных тактах принимает следующий вид:

$$\text{Тсдвига} = (n + 1) * \left\lceil \frac{s}{\lceil \log_2 (\max\ m_i) \rceil} \right\rceil \qquad (4)$$

В частном случае, когда $s = p$, то

$$\text{Тсдвига} = (n + 1) \cdot n$$

3. Операция сложения мантисс.

Сложение мантисс производится по правилам модулярной арифметики, т.е.

$$E_1 = (E_1 \bmod m_1, E_1 \bmod m_2, ..., E_1 \bmod m_n) = (u_1, u_2, ..., u_n)$$
$$E_2 = (E_2 \bmod m_2, E_2 \bmod m_2, ..., E_2 \bmod m_n) = (v_1, v_2, ..., v_n)$$
$$E_1 + E_2 = ((u_1 + v_1) \bmod m_1, (u_2 + v_2) \bmod m_2, ..., (u_n + v_n) \bmod m_n)$$

После сложения необходимо произвести проверку на переполнение, т.е выхода результаты за пределы допустимого диапазона представления мантиссы.

Известен метод нулевизации, позволяющий определить номер интервала, к которому принадлежит результат суммы чисел в нормированной модулярной системе [1].

Пусть число A задано в нормированной модулярной системе. Данный метод позволяет за n операций определить значение t ($0 \le t < m_{n+1}$) в диапазоне: $\left[t \cdot \frac{M}{m_{n+1}}, (t + 1) \cdot \frac{M}{m_{n+1}} \right)$ к

которому принадлежит число A .

По известному значению t можно определить требуется ли округление числа A и на какое количество разрядов - s_1 , т.е.

$$s1 = \text{int}\left(\log\left(t \cdot \frac{M}{m_{n+1}} \right) \right) - p ,$$

где int - целая часть числа.

При сложении двух n -разрядных чисел, в результате может получиться число, максимальная разрядность которого будет равна $n + 1$. Поэтому округление при сложении двух чисел может быть выполнено не более чем на 1 разряд.

Временные затраты в условных тактах для сложения и округления представлены ниже:

$$\text{Тслож} = \lceil \log_2 \max(m_i) \rceil ,$$

$$\text{Токругл.} = n \quad \text{тактов.}$$

Операция сложения порядков происходит также как и в позиционной системе счисления. Для повышения быстродействия сдвига мантиссы в модулярную систему счисления вводится дополнительный модуль равный $10^{\lceil \log_2 (\max m_i) \rceil}$. Это позволит найти число, полученное после сдвига As без определения Q для всех $s \le \lceil \log_2 (\max\ m_i) \rceil$.

Для проверки работоспособности и эффективности предлагаемого метода проведено следующее аналитическое исследование.

Рассмотрим представление чисел с плавающей точкой с двойной точностью по стандарту «IEEE Standard 754 Floating Point Numbers». Длина мантиссы числа при этом равна $p = 53$ разряда (16 десятичных цифр), а порядка - 11 разрядов.

Необходимо выбрать модули модулярной системы так, чтобы выполнялись следующие условия:

1) первый модуль должен быть равным двум. Это необходимо для введения в модулярную систему отрицательных чисел [1]

2) произведение первых $n - 1$ модулей превышало 10^{16} .

3) Последний модуль должен быть выбран так, чтобы система модулей была нормированной, т.е весовой коэффициент последнего модуля должен быть равным единице, что требуется для метода нулевизации [1]

Этим условиям удовлетворяет следующая система модулей.

$$m_1 = 2,\ m_2 = 999959\ ,\ m_3 = 999961\ ,$$
$$m_4 = 999979\ ,\ m_5 = 5477401$$

Сравним временные затраты сдвига и сложения мантисс чисел в модулярной системе счисления и в позиционной. Предполагается, что в позиционной системе счисления сдвиг на s -разрядов осуществляется за s -тактов. Тогда время сложения в позиционной системе счисления (ПСС) будет равным

$$\text{Тслож. (ПСС)= Тсдвига+Тслож} = s + p\ .$$

В модулярной системе счисления:

$$\text{Тслож(МСС)}=(n+1)*\left\lceil \frac{s}{\lceil \log_2(\max\ m_i)\rceil} \right\rceil + \lceil \log_2 \max(m_i)\rceil + n\ .$$

Оценку эффективности применения модулярной арифметики для реализации операции суммирования будем производить по величине достигаемого коэффициента ускорения по следующей формуле:

$$\text{Eff} = \text{Тслож. (ПСС)} / \text{Тслож (МСС)}$$

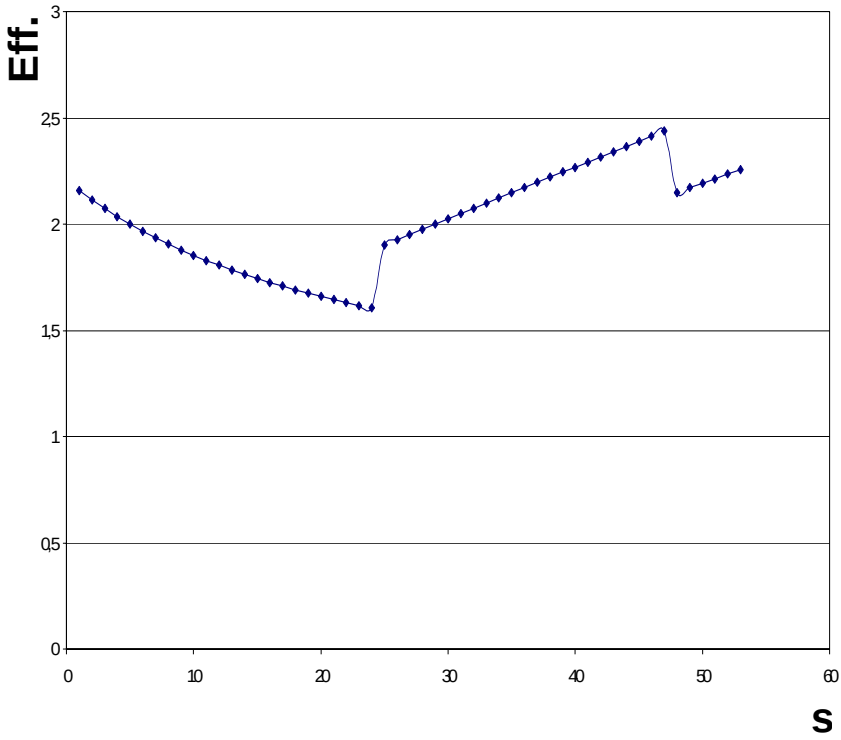
Ниже представлен полученный график зависимости эффективности Eff от числа сдвигов мантиссы s.

Из графика видно, что применение рассматриваемых методов обеспечивает ускорение модулярного сложения (вычитания) чисел с плавающей точкой в среднем в 2 раза.

Перспективными направлениями дальнейших исследований являются разработка алгоритмов, реализующих операции умножения, деления чисел с плавающей точкой.

Выводы:

1. Применение модулярной арифметики позволяет повысить точность сложения чисел в среднем в два раза без снижения быстродействия.
2. Для сложения чисел с плавающей точкой с удвоенной точностью достаточно пяти модулей модулярной системы.



Литература

1. Акушский Н.Я, Юдицкий Д.И. Машинная арифметика в остаточных классах, М. “Сов. радио ”, 1968. – 439 с.
2. Хамер К., Вранешич З., Заки С. Организация ЭВМ, 5-е изд. - СПб.: Издательская группа BHV, 2003. - 848 с.

УДК 681.325.5

Л.А. Овчаренко

Реализация немодульных операций на когерентных модулярных сумматорах

(Военный институт радиоэлектроники, г. Воронеж)

Предложен новый подход к выполнению немодульных операций в модулярной арифметике, идея которого базируется на изоморфности конечных множеств результатов аддитивных операций в кольце целых чисел и дискретных состояний фазы (временного положения) периодического колебания, кодируемых в соответствии со значениями операндов. Существо данного подхода иллюстрируется на примере устройств, реализующих немодульные операции преобразования позиционного кода в код вычета по модулю и преобразования по заданной функциональной зависимости модулярного кода в напряжение.

Одной из основных причин достаточно ограниченного применения на практике вычислительных структур в системе остаточных классов (СОК) является наличие проблемы немодульных операций. Известно [1], что базовой при выполнении таких операций является процедура формирования модульных сумм вида

$$R = \left| \sum_{i=1}^{n} \left| F_i(\alpha_i) \right|_m \right|_m , \quad (1)$$

где $\left|x\right|_y$ – вычет числа x по модулю y; $F_i(\alpha_i)$ – целочисленная функция вычета α_i по некоторому модулю m, реализуемая как унарное преобразование $\alpha_i \to F_i(\alpha_i)$.

Обычно [2] для вычисления (1) применяются *n*-местные сумматоры на полупроводниковых логических вентилях в виде пирамид двухместных сумматоров по модулю. Поскольку число ярусов здесь равно двоичному логарифму числа слагаемых, время получения модульной суммы будет прямо пропорционально как числу ярусов, так и задержке в логическом вентиле. Если ориентироваться на традиционную элементную базу, то очевиден единственный путь повышения быстродействия выполнения немодульных операций, который заключается в уменьшении времени переключения логических вентилей. Вместе с тем, учитывая то, что современная микроэлектроника подходит к пределу своих технологических возможностей, по видимому, в ближайшей перспективе не следует ожидать существенного повышения быстродействия вычислительных устройств на полупроводниковых логических вентилях.

В этой связи необходимы принципиально иные подходы к выполнению *n*-местных арифметических операций по модулю. Идея одного из таких подходов базируется на изоморфности множества результатов аддитивных операций в кольце целых чисел и множества дискретных состояний фазы (временного положения) периодического колебания, кодируемых в соответствии со значениями операндов [3]. Управляя фазой (временной задержкой) периодического колебания можно получать значение суммы любого числа слагаемых в (1), которая в данном случае будет прямо пропорциональна суммарному фазовому сдвигу (временному положению) этого колебания. Так как в настоящее время при использовании практически однотипной элементной базы реализуемая частота гармонических колебаний на один-два порядка превышает тактовую частоту наиболее быстродействующих цифровых схем, то осуществляя сжатие масштаба времени за счет использования в качестве опорного колебания сигнала СВЧ диапазона, возможно достижение достаточно высокого быстродействия выполнения *n*-местных арифметических операций по модулю.

Рассмотрим более подробно существо данного подхода.

Известно, что если гармоническое колебание с фиксированными значениями амплитуды U и частоты ω:

$$u(t)=U\ cos(\omega t) \qquad (2)$$

проходит через n последовательно соединенных фазовращателей, то фазовый набег в них суммируется и на выходе последнего (n-го) фазовращателя это колебание будет описываться выражением

$$u_n(t)=U\cos\left(\omega\cdot t-\left|\sum_{i=1}^{n}\varphi_i\right|_{2\pi}\right), \qquad (3)$$

где φ_i – сдвиг фазы в i-ом фазовращателе.

Пусть значение фазового сдвига в i-ом фазовращателе прямо пропорционально величине операнда α_i:

$$\varphi_i=\frac{2\pi}{m}\cdot\alpha_i \quad \left(\alpha_i=\overline{0,\ m-1};\quad i=\overline{1,n}\right). \qquad (4)$$

Тогда используя равенство

$$\left|\sum_{i=1}^{n}\varphi_i\right|_{2\pi}=\left|\frac{2\pi}{m}\cdot\sum_{i=1}^{n}\alpha_i\right|_{2\pi}=\frac{2\pi}{m}\cdot\left|\sum_{i=1}^{n}\alpha_i\right|_{m}, \qquad (5)$$

получим на основании (3) - (5)

$$u_n(t)=U\cdot\cos\left(\omega\cdot t-\frac{2\pi}{m}\cdot\left|\sum_{i=1}^{n}\alpha_i\right|_{m}\right) \qquad (6)$$

Из (6) видно, что после прохождения гармонического колебания через n фазовращателей, сдвиги фазы в которых прямо пропорциональны значению слагаемых α_i, $i=\overline{1,n}$, результирующий набег фазы гармонического колебания будет прямо пропорционален следующей величине:

$$X=\left|\sum_{i=1}^{n}\alpha_i\right|_{m}. \qquad (7)$$

Следовательно, для определения вычета суммы n чисел по модулю m достаточно измерить сдвиг фазы гармонического колебания $u_n(t)$ (6) относительно $u(t)$ (2).

Потенциально наиболее высокая достоверность измерения величины сдвига фазы может быть достигнута в оптимальном измерителе фазы сигнала известной формы [4], представляющем собой набор корреляторов, вычисляющих взаимокорреляционные функции гармонического колебания (6) с m опорными колебаниями, формируемыми путем соответствующей задержки во времени опорного колебания (2). По аналогии с когерентным приемом в теории связи, назовем сумматор, реализующий на основании изложенного подхода n - местную операцию сложения чисел по модулю m, *когерентным модулярным сумматором* (КМС). Структурная схема когерентного модулярного сумматора приведена на рис. 1, где символами x и ∫ соответственно обозначены аналоговые перемножители и интеграторы.

В схеме когерентного модулярного сумматора, в зависимости от значений операндов α_i, управление фазой гармонического колебания осуществляется на фиксированной частоте генератора (Γ). Поэтому реализация управляемых фазовращателей (УФ) с минимальными фазовыми ошибками технически не вызывает принципиальных трудностей, как это показано в [6] на примере манипулятора фазы с переключаемыми каналами. В таком УФ изменение фазы происходит путем коммутации отрезков передающих линий, имеющих различную длину, и, соответственно, в зависимости от длины линии l можно получить в пределах $0\ldots90^0$ сдвиг фазы: $\Delta\varphi = 2\cdot\pi\cdot l/\lambda$, где λ – длина волны в линии. Путем наращивания количества секций фазовращателей может быть обеспечен любой фазовый сдвиг в пределах $0\ldots360^0$. Для фазовращателей такого типа фазовая ошибка на фиксированной частоте не превышает десятые доли градуса [6].

В качестве базового элемента схемы выбора максимума измерителя сдвига фазы может быть использован триггерный усилитель дифференциальных сигналов, способный различать сигналы с перепадом уровня менее 50 мв [7].

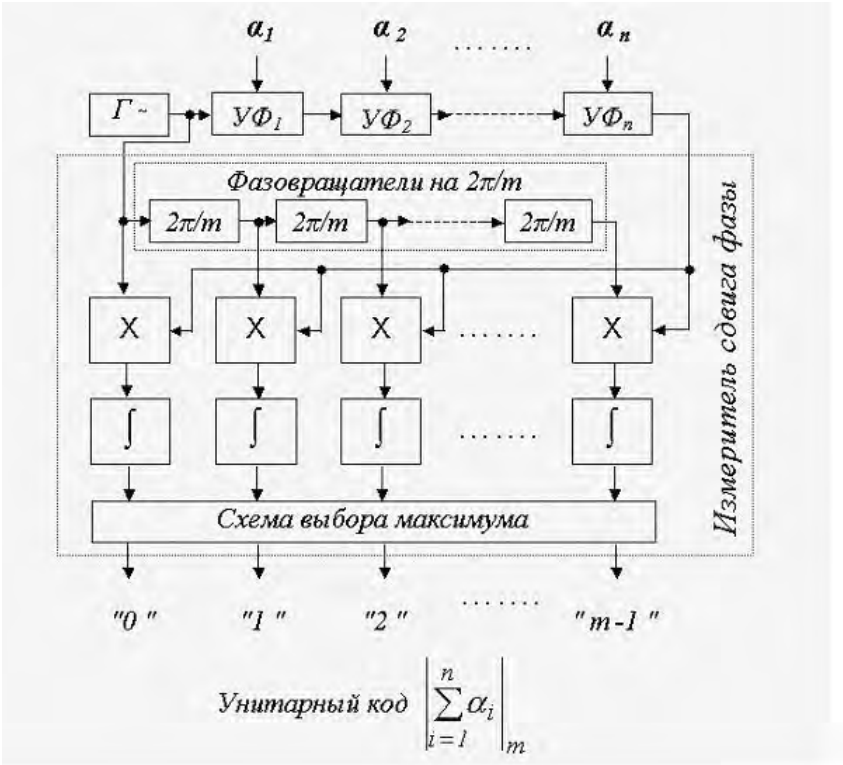


Рис. 1. Когерентный модулярный сумматор

Как уже отмечалось выше, для достижения максимального быстродействия и возможности создания КМС в интегральном исполнении, его функциональные элементы желательно реализовать в СВЧ диапазоне, где они имеют минимальные размеры, что соответственно позволяет обеспечить минимальные задержки при распространении сигналов.

Например, в коротковолновой части СВЧ диапазона (миллиметровом диапазоне длин волн) в измерителе сдвига фазы корреляторы могут быть выполнены на бескорпусных смесительных диодах и микрополосковых линиях передачи в виде кольцевого балансного фазового детектора, в нагрузку которого включен фильтр нижних частот (ФНЧ) [5]. В простейшем случае ФНЧ реализуется как интегрирующая *RC*–цепь с постоянной времени, в два-три раза превышающей длительность периода T опорного гармонического колебания. Тогда продолжительность переходного процесса в таком

корреляторе может быть оценена на уровне $5\ldots10$ периодов опорного колебания (2).

Учитывая, что максимальная задержка гармонического сигнала в управляемом фазовращателе не превышает длительности периода T этого сигнала [6], общее время расчета суммы n чисел по модулю m в схеме когерентного модулярного сумматора будет составлять величину порядка

$$t_{\Sigma} = n \cdot T + 10 \cdot T . \qquad (8)$$

Так, при частоте генератора 400 ГГц и $n = 10$, указанное время будет равно $t_{\Sigma} = 10 \cdot 0{,}25 \cdot 10^{-11} + 10 \cdot 0{,}25 \cdot 10^{-11} \approx 0{,}5 \cdot 10^{-10}$ с, что значительно меньше, чем в аналогичных схемах n – местных сумматоров на полупроводниковых логических вентилях [2].

Простота формирования модульных сумм в когерентном модулярном сумматоре свидетельствует о предпочтительности его применения в базовых структурах модулярной арифметики, предназначенных для перехода от позиционного кода к модулярному и обратно, а также для выполнения немодульных операций других типов. В отличие от традиционных вычислительных структур на полупроводниковых логических вентилях [8], КМС позволяет исключить многие последовательные вычислительные процедуры и тем самым в ряде случаев реализовать однотактный режим выполнения немодульных операций.

В качестве иллюстрации этого на рис. 2 приведена структурная схема преобразователя двоичного позиционного кода в код остатка по модулю *m* [9].

Здесь в управляемых фазовращателях $УФ_j$ осуществляется сдвиг фазы на угол $\varphi_j = \frac{2\pi}{m}\left|2^{j-1}\right|_m$ при $a_j = 1$, либо на угол $\varphi_j = 0$ при $a_j = 0$, где a_j – *j*-ый разряд *K*-значного двоичного позиционного кода, $j = \overline{1, K}$. Вариант схемы управляемого фазовращателя, в котором реализуется поворот на две градации фазы

[6], изображен на рис. 3.

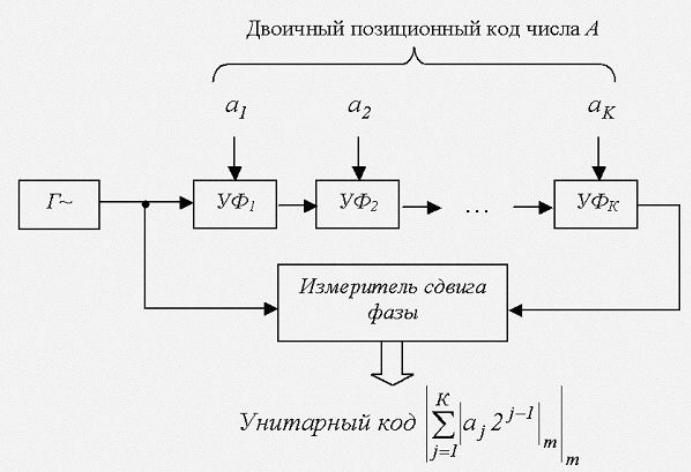


Рис. 2. Преобразователь двоичного позиционного кода в код остатка по модулю m

В этой схеме длина канала задержки l_1 выбирается из условия получения требуемого изменения фазы $\varphi_j = \frac{2\pi}{m}\left|2^{j-1}\right|_m$. Короткозамкнутый шлейф l_2 дополняет половину петли канала задержки до четвертьволновой длины $l_2 + l_1/2 = \lambda/4$ (λ - длина волны гармонического колебания в линии). Когда диоды V_1 и V_2 открыты, гармоническое колебание распространяется по опорному каналу со сдвигом фазы $\varphi_j = 0$, поскольку канал задержки совместно с дополнительным шлейфом l_2 образует два короткозамкнутых четвертьволновых шлейфа, не влияющих на прохождение сигнала по опорному каналу. Когда диоды V_1 и V_2 заперты, гармоническое колебание распространяется по каналу задержки. Указанная схема осуществляет поворот фазы на угол в пределах $0\ldots\pi/2$. Путем наращивания количества секций фазовращателей может быть обеспечен любой сдвиг фазы в пределах $0\ldots2\pi$.

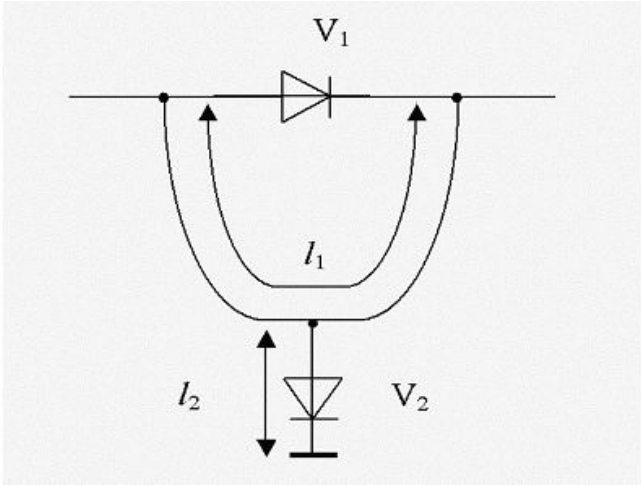


Рис. 3. Управляемый фазовращатель на две градации фазы

Таким образом, суммарный набег фазы на выходе *K*-го управляемого фазовращателя будет прямо пропорционален остатку по модулю m *K*-значного двоичного позиционного кода числа $A = \sum_{j=1}^{K} a_j \cdot 2^{j-1}$:

$$\left| \sum_{j=1}^{K} \frac{2\pi}{m} \left| a_j \cdot 2^{j-1} \right|_m \right|_{2\pi} = \frac{2\pi}{m} \left| \sum_{j=1}^{K} \left| a_j \cdot 2^{j-1} \right|_m \right|_m .$$

Величина набега фазы определяется в измерителе сдвига фазы, где формируется унитарный код искомого остатка.

Быстродействие такого преобразователя может быть оценено по формуле (8).

Еще одним примером эффективного применения КМС могут быть устройства преобразования модулярного кода в напряжение или ток, использующиеся при сопряжении разнотипных блоков информационно-измерительных систем. Традиционно [1,10], в таких устройствах преобразования получение аналоговой величины осуществляется в два этапа: вначале модулярный код переводится в по-

зиционный и затем из этого позиционного кода формируется напряжение путем суммирования токов, прямо пропорциональных весам разрядов позиционного кода, на общем сопротивлении нагрузки.

Идея способа цифроаналогового преобразования с применением КМС базируется на китайской теореме об остатках [11].

Действительно, если установить сдвиги фазы $\varphi_i \left(i = \overline{1, N}\right)$ в N последовательно соединенных управляемых фазовращателях на угол $\varphi_i = \frac{2\pi}{m_i}\left|\alpha_i \cdot \mu_{i,N}\right|_{m_i}$, где m_i – основания СОК; α_i – i-ый разряд модулярного кода; $\mu_{i,N}$ – константа, определяемая как решение сравнения $\frac{M_N}{m_i} \cdot \mu_{i,N} \equiv 1 \left(\mathrm{mod}\, m_i\right)$; $M_N = \prod_{i=1}^{N} m_i$, то суммарный набег фазы гармонического колебания на выходе N – го фазовращателя в соответствии с китайской теоремой об остатках (КТО) будет определяться выражением:

$$\Phi_\Sigma = \left|\sum_{i=1}^{N}\varphi_i\right|_{2\pi} = \left|\sum_{i=1}^{N}\frac{2\pi}{m_i}\cdot\left|\alpha_i \cdot \mu_{i,N}\right|_{m_i}\right|_{2\pi} =$$

$$= \frac{2\pi}{M_N}\left|\sum_{i=1}^{N} M_{i,N} \cdot \left|\alpha_i \cdot \mu_{i,N}\right|_{m_i}\right|_{M_N} = \frac{2\pi}{M_N} A. \quad (9)$$

Следовательно, суммарный набег фазы гармонического колебания Φ_Σ прямо пропорционален величине числа A, модулярный код которого представлен остатками $\left(\alpha_1, \alpha_2 \ldots \alpha_N\right)$ по соответствующим основаниям СОК $m_i \left(i = \overline{1, N}\right)$.

Данный способ наиболее просто позволяет выполнять функциональные преобразования кода СОК в напряжение следующего ви-

да: $A \to \cos\left(\frac{2\pi}{M_N} A\right)$ либо $A \to \sin\left(\frac{2\pi}{M_N} A\right)$, как это иллюстрируется схемами на рис. 4.

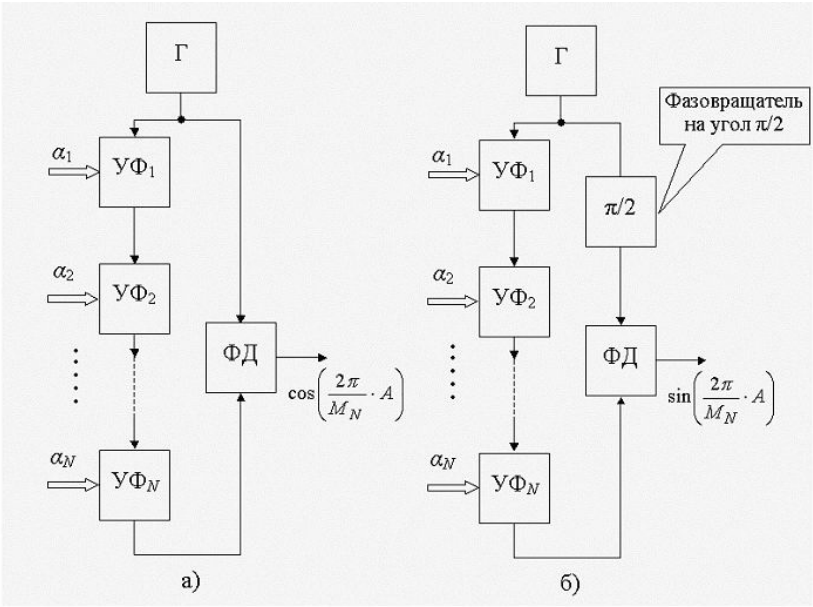


Рис. 4. Функциональные цифроаналоговые преобразователи (ЦАП) кода СОК в напряжение в соответствие с тригонометрической зависимостью

Полагая задержку в управляемых фазовращателях УФ$_i$ $\left(i = \overline{1, N}\right)$ равной периоду T частоты генератора (Г) гармонического колебания, а длительность переходного процесса в фазовом детекторе (ФД) — (5…10)T, есть основание считать, что суммарное время преобразования в ЦАП на рис. 4 будет составлять величину порядка $N \cdot T + (5\ldots10) \cdot T$. То есть, при использовании генератора миллиметрового диапазона длин волн может быть достигнуто достаточно высокое быстродействие преобразования – на уровне единиц наносекунд, что соизмеримо с быстродействием лучших образцов ЦАП в ПСС [12].

Учитывая приближенное равенство $\sin\left(\frac{2\pi}{M_N}A\right) \approx \frac{2\pi}{M_N}A$ при $|A| << M_N$, ЦАП на рис. 4б может быть использован для линейного преобразования кода СОК в напряжение. С этой целью основной диапазон изменения данных в модулярном процессоре $[0, M_N)$ необходимо расширить путем добавления дополнительных оснований СОК, так, чтобы выполнялось приведенное выше неравенство. При этом приемлемая для практики точность преобразования достигается, если диапазон представления чисел СОК превышает диапазон изменения данных не менее, чем в 30…50 раз [13].

Реализация линейного преобразования кода СОК в более широком диапазоне, а также преобразований с другим видом функциональной зависимости, может быть осуществлена на основе аппроксимации требуемой функции $F(A)$ усеченным рядом Фурье [13]:

$$F(A) = b_0 + \sum_{j=1}^{n} b_j \cdot \cos\left(\frac{2\pi}{M_N} \cdot A \cdot j + \phi_j\right), \quad (10)$$

где b_j и ϕ_j – соответственно амплитудные и фазовые коэффициенты *j* – го члена ряда Фурье.

На рис. 5 представлена схема ЦАП кода СОК, в которой в соответствии с выражением (10) реализован алгоритм функционального преобразования вида: $A \to F(A)$.

В этой схеме делитель мощности формирует из опорного гармонического колебания генератора (Г) n гармонических колебаний с амплитудами и фазами b_j и ϕ_j. В устройствах отображения $УО_{ji}$ $\left(j = \overline{1,n};\ i = \overline{1,N}\right)$ выполняются унарные преобразования унитарных кодов α_i в код $\left|j \cdot \alpha_i \cdot \mu_{i,N}\right|_{m_i}$, а в управляемых фазовращателях $УФ_{ji}$ – сдвиг фазы на угол

$\varphi_{ji} = \frac{2\pi}{m_i}\left| j \cdot \alpha_i \cdot \mu_{i,N} \right|_{m_i}$. В соответствии с КТО, на выходе j – ой линейки фазовращателей суммарный набег фазы гармонического колебания будет равен:

$$\Phi_{\Sigma j} = \left| \sum_{i=1}^{N} \varphi_{ji} \right|_{2\pi} = \left| \sum_{i=1}^{N} \frac{2\pi}{m_i} \cdot \left| j \cdot \alpha_i \cdot \mu_{i,N} \right|_{m_i} \right|_{2\pi} =$$

$$= \frac{2\pi}{M_N} \left| \sum_{i=1}^{N} M_{i,N} \cdot \left| j \cdot \alpha_i \cdot \mu_{i,N} \right|_{m_i} \right|_{M_N} = \frac{2\pi}{M_N} \cdot \left| A \cdot j \right|_{M_N} = \left| \frac{2\pi}{M_N} \cdot A \cdot j \right|_{2\pi}.$$

Следовательно, на выходе сумматора мощности сформируется суммарное колебание, прямо пропорциональное $\sum_{j=1}^{n} b_j \cdot \cos\left(\omega \cdot t - \frac{2\pi}{M_N} \cdot A \cdot j - \phi_j \right)$, которое в результате взаимодействия с опорным колебанием генератора в фазовом детекторе (ФД) преобразуется к виду $\sum_{j=1}^{n} b_j \cdot \cos\left(\frac{2\pi}{M_N} \cdot A \cdot j + \phi_j \right)$. И, наконец, на выходе сумматора, выполненного на операционном усилителе, окончательно получаем напряжение, прямо пропорциональное значению усеченного ряда Фурье (10).

В предположении того, что задержка в сумматоре мощности и в каждом управляемом фазовращателе равна периоду опорного гармонического колебания – T, а длительность переходного процесса в фазовом детекторе (ФД) — (5…10)T, находим оценку времени преобразования кода МСС: $(N+1) \cdot T + (5 \ldots 10) \cdot T$, которое может быть равно менее 1 нс при использовании генератора миллиметрового диапазона длин волн.

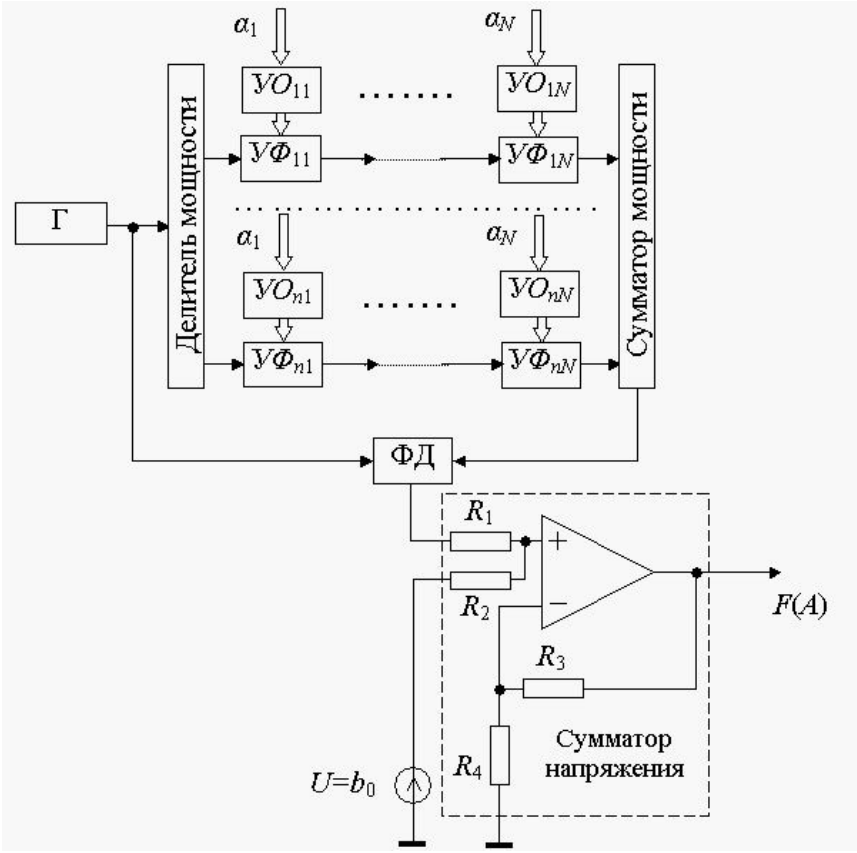


Рис. 5. Функциональный цифроаналоговый преобразователь кода СОК в напряжение в соответствии с зависимостью, аппроксимируемой усеченным рядом Фурье

Основное достоинство предлагаемого подхода к реализации цифроаналогового преобразования заключается в отсутствии необходимости промежуточного перевода модулярного кода в позиционный код, что, в конечном итоге, является предпосылкой существенного увеличения быстродействия выполнения немодульной операции преобразования «цифровой код – аналоговая величина».

- Таким образом, по сравнению с ранее применявшимися вычислительными блоками на полупроводниковых логических вентилях когерентный модулярный сумматор является более адекватным средством отображения алгоритмов немодульных операций на вычислительные структуры модулярной арифметики.

Литература

1. Чернявский А.Ф., Данилевич В.В., Коляда А.А., Селянинов М.Ю. Высокоскоростные методы и системы цифровой обработки информации. – Мн.: Белгосуниверситет, 1996. – 376 с.
2. Червяков Н.И. Организация арифметических расширителей в микропроцессорных системах, базирующихся на множественном представлении информации// Управляющие системы и машины. – 1987. – №1. – С.26-29.
3. Овчаренко Л.А. Вариант реализации основных операций в модулярном арифметическом устройстве// Телекоммуникации. – 2001. – №3. – С.8-11.
4. Тихонов В.И. Статистическая радиотехника. – М.: Сов. радио, 1966. – 678 с.
5. Микроэлектронные устройства СВЧ: Учеб. Пособие для радиотехнических специальностей вузов/ Г.И. Веселов, Е.Н. Егоров, Ю.Н. Алехин и др.; Под ред. Г.И. Веселова. – М.: Высш. шк., 1980. – 280 с.
6. Соколинский В.Г., Шейнкман В.Г. Частотные и фазовые манипуляторы. – М.: Радио и связь, 1983. – 192 с.
7. Кристовский Г.В., Погребной Ю.Л. Методика разработки КМОП БИС с малыми логическими перепадами// Успехи современной радиоэлектроники. – 2002. – №7. – С.25-35.
8. Коляда А.А., Пак И.Т. Модулярные структуры конвейерной обработки цифровой информации. – Мн.: Университетское, 1992. – 256 с.
9. Патент РФ №2192092. Устройство для преобразования n – разрядного двоичного позиционного кода в двоичный код остатка по модулю m/ Овчаренко Л.А., Турченяк В.И. – 2002. – Бюл. № 30.
10. Абрамсон И.Т., Авров О.М., Лапкин Л.Я. Кодирование электрических величин в системе остаточных классов// Автометрия. – 1975. – №2. – С.23-29.
11. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. – М.: Сов. радио, 1968. – 440 с.
12. Гитис Э.И., Пискулов Е.А. Аналого-цифровые преобразователи. – М.: Энергоиздат, 1981. – 360 с.
13. Овчаренко Л.А. Цифроаналоговый преобразователь кода системы остаточных классов контроллера управления динамическим объектом// Известия вузов – Радиоэлектроника. – 2002. – №11 – С.29-33.

УДК 681. 03

В.А. Краснобаев

Влияние формы кодирования операндов на надежность систем обработки цифровой информации

(Харьковский национальный технический университет сельского хозяйства им. Петра Василенка)

В данной статье рассматривается метод повышения надежности систем обработки цифровой информации (СОИ), основанный на использовании непозиционных кодовых структур в системе остаточных классов (СОК). Проведено исследование влияния основных свойств модулярной арифметики (МА) на структуру СОИ и принцип ее функционирования. Определено, что свойства МА создают предпосылки более эффективного, чем в позиционных системах счисления, использования одновременно трех видов резервирования: структурного, информационного и функционального.

Введены и обоснованы понятия первичной и вторичной избыточности. Определено, что первичная избыточность для СОИ совпадает с понятием естественной избыточности технических систем переработки информации, а вторичная избыточность - с понятием искусственной избыточности. Данное обстоятельство дало возможность сделать следующий основной вывод: предложенный вариант выбора формы кодирования операндов в МА позволяет создать отказоустойчивые вычислительные структуры (в частности СОИ в СОК), что особенно

важно при проектировании бортовых специализированных вычислительных комплексов баллистических ракет и космических аппаратов, а также при построении вычислителей в составе сложных и ответственных систем управления, например, для обработки информации АСУ, функционирующих в реальном времени.

In given clause the method of increase of reliability of systems of processing of the digital information (SPI), based on use of not item code structures in system of residual classes (SRC) is considered(examined). The research of influence of the basic properties modular of arithmetics (MA) on structure SPI and principle of its functioning is carried out(spent). It is determined, that the properties MA create the preconditions more effective, than in position notations, use simultaneously of three kinds of reservation: structural, information and functional.

The concepts of primary and secondary redundancy are entered and proved. It is determined, that the primary redundancy for SPI coincides with concept of natural redundancy of technical systems of processing of the information, and secondary redundancy - with concept of artificial redundancy. The given circumstance has enabled to make the following basic conclusion: the offered variant of a choice of the form of coding of operands in MA allows to create fault tolerance computing structures (in particular SPI in SRC), that is especially important at designing the onboard specialized computer complexes of ballistic rockets and space vehicles, and also at construction of calculators in structure of complex and responsible control systems (for example, for processing the information of a management information system), functioning in real time.

Введение

При традиционном подходе к созданию систем обработки цифровой информации (СОИ) к системе счисления (СС), которая используется при кодировании (представлении) операндов, предъявляются следующие основные требования:

- простота технической реализации представления кодовых слов при использовании существующей элементной базы;
- единственность представления кодовых слов в заданном числовом диапазоне;
- простота аппаратной и программной реализации методов и алгоритмов выполнения необходимых операций в заданной СС;
- выполнение условия «экономичности» СС, которая характери-

зует первичную избыточность СОИ.

Обзор литературы. В [1] применительно к средствам переработки информации введены частные понятия первичной и вторичной структурной избыточности СОИ. В общем виде данные понятия могут быть обобщены и сформулированы следующим образом.

*Определение 1***.** Первичной (структурной, информационной, функциональной) избыточностью (ПИ) СОИ будем называть существующую или искусственно вводимую избыточность данного вида, обусловленную природой создания или методом искусственного образования применяемой СС

Определение 2. Вторичной (структурной, информационной, функциональной, временной и нагрузочной) избыточностью (ВИ) СОИ будем называть избыточность искусственно вводимую в СОИ для улучшении ее отдельных характеристик (производительности, надежности, достоверности, помехоустойчивости, отказоустойчивости пр.) после того, как СС окончательно определена.

Из второго определения видно, что ВИ - это избыточность, обусловленная применением традиционных методов резервирования, широко используемых в технических системах различного назначения для улучшения их характеристик. Первичная избыточность для СОИ совпадает с понятием естественной избыточности (ЕИ) технических систем переработки информации, а ВИ - с понятием искусственной избыточности (ИИ). Необходимость введения и использования ВИ обусловлена требованиями, предъявляемыми к характеристикам создаваемых СОИ. Наряду с вышеперечисленными требованиями к СС отметим, что выбранная и используемая система счисления сама существенно влияет на следующие характеристики СОИ:

- структуру (архитектуру) СОИ;
- принципы переработки информации (в большей степени на методы и алгоритмы выполнения арифметических операций);
- требования, предъявляемые к использованию новой элементной базы;
- системную и пользовательскую производительность вычислительных структур;
- надежность, живучесть, достоверность и отказоустойчивость

СОИ;

- эксплуатационные характеристики и показатели СОИ и пр.

Количественно объем Vпи оборудования СОИ, обусловленный наличием ПИ, несколько меньше объема $V_{ЕИ}$ оборудования при наличии ЕИ. Объем Vви дополнительного оборудования, определенный наличием ВИ, полностью совпадает с объемом оборудования $V_{ИИ}$, обусловленного наличием ИИ. Анализ влияния формы кодирования операндов на структуру и отдельные характеристики различных типов СОИ показал, что для систем обработки цифровой информации вполне корректно считать, что $V_{ПИ} \approx V_{ЕИ}$.

Постановка задачи. При традиционном подходе к выбору СС СОИ в первую очередь необходимо обеспечить следующее условие:

$$V_{ПИ} = \min.$$

Однако выполнение условия (1) не всегда правомерно при разработке вычислительных структур, когда априорно возникает задача улучшения необходимых характеристик СОИ. Вполне возможно, что вариант построения СОИ, основанный на выполнении условия (1), вообще не целесообразен. Данная особенность ярко проявляется при использовании, например, непозиционной системы счисления в остаточных классах (СОК).

Действительно, безизбыточная СОИ в СОК содержит несколько большее (на ≈ 15%) количество оборудования $V_{ПИ,}$ чем СОИ в позиционной (например, двоичной, т.е. информация представляется позиционным двоичным кодом) СС (ПСС) для заданной одинаковой длине разрядной сетки (и при одних и тех же требованиях, предъявляемых к СОИ) без учета введения ВИ [2, 3]. Однако, как показали теоретические исследования и практические расчеты [4, 5], для достижения заданного (необходимого) уровня надежности СОИ (вычислительных структур) в СОК требуется гораздо меньший объем $V_{ВИ}$ оборудования, чем для СОИ в ПСС (см. табл. 1).

Предварительные расчеты показали, что суммарная структурная избыточность $V_{СИ} = V_{ПИ} + V_{ВИ}$ СОИ в СОК, обеспечивающая заданный уровень H(t) надежности (отказоустойчивости), значительно меньше, чем для дублированных и троированных мажоритарных вычислительных структур, широко используемых в ПСС [1], т. е. обеспечивается условие

$$\begin{cases} H_{сок}(t) \geq H_{псс}(t)[t = const]; \\ V_{си-сок} < V_{си-псс,} \end{cases}$$

без снижения пользовательской производительности.

Выражение (3**),** определяет условие обратное условию (2), т. е**.** при одинаковом количестве оборудования $\mathbf{V}_{СИ}$ СОК обеспечивает более высокое значение отказоустойчивости, т. е.

$$\begin{cases} H_{сок}(t) > H_{пи}(t)[t = const]; \\ V_{си-сок} \approx V_{си-псс.} \end{cases}$$

В таблице 1 представлены расчетные данные количества оборудования, необходимого для реализации методов повышения отказоустойчивости (надежности) в ПСС и СОК. Необходимое количество оборудования рассчитано по методике, представленной в [1], где:

- $V_{ПИ}^{(l)}$ - относительное количество оборудования безизбыточной СОИ в ПСС и в СОК, приведенное к одной единице l-байтовой разрядной сетки;
- $V_{СИ}^{(l)}$ - относительное количество оборудования избыточной СОИ в ПСС (троированная мажоритарная структура) и в СОК (с тремя контрольными основаниями) приведенное к одной единице l-байтовой разрядной сетки.

$\delta = \dfrac{V_{си\ (ПСС)}^{(l)} - V_{си\ (СОК)}^{(l)}}{V_{си\ (ПСС)}^{(l)}} \cdot 100\%$ - коэффициент относительного выигрыша в количестве суммарного оборудования СОИ в ПСС и в СОК.

Результаты расчетов (см. табл. 1 и [1]) показали, что с увеличением длины l-разрядной сетки СОИ, что характерно для современной тенденции развития вычислительных средств обработки цифровой информации, эффективность применения СОК резко возрастает.

Проявление основных свойств СОК [2, 6, 7] поясняет смысл выражений (2) и (3) следующим образом:

- первичная избыточность в СОК заметно и существенно положительно (с точки зрения улучшения характеристик СОИ) прояв-

ляет себя только при наличии ВИ;

- в СОК существует значительное взаимное положительное влияние отдельных видов резервирования, предусмотренных для повышения отказоустойчивости СОИ.

Таблица 1

l (n)	ПСС (двоичноя)		СОК				Выигрышв количестве оборудования, δ
	$V_{ПИ}^{(l)}$	$V_{СИ}^{(l)}$	Информационные основания	Контрольные основания	$V_{ПИ}^{(l)}$	$V_{СИ}^{(l)}$	
1 (4)	8	24	m_1=3, m_2=4, m_3=5, m_4=7	m_5=11, m_6=13, m_7=17	10	23	4
2 (6)	16	48	m_1=2, m_2=5, m_3=7, m_4=9, m_5=11, m_6=13	m_7=17, m_8=19, m_9=23, m_{10}=29, m_{11}=31	19	34	29
3 (8)	24	72	m_1=3, m_2=4, m_3=5, m_4=7, m_5=11, m_6=13, m_7=17, m_8=19	m_9=23, m_{10}=29, m_{11}=31,m_{12}=37, m_{13}=41,m_{14}=43, m_{15}=47,m_{16}=53	28	43	40
4 (10)	32	96	m_1=2, m_2=3, m_3=5, m_4=7, m_5=11, m_6=13, m_7=17, m_8=19 m_9=23, m_{10}=29	m_{11}=31,m_{12}=37, m_{13}=41,m_{14}=43, m_{15}=41,m_{16}=53, m_{17}=59,m_{18}=61, m_{19}=67	37	54	43
8 (17)	64	192	m_1=2, m_2=3, m_3=5, m_4=7, m_5=11, m_6=13, m_7=17, m_8=19 m_9=23, m_{10}=29 m_{11}=31, m_{12}=37, m_{13}=41, m_{14}=43, m_{15}=47, m_{16}=51.	m_{17} = 53, m_{18} = 57, m_{19} = 59.	67	82	57

В отличие от СОК в ПСС применение одного вида резервирования не всегда обуславливает одновременное наличие и других видов

резервирования. Отметим, что это не свидетельствует об отсутствии других видов избыточности. Так, применение информационного резервирования (введение информационной избыточности) для повышения достоверности вычислений СОИ в ПСС вызывает наличие структурной ВИ. Таким образом, применение необходимого вида резервирования в ПСС обязательно сопровождается наличием неиспользуемой («вредной») структурной избыточности, что, в конечном итоге, негативно влияет на технические и стоимостные характеристики СОИ.

В силу влияния основных свойств (независимость, равноправность и малоразрядность остатков $\mathbf{a_i \equiv A(mod\, m_i), i = \overline{1,n},}$ представляющих операнд А) СОК на особенности синтеза СОИ структурное, информационное и функциональное резервирование оказывают друг на друга одновременное взаимное положительное влияние. Например, введение вторичной структурной избыточности (применение структурного резервирования), посредством дополнительного использования **k** резервных вычислительных трактов к имеющимся **n** основным, приводит к проявлению как информационного, так и функционального резервирования. Первое из них связано с информационной избыточностью, обусловленной наличием избыточных кодовых слов и реализуемой путем использования дополнительной информации, получаемой с выходов **k** резервных вычислительных трактов. Относительно функционального резервирований отметим, что, в соответствии со свойствами СОК, один работоспособный вычислительный тракт СОИ в СОК, функционирующий по основанию m_j (при соблюдении условия $\mathbf{m_j \geq \prod_{\rho=1}^{r} m_{i_\rho}}$ (4)), может взять на себя вычислительные функции до **r** одновременно отказавших вычислительных трактов.

Практическая реализация. На рис. 3.11 представлено устройство для одновременной реализации структурного, информационного и функционального резервирования в СОК [8]. Данное устройство для резервирования в СОК содержит информационные $m_i \left(i = \overline{1,n}\right)$ и контрольный m_{n+1} вычислительные тракты, блок контроля (БК), дешифратор (ДС), логические элементы И и ИЛИ. Присутствие сигнала на **k**-й выходной БК соответствует отказу **k**-го вычислительного тракта $m_k \left(k = \overline{1, n+1}\right)$.

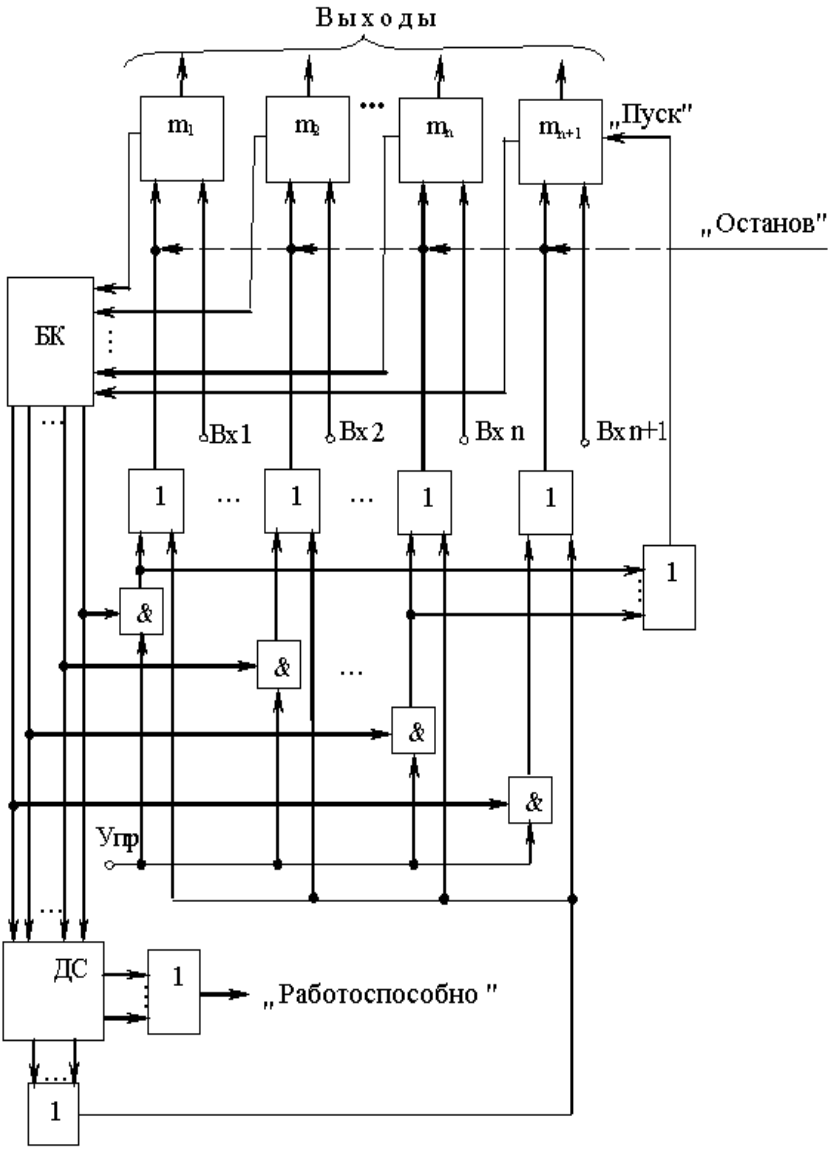


Рис.3.11 Устройство для резервирования в СОК

Таким образом, с выхода БК на вход дешифратора поступает **n**-разрядный двоичный код, несущий информацию о работоспособности рабочих трактов $m_1 \div m_4$ устройства. Присутствие единиц на

некоторых позициях этого кода соответствует отказу рабочих трактов с номерами, соответствующими номерам позиций этих единиц. В табл.2 приведен пример образования выходного кода БК для СОК, заданной информационными основаниями $m_1 = 3$, $m_2 = 4$, $m_3 = 5$, $m_4 = 7$ и контрольным основанием $m_5 = 23$.

Рассмотрим работу этого устройства.

1. Все основные тракты $m_1 \div m_4$ работоспособны. В этом случае с выхода блока контроля код 0000 поступает на вход дешифратора ДС, с выхода которого сигнал на нулевой шине через элемент ИЛИ поступает на выход «работоспособно», что свидетельствует о том, что устройство работоспособно.

Таблица 2

Выходной код БК				Номер отказавшего тракта	Соотношение оснований СОК	Работоспособность устройства
m_1	m_2	m_3	m_4			
0	0	0	0	-	-	+
0	0	0	1	4	$m_4 < m_5$	+
0	0	1	0	3	$m_3 < m_5$	+
0	0	1	1	3,4	$m_3 m_4 > m_5$	-
0	1	0	0	2	$m_2 < m_5$	+
0	1	0	1	2,4	$m_2 m_4 > m_5$	-
0	1	1	0	2,3	$m_2 m_3 < m_5$	+
0	1	1	1	2,3,4	$m_2 m_3 m_4 > m_5$	-
1	0	0	0	1	$m_1 < m_5$	+
1	0	0	1	1,4	$m_1 m_4 < m_5$	+
1	0	1	0	1,3	$m_1 m_3 < m_5$	+
1	0	1	1	1,3,4	$m_1 m_3 m_4 > m_5$	-
1	1	0	0	1,2	$m_1 m_2 < m_5$	+
1	1	0	1	1,2,4	$m_1 m_2 m_4 > m_5$	-
1	1	1	0	1,2,3	$m_1 m_2 m_3 > m_5$	-
1	1	1	1	1÷4	$\prod_{i=1}^{4} m_i \succ m_5$	-

2. Часть основных трактов неработоспособно.

а) *Допустим, что отказали тракты по основаниям* m_1 *и* m_2. С выхода БК код 1100 поступает на вход дешифратора ДС, с выхода которого сигнал по двенадцатой ($1100_2 = 12_{10}$) выходной шине через

элемент ИЛИ 8 поступает на выход «работоспособно». Одновременно код 1100 открывает первый и второй элементы И (единицы на первой и второй позициях кода 1100) и сигнал шины «Управление» через первый и второй элементы ИЛИ поступает на управляющие входы «Останов» соответственно первого m_1 и второго m_2 трактов устройства, а также через элемент ИЛИ поступает на управляющий вход «Пуск» контрольного тракта m_5 . Таким образом, устройство работоспособно и информация обрабатывается блоками m_3- m_5 (тракты m_1 и m_2 отключены);

б) Допустим, что отказали тракты по основанию m_3 и m_4.

С выхода БК код 0011 поступает на вход дешифратора. С выхода дешифратора сигнал по третьей ($0011_2=3_{10}$) выходной шине через элементы ИЛИ поступает на входы «Останов» трактов $m_1 \div m_5$ ($m_3 \cdot m_4=35>m_5=23$). В этом случае устройство неработоспособно и сигнал шины «Работоспособно» отсутствует.

Особенность функционирования данного устройства состоит в расширении функциональных возможностей за счет замены одним исправным контрольным трактом не одного, а одновременно нескольких неработоспособных рабочих трактов при выполнении условия (4). Это позволяет существенно повысить надежность и отказоустойчивость вычислительных структур за счет возможности одновременного использования трех видов резервирования: структурного (за счет введения контрольного вычислительного тракта по основанию m_n , параллельно функционирующего с основными вычислительными трактами), информационного (за счет использования дополнительной выходной информации контрольного вычислительного тракта, обеспечивающей возможность коррекции искаженной информации) и функционального (за счет выполнения условия (4)).

Приведенный пример показывает, что в СОК, в отличие от ПСС, введенная ВИ максимально полно используется для улучшения характеристик СОИ. Действительно, использование любого вида резервирования в конечном итоге приводит к структурной (аппаратной) избыточности, которая в СОК (в отличие от ПСС) используется для организации одновременно нескольких различных видов резервирования, что повышает коэффициент использования вводимого избыточного и общего суммарного оборудования СОИ.

Данная организация одновременно различных видов резервирова-

ния за счет введения структурной избыточности характерна для структурно-функциональной организации деятельности мозга человека и может обеспечить сверхвысокую надежность, отказоустойчивость и живучесть вычислительных структур , а также большую скорость обработки огромных массивов информации. В этом аспекте деятельность человеческого мозга близка к голографическим принципам обработки информации, что, в свою очередь, согласуется с методами и алгоритмами переработки информации в СОК [9].

Исходя из вышеизложенного, при создании (проектировании) СОИ необходимо учитывать не только влияние СС на объем $V_{ПИ}$, а в первую очередь оценивать значение $V_{СИ}$ (при учете влияния СС на остальные характеристики СОИ) , т. е. целесообразно выбирать СС с учетом ее дальнейшего влияния на выбор методов улучшения необходимых характеристик СОИ. По-видимому, при построении высокоотказоустойчивых вычислительных структур можно отказаться от традиционного критерия «экономичности» выбора позиционных СС по критерию (1), при котором необходимо обеспечить выполнение условия $f(q) = q \cdot \log_q N = \min,$ где q - основание данной позиционной СС; N - длина машинного слова (разрядная сетка СОИ). Для этого критерия оптимальное числовое значение основания СС будет равно $q_{опт} = e \approx 2{,}72$.

Как показали исследования и расчеты [1, 4, 5, 10] выбор, с точки зрения обеспечения заданного уровня надежности СОИ, СС целесообразно проводить по критерию

$$V_{СИ} = \min, \qquad (5)$$

а не по критерию (1), при заданном уровне требований к отдельным характеристикам ЭВМ. Данная задача близка к задаче оптимального резервирования в теории надежности, что нашло подтверждение при практическом создании блоков и узлов отказоустойчивых СОИ в СОК [4].

Выводы. Таким образом, предложенный вариант выбора формы кодирования операндов по критерию (5) позволяет создать природотказоустойчивые вычислительные структуры (в частности СОИ в СОК), что особенно важно при проектировании бортовых специализированных вычислительных комплексов баллистических ракет и космических аппаратов, а также при построении вычислителей в

составе сложных и ответственных систем управления (например, для обработки информации АСУ ТП), функционирующих в реальном времени.

Литература

1. Краснобаев В.А. Надежностная модель ЭВМ в системе остаточных классов // Электрон. моделирование.-1985.-№ 4. - С 44-46.
2. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. - М.: 1968. - 440 с.
3. Торгашов В.А. Система остаточных классов и надежность ЦВМ. - М.: 1973. - 118 с.
4. Ирхин В.П. Проектирование непозиционных специализированных процессоров. - Воронеж: Воронежский государственный университет, 1999. -136с.
5. Краснобаєв В.А. Синтез та оптимізація обчислювальних структур у системі залишкових класів // Інформаційно-керуючи системи на залізничному транспорті. – 2000. - № 2.- С. 36 – 37.
6. Фурман И.А., Краснобаев В.А. Новые возможности использования системы счисления в остаточных классах для построения высокоэффективных устройств обработки данных и управления // Вісник ХДТУСГ. 2000. Вип. 3.С. 27 – 31.
7. Краснобаев В.А.Основы создания вычислителей на основе остаточных классов // Системи обробки інформації. – Харків: НАНУ, ПАНМ, ХВУ. – 2001. - Вип. 1 (11). – С. 3 – 7.
8. А.с. 1168947 СССР. Устройство для резервирования / В.А. Краснобаев. – Опубл. 1985. БИ. № 27.
9. Краснобаев В.А., Илюшко Я.В. Построение систем искусственного интеллекта на основе использования непозиционного кодирования информации // Открытые информационные и компьютерные интегрированные технологии. – Х.: НАКУ (ХАИ). – 2004. – Вып. 24. – С. 286-298.
10. Жихарев В.Я., Илюшко Я.В., Краснобаев В.А. Влияние системы счисления на надежность ЭВМ. // Радіоелектронні і комп’ютерні системи. – 2004. - № 1(5).-С. 98 – 104.

УДК 621.29

И.И. Дзегеленок, Ш.А. Оцоков

Метод ускорения модулярной арифметики с самоисключением ошибок округления

(Московский энергетический институт)

Разработан метод для ускорения выполнения арифметических операций без ошибок округления в системе остаточных классов на поле рациональных чисел. Данный метод позволяет существенно расширить диапазон представления дробных чисел в системе остаточных классов и повысить быстродействие вычислений за счет снижения разрядности модуля, по которому проводятся вычисления с исключением ошибок округления.

Системы компьютерной алгебры находят широкое применение во многих областях науки и техники. Возможности таких систем как, например, выполнение преобразований, упрощений произвольных математических выражений, расчеты без ошибок округления, помогают найти точные или аналитические решения многих научных задач.

В современных математических системах (Mathcad, Maple и др) для реализации вычислений, в которых исключаются ошибки округления, как правило, используется рациональная арифметика на основе схемы приведения дробей. Для выполнения арифмети-

ческих операций над дробями по этой схеме требуется определение наибольшего общего делителя.

Другой метод для реализации вычислений без ошибок округления основан на использовании арифметики системы остаточных классов (модулярная арифметика).

Все арифметические операции с дробными числами выполняются в поле целых чисел системы остаточных классов. Результаты вычислений являются целыми числами, которые можно представить в виде простых дробей. Такое представление единственно, так как существует взаимно-однозначное соответствие между целыми числами системы остаточных классов и конечным множеством несократимых дробей Фарея [2].

Численные эксперименты на ЭВМ показывают, что быстродействие вычислений в одномодульной системе остаточных классов выше чем по схеме приведения дробей.

Пусть необходимо вычислить значение некоторого арифметического выражения с исключением ошибок округления. Допустим, что в одномодульной системе остаточных классов возникла ошибка псевдопереполнения конечного результата (выход результата за пределы допустимого диапазона). Тогда, это приводит к получению неверного результата при отображении в дробь.

Рассмотрим метод организации вычислений с исключением ошибок округления и с более широким диапазоном представления дробей, чем в одномодульной системе остаточных классов.

Предлагаемый метод основан на том, что вне зависимости от ошибки псевдопереполнения искомые и полученные результаты принадлежат одному классу вычетов. В процессе вычислений найдем произведение всех знаменателей дробей модулю СОК. Можно показать, что искомый результат A / B определяется следующим образом:

$$\frac{A}{B} = N_1 \frac{\left| N_1 \cdot U \cdot B_2 \right|_m}{B_2} \qquad (1)$$

где U - конечный результат вычислений в СОК,

m - модуль СОК,

B_2 – произведение знаменателей всех дробей по модулю P ,

N_1 - знак результата.

Рассмотрим пример для вычисления 1-1/2-1/3-1/4.

1) $1 - \frac{1}{2} = \frac{1}{2}$, 2) $\frac{1}{2} - \frac{1}{3} = \frac{1}{6}$, 3) $\frac{1}{6} - \frac{1}{4} = -\frac{1}{12}$.

Пусть модуль СОК m=127, тогда

1) $\left|1 - \frac{1}{2}\right|_{127} = 1 - 64 = -63$,

2) $\left|\frac{1}{2} - \frac{1}{3}\right|_{127} = -63 + 42 = -21$,

3) $\left|\frac{1}{6} - \frac{1}{4}\right|_{127} = -21 + 95 = 74$

Конечное множество дробей Фарея p/q , $((p,q) = 1, p/q > 0)$ соответствующих классу вычетов 74 по модулю 127 включает следующие дроби:

2/103 3/91 4/79 5/67 6/55 7/43 8/31 9/19 10/7 11/122 12/110 13/98 14/86 15/74 16/62 17/50 18/38 19/26 20/14 21/2 22/117 23/105 24/93 25/81 26/69 27/57 28/45 29/33 30/21 31/9 32/124 33/112 34/100 35/88 36/76 37/64 38/52 39/40 40/28 41/16 42/4 43/119 44/107 45/95 46/83 47/71 48/59 49/47 50/35 51/23 52/11 53/126 54/114 55/102 56/90 57/78 58/66 59/54 60/42 61/30 62/18 63/6 64/121 65/109 66/97 67/85 68/73 69/61 70/49 71/37 72/25 73/13 75/116 76/104 77/92 78/80 79/68 80/56 81/44 82/32 83/20 84/8 85/123 86/111 87/99 88/87 89/75 90/63 91/51 92/39 93/27 94/15 95/3 96/118 97/106 98/94 99/82 100/70 101/58 102/46 103/34 104/22 105/10 106/125 107/113 108/101 109/89 110/77 111/65 112/53 113/41 114/29 115/17 116/5 117/120 118/108 119/96 120/84 121/72 122/60 123/48 124/36 125/24 74/1

Докажем, что в этом списке нет двух дробей с одинаковыми знаменателями. Предположим противное, т.е. существуют две дроби p_1/q и p_2/q, $p_1 \neq p_2$, $p_1 < m, p_2 < m$ принадлежащие одному и тому же классу вычетов r по модулю m .

Тогда

$p_1 \cdot q^{-1} \equiv p_2 \cdot q^{-1} (\mathrm{mod}\ m)$, $p_1 \equiv p_2 (\mathrm{mod}\ m)$, но $p_1 \neq p_2$ по условию. Т.е. пришли к противоречию. Ч.т.д.

Таким образом, в этом списке нет двух положительных дробей с одинаковым знаменателем.

В работе [1] для вычислений с отрицательными числами в системе остаточных классов используется искусственное представление чисел. Суть которого заключается в том, что вводится дополнительное основание СОК равное двум и диапазон представления чисел в СОК удваивается. При этом положительным числам соответствует диапазон от m до $2m$, а отрицательным от 0 до $m - 1$.

В соответствии с описанными в работе [1] правилами выполнения арифмети-ческих операций с числами в искусственной форме получим результат в двухмодульной СОК. Преобразуя его в одномодульную СОК и в зависимости от диапазона определим знак результата вычислений в СОК.

Проблемой вычислений в СОК является ошибка пседопереполнения. Без обнаружения этой ошибки нет гарантий в том, что полученный результат является правильным.

Одним их возможных способов решения этой проблемы заключается в следующем. В процессе вычислений в СОК параллельно и независимо друг от друга ведутся вычисления в позиционной системе счислений (ПСС). И если результаты этих вычислений отличаются друг от друга более чем на порядок, то имеет место ошибка пседвопереполнения. В зависимости от знака результата вычислений в ПСС определяется N_1 в формуле (1). Иллюстрируем это на примере.

В данном примере $B_2 = 24$, $U = 74$ и $N_1 = -1$.

Поэтому,

$$\frac{A}{B} = -\frac{\left|-74 \cdot 24\right|_{127}}{24} = -\frac{1}{12}$$

Т.к. величины B_2, U и N_1 не зависят друг от друга их можно вычислять параллельно.

Пусть N - порядок дроби Фарея искомого результата, тогда с использованием одномодульной СОК необходимо выбрать модуль $m \geq 2N^2 + 1$, а в предлагаемом методе - $m \geq N + 1$ [2,3].

Эффект повышения быстродействия вычислений по сравнению с одномодульной СОК достигается тем, что все арифметические операции проводятся над числами имеющими разрядность $]\log_2(N+1)[$, в то время как в одномодульной СОК разрядность чисел будет больше чем $]\log_2(2N^2+1)[$.

Разработанный метод позволяет повысить быстродействие вычислений с исключением ошибок по сравнению с одномодульной системой остаточных классов за счет существенного расширения диапазона представления дробей Фарея. Хотя в данном требуется кроме U требуется определить еще и B_2, N_1, но это приводит к снижению быстродействия, т.к величины U, B_2, N_1 могут быть вычислены параллельно и независимо друг друга.

Краткое описание программного модуля безошибочной обработки чисел в одномодульной системе остаточных классов

Разработан программный модуль безошибочной обработки чисел, который может использоваться при решении различных вычислительных задач, требующих высокой точности вычислений. Основу программного модуля составляют алгоритмы преобразования и вычислений в одномодульной СОК. Рассмотрим структуру программного модуля. В его состав входят следующие процедуры:

1. Init (P) - процедура инициализации модуля, входной параметр : основание СОК - P;

2. Add_Sok (A, B ; Result) - процедура сложения двух чисел, входные параметры: два числа, выходные - результат суммирования.

3. Sub_Sok (A, B ; Result) - процедура вычитания двух чисел, входные параметры: два числа, выходные - разность двух чисел.

4. Mul_Sok(A, B ; Result) - процедура умножения двух чисел,

входные параметры: два числа, выходные - результат произведения двух чисел.

5. Div_Sok(A, B ; Result) - процедура деления двух чисел, входные параметры: два числа, выходные - результат деления двух чисел.

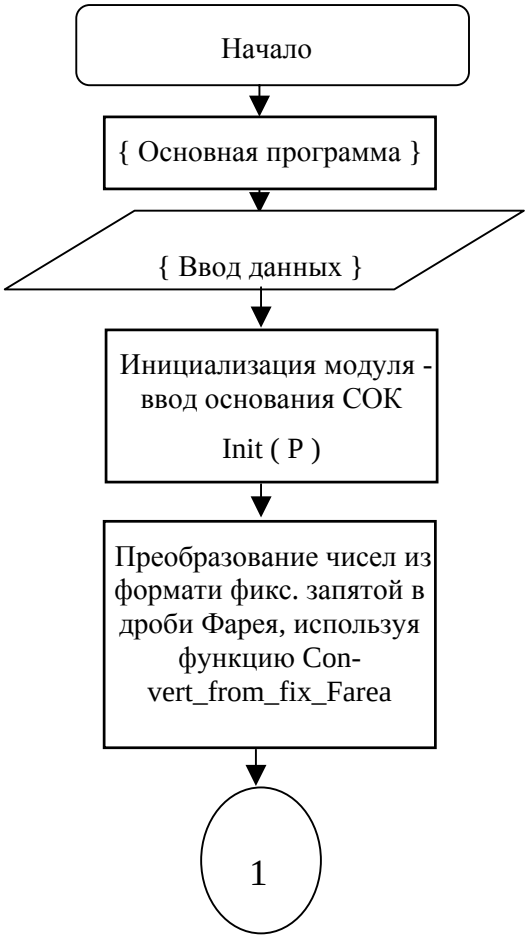
Процедуры для работы с числами сверх большой величины длиной до 100 цифр:

6. Mul_Long_Sok(A, B ; Result) - процедура умножения двух чисел A и B, входные параметры: два числа, выходные - результат произведения двух чисел.

7. Div_Long_Sok(A, B ; Result) - процедура деления двух чисел A и B, входные параметры: два числа, выходные - результат деления двух чисел.

8. Add_Long_Sok (A, B ; Result) - процедура сложения двух чисел A и B, входные параметры: два числа, выходные - результат суммирования.

9. Sub_Long_Sok (A, B ; Result) - процедура вычитания двух чисел A и B, входные параметры: два числа, выходные - разность двух чисел.



Процедуры преобразования чисел и работы с дробями Фарея:

10. Convert_from_Sok_Farea(A ; Result) - процедура преобразова-

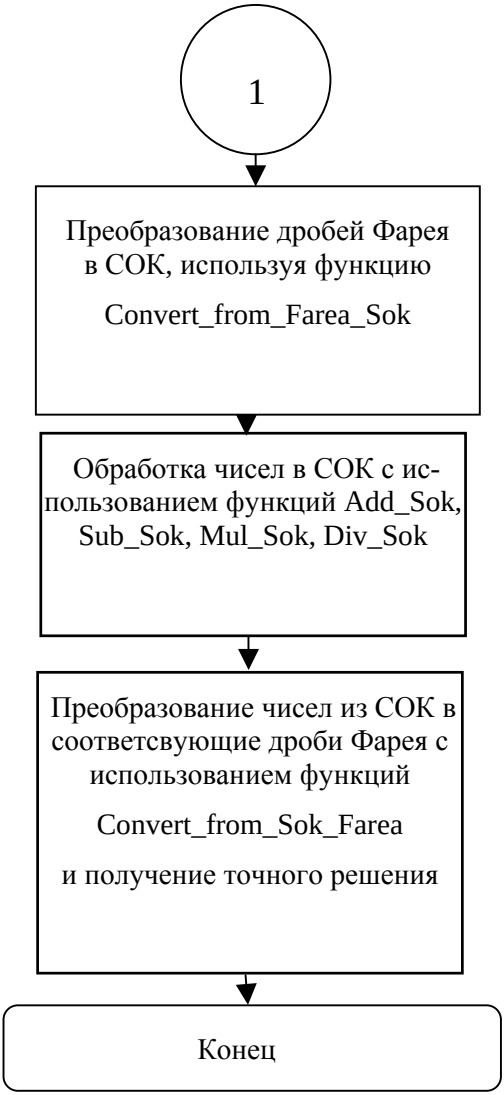
ния числа A из СОК в соответствующую дробь Фарея Result ;

11. Sokr (A; Result) - процедура для сокращения дроби A

12. Convert_from_fix_Farea (A ; Result) - процедура преобразования числа A с фиксированной запятой в дробь Фарея Result.

Для безошибочных вычислений необходим именно следующий порядок вызовов процедур программного модуля, показанный на рис 1.

Программный модуль реализованный в виде dll-библиотекии и динамически подключается к прикладной программе. Реализация в виде dll-библиотеки позволяет использовать ее практически в любых современных средах программирования для прикладных параллельных программ, в которых требуется повышенная точность вычислений.



Литература

1. *Акушский Н.Я, Юдицкий Д.И.* Машинная арифметика в остаточных классах, М. “Сов. радио ”, 1968. – 439 с.
2. *Грегори Р, Кришнамурти Е.* Безошибочные вычисления. Методы и приложения.М.:Мир,1988. –207 с.
3. *Дзегелёнок И.И, Оцоков Ш.А.* Подход к решению проблемы безошибочных вычислений с использованием ускоренного алгоритма отображения дробей Фарея. // Труды научной конференции, посвященной 75-летию со дня рождения академика В.А.Мельникова. РАН. М. 2004.

УДК. 681.3

А.И. Корнилов, М.Ю. Семенов,
О.В. Ласточкин, В.С. Калашников

Применение современных методов проектирования при реализации модулярных вычислительных процедур

(Институт проблем проектирования в микроэлектронике РАН)

Показано применение современных методов проектирования и логического синтеза при построении эффективных, с точки зрения быстродействия и занимаемой площади, модулярных вычислительных блоков для отдельных значений модулей.

К цифровым устройствам постоянно предъявляются требования повышения быстродействия. Применение аппарата модулярной арифметики при построении специализированных вычислителей позволяет повысить производительность таких систем за счет естественного распараллеливания трактов обработки данных без какого-либо изменения существующих технологий. В 60-70-е годы были проведены значительные теоретические исследования в области модулярной арифметики (в том числе и в России) и реализован ряд высокоэффективных вычислительных систем на ее основе. Однако данное направление не получило дальнейшего широкого развития во многом из-за проблем в реализации этих устройств, связанных с элементной базой, принципиально ориентированной на двоичную булеву арифметику. В настоящее время с развитием интегральной схемотехники появляются возможности по использованию современных методов проектирования при реализации модулярных вы-

числительных процедур. В свою очередь, эффективная реализация отдельных модулярных вычислительных блоков приводит к повышению эффективности всего устройства в целом.

Рассмотрим более подробно, какие методы проектирования могут быть использованы при реализации модулярных вычислительных процедур.

Методы логического синтеза быстродействующих модулярных сумматоров на основе BDD-технологии.

Одним из возможных методов логического синтеза быстродействующих цифровых схем является использование представления логических функций на основе Диаграмм Двоичных Решений (ДДР) или в англоязычной терминологии - Binary Decision Diagram (BDD). Применение диаграмм двоичных решений может быть продемонстрировано при построении быстродействующих сумматоров по модулям вида (2^n-1) и (2^n+1). Модули данного типа находят широкое применение при построении систем с использованием аппарата модулярной арифметики.

Построение быстродействующих сумматоров по модулю вида (2^n-1) на основе BDD.

Сумматоры по модулям вида (2^n-1) могут быть построены на основе сумматоров с циклическим переносом из высшего в низший разряд. Это связано с тем, что суммирование по модулю (2^n-1) может быть преобразовано в суммирование по модулю 2^n в соответствии со следующей формулой:

$$\left|a+b\right|_{2^n-1} = \left|a+b+c_{out}\right|_{2^n} \qquad (1)$$

В классической BDD каждое звено диаграммы соответствует разложению Шеннона, а в аппаратном исполнении реализации мультиплексора. Однако для традиционных КМОП схем целесообразно использовать монотонный базис, применяя известные булевы функции генерации (g_i) и распространения переноса (p_i):

$$p_i = a_i \vee b_i \qquad (2)$$

$$g_i = a_i \wedge b_i \qquad (3)$$

$$c_{i+1} = g_i \vee p_i c_i \qquad (4)$$

В таких сумматорах быстродействие определяет критический путь при вычислении функции выходного переноса. Построим функциональную диаграмму для системы функций переноса сумматоров по модулю (2^n-1) в соответствии с формулой (1), основываясь на функциональной диаграмме обычного двоичного сумматора в соответствии с формулами (2) – (4). В этом случае каждое звено диаграммы соответствует аппаратной реализации КМОП-элемента 2И-ИЛИ. Для обозначения таких диаграмм, которые, фактически, являются одной из разновидностей BDD диаграмм, будем использовать название FDD - Functional Decision Diagram.

На рис. 1(a) приведена FDD для n-разрядного двоичного сумматора с учетом формул (2)-(4). Для сумматоров по модулю (2^n-1), в которых выходной перенос равен входному (см. формулу (1)), FDD принимает вид, показанный на рис. 1(b).

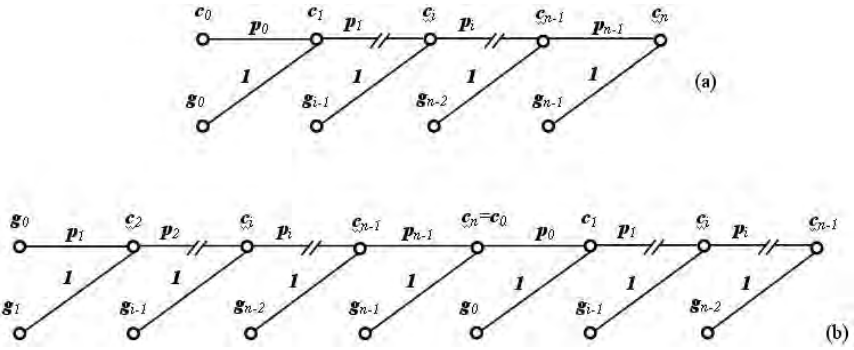


Рис. 1. FDD для n-разрядного двоичного сумматора (a) и преобразованная FDD для сумматора по модулю (2^n-1) (b).

Применим к полученной диаграмме для сумматоров по модулю (2^n-1) алгоритм декомпозиции, описанный в работах [1], [2]. Правило преобразования заключается в том, что длинная структура FDD может быть сведена обрывом ребра к двум диаграммам, при этом эти декомпозированные части реализуются параллельно (примени-

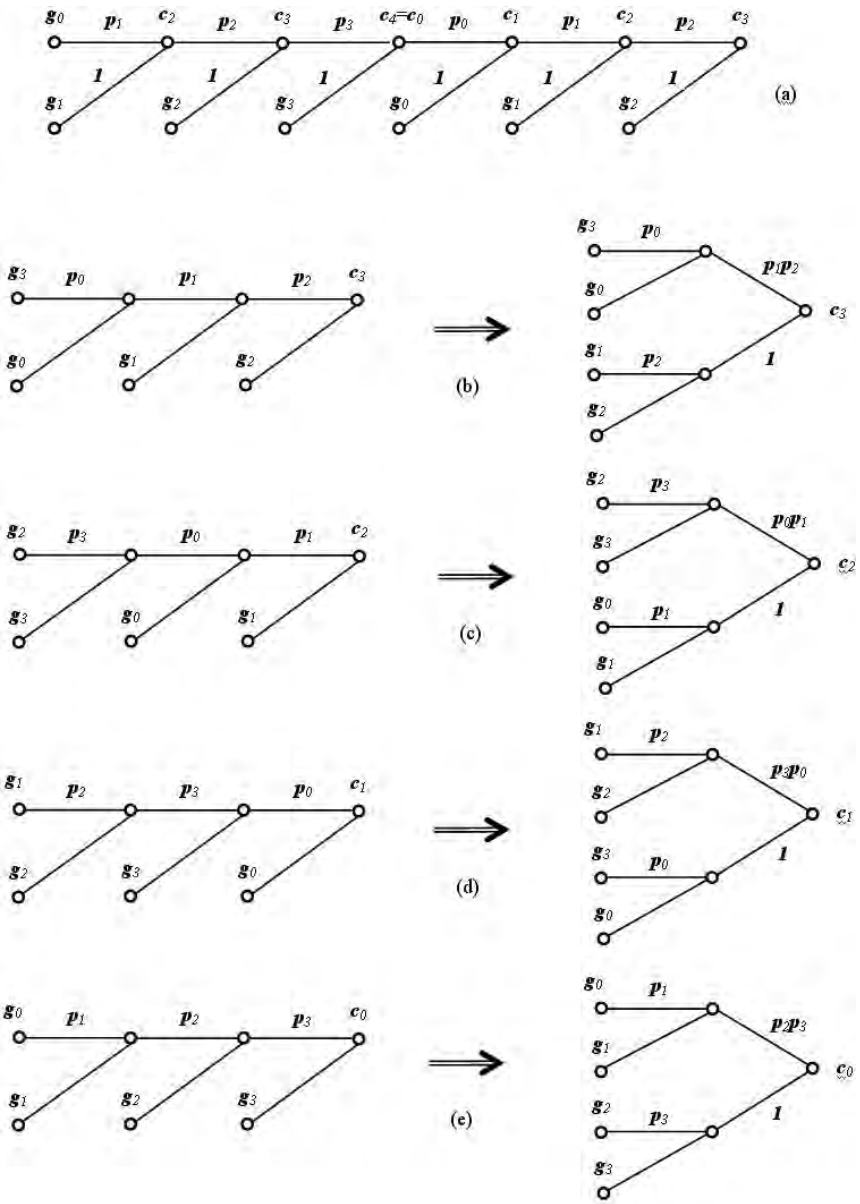


Рис.2. FDD для сумматора по модулю 15 и этапы декомпозиции: (a) – исходная FDD,
(b) – перенос c_3, (c) – перенос c_2, (d) - перенос c_1, (e) - перенос c_0.

тельно к функциям переноса сумматора по модулю (2^n-1) это соответствует левой и правой частям FDD), а также вводится дополнительная логика. Параллельная реализация обеспечивает более быстродействующую структуру с некоторым увеличением аппаратных затрат.

Таким образом, исходя из общей диаграммы, изображенной на рис. 1(b), возможно построить FDD для каждой из функций переноса c_i для сумматора по модулю (2^n-1). При этом необходимо руководствоваться основным принципом, согласно которому, при построении диаграммы для переноса c_i разрез должен осуществляться по ребру p_i [3]. Данное правило является определяющим при построении FDD для функций переноса сумматоров по модулю (2^n-1).

В качестве примера, на рис. 2(a) приведена общая FDD для сумматора по модулю m=15. Диаграммы для функций переноса c_3, c_2, c_1 и c_0 формируются разрывом соответствующих ребер p_3, p_2, p_1 и p_0. На рис. 2(b), 2(c), 2(d), 2(e) показаны FDD для каждого из переносов и полученные из них декомпозированные диаграммы.

Схемная реализация для переноса c_3, соответствующая построенной оптимизированной FDD, приведена на рис. 3. При этом каждое звено диаграммы соответствует реализации на основе элемента 2И-ИЛИ (рис. 3(a)). Для КМОП-базиса, используя правила де Моргана, можно легко получить схему с использованием элементов 2И-ИЛИ-НЕ/2ИЛИ-И-НЕ(см. рис. 3(b)).Аналогичным образом, можно получить схемные реализации для других функций переноса.

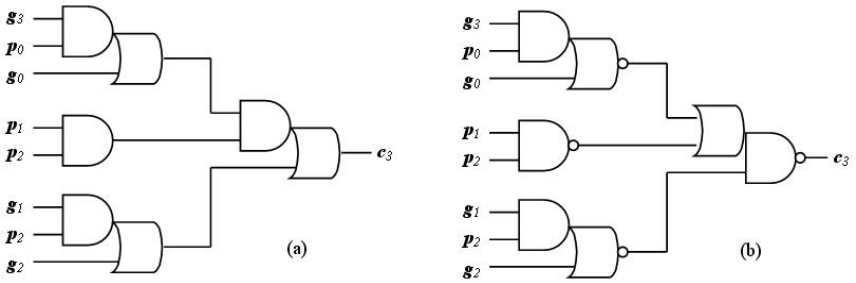


Рис.3. Аппаратная реализация функции переноса:
• c_3 в базисе элементов 2И-ИЛИ (a),
• в базисе элементов 2И-ИЛИ-НЕ/2ИЛИ-И-НЕ (b).

Следует отметить, что для полной реализации модулярного сумматора необходимо введение дополнительной логики, обеспечивающей единственное представление нуля. Эта логика практически не приводит к увеличению задержки на критическом пути, так как реализуется параллельно с основной структурой.

Таким образом, сумматор по модулю (2^n-1) включает следующие основные стадии:

- блок, реализующий функции генерации (g_i) и распространения переноса (p_i) в соответствии с формулами (2) и (3);

- блок, реализующий функции переноса c_i и построенный на основе декомпозированных FDD;

- блок, вычисляющий финальное значение модулярной суммы ($S_i = a_i \wedge b_i \wedge c_i$).

Представленная структура означает, что сумматор по модулю (2^n-1) не уступает по быстродействию двоичным сумматорам такой же разрядности.

Построение быстродействующих сумматоров по модулю вида (2^n+1) на основе BDD.

Аналогичным образом могут быть построены сумматоры по модулям вида (2^n+1). Суммирование по модулю (2^n+1) преобразовывается в суммирование по модулю 2^n следующим образом [4]:

$$\left|a+b\right|_{2^n+1} = \left|a+b+z\right|_{2^n} + \bar{c}_{out}, \quad (5)$$

где $z = 2^{n+1} - \left(2^n+1\right) = 2^n - 1$ – дополнение значения модуля до 2^{n+1}, а c_{out} - старший бит суммы $\left(a+b+z\right)$. Из соотношения (5) следует, что для определения суммы $\left|a+b\right|_{2^n+1}$ необходимо сначала вычислить сумму $\left(a+b+z\right)$, а затем прибавить к ней инверсию старшего бита, задействовав при этом, лишь младшие n бит (т.е. взяв ее по модулю 2^n). Следует отметить, что старший бит результата определяется специальным образом для обеспечения единственности представления нуля.

Добавление константы z целесообразно реализовывать с помощью сумматора с запоминанием переносов (CSA-carry-save adder), который сводит сложение трех чисел к сложению двух за время за-

держки полного одноразрядного сумматора [5], [6]. Поскольку данная константа известна заранее для каждого конкретного значения модуля и имеет вид 2^n-1, то одноразрядные сумматоры могут быть заменены на полусумматоры (half-adder, HA), из которых *n* имеют в качестве входного переноса “1” а один является обычным полусумматором (рис. 4).

Следующим шагом необходимо сложить полученные два числа *u* и *v*, определив тем самым значение $\bar{c}_{out}$. Для этого снова воспользуемся известными булевыми функциями генерации и распространения переноса (2) и (3).

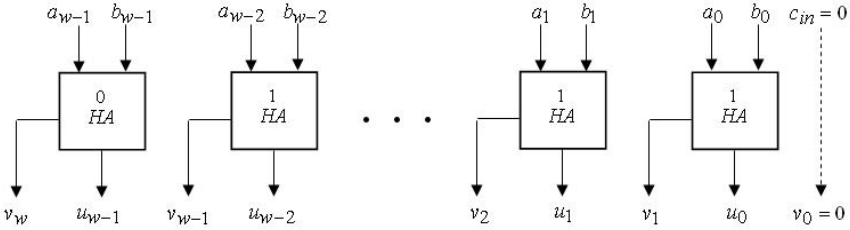


Рис.4. Блок добавления константы z в сумматорах по модулю (2^n+1).

Как и в сумматорах по модулю вида (2^n-1), воспользуемся диаграммами двоичных решений для вычисления c_{out}. Поскольку теперь, в качестве входного переноса необходимо использовать инверсию старшего бита $\bar{c}_{out}$, то реализация циклического переноса на основе BDD в данном случае представляется невозможной, вследствие немонотонного базиса. Поэтому для вычисления c_{out} используется обычная FDD для двоичного сумматора на рис. 1(а). Очевидно, что инверсия $\bar{c}_{out}$ может быть легко получена выбором соответствующих КМОП-элементов при реализации данного блока.

Вычислив $\bar{c}_{out}$, необходимо определить сигналы переноса из младших разрядов с учетом $c_{in} = \bar{c}_{out}$. Для этого достаточно найти их значения исходя из условия $c_{in} = 0$:

$$c'_1 = g_0 = 0,$$

$$c'_2 = g_1,$$

$$c'_3 = g_2 \vee (p_2 \wedge c_2) = g_2 \vee (p_2 \wedge g_1),$$

$$\dots$$

$$c'_n = g_{n-1} \vee (p_{n-1} \wedge c_{n-1}) = g_{n-1} \vee (p_{n-1} \wedge g_{n-2}) \vee (p_{n-1} \wedge p_{n-2} \wedge g_{n-3}) \vee \dots \vee (p_{n-1} \wedge p_{n-2} \wedge \dots \wedge p_2 \wedge g_1),$$

а затем скорректировать эти промежуточные переносы следующим образом:

$$c_1 = c'_1 \vee (p_0 \wedge \overline{c}_{out}) = p_0 \wedge \overline{c}_{out},$$

$$c_2 = c'_2 \vee ((p_1 \wedge p_0) \wedge \overline{c}_{out}) = c'_2 \vee (p_{10} \wedge \overline{c}_{out}),$$

$$c_3 = c'_3 \vee ((p_2 \wedge p_1 \wedge p_0) \wedge \overline{c}_{out}) = c'_3 \vee (p_{210} \wedge \overline{c}_{out}),$$

$$\dots$$

$$c_n = c'_n \vee ((p_{n-1} \wedge p_{n-2} \wedge \dots \wedge p_0) \wedge \overline{c}_{out}) = c'_n \vee (p_{n-1\dots0} \wedge \overline{c}_{out}),$$

где значения $c'_1, \dots, c'_n$ и соответствующие конъюнкции $p_{10}, p_{210}, \dots, p_{n-1\dots0}$ вычисляются параллельно с $\overline{c}_{out}$ на основе той же самой FDD и не вносят дополнительную задержку. Таким образом, вычисление сигналов переноса при условии $c_{in} = \overline{c}_{out}$ реализуется за одну стадию с помощью КМОП-элемента 2И-ИЛИ, как показано на рис. 5.

Стоит также отметить, что как и в сумматоре по модулю вида (2^n-1) в данном случае требуется введение дополнительной логики для вычисления старшего бита результата и обеспечения единственного представления нуля, которая также функционирует параллельно с основной структурой и не вносит дополнительной задержки.

Таким образом, сумматор по модулю (2^n+1) включает следующие основные стадии:

- блок, реализующий добавление константы z к сумме входных операндов на основе CSA;

- блок, реализующий функции генерации (g_i) и распространения переноса (p_i) в соответствии с формулами (2) и (3);

- блок, реализующий функции переноса c'_i и $\overline{c}_{out}$ (при условии

$c_{in} = 0$) и построенный на основе декомпозированных FDD;

- блок, реализующий коррекцию сигналов переноса с учетом $c_{in} = \overline{c}_{out}$ на основе КМОП-элемента 2И-ИЛИ;

- блок, вычисляющий финальное значение суммы ($S_i = a_i \wedge b_i \wedge c_i$).

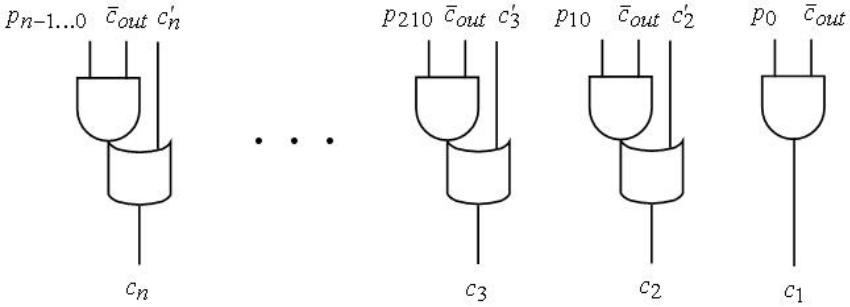


Рис.5. Блок вычисления переносов с учетом $c_{in} = \overline{c}_{out}$.

Представленная структура означает, что задержка сумматора по модулю (2^n+1) отличается от задержки обычных двоичных сумматоров такой же разрядности лишь на сумму задержек одного полусумматора и КМОП-элемента 2И-ИЛИ.

Методы построения быстродействующих модулярных умножителей на основе алгоритма Бута.

Существуют различные методы построения модулярных умножителей. Так, например, одним из распространенных методов является метод индексного (или дискретно-логарифмического) модулярного умножения [7]. Данный подход основан на преобразовании операндов в соответствующие индексы и заменой операции модулярного умножения на операцию модулярного сложения над полученными индексами. К недостаткам индексных умножителей можно отнести то, что они могут быть построены только для модулей, являющихся простыми числами, а также значительное возрастание общей площади умножителя при увеличении значения модуля [8].

Построение быстродействующих умножителей по модулю вида (2^n-1)

Для отдельных значений модулей при построении быстрых умножителей могут быть эффективно использованы такие же принципы

реализации, что и для быстродействующих двоичных умножителей, а именно, модифицированный алгоритм Бута [9], [10] и дерево Уоллеса [6].

В общем случае умножитель Бута состоит из селектора, генератора частичных произведений (декодер Бута), массива сумматоров для сложения частичных произведений. С помощью селектора осуществляется разложение множителя на отдельные группы бит (например, для разрежения спектра степеней в сумме частичных произведений через разряд используется группировка по три бита). Декодер Бута, в зависимости от значений полученных после селектора, осуществляет необходимые преобразования над вторым операндом. Массив сумматоров целесообразно реализовывать на основе дерева Уоллеса с финальным сумматором, построенным на основе алгоритмов быстрого сложения (например, на основе BDD-технологии). Обобщенная структура умножителя Бута представлена на рис. 6.

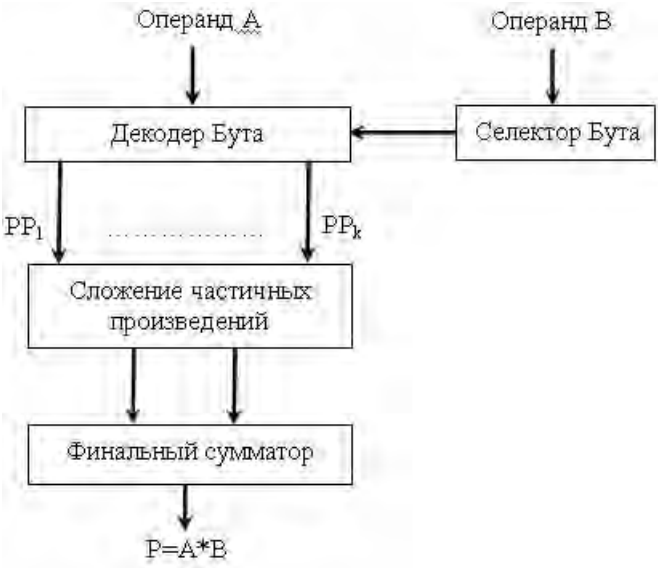


Рис.6. Обобщенная структурная схема умножителя на основе алгоритма Бута.

Особенности операций по модулю вида (2^n-1) позволяют эффективно реализовывать каждый элемент умножителя. В этом случае, частичные произведения $PP_1,\ldots,PP_k$ и конечное произведение P будут вычислены, соответственно, по модулю (2^n-1). Селектор для модулярного умножителя строится таким же образом, как и селектор для обычного позиционного умножителя.

1. Реализация декодера Бута в умножителях по модулю (2^n-1)

Декодер Бута формирует значения:

$$\left\{|A|_{2^n-1},|2\cdot A|_{2^n-1},|-2\cdot A|_{2^n-1},|-A|_{2^n-1},0\right\} \quad (6)$$

при условии, что анализ операнда B выполняется на основе триад бит.

Покажем, что получить значения частичных произведений $\left\{|2\cdot A|_{2^n-1},|-2\cdot A|_{2^n-1},|-A|_{2^n-1}\right\}$ для декодера Бута можно комбинируя операцию инверсии каждого бита операнда A и/или его циклический сдвиг влево на соответствующее количество бит.

Рассмотрим генерацию частичных произведений в (6) более подробно. Пусть есть число $A=\sum_{i=0}^{n-1}a_i\,2^i$, лежащее в диапазоне $\left[0,2^n-1\right)$. Выражение соответствующее умножению числа A на число 2, можно записать в следующем виде:

$$A\cdot 2=\left(\sum_{i=0}^{n-1}a_i\,2^i\right)2=\sum_{i=0}^{n-1}a_i\,2^{i+1}=a_{n-1}2^n+\sum_{i=0}^{n-2}a_i\,2^{i+1} \quad (7)$$

Из формулы (7) можно найти значение выражения $|A\cdot 2|_{2^n-1}$:

$$|A\cdot 2|_{2^n-1}=\left|a_{n-1}2^n+\sum_{i=0}^{n-2}a_i\,2^{i+1}\right|_{2^n-1}=\left|\left|a_{n-1}2^n\right|_{2^n-1}+\left|\sum_{i=0}^{n-2}a_i\,2^{i+1}\right|_{2^n-1}\right|_{2^n-1} \quad (8)$$

Вычислим значение каждого слагаемого формулы (8), принимая во

внимание, что $\left|2^n\right|_{2^n-1} = 2^0$, а $\left|a_{n-1}\right|_{2^n-1} = a_{n-1}$:

$$\left|a_{n-1}2^n\right|_{2^n-1} = \left|\left|a_{n-1}\right|_{2^n-1} \cdot \left|2^n\right|_{2^n-1}\right|_{2^n-1} = a_{n-1}2^0 \quad (9)$$

$$\left|\sum_{i=0}^{n-2} a_i 2^{i+1}\right|_{2^n-1} = \sum_{i=0}^{n-2} a_i 2^{i+1} \quad (10)$$

На основании полученных формул (9) и (10), умножение некоторого числа A на 2 по модулю (2^n-1) соответствует циклическому сдвигу влево на один бит.

Используя свойство модулярной арифметики для отрицательных чисел [6]: $\left|-Y\right|_X = \left|X-Y\right|_X$, можно показать, что значение выражения $\left|-A\right|_{2^n-1}$ соответствует инверсии всех битов исходного числа, т.е.

$$\left|-A\right|_{2^n-1} = \overline{a_{n-1}}\,\overline{a_{n-2}}\ldots\overline{a_0}$$

Выражение $\left|-A\cdot 2\right|_{2^n-1}$ соответствует циклическому сдвигу влево исходного слова, с инверсией каждого бита которого:

$$\left|-A\cdot 2\right|_{2^n-1} = \overline{a_{n-2}}\,\overline{a_{n-3}}\ldots\overline{a_0}\,\overline{a_{n-1}}$$

Если значение A равно нулю, тогда $\left|A\right|_{2^n-1} = 0 = \underbrace{000\ldots00}_{n\ бит}$.

2. Реализация блока суммирования частичных произведений (мультиоперандный сумматор по модулю вида (2^n-1))

При анализе возможных методов реализации данного блока важно отметить, что частичные произведения, полученные на выходах декодера, имеют одинаковую разрядность. Этот факт позволяет для сложения частичных произведений эффективно использовать дерево Уоллеса. Как известно, дерево Уоллеса для суммирования чисел в двоичной системе счисления строится на основе сумматоров с запоминанием переносов (CSA). Учитывая свойство (1), аналогичным образом может быть построен сумматор с запоминанием пе-

реносов для модулей вида $2^n - 1$. Для этого достаточно использовать в качестве входного переноса следующего каскада выходной разряд переноса сумматора с запоминанием переносов, как показано на рис. 7. Очевидно, что задержка такого сумматора, как и его двоичного варианта, определяется максимальной задержкой одноразрядного полного сумматора (full-adder, FA), которую мы обозначим через $t_{з\,\max}^{FA}$.

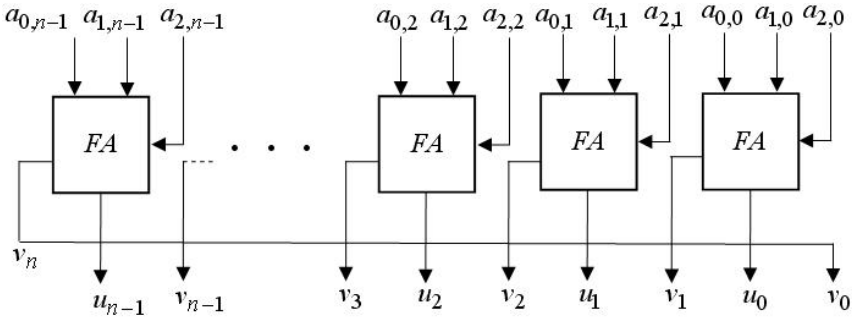


Рис. 7. Сумматор с запоминанием переносов для модулей вида $2^n - 1$.

На выходе такого сумматора формируются два числа одинаковой разрядности, что дает возможность построить регулярную структуру дерева Уоллеса, поскольку все входные и выходные операнды на каждом из каскадов дерева имеют одинаковую разрядность (рис. 8). Быстродействие данной структуры блока суммирования частичных произведений зависит только от количества операндов и не зависит от их разрядности, а значит и от значения модуля. Поскольку глубина дерева или количество этапов суммирования определяется как $\Theta(\log_2 k)$, а задержка любого из уровней структуры равна $t_{з\,\max}^{FA}$, то общую оценку быстродействия дерева Уоллеса для модулей вида $2^n - 1$ можно представить в виде $\Theta(\log_2 k) \cdot t_{з\,\max}^{FA}$, где k – количество входных операндов, $t_{з\,\max}^{FA}$ - задержка одноразрядного полного сумматора.

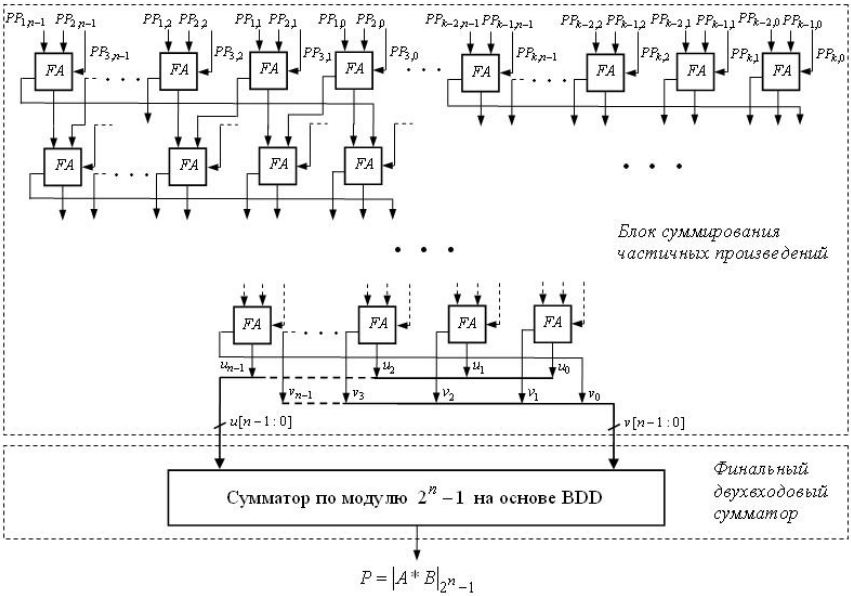


Рис. 8. Реализация блока суммирования частичных произведений (мультиоперандного сумматора по модулю вида 2^n-1) на основе дерева Уоллеса.

Таким образом, преодолеваются основные недостатки дерева Уоллеса: нерегулярность структуры и резкое возрастание площади при увеличении количества операндов, которые необходимо сложить. Финальный сумматор может быть построен на основе BDD-технологии, как это показано ранее в данной работе.

Из проведенного анализа можно сделать вывод, что умножители по модулю вида 2^n-1 могут быть построены также эффективно с точки зрения быстродействия и занимаемой площади, как и обычные двоичные умножители такой же разрядности.

Следует отметить, что принципы построения быстрых умножителей на основе алгоритма Бута также применимы при реализации умножителя по модулю вида 2^n+1.

Литература

1. Kornilov A., Isaeva T., Syngaevsky V. Carry Circuit Depth Optimization by BDD Based Decomposition // Proc. of PATMOS'97 Workshop – Louvain-la-Neuve, Belgium, Sep. 8-10. – 1997.-P.89-98.

2. Исаева Т.Ю., Корнилов А.И. Алгоритм декомпозиции логических функций, ориентированный на синтез быстродействующих цифровых устройств// Информационные технологии. - №4, 2001. - С.26-31.
3. Корнилов А.И., Исаева Т.Ю., Семенов М.Ю. Методы логического синтеза сумматоров с ускоренным переносом по модулю (2^n-1) на основе BDD-технологии // Известия ВУЗов. Электроника. – 2004. - №3. – С. 54-60.
4. A. A. Hiasat. High-Speed and Reduced-Area Modular Adder Structures for RNS // IEEE Transactions on Computers, vol. 51, no. 1, January 2002.
5. T. Kim, W. Jao, S. Tjiang. Circuit Optimization Using Carry-Save-Adder Cells // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 17, no. 10, October 1998.
6. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. - М: МЦНМО, 2001. – 960с.
7. D. Radhakrishnan, Yong Yuan. Novel Approaches to the Design of VLSI RNS Multipliers // IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing, Vol. 39. No 1. January 1992. P.: 52-57.
8. Корнилов А.И., Семенов М.Ю., Ласточкин О.В. Принципы построения модулярных индексных умножителей // Известия ВУЗов. Электроника. – 2004. - №2. – С. 48-55.
9. Угрюмов Е. Цифровая схемотехника – СПБ.: БХВ-Петербург, 2001. – 528с.
10. H.I. Saleh, A.H. Khalil, M.A. Ashour, A.E. Salama. Novel serial-parallel multipliers // IEEE Proc.-Circuits Devices Systems, Vol. 148, No. 4, August 2001, P.: 183-189.

УДК. 681.3

А.А. Евдокимов

Реализация модулярных нейронных вычислительных структур на базе ПЛИС

(Невинномысский технологический институт (филиал) Северо-Кавказского государственного технического университета)

Предложена реализация Нейронной Сети Конечного Кольца (НСКК) на основе ПЛИС Xilinx. Показано, что данная реализация НСКК имеет преимущества по времени преобразования и аппаратным затратам по сравнению с известными схемными решениями на базе ПЛИС.
Realization of Finite Ring Neural Network (FRNN) on the basis of FPGA Xilinx is offered. It is shown, that the given realization of FRNN has advantages on time of transformation and apparatus expenses in comparison with known circuit decisions on the basis of FPGA.

С развитием распределенных вычислительных сетей появляются новые приложения системы остаточных классов (СОК): системы порогового доступа [1, 2], надежного хранения информации [3], динамического управления загрузкой компьютерной сети [3], секретной передачи информации. Новый этап развития систем, функционирующих в системе остаточных классов, связан с широкими логическими возможностями современных ПЛИС и использованием аппарата нейронных сетей в немодульных операциях. Исследования в области разработки нейросетевых моделей СОК ведутся,

как на межразрядном [4], так и на внутриразрядном [5, 6] уровнях системы остаточных классов.

В соответствии с итеративным алгоритмом понижения разрядности числа, представленного в позиционной системе счисления (ПСС)

$$A(j+1)=\sum_{i=0}^{\lceil \log_2 A(j)\rceil}\left|2^i\right|_p^+\left\{A(j)\right\}^{[i]}, \qquad (1)$$

где $\{A(j)\}^{[i]}$ – оператор извлечения *i*-го разряда двоичного представления $A(j)$,

$\left|2^i\right|_p^+$ – операция вычисления остатка целочисленного деления по модулю *p*, процесс модулярной редукции может быть реализован на нейронной сети (НС) прямого распространения сигнала [4]. Перевод задачи модулярного сокращения в нейросетевой логический базис позволяет дополнительно распараллелить вычисления внутри разрядов системы остаточных классов. Однако характеристики реализации НС на базе комбинационного сумматора и ПЗУ с весовыми коэффициентами хуже, чем у табличного вычислителя. Так, один слой данной НС для p = 13 и n = $\left\lceil \log_2 A(0)\right\rceil+1$ = 8 может быть реализован на базе ПЛИС фирмы Xilinx на 29 просмотровых таблицах (LUT) с временем на полное сокращение по модулю 13 равным 648 нс [7], когда аналогичный табличный преобразователь с декомпозицией числа на группы младших и старших разрядов имеет быстродействие 26 нс при аппаратурных затратах в 80 LUT [8].

Поскольку одна итерация (1) соответствует одному слою НС, то избавиться от межразрядных переносов в слое невозможно. Однако использование ПЛИС позволяет распределить весовые коэффициенты $\left|2^i\right|_p^+$ на поцифровом уровне между слоями НС (рисунок 1). Тогда k-ый нейрон в каждом слое данной сети аккумулирует одноразрядные операнды, соответствующие цифрам числа $\left|2^i\right|_p^+\left\{A(j)\right\}^{[i]}$ в двоичном представлении, т. е.

$$\left\| \left[\frac{\left| 2^i \right|_p^+ \left\{ A(j) \right\}^{[i]}}{2^k} \right] \right\|_2^+ .$$

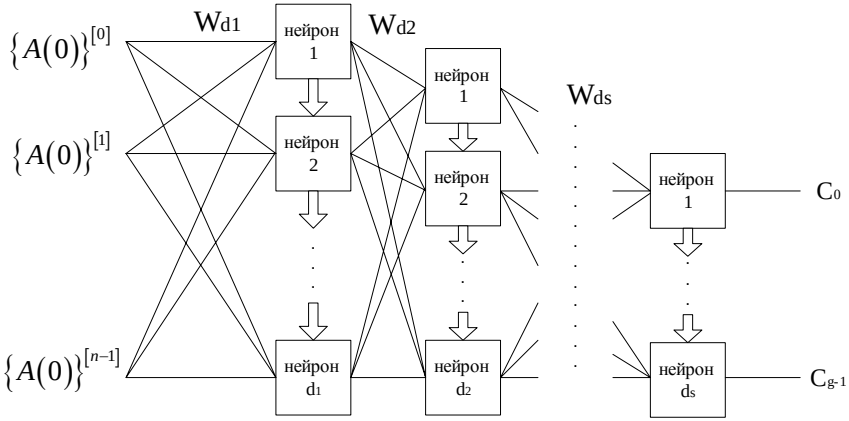


Рис. 1. Нейронная сеть конечного кольца с латеральными связями

Рассмотрим принципы построения нейронной сети конечного кольца (НСКК) с одноразрядными весовыми коэффициентами.

Число нейронов в *l*-ом слое $d = \left[\log_2 \left(\sum_{i=0}^{n-1} \left| 2^i \right|_p^+ \right) \right] + 1,$ где $n = \left[\log_2 A(l-1) \right] + 1$. Пронумеруем нейроны в *l*-ом слое от 0 до (d – 1). Тогда связи между входным слоем НС и нейронами первого слоя определяется матрицей $M = (m_{ik})$, где

$m_{ik} = \left\| \left[\frac{\left| 2^i \right|_p^+}{2^k} \right] \right\|_2^+$, где $i = 0 \div n-1,\ k = 0 \div d-1.$

Пример: для p = 13, n = 8, d = 6 матрица весовых коэффициентов

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Нейрон представляет собой многоместный сумматор одноразрядных операндов a_i. Нейрон имеет h входов и $\lceil \log_2 h \rceil + 1$ выходов. Значение j-ого выхода k-ого нейрона в *l*-ом слое определяется

$${}^{l}a_f^k = \left\| \left\lfloor \frac{\sum_{i=0}^{h-1} a_i}{2^f} \right\rfloor \right\|_2^+, \quad f = 0 \div h - 1.$$

Из выходов ${}^{l}a_0^k$ формируется входное значение на (*l* + 1) слой НС. Выходы ${}^{l}a_1^k$, ${}^{l}a_2^k$, …, ${}^{l}a_{h-1}^k$ заводятся на входы нейронов под номерами k + 1, k + 2, …,

k + h – 1, соответственно, в *l*-ом слое. Так как число нейронов в *l*-ом слое равно d, то при переносе разрядов с нейронов d – 2, d – 3 и т. д. на вход нейрона под номером

d – 1 последний реализуется в виде логического элемента «ИЛИ». При отсутствии переносов на (d – 1)-ый нейрон числом больше 1 выход ${}^{l}a_1^{d-2}$ с нейрона под номером

(d – 2) заводится непосредственно на вход (*l* + 1)-го слоя.

Для построения на базе ПЛИС типа FPGA фирмы Xilinx основных функциональных блоков модулярного нейросопроцессора разрабо-

таны VHDL-описания данных устройств: один слой НСКК на основе комбинационного сумматора (FRNN_Perform) [7], НСКК на основе однобитных нейронов (слой frnnmain (рисунок 2) и сеть frnn13full) [7], табличный преобразователь чисел из позиционного двоичного представления в СОК (pns2rns) [8], преобразователь чисел из СОК в полиадическую систему счисления на основе ПЗУ (rns2mrs) [9]. Для синтеза функциональных блоков модулярного нейропроцессора на основе разработанных описаний устройств на языке VHDL получены соответствующие методики. Разработанные методики дополняют предлагаемые фирмой Xilinx методы построения вычислителей на базе ПЛИС.

Рассмотрим методику построения НСКК с однобитовыми весовыми коэффициентами на основе разработанного объекта frnnmain:

1. Определяется разрядность модуля НСКК g и разрядность преобразуемых данных n. Вычисляется максимальная разрядность результата d первой итерации при разрядности входного вектора n по модулю p: $d = \left\lceil \log_2 \left(\sum_{i=0}^{n-1} \left| 2^i \right|_p^+ \right) \right\rceil + 1.$

2. Вычисляются значения весовых коэффициентов НСКК $\left| 2^i \right|_p$, i = 0, 1, 2, …, n – 1, в двоичном представлении и заносятся во второй столбец таблицы 1.

3. Заполнение таблицы 1 производится последовательно по строкам, соответствующим номеру входа НСКК, по следующему правилу: ячейка отмечается (крестиком), если номеру входа соответствует весовой коэффициент, в двоичном представлении которого разряд под номером нейрона в слое НСКК равен единице (нумерация разрядов начинается с 1).

4. Ячейки строки «Разрядность выходных данных нейрона» и последних d – 1 строк таблицы 1 заполняются по столбцам, соответствующим номеру нейрона в слое НСКК. В ячейку по первому нейрону заносится число равное разрядности суммы вышележащих по столбцу отметок U. Если U больше единицы, то последовательно отмечаются ячейки строки № 0 (латеральной связи) столбцов 2, 3, …, U. Разрядность выходных данных нейрона по второму столбцу номера нейрона подсчитывается посредством раз-

рядности суммы отмеченных ячеек, как вышележащих по столбцу № 1, так и нижележащих (боковые связи) по данному столбцу. Если разрядность суммы V отмеченных ячеек больше 1, тогда последовательно отмечаются ячейки строки № 1 (латеральной связи) столбцов 3, 4, …, V+1. Процесс заполнения ячеек продолжается до столбца под номером d – 1. Ячейка, соответствующая строке «Разрядность выходных данных нейрона» и столбцу d – 1 не заполняется.

Таблица 1.

Таблица для построения НСКК на основе однобитных нейронов

Номер входа НСКК	Весовой коэффициент НСКК	Номер нейрона в слое НСКК			
		0	2	...	d – 1
0	$\left\|2^0\right\|_p$				
1	$\left\|2^1\right\|_p$				
2	$\left\|2^2\right\|_p$				
...	...				
n – 1	$\left\|2^{n-1}\right\|_p$				
Разрядность выходных данных нейрона					
Номер нейрона латеральной связи	0				
	2				
	...				
	d – 2				

5. В описании объекта (entity) и компонентов модели frnnmain изменяем значения настроечных констант входного регистра-защелки со сбросом regsmpl и тристабильного буфера regdef: для regsmpl присваиваем j = n – 1, для regdef изменяем j = d – 1.

6. В соответствии с данными таблицы 1 производим построение нейронов, последовательно начиная с нейрона под номером 1. Последовательность необходима по причине возможного

повторения структур нейронов. В объекте (entity) нейрона создается два порта, определяющих вход и выход нейрона. Число входов определяется количеством отмеченных ячеек по соответствующему столбцу, а число выходов – значению строки «Разрядность выходных данных нейрона». В архитектуре (architecture) объекта нейрона создается блок, в котором посредством оператора условного параллельного присваивания (<= … when … else) описывается поведение объекта: выход равен числу в двоичном представлении активных входов нейрона. Столбец d – 1 может содержать одну или две отметки, поскольку d – 1 есть последний нейрон в слое. При двух и более входах его можно реализовать на логическом элементе «ИЛИ».

7. В объявлении блока архитектуры (architecture) устройства frnnmain описываются созданные нейроны в качестве компонентов, а в описании поведения объекта (после оператора begin) описывается структура слоя НСКК с помощью оператора соединения портов (port map) в соответствии с таблицей 1.

Рассмотрим пример построения НСКК для модуля p = 13: g = 4, n = 8, M = 3, d = 6. В таблице 2 первая часть таблицы (до строки «Разрядность выходных данных нейрона») заполнена в соответствии с двоичным представлением весовых коэффициентов НСКК: 0001 – отмечается ячейка № 0 строки № 0; 0010 – отмечается ячейка № 1 строки № 1; 1011 – отмечаются ячейки № 0, 1, 3 строки № 7. Процесс заполнения строки «Разрядность выходных данных нейрона»: сумма отмеченных ячеек по первому столбцу равна 3, соответственно разрядность равна 2; следовательно, в строке № 0 латеральных связей нейронов отмечаем ячейку по столбцу № 1. Число однобитных выходов второго нейрона равно 3, следовательно, в строке латеральных связей № 1 отмечаем ячейки по 2, 3 столбцам. На рисунке 2 изображена структура разработанного слоя НСКК.

Моделирование разработанных преобразователей по модулю 13 в среде Xilinx ISE v.5.2 и ModelSim показало, что наилучшее соотношение аппаратурных затрат и быстродействия характерно для модели НСКК (таблица 3), построенной на основе однобитных весовых коэффициентов, что объясняется относительно высоким коэффициентом использования просмотровых таблиц за счет индивидуального подхода к проектированию на уровне элементарных примитивов ПЛИС. Из результатов моделирования, представленных на рисунке 3, видно, что с ростом разрядности чисел преиму-

щество модели frnnmain в смысле аппаратурных затрат возрастает приблизительно в два раза по сравнению с FRNN_Perform.

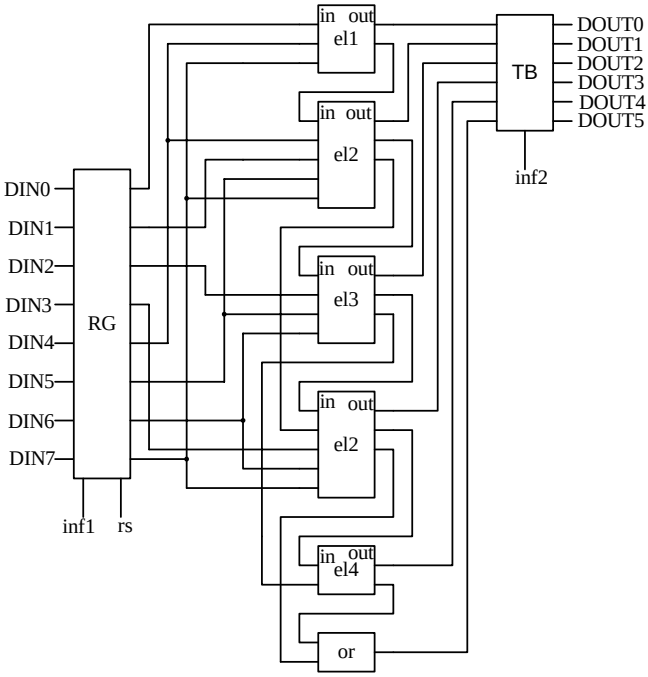


Рис. 2. Слой НСКК на базе одноразрядных весовых коэффицентов

Таким образом, для повышения производительности модулярного нейропроцессора необходимо использовать нейронную сеть, а не слой нейронов. Время работы слоя и сети равны 51 нс и 22 нс, соответственно, когда аппаратурные затраты 19 LUT и 30 LUT, соответственно. Произведение затрачиваемых ресурсов дает 969 LUT·нс для слоя и 660 LUT·нс для сети, что говорит об эффективности использования нейронной сети.

Сравнение характеристик преобразователей по модулю на базе НСКК с одноразрядными весами с известными схемными решениями на базе ПЛИС фирмы Xilinx (сумматор с разрядно-кристальным управлением [10], модификация Кима-Собельмана [11]) показало преимущество разработанной нейронной сети (таблица 4).

Таблица 2.

Таблица для построения НСКК на основе однобитных нейронов при p = 13

Номер входа НСКК	Весовой коэффициент НСКК	Номер нейрона в слое НСКК					
		0	1	2	3	4	5
0	0001	X					
1	0010		X				
2	0100			X			
3	1000				X		
4	0011	X	X				
5	0110		X	X			
6	1100			X	X		
7	1011	X	X		X		
Разрядность выходных данных нейрона		2	3	3	3	2	
Номер нейрона латеральной связи	0		X				
	1			X	X		
	2				X	X	
	3					X	X
	4						X

Таблица 3.

Временные и аппаратурные затраты устройств сокращения по модулю 13 восьмиразрядного числа

	FRNN_Perform	frnnmain/frnn13full	pns2rns
Flip Flop, шт.	22	8/8	8
LUT, шт.	29	19/30	80
TBUF, шт.	4	6/4	12
Время работы устройства, нс	648	51/22	26

а) б)

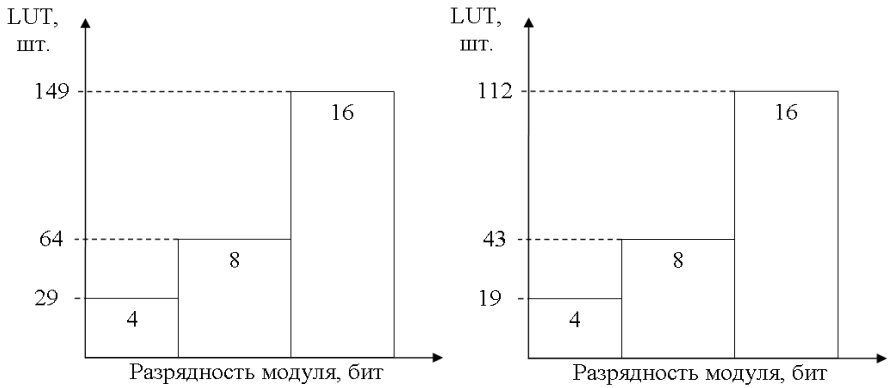


Рис. 3. Зависимость аппаратурных затрат НСКК FRNN_Perform (а) и frnnmain (б) от разрядности модуля

Таблица 4.

Сравнительный анализ преобразователей по модулю, реализованных на базе ПЛИС фирмы Xilinx

Параметры устройств	Умножители по модулю				Преобразователь с разрядно-кристальным управлением (основание 31)	НСКК на основе однобитных весовых коэффициентов (основание 31)
	Модификация метода Кима - Собельмана		НСКК на основе однобитных весовых коэффициентов			
	Разрядность модуля		Разрядность модуля			
	8	16	8	16		
Аппаратурные затраты,LUT шт.	192	640	107	372	174	35
Время работы, нс	80	192	67	115	34.8	22

Таким образом, подход в построении модулярных вычислителей в нейросетевом логическом базисе на базе ПЛИС является более эффективным по сравнению с традиционным подходом разрядно-кристального управления и табличной арифметики.

Литература

1. Червяков Н. И., Евдокимов А. А. Нейросетевой генератор криптографических ключей пороговой схемы разделения секрета// Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2004, № 10. – С. 62 – 67.
2. Червяков Н. И., Евдокимов А. А. Динамическая система пролонгированной безопасности // Инфокоммуникационные технологии. – Самара: Изд-во ПГАТИ, 2004, № 4. – С. 31 – 35.
3. Червяков Н. И., Евдокимов А. А. Пороговое разделение файла на базе китайской теоремы об остатках // Инфокоммуникационные технологии. – Самара: Изд-во ПГАТИ, 2004, № 1. – С. 38 – 43.
4. Червяков Н. И., Сахнюк П. А., Шапошников А. В., Макоха А. Н. Нейрокомпьютеры в остаточных классах. Кн. 11: Учеб. пособие для вузов. – М.: Радиотехника, 2003. – 272 с.
5. Червяков Н.И., Ремизов С.И. Структуры нейронных сетей конечного кольца// Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2004, № 12. – С. 21 – 30.
6. Евдокимов А. А. Согласованность геометрических моделей системы остаточных классов и нейронной сети CMAC// Труды участников международной школы-семинара по геометрии и анализу памяти Н. В. Ефимова. – Ростов-на-Дону: Изд-во ООО «ЦВВР», 2004. – С. 188 – 190.
7. Евдокимов А.А. Свидетельство об отраслевой регистрации разработки № 4086 от 03 декабря 2004 г. на разработку «Описания нейронной сети конечного кольца «VHDL»». Номер гос. регистрации 50200401423 от 09 декабря 2004 г. (Министерство образования РФ. Государственный координационный центр информационных технологий. Отраслевой фонд алгоритмов и программ)
8. Евдокимов А.А. Свидетельство об отраслевой регистрации разработки № 4087 от 03 декабря 2004 г. на разработку «Описание устройства для преобразования чисел из позиционного представления в систему остаточных классов «VHDL»». Номер гос. регистрации 50200401424 от 09 декабря 2004 г. (Министерство образования РФ. Государственный координационный центр информационных технологий. Отраслевой фонд алгоритмов и программ)
9. Евдокимов А.А. Свидетельство об отраслевой регистрации разработки № 4088 от 03 декабря 2004 г. на разработку «Описание устройства для перевода чисел из системы остаточных классов в обобщенную позиционную систему счисления «VHDL»». Номер гос. регистрации 50200401425 от 09 декабря 2004 г. (Министерство образования РФ. Государственный координационный центр информационных технологий. Отраслевой фонд алгоритмов и программ)
10. Venkatesan R. FPGA implementation of RNS structure// A Thesis Sub-

mitted to the Faculty of Graduate Studies through the Department of Electrical Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science at the University of Windsor, 1994. – 124 p.
11. Beuchat J.-L., Muller J.-M. Modulo M multiplication-addition: algorithms and FPGA implementation// ELECTRONICS LETTERS 27th May 2004 Vol. 40 No. 11.

УДК. 681.3

Д.Б. Малашевич, П.Р. Машевич

Элементная база модулярных и троичных ЭВМ

(Московский государственный институт электронной техники (технический университет), ОАО «Ангстрем»)

Рассматривается историческая ретроспектива и современное состояние обеспечения электронной элементной базой разработок модулярных и троичных ЭВМ.

В материалах конференции некоторые ее участники утверждают, что одним из препятствий в развитии модулярных ЭВМ было и остается отсутствие элементной базы. Некоторые из них видят светлое будущее модулярной арифметики в программируемых логических интегральных схемах (ПЛИС). Существует так же мнение, что созданию троичных ЭВМ также препятствует отсутствие специальной элементной базы. Даже такой общепризнанный авторитет, как Д. Кнут в своей известной монографии [1] в 1976 г. отметил: «…*что ее* (троичной системы) *симметричность и простая арифметика окажутся в один прекрасный день весьма существенными (когда «флип-флоп» заменится на «флип-флэп-флоп»)*», т.е. когда появится троичный триггер. Очевидно, он не знал о троичных ЭВМ Н.П. Брусенцова, первая из которых – Сетунь, появилась на свет задолго до написания Д.Кнутом книги.

Такие утверждения представляются неубедительными. Попытаемся рассмотреть этот вопрос подробнее.

Элементная база первых модулярных и троичных ЭВМ

На всех этапах развития вычислительной техники одним из важнейших факторов, определяющих научно-технический уровень электронных вычислительных машин (ЭВМ) была ее электронная элементная база. Первые ЭВМ строились на основе вакуумных электронных ламп, поэтому их физические объемы измерялись многими кубометрами, а энергопотребление киловаттами. Затем появились полупроводниковые диоды и транзисторы, а с ними феррит-диодные и феррит-транзисторные ячейки. Это привело к существенному увеличению плотности компоновки аппаратуры ЭВМ, но их физические характеристики не изменились – в те же кубометры стали помещать большее функциональное содержание. Даже появление интегральных схем (ИС) мало отразилось на физических характеристиках ЭВМ – на всех этапах развития вычислительной техники характеристики ЭВМ определялись компромиссом возможного и желаемого, причем в ущерб желаемому. Так что появление ИС низкой и даже средней интеграции кубометров ЭВМ не убавило. Только с появлением микропроцессоров и других интегральных схем высокой сложности – больших и сверхбольших интегральных схем (БИС и СБИС), пожелания многих разработчиков и потребителей (далеко не всех) оказалось возможным удовлетворить. ЭВМ из продукции штучного или малосерийного производства превратились в массовые компьютеры. Это привело к резкому расширению областей применение компьютеров и, как следствие, появлению новых задач, для решения которых возможностей компьютеров оказывалось недостаточно. На негативном влиянии БИС и СБИС мы остановимся ниже.

Элементы модулярных ЭВМ

Первые модулярные ЭВМ Т-340А и К-340А [2] строились на основе полупроводниковых дискретных транзисторов и диодов. Разработка их предшественницы, двоичной позиционной ЭВМ А-340, первой полупроводниковой ЭВМ в НИИ-37, проходила в два этапа. Малые, по сравнению с лампами, размеры диодов и транзисторов породили соблазн построить ЭВМ на больших

печатных платах, что бы еще больше увеличить плотность компоновки аппаратуры. Но возникла проблема, которой не ожидали: каждая из этих плат была, как правило, уникальной, настроить ее, заставить работать оказалось практически невозможно. Эта ошибка стоила карьеры главному конструктору ЭВМ. Пришедший на его смену Д.И. Юдицкий быстро понял проблему и «рассыпал» машину на маленькие стандартные ячейки, каждая из которых содержала один или несколько типовых логических элементов. Эти ячейки выпускались уже в условиях серийного производства, легко настраивались, устанавливались на большие коммутационные платы (субблоки) и в дальнейшем хорошо на них совместно работали. Так появилась библиотека стандартных элементов. Этот принцип стандартных ячеек был положен и в конструкцию модулярных ЭВМ Т-340А и К-340А, что и позволило их быстро разработать, изготовить и отладить.

Идея небольших стандартных ячеек была положена и в основу конструкции разработанных в зеленоградском Центре микроэлектроники модулярных ЭВМ «Алмаз» и 5Э53. Только в качестве элементов в ячейках использовались уже первые в стране интегральные схемы типа Тропа, Посол, Конус и другие. Каждая ИС по своему составу и назначению примерно соответствовала стандартной ячейке Т-340А и К-340А, т.е. содержала один или несколько, как правило однотипных, элементов, реализующих функции булевой алгебры. Но на ячейках теперь были реализованы более сложные элементы – простейшие комбинационные схемы, типа триггеров, фрагментов регистров, счетчиков, дешифраторов и т.п. Всего в 5Э53 использовалось 160 типов ячеек – библиотека весьма обширная и по нынешним временам. Ячейки так же монтировались на больших коммутационных платах субблоков, которые в свою очередь устанавливались рядами в стойку ЭВМ. И здесь идея сработала, обе ЭВМ в короткое время были разработаны, изготовлены и настроены.

Параллельно с завершением работ по созданию 5Э53, ее разработчики выполняли задельную поисковую работу над элементной базой и принципами конструирования ЭВМ VI-поколения, о которых тогда еще только начинали говорить, как о светлой перспективе вычислительной техники. Планировалось создание модулярной ЭВМ с табличной реализации арифметики. Одним из высших достижений микроэлектроники того времени

была полупроводниковая диодная матрица из 256 диодов на диэлектрической подложке – ДМР-256, производство которой начиналось на заводе «Микрон». На ее основе и планировали строить и ПЗУ модулярных таблиц и некоторые другие узлы ЭВМ. Одновременно предусматривалось и использование новейших тогда ИС средней интеграции. Все ИС основных устройств ЭВМ планировалось применять в бескорпусном исполнении. Для этого была продумана специальная 4-х уровневая конструкционная система:

- На первом уровне кристаллы ДМР-256 и других ИС монтировались на квадратную ситаловую плату, размером примерно 8х8 см с внешними выводами на все четыре стороны платы.
- На втором уровне платы собирались в многоэтажную этажерку с межплатным монтажом по четырем граням этажерки и внешними выводами в нижний торец этажерки, получившей название МФБ – многофункциональный блок.
- На третьем уровне этажерки устанавливались на большую печатную кросс-плату субблока.
- Завершал эту иерархию блок, объединяющий несколько субблоков – металлический герметичный корпус которого заполнялся фреоном. Для вывода тепла из блока в него устанавливались тепловые трубки. В коллективе этот корпус получил название «чемодан».

Этот проект ЭВМ VI-поколения не был реализован в связи с отсутствием тогда заказчика на модулярную ЭВМ подобного класса. На подобную ЭВМ в то время нашелся только один заказчик, алгоритмы решения задач которого содержали высокий процент немодульных операций. Модулярная арифметика на его задачах была неэффективна и от нее пришлось отказаться. Это была последняя попытка разработки в Зеленограде высокопроизводительной многоразрядной ЭВМ, в которых модулярная арифметика эффективна. Все работы в этой области были директивно пресечены.

Примерно в 1973-1975 гг. в г. Перми, по-видимому на Пермском моторной заводе, коллективом во главе с Борисом Сергеевичем Гаспером была разработана модулярная бортовая управляющая

ЭВМ для управления режимами работы авиационными двигателями. Ее образец был изготовлен, настроен и прошел испытания на одном из авиадвигателей, но по каким-то причинам, установит которые помешала безвременная смерть Б.С. Гаспера, не принят в серийное производство. ЭВМ была построена на интегральных схемах.

В середине 80-х годов коллективом во главе с Е.К. Лебедевым были разработаны автокорреляционной функции спецпроцессоры «Вычет-1» и «Вычет-2», удостоенные серебряной медали ВДНХ СССР в 1989 г. Операционное устройство спецпроцессоров построено на основе ПЛМ с применением БИС К1518ВЖ1 и К1804ВС1.

Примерно в 1985 – 1990 г. группа специалистов во главе с В.Г. Евстигнеевым предприняла попытку создания комплекта БИС на основе БМК «Триумф» воронежского ПО «Электроника». Разработка велась на основе средств САПР тех времен на ЭВМ типа “μWAX” фирмы DEC в течение примерно 1,5 лет, но была прервана начавшимися в стране реформами.

Имеются отрывочные данные о разработках других модулярных ЭВМ и устройств, но об их элементной базе у авторов нет информации. Однако сам факт таких разработок свидетельствует о том, что элементная база была.

Элементы троичных ЭВМ

При создании первых троичных ЭВМ с элементной базой неразрешимых проблем так же не было. Мало того, именно элементная база натолкнула на мысль об использовании троичной системы счисления. Вот как об этом вспоминает Н.П. Брусенцов, главный конструктор первых троичных ЭВМ «Сетунь» и «Сетунь-70» [3]: «*Юлий Израилевич Гутенмахер строил машину ЛЭМ-1 на феррит-диодных элементах. Мне пришла в голову мысль, что раз транзисторов нет, то можно попытаться делать ЭВМ на этих элементах. Соболев, которого все очень уважали, договорился, чтобы я побывал на стажировке у Гутенмахера. Я все детально изучил. Поскольку по образованию я радиоинженер, то сразу увидел: не все нужно делать так, как делают они. Главное, что я увидел: они используют пару сердечников под каждый бит - рабочий и компенсационный. И мне пришла в голову идея: а что,*

если заставить компенсационный сердечник работать. Тогда каждая ячейка становится трехзначной. В результате получилось, что в "Сетуни" количество сердечников было в семь раз меньше, чем в ЛЭМ-1. При этом "Сетунь" имела почти вдвое большую разрядность».

Таким образом, для построения ЭВМ «Сетунь» были созданы феррит-диодные логические элементы, выполненные на нелинейных трансформаторах импульсов тока с диодами (быстродействующие магнитные усилители с питанием импульсами тока). Это были двухпроводные элементы троичной логики с электрическими сигналами двух уровней (Е, 0) и бибитным представлении тритов, в котором значения трита представлены двумя неразделяемыми битами (бибитами): «+1» → «01», «0» → «00» и «-1» → «10». Четвертое состояние бибитов «11» блокировалось. Несмотря на такую, казалось бы, избыточность, оказалось, что эти элементы не только весьма удобны для построения троичных цифровых устройств, но и что построенные на них троичные устройства получаются структурно более простыми, существенно более экономными по объему оборудования и потребляемой мощности и более быстрыми, чем двоичные устройства, реализованные на аналогичных элементах [4]. В условиях значительного разброса значений физических параметров, примененных в логических элементах диодов и ферритовых сердечников (которые поставщиками по существу не контролировались), была введена сортировка тех и других на попарно соответствующие друг другу группы, благодаря чему производство элементов было практически безотходным, а параметры элементов жестко стандартизованными. При дальнейшей сборке из таких элементов логических узлов (субблоков) и блоков машины требовалась только правильность проводных соединений, проверяемая на стендах логического контроля.

Такой выбор элементной базы для того времени оказался очень удачным. Даже в опытном образце ЭВМ, который проработал в МГУ более 15 лет, из 4 тыс. использованных в нем элементов было только 3 отказа (все 3 на первом году эксплуатации): 2 из-за пробоя диодов типа Д1 и 1 из-за нарушения изоляции между обмотками импульсного трансформатора. Сетунь устойчиво работала при значительной нестабильности напряжения питающей электросети и

в достаточно широком диапазоне температур окружающей среды (от +15 до +30° С). Серийные экземпляры машин "Сетунь" успешно эксплуатировались в различных климатических зонах с холодным, с жарким, а также с резко континентальным климатом (например, в Ашхабаде, Душанбе, Махачкале, Иркутске, Якутске, Одессе и т.п.), причем без какого-либо сервисного обслуживания и практически без запасных частей. Едва ли это может свидетельствовать о плохой надежности аппаратуры и ее элементной базы.

В 1970 г. тем же коллективом была разработана и изготовлен опытный образец второй троичной ЭВМ «Сетунь-70» [5]. Эта двухстековая машина была построена также на феррит-диодных пороговых логических элементах, но уже с однопроводной передачей трехуровневого троичного сигнала. Если о Сетуни можно сказать, что это была аппаратная эмуляция троичной ЭВМ на двоичных элементах, то Сетунь-70 была полностью троичной машиной. Опытный образец практически безотказно проработал в МГУ в течение 17 лет, но в серийное производство запущен так и не был. Специализирующимся в производстве ЭВМ Минрадиопрому и Минприбору чужие разработки были не нужны, тем более с непонятной троичной системой.

Таким образом, мы имеем полное основание констатировать, что даже на первых этапах развития вычислительной техники, в условиях дискретной или с низкой степенью интеграции элементной базы, проблем с элементами для модулярных и троичных ЭВМ не было. Все, кто хотел делать такие машины, их и делал на современной соответствующему периоду времени элементной базе. Проблема, во всяком случае в нашей стране, была совершенно в другом: те, кто мог и хотел разрабатывать модулярные и троичные ЭВМ оказались вне ведомств, специализирующихся в производстве ЭВМ, вне Минрадиопрома и Минприбора. У тех были свои главные конструкторы ЭВМ и коллективы разработчиков, свои программы разработок. «Чужакам», тем более успешно с ними конкурирующим, места на их заводах не находилось. Пока Д.И. Юдицкий и И.Я. Акушский работали в Минрадиопроме (в НИИ-37), их машина имела право на существование и производилась серийно. Стоило им перейти в Минэлектронпром, и самую высокопроизводительную в мире ЭВМ «5Э53» похоронили. К этому времени в стране уже имелась

довольно развитая научная школа Акушского-Юдицкого, объединяющая приверженцев системы остаточных классов в разных городах и республиках страны. Но, в результате директивного прекращения работ над модулярными ЭВМ, единый центр научной школы был разрушен, школа распалась, ее активность резко ослабла, исследования в области модулярной арифметики переместились в учебные и, частично, академические институты, т.е. перешли в область сугубо теоретических исследований.

Аналогичная судьба сложилась и в троичной системе. Завода для производства «плода университетской фантазии» так и не нашлось. В результате Н.П. Брусенцову в МГУ, в этом храме науки, просто запретили заниматься разработкой ЭВМ. С позиции руководства МГУ такое решение было понятно и логично: зачем тратить силы на такие разработки, практически реализовать которые все равно не дадут. Других сторонников троичной системы в стране не было и работы по ее развитию фактически прекратились, не успев сформироваться в научную школу. Остался ее энтузиаст – Н.П. Брусенцов, который углубился в трехзначную логику и развил ее до теоретических основ троичной информатики.

Этими двумя акциями в начале семидесятых годов прошлого века какие-либо активные работы по созданию модулярных и троичных, а заодно и любых иных не двоичных ЭВМ, в стране были насильственно прекращены. Два перспективных направления развития вычислительной техники, свершившие довольно успешный старт, имевшие определенные существенные преимущества перед двоичными позиционными ЭВМ, были ликвидированы по причинам, далеким от науки, техники и экономики. Причины оттого хорошо иллюстрируются следующими эпизодами:

- Зам. министра Минрадиопромышленности (МРП) В.И. Марков об ЭВМ «5Э53», разработанной в Минэлектронпроме (МЭП): «*…мы должны обязательно защищать свою ЭВМ (МРП), а не какой-то там МЭП…*» [2].

- Н.П. Брусенцову в ГКРЭ (Госкомитет по радиоэлектронике, позже Минрадиопром) довелось услышать иной решающий довод нелюбви к «плоду университетской фантазии»: “*от двоичных и десятичных голова болит, а они еще с троичной*

лезут" [6].

Современная элементная база

Появление и широкое распространение микропроцессоров и других БИС и СБИС высокой сложности имело и отрицательное влияние на развитие вычислительной техники. Разработчики ЭВМ потеряли возможность практической реализации своих новых идей, если не могли обеспечить достаточную для производителя БИС потребность в своей разработке, а достаточная потребность исчисляется миллионами БИС. Это во всем мире привело к резкому сокращению количества коллективов, разрабатывающих ЭВМ. Центр создания архитектур микропроцессоров переместился в полупроводниковые фирмы, специализирующиеся на разработке и производстве интегральных схем. Приверженцев модулярной арифметики, троичной системы или других недвоичных систем среди них не оказалось. Коллективы, ранее разрабатывающие ЭВМ либо перешли к созданию систем с более высокого начального уровня проектирования, используя в качестве электронных компонентов покупные стандартные микропроцессоры, либо сменили вид деятельности, либо распались.

Однако и в этот период состояние развития элементной базы тормозом развития модулярных и троичных ЭВМ, во всяком случае в нашей стране, не было – тормозить было нечего. Активных разработок не велось.

Одновременно с микропроцессорами появились и широко применялись БИС другого класса – различного типа программируемые матричные микросхемы, позволяющие относительно простым способом реализовать практически любые авторские схемотехнические решения. Примером таких микросхем являются полузаказные БИС, на которых мы остановимся ниже. Матричные ИС принципиально дороже заказных микросхем, тех же микропроцессоров, и их применение экономически нецелесообразно в продукции массового производства. Но для опытных образцов и для мелкосерийного производства они прекрасно используются.

Некоторые участники конференции видят перспективу широкого применения модулярной арифметики в программируемых логических интегральных схемах – (ПЛИС). Действительно,

ПЛИС в настоящее время являются наиболее удобным средством разработки интегральных схем. Но в своей экономической нише.

А таких ниш имеется три, т.к. во всем многообразии интегральных схем по принципам проектирования, производства и применения различают 3 вида: заказные, полузаказные и программируемые ИС.

Заказные ИС предназначены для массового производства. Их изготовление осуществляется в ходе одного полного технологического цикла по индивидуальному комплекту конструкторской документации. Проектирование заказных ИС осуществляется на основе библиотек стандартных ячеек с максимальной плотностью заполнения площади кристалла. В результате размеры и себестоимость кристалла заказной ИС, по сравнению с функциональными аналогами ИС других типов, минимальны. Но цикл «разработка-поставка» для заказных ИС является самым длительным и дорогостоящим, поэтому их проектирование и производство целесообразно только при большой применяемости, исчисляемой миллионами.

В распоряжении заказчика имеется ряд вариантов маршрутов проектирования заказных ИС. В зависимости от квалификации или по другим причинам заказчик может выбрать любой из них. Во всех маршрутах используется библиотека стандартных ячеек производителя БИС и определенные набор программных средств САПР. Возможны следующие варианты маршрутов проектирования заказных БИС:

- По техническому заданию (ТЗ) заказчика. Весь цикл проектирования выполняет дизайн-центр производителя или специализированной фирмы, работающей в режиме “fables” (без фабрики). Во втором случае производство БИС разработчиком размещается в специализированной фирме, работающей в режиме “foundry” (кремниевая мастерская), т.е. изготавливающей БИС по проектам заказчиков.

- Процесс проектирования БИС может быть разделен на два этапа между двумя исполнителями. На первом этапе (режим “Front-End”) выполняются начальные этапы проектирования, завершающиеся созданием электрической схемы БИС, модели ее функционирования и контрольных тестов. Этот этап выполняет либо сам заказчик, либо специализированная

фирма, работающая в режиме “fables”. Второй этап (режим “Back-End”) начинается с разработки топологии БИС, а завершается сдачей готового проекта изготовителю, проверкой поученных образцов и сдачей их заказчику. Исполнителем второго этапа может быть fables-фирма или дизайн-центр производителя.

- И, наконец, заказчик имеет возможность самостоятельно выполнить весь цикл проектирования БИС, т.е. выступить в качестве fables-фирмы и разместить изготовление в foundry-фабрике.

Термины “Front-End” и “Back-End” применяются и в производстве интегральных схем в схожем смысле. Этап “Front-End” включает все технологические процессы, связанные с формировании полупроводниковых структур в пластине монокремния. На этапе “Back-End” наносятся только верхние слои металлизации (в зависимости от технологии от $1 \div 5$ и более), реализующие межэлементные соединения согласно спроектированной на одноименном этапе топологии.

Таким образом современная микроэлектроника представляет потребителю любой вариант действий: покупку стандартных БИС, заказ необходимой ему БИС, частичное (Front-End”) участие в проектирование заказанной БИС или самостоятельное выполнение всего проекта. Но в любом случае этот процесс потребует больших затрат времени и средств.

Полузаказные ИС предназначены для средне- и малосерийного производства. Технологический цикл их разработки и изготовления разбит на два этапа. На первом этапе осуществляется создание базовых матричных кристаллов (БМК) в неразделенной пластине.

БМК представляет собой совокупность полупроводниковых структур одного или нескольких типов с элементарными соединениями внутри групп транзисторов или без них. В случае КМОП технологии группы транзисторов (вентили БМК) состоят из одной или нескольких пар комплиментарных транзисторов с общим поликремневым затвором для каждой пары. Комплиментарные пары в каждой группе транзисторов могут отличаться размером для обеспечения различной нагрузочной способности. По внешним сторонам кристалла БМК располагаются

контактные площадки и усилительные буферы для внешних выводов микросхемы. Совокупность ячеек БМК представляет собой периодическую структуру с «каналами трассировки», по которым на втором этапе разработки и изготовления будут выполняться межсоединения транзисторов. БМК изготавливаются в технологическом маршруте массового производства и в виде неразделённых пластин складируются.

Разработка непосредственно полузаказной ИС на базе БМК осуществляется на основе библиотеки элементов, представляющих собой описание соединений полупроводниковых приборов, размещённых в БМК, согласно электрической схеме соответствующего элемента. При проектировании полузаказной БИС по определенным правилам осуществляется соответствующая схеме устройства «расстановка» библиотечных элементов на поверхности кристалла с их библиотечной топологией, и разрабатывается топология межэлементных соединений согласно электрической схеме проектируемого устройства.

На втором этапе изготовления полузаказной ИС со склада берутся пластины с готовыми БМК и в условиях штучного или малосерийного (на уровне пластин) производства на них наносятся слои металлизации, реализующие топологии библиотечных элементов и межэлементных соединений.

При проектировании полузаказных БИС никогда не удается использовать все имеющиеся в кристалле вентили. Практически максимальный процент заполнения БМК (использования его вентилей) для схем со средней регулярностью разводки типа процессоров составляет 40 – 50 %. Средне-статистическое заполнение еще ниже – 20 – 25 %. Но бывали случаи создания многовыводных схем с заполнением в 2 – 3 %. Эти данные основаны на статистике реальных проектов полузаказных БИС на основе КМОП БМК 1592ХМ1 ОАО «Ангстрем». Низкий процент заполнения БМК определяется рядом причин, в т.ч.:

- При самой оптимальной разработке никогда не удаётся использовать все имеющиеся в БМК вентили, т.к. их заранее заданные структура и физическое размещение на поверхности кристалла накладывают определенные ограничения на топологию, как библиотечных элементов, так и на их межсоединения. А при необходимости более

мощного, чем имеются, транзистора приходится использовать несколько параллельных.

- БМК выпускается в виде рядов с существенно различным числом вентилей. При выборе БМК, естественно, берется тот, у которого число вентилей больше требуемого. Например, имеется ряд БМК 1592ХМ1, 1592ХМ2, 1592ХМ3 и 1592ХМ4. Эти БМК полностью идентичны во всем, кроме емкости: числа вентилей внешних и выводов. Они, соответственно, содержат 100, 60, 30 и 10 тыс. вентилей. Следовательно, если устройство содержит около 6 тыс. вентилей, с учетом процента заполнения придется применить БМК на 30 тысяч, т.е. заполнение БМК будет ниже 20%., а кристалл более, чем впятеро превысит функционально аналогичный заказной.

Следовательно, себестоимость полузаказной ИС всегда существенно выше себестоимости заказной ИС, что и ограничивает области их применения мелкосерийной продукцией.

Процесс проектирования полузаказной БИС на основе БМК так же может распределятся между заказчиком и изготовителем аналогично описанному для заказных БИС. Технологический цикл «разработка-производство» для полузаказных ИС измеряется неделями и значительно короче такого цикла для заказных ИС.

Между процессами создания заказных и полузаказных ИС много общего. Оба вида разрабатываются на основе похожих библиотек стандартных элементов, на одних и тех же средствах САПР и могут изготовляться по одной технологии. У обоих видов стандартный библиотечный элемент имеет фиксированные схему и топологию и как незыблемый фрагмент используется при проектировании микросхемы. Имеется и еще ряд общих черт. Главное отличие заключается в том, что в полузаказных ИС набор транзисторов, их группировка в элементарные вентили и размещение вентилей на поверхности кристалла заданы единожды и навсегда. И если, например, при проектировании стандартной ячейки нагрузочной способности транзистора оказывается недостаточно, то используется два или более параллельно соединенных. При проектировании библиотечного элемента для заказной ИС разработчик имеет возможность использовать любые транзисторы (в пределах возможности технологии) и ставить их в произвольное

место на поверхности кристалла (в пределах определенных правил). И может использовать эти транзисторы произвольно, а не в виде заранее заданных структур в виде элементарных вентилей. Т.е. если недостаточно нагрузочной способности имеющихся транзисторов, разработчик имеет возможность сотворить соответствующий новый транзистор. Иными словами, в полузаказных БИС более половины транзисторов, как правило, не используются, а в заказных БИС лишних транзисторов просто нет. Аналогично и с внешними выводами, в заказной ИС их количество всегда оптимально, в полузаказной заранее задано и, как правило, больше требуемого. Все это приводит к тому, что у заказной ИС и размеры кристалла, и временные задержки, и число выводов, и потребляемая мощность, а значит и стоимость, всегда значительно меньше, чем у функционально эквивалентной полузаказной.

Программируемые логические схемы это завершённые электронные изделия, не требующие для реализации конкретной схемы дополнительных промышленных технологических операций. ПЛИС представляют собой массивы конфигурируемых ячеек (не путать с термином библиотечной ячейки), с выполненными программно перекоммутируемыми межсоединениями. Ячейки ПЛИС состоят из нескольких управляющих элементов (мультиплексоров, коммутирующих соединения, триггеров и т.д.) и одного или нескольких табличных элементов с таблицами коммутации межсоединений в виде долговременной памяти, значения которых могут быть изменены. Конфигурирование такой схемы под конкретную схему выполняется с помощью программирования – записи во внутреннюю память микросхемы конфигурации ячеек. Программирование и перепрограммирование ПЛИС выполняется с помощью специального устройства – программатора, подключаемого к ПК, и легко может быть выполнено самим проектировщиком.

Размер кристалла ПЛИС при одинаковой технологии их производства всегда существенно больше кристалла БМК. Это происходит потому, что ПЛИС представляет собой функционально двухслойный пирог. Один слой – типовые вентили для потребителя, аналогичные вентилям БМК, из которых строится схема проектируемого устройства. Второй слой – система

программируемой коммутации этих вентилей для их соединения согласно схеме проектируемого устройства. Причем доля второго слоя в объеме оборудования кристалла существенно превышает долю первого, даже с учетом того, что на самом деле они перемешаны между собой и явно их выделить невозможно, да и не нужно.

Процент заполнения кристалла у ПЛИС немного ниже, чем у БМК, т.к. возможности системы коммутации вентилей несколько меньше, чем возможности топологической разводки в БМК.

Все это в совокупности определяет высокую стоимость ПЛИС и определяет области их применения:

- при макетировании устройств с последующим переводом проектов, в зависимости от планируемых объемов производства, в заказные или полузаказные ИС,
- при штучном или мелкосерийном производстве разрабатываемого устройства.

Таким образом можно грубо оценить требования к емкости БМК или ПЛИС в зависимости от объема оборудования разрабатываемого устройства с регулярностью схемы и топологии типа процессора. Приведенные в таблице данные для БМК и ПЛИС минимальны. Реально нужно выбирать процентов на 30 - 50 более.

Параметр	Заказные	Полу-заказные	ПЛИС
Максимальное заполнение, %	100	50	45
Минимальное число необходимых вентилей для эквивалентных БИС	50	100	110

Для иллюстрации затрат времени и средств на создание БИС различных типов рассмотрим условный проект БИС с объемом оборудования до 100 000 вентилей, что соответствует характеристикам отечественного БМК 1592ХМ1. Поскольку этап Front-End для всех типов БИС одинаков, в таблице он не учтен. Не учтены так же затраты на приобретение оборудования и программного обеспечения, по умолчанию принято, что все исполнители ими полностью оснащены. Иными словами, в расчетах заложены только прямые затраты на разработку топологии БИС их изготовление и испытания, на материалы,

энергию и т.п.

Тип ИС	Цикл проект – поставка, недель	Цена проект-поставка, руб.	Число опытн. образ-цов	Цена 1 БИС опытной, руб.	Тираж-ность, шт/год	Цена 1 БИС далее, руб.
ПЛИС	1	5 000	1	10 000	30	10 000
П/зак. БИС	13	600 000	30	20 000	>100	2 000
Зак. БИС	40	5 000 000	100	50 000	>10 000	150

Приведенные данные позволяют грубо оценить, какой тип ИС следует выбрать, в зависимости от планируемого объема производства разрабатываемого изделия.

Таким образом, надежды некоторых приверженцев модулярной арифметики на ПЛИС, как на панацею, не совсем оправданы. ПЛИС действительно идеальны в настоящее время для макетирования и изготовления экспериментальных образцов аппаратуры. Но если речь пойдет о ее серийном производстве, то проект ИС с ПЛИС придется переводить в полузаказную или заказную ИС. Эта технология хорошо отработана, но потребует определенных затрат времени и средств.

Библиотеки

Основным принципом проектирования, позволяющим разрабатывать заказные и полузаказные ИС любой сложности, является принцип декомпозиции – разбиение объекта проектирования на составные части. Принцип декомпозиции является общим принципом проектирования и применяется не только при разработке ИС, но и в программировании, и во многих других задачах, в т.ч. и не связанных с электроникой.

При разработке ИС применяется термин «функциональная декомпозиция». Этот термин отражает суть применяемой декомпозиции – выделение функционально завершённых блоков (фрагментов схемы). Под функциональной завершённостью понимается очевидная, легко формализуемая функция, выполняемая выделенным блоком устройства. При этом, функциональная декомпозиция выполняется на различных уровнях иерархии проекта:

- Сложные функциональные блоки,

- Модули высокоуровневого описания,
- Библиотечные стандартные ячейки.

Применение декомпозиции проекта позволяет эффективно разделить задачу проектирования между несколькими разработчиками или коллективами. Функциональная декомпозиция позволяет выделять многократно используемые фрагменты схемы устройства, и, используя механизм ссылок, многократно применять единожды созданный и отлаженный фрагмент.

Верхним уровнем декомпозиции в этой иерархии, являются сложные функциональные блоки (СФБ, или IP-блоки в англоязычном варианте, IP – intellectual proprietary). СФБ представляют собой самостоятельные устройства, такие как: ядра микроконтроллеров и микропроцессоров, сопроцессоров, интерфейсов, устройств цифровой обработки сигналов, памяти и т.п., которые могут быть объединены в одном кристалле. СФБ являются не только составными частями конечного проекта, но и сами по себе являются законченным продуктом и товаром, востребованным на рынке. Как правило, они разрабатываются в соответствии с определенными правилами межблочной совместимости, что позволяет объединять их в одном кристалле БИС в требуемую потребителю систему. Такие БИС, представляющие собой функционально и конструктивно законченный продукт, и технология их создания на основе СФБ получили название «системы на кристалле» (СнК).

Примером декомпозиции на средних уровнях может служить применение Verilog-модулей (Verilog – язык описания цифровых схем в системе автоматизации проектирования ИС (САПР)). Разработчик описывает и отлаживает относительно небольшие части проекта, а затем использует их на более высоких уровнях иерархии.

Декомпозицией на низшем уровне можно считать применение библиотек стандартных ячеек. В качестве стандартных ячеек (элементов) выбираются простейшие логические вентили, выполняющие функции (обычно на 2-4 входа) типа И-НЕ, ИЛИ-НЕ, исключающее ИЛИ, а также некоторые их комбинации и более сложные элементы, такие как одноразрядные сумматоры, элементы памяти – триггеры, мультиплексоры и т.п. Состав элементов разных библиотек индивидуален.

Для разработки модулярных устройств вполне применимы существующие библиотеки стандартных элементов. А если будет обнаружена необходимость применения каких-либо специфичных элементов (что маловероятно), то нет никакой технической проблемы в расширении соответствующей библиотеки, это вопрос сугубо экономический и организационный.

Учитывая ограниченность областей применения модулярной арифметики, их специфичность, имеет смысл создание модулярных СФБ, например процессора обработки сигналов или криптопроцессора, для использования их в СнК вместе с другими СФБ. На первом этапе разработки таких СФБ, для натурального макетирования, вполне применимы ПЛИС.

Такой подход открывает большие возможности приверженцам модулярной арифметики для ее реального внедрения и демонстрации преимуществ там, где они есть. СнК освобождает их от необходимости искать заказчика системы, заниматься разработкой и организацией производства системы в целом со всеми связанными с этим хлопотами и проблемами. Им достаточно только подобрать работающую в технологии СнК фирму, например Элвис, и разработать модулярный СФБ, взаимозаменяемый по присоединительным характеристикам с применяемым этой фирмой позиционным СФБ, например ЦОС (DSP) системы «Мультикор». Если он окажется существенно лучше, то его, скорее всего, с удовольствием воспримут. Если нет, значит оптимизм приверженцев СОКа оказался несколько преувеличенным.

Многое из вышесказанного справедливо и по отношению к троичным двухпроводным устройствам. Поскольку троичная логика включает в себя двоичную, все стандартные ячейки существующих библиотек в ней применимы. Но очевидна необходимость пополнения библиотек стандартными ячейками, реализующими специфичные функции троичной логики. Что касается однопроводных троичных устройств, то для них потребуется создание новых библиотек и новых САПР. Эта проблема подробнее рассмотрена в [7].

Таким образом, мы приходим к неизбежному выводу, что ныне существующей элементной базы вполне достаточно для создания любых модулярных и троичных (с двухпроводными элементами) устройств и систем. Этого вполне достаточно для реализации

реальных проектов, воплощающих позитивные свойства модулярности и троичности. Такие проекты, наряду с решением своих задач по назначению, могут быть использованы и для пропаганды этих перспективных направлений развития вычислительной техники и способствовать их дальнейшему развитию.

Литература

1. **Кнут Д.** Искусство программирования для ЭВМ - Получисленные алгоритмы// М.: Мир, 1977. – С. 724
2. **Малашевич Б.М.** Разработка вычислительной техники в Зеленограде: Неизвестные суперЭВМ. Журнал «ЭЛЕКТРОНИКА: наука, технология, бизнес, № 2, 4 и 7, 2004
3. **Румянцев Д.** Долой биты! Интервью с конструктором троичной ЭВМ. Журнал Upgrade. Ноябрь 30, 2005
4. **Брусенцов Н.П., Маслов С.П., Розин В.П., Тишулина А.М.** Малая цифровая вычислительная машина "Сетунь"/. - М.: Изд-во МГУ, 1965.
5. **Брусенцов Н. П.** и др. Общая характеристика малой цифровой машины «Сетунь-70». В кн.: Вычислительная техника и вопросы кибернетики, вып.10. Л., 1973, – С. 3-21.
6. **Брусенцов Н. П., Жоголев Е. А., Маслов С. П., Рамиль Альварес Х.** «Опыт создания троичных цифровых машин». В кн.: Компьютеры в Европе — прошлое, настоящее и будущее. Труды международного симпозиума. — Киев: «Феникс», 1998.
7. **Малашевич Д.Б.** Недвоичные системы в вычислительной технике. Юбилейная международная научно-техническая конференция «5э лет модулярной арифметике», Сборник научных трудов, М. 2005

УДК. 681.3

В.К. Зольников, П.Р. Машевич

Структурная декомпозиция блоков микропрограммного управления

(*Воронежская лесотехническая академия, ОАО «Ангстрем»*)

В работе исследованы особенности проектирования блоков микропрограммного управления (БМПУ) СБИС минимальной площади на базе регулярных структур типа программируемые логические матрицы (ПЛМ) и ПЗУ. Рассмотрена структурная декомпозиция БМПУ: изменение количества входов, выходов, введение мультиплексоров на входах и демультиплексоров на выходах, введение счетчиков, раздельных полей управляемых регистров и др. и сделаны оценки изменения площади БМПУ на кристалле при структурной декомпозиции.

При построении блоков микропрограммного управления (БМПУ) СБИС основной задачей является минимизация аппаратных затрат, а именно – минимизация площади, занимаемой БМПУ на кристалле. Рассмотрим различные варианты структурной декомпозиции БМПУ на базе программируемых логических матриц (ПЛМ) и ПЗУ.

Как отмечено в [1] структурная схема БМПУ на ПЛМ, без декомпозиции, реализуется на структуре ПЛМ с регистровой памятью (рис.1), а функционирование определяется множеством входных

переменных $X=\{x_{i|i=1\div L}\}$, множеством состояний $S=\{s_{i|i=1\div s}\}$, множеством выходных переменных $Y=\{y_{i|i=1\div N}\}$ (микрокоманд) и порядком их следования [2].

Площадь, занимаемая таким блоком на кристалле, определяется следующей зависимостью [3]:

$$S \sim (2\cdot(L+r)+N)\cdot p \qquad (1)$$

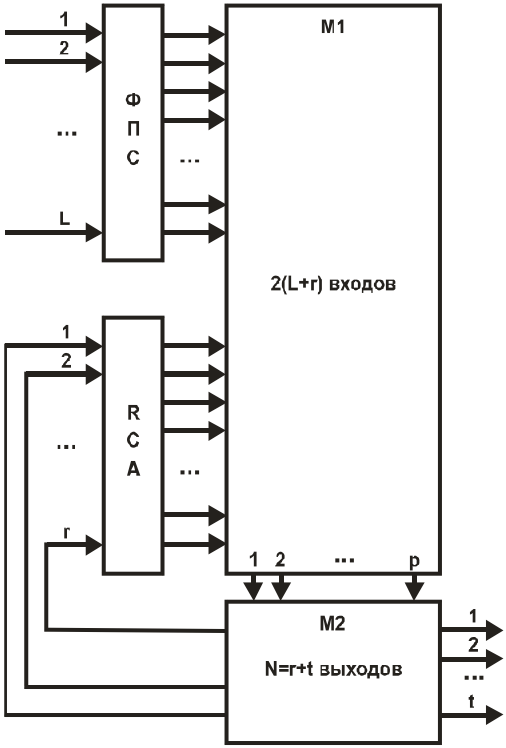


Рис.1.Структурная схема БМПУ на ПЛМ
с регистровой памятью

Рассмотрим методы структурной декомпозиции БМПУ, позволяющие уменьшить площадь блока.

Одним из приемов получения БМПУ минимальной площади является его декомпозиция. Рассмотрим вариант, при котором ПЛМ

разделяется на две последовательно включенные ПЛМ1 и ПЛМ2 (рис. 2.) для конъюнктивной формы логических условий $X_{i,j}=F_{i,j}\wedge\Psi$ [4], при этом F_{ij} будем реализовывать на ПЛМ1, а ψ - на ПЛМ2.

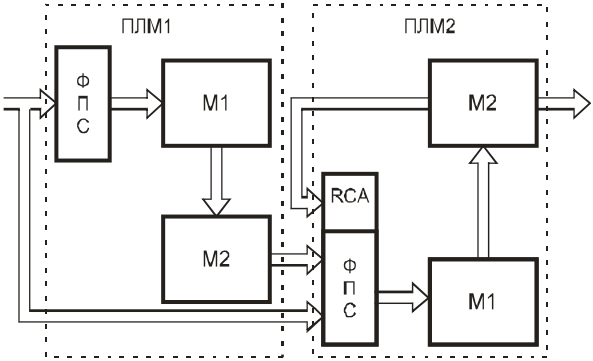


Рис. 2. Вариант декомпозиции БМПУ

Тогда для варианта логических условий, для которых реализация F_{ij} требует m логических произведений, а для реализации ψ - h (см. [4].) получим следующее выражение для увеличения общей площади - ΔS_3, причем без введения промежуточных состояний

$$\Delta S_3 \sim (2\cdot L+1)\cdot m + (2(L+r)+N)\,h+2(p+h) \qquad (11)$$

Рассмотрим вариант реализации фрагмента графа переходов на структуре, изображенной на рис.2., при кодировании команд методом расширения кодов операций [5], при котором логические условия $x_{i,jk}$, вызывающие переход из состояния S_{i0} в состояния $S_{jk|k=1\div h}$. , можно представить в виде

$$X_{i,jk} = \prod_{n=1}^{k-1} \Psi_n \bullet \overline{\Psi_k}$$

Для тех же условий [4], при которых были получены ΔS_2

$$\Delta S_3^{\,1} \sim (2\cdot(L+r)+N)\cdot h + (2L+h)h+2h(p+h) \qquad (12)$$

При анализе оптимальности декомпозиции БМПУ необходимы оценки площади ФПС, а также выходных буферных элементов. Так как ФПС размещаются с двух сторон матрицы М1 и по геометрическим размерам сопрягаются с ней, а его длина ($l_{фпс}$) зависит от конкретной схемотехнической и топологической реализации и на практике составляет от 5 до 12 размеров ячеек матрицы то $\Delta S_{фпс}$ ~4

($l_{фпс}$) L, где L – количество разрядов ФПС.

Декомпозиция БМПУ часто имеет еще одну цель – увеличение быстродействия процессора. Например, ПЛМ1 осуществляет предварительный разбор i+1-ой команды, а ПЛМ2 формирует при этом микрокоманду i-ой команды, реализуя тем самым конвейерный принцип управления.

Методы изменения количества входов и выходов ПЛМ для уменьшения площади БМПУ. Часто при построении блоков БМПУ реального микропроцессора количество входных переменных достаточно велико. Это и разряды команды, и разряды регистра состояний процессора, и коды причин прерываний, и коды временных признаков из блока синхронизации и др. По аналогии с терминологией программирования будем различать входные переменные, являющиеся глобальными (влияют на весь граф переходов БМПУ) и локальными (влияют на отдельный участок графа переходов БМПУ) логическими условиями. Если переходы по локальным логическим условиям осуществляются из различных состояний графа переходов, и подмножества входных переменных, соответствующих этим локальным логическим условиям, не содержат одинаковые переменные, то можно подключать источники входных сигналов различных подмножеств в различные моменты времени на одни и те же входы ПЛМ (мультиплексирование информации на входе).

Пусть функционирование БМПУ определяется множеством входных переменных $X_{глj}=\{x_{глi}|_{i=1\div L}\}$, соответствующих глобальным логическим условиям, и множествами $X_{л1}=\{x_{лi}|_{i=1\div m_1}\}$, $X_{л2}=\{x_{лi}|_{i=1\div m_2}\}$, . . . $X_{лk}=\{x_{лi}|_{i=1\div m_k}\}$ входных переменных, соответствующих локальным логическим условиям.

Пусть пересечение множеств локальных логических условий равно пустому множеству $X_{лi}\ \ X_{лj}=\varnothing$, при $i\neq j$, $i,j=1\div k$.

Пусть общее количество логических произведений для реализации графа переходов равно p, а количество выходов ПЛМ – N. Тогда площадь ПЛМ:

$$S_1 \sim (2\cdot(L+r+\cdot\sum_{i=1}^{k} m_i)+N)\cdot p \qquad (13)$$

При введении мультиплексора на входе необходимо выбрать мак-

симальное количество входных переменных и выходов, т.е.

$$\left]\sum_{i=1}^{K} m_i\right[\quad \text{и} \quad]\log_2 k[.$$

На рис.3. изображена структурная схема БМПУ с мультиплексором на входе. Его площадь составляет:

$$S_2 \sim (2\cdot(L+r+\left]\sum_{i=1}^{K} m_i\right[)+N+]\log_2 k[) \cdot p \qquad (14)$$

Так как величина $2\cdot\cdot\sum_{i=1}^{k}\mathbf{m}_i > 2\cdot\left]\sum_{i=1}^{k}\mathbf{m}_i\right[+]\log_2 k[$, то всегда

мультиплексирование информации на входе ПЛМ в тех случаях, когда это возможно, приводит к уменьшению площади.

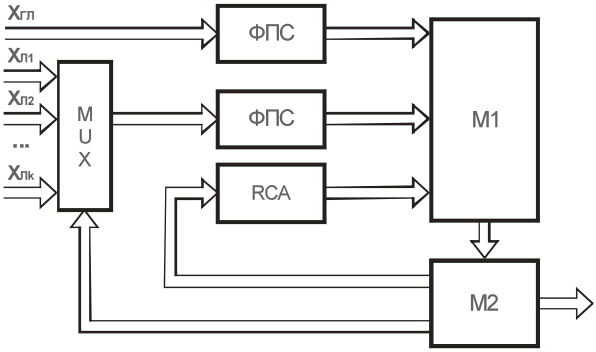


Рис. 3. Структурная схема БМПУ с мультиплексором на входе

Разновидностью метода мультиплексирования информации на входе БМПУ можно считать метод, при котором 2m – разрядный мультиплексор подключается своими выходами непосредственно на входы ПЛМ. Первые 2m входов мультиплексора подключаются к группе 2m выходов ФПС, соответствующих некоторому множеству входных переменных $X_л=\{x_{лi}\,|\,i=1\div m\}$, а вторые 2m входов к некоторой константе (рис.4). . При этом могут использоваться две различные константы, применение каждой из которых приводит к различным вариантам.

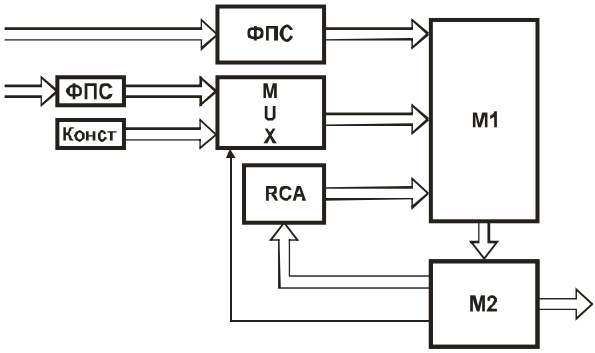


Рис. 4. Структурная схема БМПУ с мультиплексором на выходе ФПС

1. Константа С=0. При подаче соответствующего управляющего сигнала на мультиплексор на m прямых и m инверсных входах ПЛМ устанавливается “0”. Это приводит к тому, что на выходах множества логических произведений, в которые входит, как существенная, хотя бы одна переменная $x_{лi} \in X_л$. С помощью этого метода можно осуществлять взаимную синхронизацию БМПУ и операционного автомата в случае, если множество входных переменных $X_л$ формируется в операционном автомате и записывается в регистр $RX_л$ БМПУ. При необходи мости осуществлять переход по логическим условиям $X_л$ нужно знать, произошла ли запись в регистр $RX_л$ кода новых логических условий из операционного автомата, либо там хранится код старых логических условий. Если при этом необходимо продолжать циклы работы ПЛМ, то устанавливают соответствующий сигнал управления мультиплексором, снятие которого осуществляется при подаче кода логических условий из операционного автомата.

2. Константа С=1. При подаче соответствующего управляющего сигнала на мультиплексор на m прямых и m инверсных входах ПЛМ устанавливается “1”. Это приводит к тому, что переходы в графе переходов выполняются независимо от локальных логических условий $X_л$. Причем, в случае ветвлений по $X_л$ одновременно осуществляется переход по всем ветвям графа. Микрокоманда и состояния, которые формируются на выходах БМПУ при таком обобщенном переходе, равны дизъюнкции микрокоманд и состояний соответственно этих ветвей в графе переходов.

Для уменьшения площади БМПУ можно применять также метод эквивалентной замены некоторого количества входов на некоторое количество выходов. Пусть функционирование БМПУ определяется множеством логических условий $X_{гл}=\{x_{гли}|_{i=1\div L}\}$ и $X_{л}=\{x_{лi}|_{i=1\div m}\}$. Пусть $Si=\{s_{ik}|_{k=1\div h}\}$ – некоторое состояние БМПУ, из которого по воздействием множества $X_{лj}$ автомат переходит в множество состояний $Sj=\{s_{jk}|_{k=1\div h}\}$, причем 2^m»h. Количество логических произведений при этом равно $P_j=\{p_{jk}|_{k=1\div h}\}$. Площадь ПЛМ при реализации этого варианта

$$S_1 \sim (2\cdot(L+r+\cdot m\ )+N)\cdot(p+\cdot\cdot\sum_{k=1}^{h} p_{jk}\ ) \qquad (15)$$

На рис.5. изображена структурная схема БМПУ, в которой введено $]\log_2 h[$ выходов, на которых при переходе БМПУ в состояние S_i появляется код, указывающий, по какому подмножеству логических условий $x_{лjk}$ будет осуществляться переход. В БМПУ также введена логическая схема, на входы которой поступают входные переменные $X_л$ и введенные выходы ПЛМ. На выходе логической схемы формируется признак истинности логических условий. Выход логической схемы подключен к дополнительно введенному входу ПЛМ (вместо m входов для приема логических условий $X_л$..

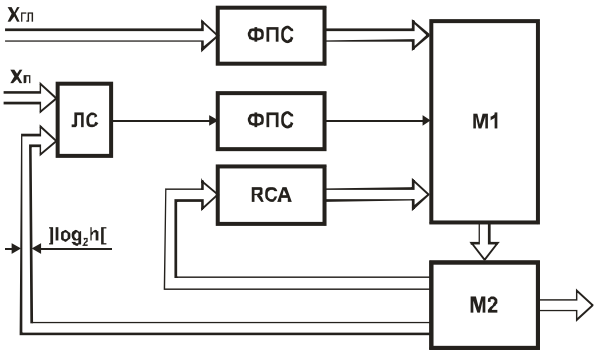


Рис. 5. Структурная схема БМПУ в которой внедрено $]\log_2 h[$ выходов

Площадь этих ПЛМ при реализации равна:

$$S_1 \sim (2\cdot(L+r+\cdot 1\ )+N+\ ]\log_2 h[)\cdot(p+h)+(2(m+]\log_2 h[+1)\cdot\sum_{k=1}^{h} p_{jk} \qquad (16)$$

увеличение площади в первом и втором вариантах равно:

$$\Delta S_1 \sim (2\cdot(L+r)+N) \cdot\cdot \sum_{k=1}^{h} p_{jk} + 2m(p+\cdot \sum_{k=1}^{h} p_{jk}) \qquad (17)$$

$$\Delta S_2 \sim (2\cdot(L+r)+N)\ h+(2+]\log_2 h[)(p+h)+(2(m+]\log_2 h[)+1) \sum_{k=1}^{h} p_{jk} \qquad (18)$$

Метод формирования некоторых обобщенных, ожидаемых в данном состоянии логических условий, можно применять при реализации команд условных переходов. В таких командах указаны логические условия (состояния), по которым необходимо осуществлять переход. Сами логические условия (состояния) представляют собой некоторый четырех- или более разрядный код, а все переходы осуществляются идентично. В этом случае количество произведений в ПЛМ1 для реализации перехода равно 1, т.е.

$$\Delta S_2 \sim (2\cdot(L+r)+N)\ h+(2+]\log_2 h[)(p+1)+(2(m+]\log_2 h[)+1) \sum_{k=1}^{h} p_{jk} \qquad (19)$$

при этом выигрыш в площади БМПУ будет еще более значителен.

Демультиплексирование информации на выходе ПЛМ. Пусть разрядность микрокоманды на входе операционного автомата равна N, причем микрокоманда состоит из полей разрядностью N_1 и $N_2 \approx N/2$. Пусть также множества выходных переменных $Y_1=\{y_{1i}|_{i=1\div N1}\}$ и $Y_2=\{y_{2i}|_{i=1\div N2}\}$ определяют микрооперации в этих полях, причем для реализации множества Y_1 выходных переменных требуется p_1 логических произведений, а для реализации Y_2 - p_2 логических произведений.

При демультиплексировании информации на выходе необходимо ввести вход, в зависимости от состояния которого можно различать, какое множество выходных переменных формируется в текущем цикле ПЛМ (рис.6)

Площадь ПЛМ с демультиплексированием информации на выходе

$$S_2 \sim (2\cdot(L+r+1)+]N_{12}[)\ (p_{1+}\ p_2) \qquad (20)$$

В наихудшем случае множества выходных переменных Y_1 и Y_2 сильно связаны и количество логических произведений $p_1 \approx p_2 \approx p$.

Тогда вариант демультиплексирования имеет преимущество

$$\Delta S \sim (2\cdot(L+r)+4p+]N_{12}[-N/2)2p \qquad (21)$$

В наилучшем случае выходные переменные не связаны и количество логических произведений $p_{1+}\ p_{2=}p$, тогда

$$\Delta S \sim (N/2-2)p \qquad (22)$$

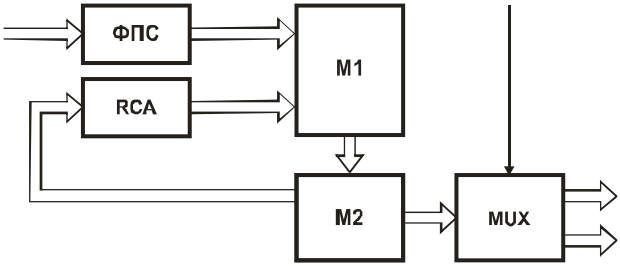


Рис. 6. Структурная схема БМПУ с демультиплексированием

Демультиплексирование информации на выходе применяют, когда необходимо управлять разнородными не связанными между собой объектами, например, операционным блоком и блоком прерываний, тогда и реализуется условие $p_{1+}p_{2=}p$.

Также эффективно применение демультиплексирования информации на выходе ПЛМ в случае, когда частота подачи микрокоманды на управляющие входы одного блока существенно ниже, чем на другого. При этом нет необходимости увеличивать количество входов в ПЛМ, а можно использовать увеличение количества состояний БМПУ. Практически это не увеличивает разрядность RCA

$$\Delta S \sim N/2\cdot p \qquad (23)$$

Для уменьшения площади можно использовать методику маскирования информации на выходе. Она дает эффект в случае объединения нескольких ПЛМ и реализации на выходах различных “проводных” функций – “проводное И”, либо “проводное ИЛИ”. Для этого в ПЛМ вводится дополнительный выход маскирования информации (запрет выдачи (ЗВ)), что позволяет при значении лог.1 на этом выводе получать не полностью определенные функции, тем самым уменьшать площадь БМПУ за счет логической минимизации.

Уменьшение площади БМПУ за счет изменений RCA. Проведем оценку модификации RCA для повторяющихся логических условий.

Пусть S_{i0} – некоторое состояние БМПУ, из которого под воздействием множества логических условий $X_{i0,j0}=\{x_{i0,j0k}|_{k=1\div l}\}$ он переходит в множество состояний $S_{j0}=\{s_{j0k}|_{k=1\div l}\}$ и на его выходах реализуется множество выходных функций $У_{i,j}=\{y_{i,jk}|_{k=1\div l}\}$. Пусть каждому подмножеству логических условий, вызывающих эти переходы, соответствует некоторая кратчайшая ДНФ, которая включает m_k элементарных конъюнкций. Пусть далее через какое-то количество состояний в соответствии с графом переходов БМПУ переходит в состояние S_{i1}, из которого под воздействием множества логических условий $X_{i1,j1}=\{x_{i1,j1k}|_{k=1\div l}\}$ автомат переходит в множество состояний $S_{j1}=\{s_{j1k}|_{k=1\div l}\}$, причем множества логических условий $X_{i0,j0}$ и $X_{i1,j1}$ равны. Фрагмент графа переходов БМПУ с повторяющимися логическими условиями изображен на рис. 7.

Для реализации на ПЛМ переходов из состояния S_{i0} в состояние S_{j0} требуется $\cdot\Sigma m_k$ логических произведений. Такое же количество логических произведений необходимо для реализации переходов из состояния S_{i1} в состояние S_{j1} . Увеличение площади БМПУ при этом будет

$$\Delta S_2 \sim (2\cdot(L+r)+N)\ 2\cdot\sum_{k=1}^{h} m_k \qquad (24)$$

Рассмотрим иную реализацию фрагмента графа переходов с повторяющимися логическими условиями. При выполнении переходов БМПУ между состояниями S_{j0} и S_{i1} закодируем состояния следующим образом. Мысленно разделим поле регистра RCA на две части, длина одной из которых равна $]\log_2 h[$ и будем на всех переходах из состояний S_{j0} в состояние S_{i1} сохранять в этой части регистра код, соответствующий состоянию $S_{j0}=\{s_{j0k}|_{k=1\div h}\}$. Количество логических произведений , необходимое для этого , равно $]\log_2 h[$. При этом необходимо обеспечить такое количество состояний, чтобы различать, что БМПУ находится именно на переходе из состояний S_{i0} в состояние S_{i1} . Пусть до введения разбиения поля RCA на две части общее количество состояний было M, т.е. $r=]\log_2 M[$, пусть также при различных путях переходов количество состояний равно M1.

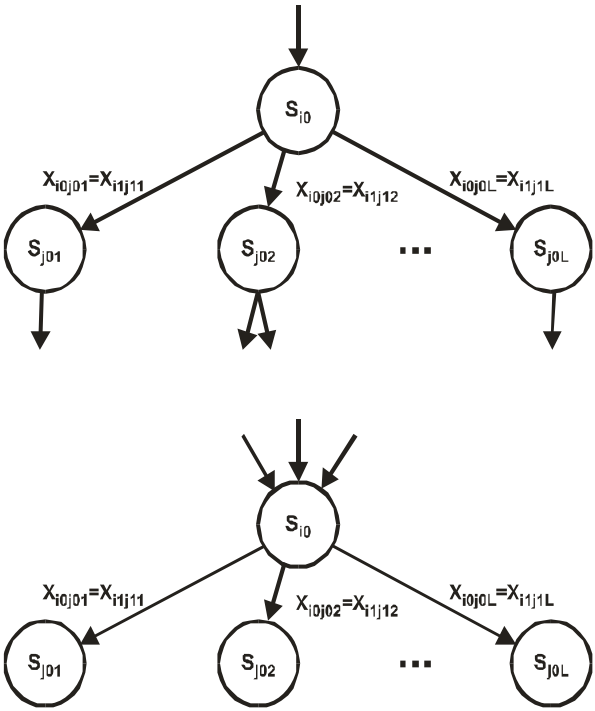


Рис.7. Фрагмент графа переходов БМПУ
с повторяющимися логическими условиями

Тогда общее количество состояний БМПУ увеличится на M1(h-1) состояний, а разрядность регистра RCA станет равной $]\log_2(M+M1(h-1))[$. Увеличение площади БМПУ для такой реализации фрагмента графа переходов равно:

$$\Delta S_2^1 \sim (2\cdot(L+r)+N)\cdot(\sum_{k=1}^{h} m_k+]\log_2 h[+3(]\log_2(M+M1(h-1))[-]\log_2 M[))(p+\sum_{k=1}^{h} m_k+]\log_2 h[) \quad (25)$$

Введем некоторые структурные изменения в регистр RCA, разобьем регистр RCA на два регистра RCA1 и RCA2, разрядность которого $r1 \geq]\log_2 h[$ (рис.8). Изменение информации в регистре RCA1 будем, как и ранее, осуществлять в каждом цикле ПЛМ. Изменение информации в регистре RCA2 будем осуществлять при появлении “1” на введенном дополнительном выходе записи следующего адреса ЗСА. В этом случае необходимо при переходах из состояния S_{i0} в состояние S_{j0} закодировать ЗСА=1 и, тем самым, осуществить

запись в регистр RCA1 кода, соответствующего состоянию $S_{j0}=\{s_{j0k}|_{k=1\div h}\}$. Затем на всех переходах из состояния S_{j0} в состояние S_{j0} сохранять ЗСА =0. Переходы из состояния S_{j0} в состояние S_{j0} необходимо осуществлять в зависимости от кода регистра RCA1. Увеличение площади составит:

$$\Delta S_2^2 \sim (2\cdot(L+r)+N)\cdot\sum_{k=1}^{h} m_k + 3(]\log_2(M+M1(h-1))[-]\log_2 M[+1))(p+\sum_{k=1}^{h} m_k) \qquad (26)$$

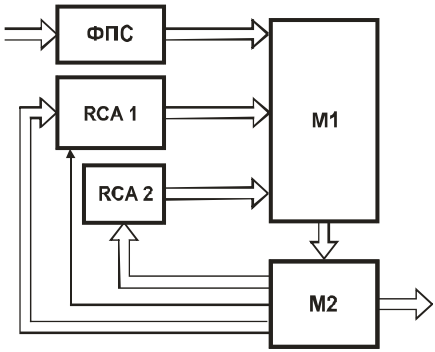


Рис. 8. Структура БМПУ с разбиением RCA на две части

Эффективным изменением структуры для рассматриваемого варианта является введение стека для хранения адресов. Очевидно, что разрядность стека r_s должна быть меньше r, так как необходимо различать переходы БМПУ из состояний S_{i0} и S_{i1} . Для управления стеком необходимо ввести два дополнительных выхода ПЛМ, в которых следует закодировать три команды управления:

запись содержимого регистра RCA в стек;

чтение верхушки стека на входы ПЛМ;

перепись регистра RCA на входы ПЛМ.

Увеличение площади ПЛМ в этом случае будет:

$$\Delta S_2^2 \sim (2\cdot(L+r)+N)\cdot\sum_{k=1}^{H} m_k + 2(p+\sum_{k=1}^{h} m_k) \qquad (27)$$

Для рассматриваемого графа переходов достаточна глубина стека,

равная 1. Варианты 3 и 4 имеют преимущество перед вариантами 1 и 2. Они могут также эффективно использоваться и в случае графа переходов, который соответствует переходам из разных микропрограмм в одну и, после ее выполнения, возвратом в исходные.

Пусть S_i - некоторое состояние БМПУ, из которого он под воздействием логических условий X_{ij} переходит в состояние S_j. При таком переходе на выходах БМПУ образуется микрокоманда Y_{ij}. Пусть также далее необходимо повторить микрокоманду Y_{ij} к раз вне зависимости от входных логических условий. Такой фрагмент встречается в микропрограммах умножения, деления, параметрических сдвигов и др. Для реализации такого варианта основное количество логических произведений требуется для обеспечения подсчета количества циклов повторений.

Исходя из формул разрядов и переносов счетчика, легко показать, что для реализации n-разрядного счетчика необходимо $n\cdot(n+1)/2$ логических произведений. Увеличение площади БМПУ при кодировании на ПЛМ состояний переходов двоичного счетчика получим

$$\Delta S \sim (2\cdot(к+r)+N)\cdot n\cdot(n+1)/2$$

Так как для обеспечения счета количества пройденных состояний не требуется обеспечивать, чтобы при переходе БМПУ из состояния S_i в состояние S_{i+1} разность кодов была равна 1, можно применять другое кодирование состояний. Например, реализовать счет циклов путем кодирования структуры сдвигового регистра. Циклический сдвиг n-разрядного слова требует n логических произведений. Если при этом передача информации между крайними разрядами осуществляется с инверсией, то при этом реализуется 2n состояний.

При увеличении количества циклов и при большом количестве микропрограмм, в которых требуется повторение микрокоманды либо группы микрокоманд, целесообразно ввести в БМПУ аппаратный счетчик. Для управления таким счетчиком целесообразно закодировать следующие команды:

- загрузить счетчик;
- передать содержимое счетчика на входы ПЛМ.

Для введения этих команд управления счетчиком достаточно увеличить ПЛМ на 1 выход, тем самым, увеличение площади ПЛМ со-

ставит:

$$\Delta S_3 = p \qquad (28)$$

Построение МПА с применением ПЗУ и ПЛМ. Рассмотрим реализацию БМПУ на ПЗУ, а также ПЗУ и ПЛМ совместно с точки зрения обеспечения минимума его общей площади.

Пусть N – разрядность микрокоманды, а M – общее количество микрокоманд в микропрограммах. Для реализации накопителя микропрограмм потребуется ПЗУ объемом N·M бит информации. Разрядность кода адреса, поступающего на входы дешифраторов такого ПЗУ для обеспечения выборки любой из M микрокоманд, равна r=]$\log_2$M[. Наиболее экономичным и часто применяемым дешифратором для ПЗУ является прямоугольный дешифратор, т.е. имеется дешифратор строк на L входов и дешифратор столбцов на m входов (m+L=r). Так как дешифратор по своей логике работы является некоторой матрицей полного перебора всевозможных конъюнкций от входных переменных и их инверсий, то площадь дешифратора можно представить следующим выражением:

$$S_{дш} = S_{яд1}\cdot 2\cdot L\cdot 2^{L} + S_{яд2}\cdot 2\cdot m\cdot 2^{m}\cdot N$$

Где $S_{яд}$ - площадь топологии ячейки дешифратора. Примем $S_{яд1}=S_{яд2}=S_{яд.}$ Общая площадь ПЗУ вместе с прямоугольным дешифратором равна:

$$S_{пзу} \sim S_{яд}\cdot 2\cdot(L\cdot 2^{L}+N\cdot(r-L)\cdot 2^{r-L})+N\cdot 2^{r} \qquad (29)$$

Найдем оптимальное разбиение по количеству входов дешифраторов строк и столбцов такого ПЗУ с точки зрения минимума его площади, т.е.

$$\frac{\partial(S_{ПЗУ})}{\partial L} = 0$$

Продифференцировав и прологарифмировав выражение 29, получим:

$$2L-\log_2\left(\frac{1+r-L}{1+L}\right) = r+\log_2 N \qquad (30)$$

Так как величина L находится в пределах $0 \le L \le r$, то из формулы 30

следует, что при больших r и малых N, $L \to r/2$, а при малых r и больших N, $L \to r$.

Сравним для этих двух вариантов площади ПЛМ и ПЗУ. Примем $S_Я = S_{ЯПЗУ} = S_{ЯПЛМ}$

1. $L=r$ $\quad S_{пзу} \sim S_я \cdot (2 \cdot r \cdot 2^r + N \cdot 2^r)$

$$S_{плм} \sim S_я \cdot (2 \cdot r + N) \cdot 2^r / K_{сж}$$

$K_{сж}$- коэффициент сжатия информации в ПЛМ, т.е. считаем, что в исходном представлении ПЛМ имела r входов, N выходов и 2^r логических произведений, а после минимизации количество логических произведений уменьшилось в $K_{сж}$ раз. В этом случае ПЛМ имеет преимущество перед ПЗУ. Однако и в этом случае иногда применяют ПЗУ в качестве накопителя микрокоманд с целью сокращения времени выполнения микропрограмм. Одним из таких характерных случаев является необходимость обеспечения передачи в блок обработки из БМПУ констант, что приводит к структуре БМПУ, включающей и ПЛМ и ПЗУ.

2. $L=r/2$ $\quad S_{пзу} \sim S_я \cdot (\cdot r \cdot 2^{r/2} \cdot (N+1) + N \cdot 2^r)$

$$S_{плм} \sim S_я \cdot (2 \cdot r + N) \cdot 2^r / K_{сж}$$

Площадь ПЛМ равна площади ПЗУ с учетом того, что r – велико при значении $K_{сж} = 1 + 2r/N$ (31).

Формула 31 показывает, что при увеличении объема блока микропрограммного управления, т.е. при росте r и сохранении разрядности микрокоманды N, коэффициент сжатия $K_{сж}$ для обеспечения преимущества ПЛМ должен возрастать. Однако, количество логических произведений ПЛМ зависит от логических условий, по которым формируются микрокоманды. В практических случаях количество логических произведений возрастает почти линейно с ростом объема микропрограмм и, следовательно, коэффициент сжатия остается постоянным. Т.е. для БМПУ с большим объемом микропрограмм целесообразно применять ПЗУ в качестве накопителя микропрограмм.

Выводы

Рассмотренные в работе методы логического и структурного анализа и минимизации аппаратных затрат при описании БМПУ в ви-

де графа переходов позволяют сделать следующие выводы:

- для снижения аппаратных затрат эффективно введение промежуточных состояний в граф переходов БМПУ;
- структурная декомпозиция БМПУ (введение мультиплексирования на входе, демультиплексирования на выходе, разбиение входного регистра и регистра следующего адреса на части, с различными алгоритмами смены информации в них, введение управляющего счетчика) позволяет существенно сократить объем накопителя микропрограмм типа ПЛМ;
- в качестве накопителя микропрограмм при малых и средних объемах предпочтительно использовать накопитель типа ПЛМ, а при больших – типа ПЗУ.

Литература:

1. **Баранов С.И., Синев В.Н.** «Программируемые логические матрицы в цифровых системах». Зарубежная радиоэлектроника, №1, 1979.
2. **Горбатов В.А**. Схемы управления ЦВМ и графы. М.«Энергия», 1971.
3. **Галицин А.А.** Минимизация количества состояний графа переходов и ее влияние на размерность автоматов управления на основе СБИС ПЛМ. Микроэлектроника, т.11, вып.3, 1982.
4. **Зольников В.К., Машевич П.Р.** Логическая оптимизация блоков микропрограммного управления СБИС. Юбилейная международная научно-техническая конференция «50 лет модулярной арифметике», Сборник научных трудов, М. 2005.
5. **Таннбаум Э.** Многоуровневая организация ЭВМ, М. Мир, 1979г.

УДК. 681.3

В.К. Зольников, П.Р. Машевич

Логическая оптимизация блоков микропрограммного управления СБИС

(*Воронежская лесотехническая академия, ОАО «Ангстрем»*)

Исследованы вопросы введения промежуточных состояний для уменьшения площади БМПУ на кристалле при реализации различных вариантов графа переходов. Проведена логическая минимизация ПЛМ для типовых фрагментов графа переходов БМПУ при конъюнктивной форме логических условий, кодировании команд методом расширения кодов операций и др. Сделаны оценки площади БМПУ на кристалле и изменения площади при оптимизации графа переходов..

При построении блоков управления СБИС микропрограммное управление при прочих равных условиях является более предпочтительным, так как обеспечивает возможность перестройки алгоритма работы СБИС на более поздних этапах проектирования, а также выпуск некоторого количества типов модифицированных вариантов СБИС с минимальными затратами на новый проект. В частности, микропрограммное управление позволяет перестроить работу операционного блока двоичной арифметики в блок обработки в модулярной либо в троичной арифметике [1].

Функционирование БМПУ определяется множеством входных переменных $X=\{x_{i|i=1\div L}\}$ (логических условий), множеством выходных переменных $Y=\{y_{i|i=1\div N}\}$ (микрокоманд) и порядком их следования. Порядок следования микрокоманд определяется микропрограммой. Микропрограмме ставится в соответствие граф переходов $G_n=<V,U>$, где носитель графа $V=\{s_{i|i=1\div s}\}$ представляет собой множество состояний БМПУ, а сигнатура $U=\{u_{ij|ij=1\div s}\}$ – множество переходов БМПУ [2]. Каждая вершина графа переходов соответствует внутреннему состоянию S_i БМПУ, а переходы из состояния S_i в состояние S_j представлены в виде ребер U_{ij} графа, причем, каждому ребру представлена в соответствие пара векторов X_{ij} (входные переменные) и $Y_{ij.}$ (выходные переменные).

Рассмотрим реализацию БМПУ на структуре ПЛМ с регистровой памятью (рис.1). Он состоит из формирователей парафазных сигналов (ФПС), матрицы (М1), которая формирует p конъюнктивных термов (логических произведений) P_1, P_2,…Pp от L+r переменных (или их отрицаний), матрицы (М2), которая реализует N дизъюнкций (N=r+t, где N – общая разрядность микрокоманды, r – разрядность регистра следующего адреса RCA, включенного в цепи обратной связи между матрицами М1 и М2, t – разрядность микрокоманды управления операционным автоматом) от p логических произведений, полученных в М1. Площадь, занимаемая таким блоком на кристалле, определяется следующей зависимостью [3]:

$$S \sim (2\cdot(L+r)+N)\cdot p \qquad (1)$$

Основную площадь при изготовлении БМПУ на кристалле занимает именно ПЛМ, так как при увеличении количества логических произведений p возрастает площадь матриц М1 и М2 и не изменяется площадь ФПС и RCA. Таким образом, при выполнении требований по быстродействию, оптимальным можно считать БМПУ, площадь которого минимальна.

Рассмотрим методы логической декомпозиции БМПУ, позволяющие уменьшить площадь блока.

Оптимизация БМПУ для конъюнктивной формы логических условий. Допущение о конъюнктивной форме логических условий является отражением того часто встречающегося факта, что коды команд, поступающие на входы БМПУ, разбиты на ряд полей. Между полями существует логическая связь, т.е. операции, закодиро-

ванные в одном поле команды, выполняются лишь при появлении некоторых кодов в других полях команды [3]. Это можно представить в виде конъюнкций логических условий в соответствующих полях команды.

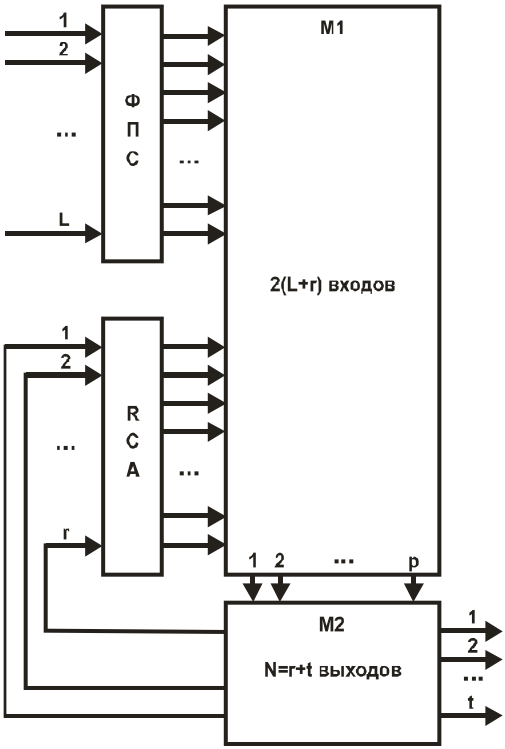


Рис.1.Структурная схема БМПУ на ПЛМ с регистровой памятью

Пусть S_{i0} - некоторое состояние БМПУ, из которого под воздействием множества логических условий $X_{i,j}=\{x_{i,jk}|_{k=1\div l}\}$ блок переходит в множество состояний $Sj=\{s_{jk}|_{k=1\div L}\}$. При таком переходе на выходах образуется множество выходных функций $Y_{i,j}=\{y_{i,jk}|_{k=1\div l}\}$.

Введем некоторые ограничения:

- Среди множества всевозможных микрокоманд $Y=\{y_i|_{i=1\div t}\}$ существует такая микрокоманда $Y\_$, подача которой не изменяет состояние операционного автомата.
- Множество входных логических условий $X_{i,j}=\{x_{i,jk}|_{k=1\div l}\}$ можно представить в виде конъюнктивной формы двух подмно-

жеств логических условий $X_{i,j}=F_{i,j}\wedge\Psi$ ($\Psi=\{\Psi_{k|k=1\div h}\}$).

$$X_{i,jk}=F_{i,j}\wedge\psi_k$$

Подмножеству $F_{i,j}$ соответствует некоторая кратчайшая дизъюнктивная нормальная форма (ДНФ), которая включает в себя m элементарных конъюнкций

$$\mathbf{F}_{i,j}=\bigvee_{q=1}^{m} f_q$$

$$f_q \{ f_q=\bigwedge_{i=1}^{\mathbf{L}} X_i^{e_{qi}} \quad e_{qi}\in(-1,1,0)\},$$

$x_i=x_i$ ($e_{qi}=-1$), $x_i=x_i$ ($e_{qi}=1$), x_i не входит в конъюнкцию ($e_{qi}=0$)

Элементы подмножества Ψ представляют собой различные элементарные конъюнкции.

$$\Psi_k=\bigwedge_{i=1}^{h} X_i^{e_{ki}}$$

Фрагмент графа переходов БМПУ для этого случая изображен на рис.2. Для реализации такого варианта фрагмента графа переходов на ПЛМ потребуется h·m логических произведений.

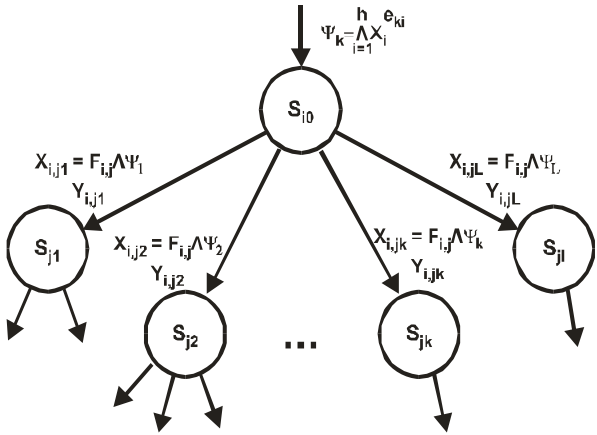


Рис. 2. Фрагмент графа переходов БМПУ
для конъюнктивной формы логических условий

Если площадь БМПУ определяется (1), увеличение площади при непосредственном кодировании такого варианта фрагмента графа переходов составляет:

$$\Delta S_1 \sim (2\cdot(L+r)+N)\cdot h\cdot m \qquad (2)$$

Введем в фрагмент графа переходов некоторое промежуточное состояние S_{i1} (рис.3.), переход в которое из состояния S_{io} осуществляется под воздействием подмножества логических условий $F_{i,j}$. На выходах БМПУ сформируем при этом микрокоманду с кодом $Y_{\_}$, не изменяющую состояние операционного автомата. Переход БМПУ из состояния S_{i1} в множество состояний $S_j=\{s_{jk}|_{k=1\div h}\}$ осуществляется под воздействием логических условий $\Psi=\{f_{k}|_{k=1\div h}\}$.

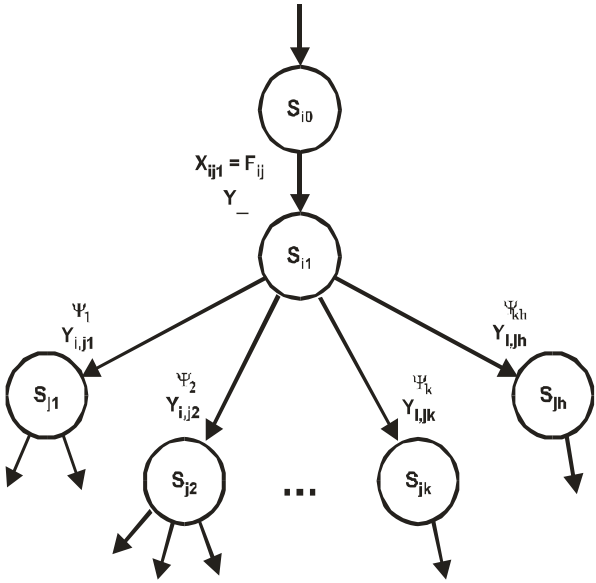


Рис.3. Фрагмент графа переходов БМПУ для конъюнктивной формы логических условий с введением одного промежуточного состояния

Для реализации на ПЛМ перехода из состояния S_{i0} в состояние S_{i1} требуется m логических произведений. Во втором цикле работы ПЛМ для реализации переходов из состояния S_{i1} в состояние $S_j=\{s_{j,k}|_{k=1\div h}\}$ требуется h логических произведений. Общее количество логических произведений для реализации такого варианта графа переходов равно h+m.

Введение промежуточного состояния приводит к увеличению общего количества состояний БМПУ. Пусть до введения промежуточного состояния S_{i1} общее количество состояний было M. Разрядность r регистра RCA была выбрана исходя из следующего соотношения $r =]\log_2 M[$. При этом необходимо рассматривать два варианта:

а) $M<2^r$, тогда при введении состояния S_{i1} не требуется увеличения разрядности регистра RCA;

б) $M=2^r$, тогда разрядность регистра RCA увеличится на 1.

В общем случае увеличение площади ПЛМ можно представить в виде:

$$\Delta S_2 \sim (2\cdot(L+r)+N)\cdot(h+m)+3\cdot(]\log_2 (M+1)[-]\log_2 M[)\cdot(p+h+m)$$

В реальных случаях условие а) всегда выполняется, т.е. $M<2^r$, так как разрядность r регистра RCA выбирают с некоторым запасом, т.е. увеличение площади ПЛМ можно представить в виде:

$$\Delta S_2 \sim (2\cdot(L+r)+N)\cdot(h+m) \qquad (3).$$

$$\Delta S_1 > \Delta S_2 \text{, для условия } l \geq 2 \text{ и } m \geq 2 \qquad (4).$$

Таким образом, кодирование БМПУ с введением промежуточного состояния будет давать выигрыш в площади ПЛМ при выполнении условия (4).

Рассмотрим оценки площади ПЛМ при конъюнктивной форме логических условий для некоторого общего случая:

- подмножество логических условий $F_{i,j}$ представим в виде конъюнктивной формы из **c** конъюнкций подмножеств логических условий, т.е.

$$F_{i,j} = \bigwedge_{i=1}^{c} F^q_{i,j} \text{ ,}$$

  а каждое подмножество $F^q_{i,j}$, представленное в виде кратчайшей ДНФ, состоит из m_q элементарных конъюнкций;

- элементы f_k подмножества условий, представленные в виде кратчайшей ДНФ, содержат каждый t_k элементарных конъюнкций.

Увеличение площади при этом составляет:

$$\Delta S_1 \sim (2(L+r)+N) \sum_{k=1}^{l} t_k \cdot \prod_{q=1}^{C} m_q \qquad (5)$$

Применив метод введения промежуточных состояний можно рассмотреть два варианта фрагмента графа переходов БМПУ.

1. Введем одно промежуточное состояние S_{i1} и учитывая выполнение условия а), увеличение площади составит:

$$\Delta S_2 \sim (2\cdot(L+r)+N)\cdot\left(\sum_{k=1}^{l} t_k + \prod_{q=1}^{C} m_q\right) \qquad (6)$$

2. Введем С промежуточных состояний и учитывая выполнение условия а), увеличение площади составит:

$$\Delta S_2 \sim (2\cdot(L+r)+N)\cdot\left(\sum_{k=1}^{l} t_k + \sum_{q=1}^{C} m_q\right) \qquad (7)$$

Однако, при такой декомпозиции БМПУ существенно увеличивается время перехода из состояния S_{i0} в множество состояний S_j. Оно составляет С+1 цикл работы блока. Если такие временные затраты недопустимы, может быть реализован вариант декомпозиции, являющийся объединением рассмотренных ранее вариантов.

Кодирование команд методом расширения кодов. При проектировании МК и МП СБИС стоит задача разработки системы команд, при этом стремятся максимально сократить разрядность команды для уменьшения объема памяти и разрядности магистрали между процессором и памятью. Поэтому для кодирования систем команд часто применяют метод расширенных кодов операций, дающий хорошие результаты по уменьшению разрядности команды [4]. Пример кодирования команд методом расширения кодов операций приведен в таблице 1.

При расширении кодов операций переход к тому или иному типу адресации осуществляется в том случае, если в части полей кодов операций, имеющих более высокий уровень иерархии, коды совпадают с выделенными кодами, а код в поле, соответствующем именно этому типу адресации, не равен выделенному коду.

Таблица 1

КОП1	Поле адресации по КОП1		
1 1 1 1	КОП2	Поле адресации по КОП2	
1 1 1 1	1 1 1	КОП3	
1 1 1 1	1 1 1	1 1 1	КОП4

КОП1 – 2^4 –1 - операций
КОП2 – 2^3 –1 - операций
КОП3 – 2^3 –1 - операций

Применим методику введения промежуточных состояний для такого варианта логических условий и оценим при этом площадь.

Введем некоторые ограничения:

- Среди множества всевозможных микрокоманд $Y=\{y_{i|i=1\div t}\}$ существует такая микрокоманда $Y_{\_}$, подача которой не изменяет состояние операционного автомата.
- Множеству входных логических условий $X_{i,j}=\{x_{i,jk|k=1\div h}\}$ соответствует некоторое множество логических условий $\Psi=\{\Psi_{k|k=1\div h}\}$, причем каждое логическое условие $x_{i,jk}$, вызывающее переход из состояния S_{i0} в состояние S_{jk} равно:

$$X_{i,j1}= \bar{\psi}_1$$

$$X_{i,j2}=\psi_1 \cdot \bar{\psi}_2$$

$$\bullet\bullet\bullet$$

$$X_{i,jh}= =\psi_1\cdot\psi_2\cdot\ldots\cdot \bar{\psi}_h$$

Элементы множества Ψ представляют собой различные элементарные конъюнкции. Фрагмент графа переходов с такими логическими условиями изображен на рис.4. Отметим, что для реализации на ПЛМ функции ψ_k потребуется m_k логических произведений, что следует из теоремы де Моргана. Таким образом увеличение площади ПЛМ составляет

$$\Delta S_1 \sim (2\cdot(L+r)+N) \cdot \sum^{h} m_k \qquad (8)$$

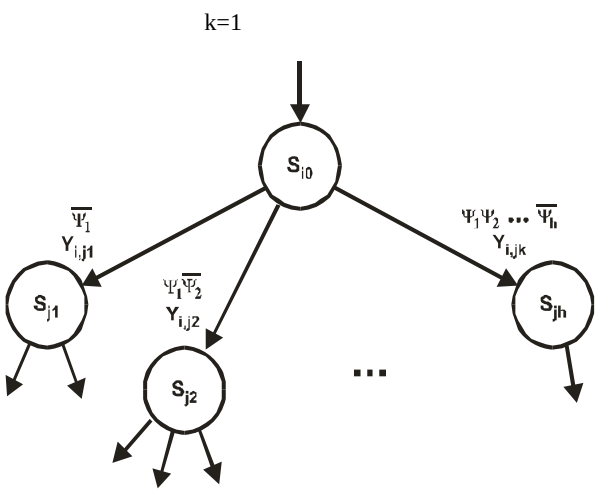


Рис.4. Фрагмент графа переходов БМПУ
для кодирования команд методом расширения кодов операций

Введем в фрагмент графа переходов промежуточные состояния, переход в которые осуществляется под воздействием множества логических условий ψ, а именно переход в состояние S_{i1} осуществляется под воздействием условия ψ_1, в состояние S_{i2} по воздействием $\psi_1.\psi_2$ и т.д. На выходах БМПУ сформируем при этом микрокоманду Y_. Для реализации на ПЛМ переходов из состояния S_{i0} в состояние $S_i = \{s_{ik}|_{k=1\div L}\}$ требуется L логических произведений, увеличение разрядности RCA при этом будет не $]\log_2 h[$, а h. Коды на h выходах RCA сформируем следующим образом – истинность логического условия ψ_1 вызывает появление “1” в первом разряде, ψ_k - “1” в k-ом разряде. Тогда состояния $S_i = \{s_{ik}|_{k=1\div h}\}$ будут закодированы следующим образом:

	1	2	3		h
S_{i1} =	0	X	X	…	X
S_{i2} =	1	0	X	…	X
…					
S_{ih} =	1	1	1	…	0

Переход из состояний S_i в состояния S_j, который выполняется во втором цикле работы ПЛМ, требует также h логических произведений, причем он вызван лишь кодом внутреннего состояния и не

зависит от входных логических условий. Увеличение площади ПЛМ равно:

$$\Delta S_2 \sim (2\cdot(L+r)+N)\cdot 2h + 3h(p+2h) \qquad (9)$$

Так как количество полей в команде при расширенном кодировании невелико (2÷3) и количество разрядов в поле (m_k) также невелико (3÷5), то площади ΔS_1 и ΔS_2 могут быть сравнимы, поэтому необходимо анализировать конкретную ситуацию.

Если допустимо снижение быстродействия БМПУ, можно ввести многократное выполнение переходов из состояния S_{i0} в состояние S_j. Введем для этого 2h промежуточных состояний. Переход из некоторого состояния $S_{i,k,1}$ в состояние $S_{i,k+1,1}$ или $S_{i,k+1,0}$ выполняется под воздействием логического условия ψ_{k+1}. На выходах БМПУ при этом формируется микрокоманда Y_. Переход из состояния $S_{i,k,0}$ в состояние $S_{j,k}$ выполняется безусловно. Кодирование промежуточных состояний S_i БМПУ можно осуществлять различными способами, при этом необходимо обеспечивать при переходе из состояния $S_{i,k,1}$ в любом разряде RCA появление “1” при истинности логического условия ψ_{k+1} и “0” – в противном случае. Пример такого кодирования состояний приведен на рис.5.

Количество логических произведений, как и в предыдущем случае, равно 2h, а количество разрядов, на которое необходимо увеличить регистр RCA , определяются по формуле:

$$]\log_2 (M+2h)[-] \log_2 M[,$$

общее увеличение площади ПЛМ составляет:

$$\Delta S_2 \sim (2\cdot(L+r)+N)\cdot 2h+3\cdot(]\log_2 (M+2h)[-]\log_2 M[)\cdot(p+2h) \qquad (10)$$

S_{i10}	•••••	0	0	0	0
S_{i20}	•••••	0	0	1	0
•••					
S_{iL0}	•••••	1	1	1	0

S_{i11}	•••••	0	0	0	1
S_{i21}	•••••	0	0	1	1
•••					
S_{iL1}	•••••	1	1	1	1

Рис. 5. Пример кодирования промежуточных состояний **S_i БМПУ**

Выводы

Рассмотренные в работе методы логического анализа и минимизации аппаратных затрат при описании БМПУ в виде графа перехо-

дов позволяют сделать следующий вывод – для снижения аппаратных затрат эффективно введение промежуточных состояний в граф переходов БМПУ;

Литература

1.. **Малашевич Д.Б, Машевич П.Р.** Элементная база модулярных и троичных ЭВМ. Юбилейная международная научно-техническая конференция «50 лет модулярной арифметике», Сборник научных трудов, М, Зеленоград. 2005.

2. **Горбатов В.А**. Схемы управления ЦВМ и графы. М.«Энергия», 1971.

3. **Баранов С.И., Синев В.Н.** «Программируемые логические матрицы в цифровых системах». Зарубежная радиоэлектроника, №1, 1979г.

4. **Таннбаум Э.** Многоуровневая организация ЭВМ, М. Мир, 1979г.

УДК 62-507.15

О.Н. Музыченко

Методы синтеза логических схем модульного контроля в натуральных двоичных кодах

(*Балтийский государственный технический университет «Военмех» им. Д.Ф. Устинова, Санкт-Петербург*)

Рассмотрены методы синтеза логических схем модульного контроля в натуральных двоичных кодах. Получены оценки сложности и быстродействия схем, синтезируемых различными методами.

1. Введение

Устройства модульного контроля параллельных двоичных кодов находят достаточно широкое применение [1-8] в современных цифровых системах управления, передачи и переработки дискретной информации, и, прежде всего, в системах специального назначения, к надежности, достоверности функционирования и контролепригодности которых предъявляются высокие требования. Они применяются в качестве средств аппаратного контроля в системах, использующих контроль по модулю [1-5], арифметические [6-8] и ряд других кодов, в устройствах, работающих в системе остаточных классов [4], в аппаратуре кодирования и декодирования помехозащищенных кодов в системах передачи информации.

В современных цифровых системах используется как числовой, так и кодовый модульный контроль. В случае числового модульного

контроля синтезируемая схема должна формировать остаток параллельного двоичного кода по модулю К с учетом весов его разрядов. При кодовом модульном контроле схема формирует остаток по модулю К числа единичных разрядов контролируемого кода, при этом веса всех его разрядов считаются единичными.

Известны устройства модульного контроля, работающие в унитарных позиционных и непозиционных двоичных кодах . В работах [9-11] исследованы методы их синтеза, получены оценки сложности и быстродействия схем, синтезируемых различными методами. Достоинством таких устройств является их высокое быстродействие, а основным недостатком - большая сложность, что ограничивает возможности их использования, в особенности при значениях модуля К>5. В данной статье рассматриваются и исследуются методы синтеза логических схем модульного контроля, функционирующих в натуральных двоичных кодах. Несмотря на то, что такие устройства широко применяются на практике, а вопросам их проектирования посвящено большое количество работ [1-5, 12, 14, 27] , проблема не получила достаточно полного теоретического решения.

2. Синтез логических схем формирования остатка двоичного кода по модулю К методом последовательной свертки

Анализ известных схемных реализаций устройств модульного контроля двоичных кодов показывает, что большинство из них может быть сведено [12] к обобщенной структуре, состоящей из двух последовательно соединенных блоков. Первый блок осуществляет операцию свертки входного n-разрядного кода $X=\{x_1, x_2,...,x_n\}$, в результате формируется $p \le (2\lambda+1)$-разрядный код остатка Z, где λ- мощность множества значений остатков весов разрядов входного кода по модулю K. Второй блок формирует натуральный двоичный код остатка $Q=(Z)\mathrm{mod}K=(X)\mathrm{mod}K$. В общем случае его реализация требует специальных методов синтеза, рассматриваемых далее.

Введем понятие (m,k)-оператора C_m, где $k<m$, который по набору X формирует набор $X' = \{x'_1, x'_2,..., x'_k\}$ такой, что

$$\left(\sum_{i=1}^{m} \omega_i x_i\right) \mathrm{mod}\, K = \left(\sum_{i=1}^{k} \omega_i x'_i\right) \mathrm{mod}\, K \quad . \tag{1}$$

Очевидно, что осуществляя последовательную свертку кода X с использованием оператора C_m, получим $p \le \lambda+1$ - разрядный код $Z=\{z_1, z_2,...,z_p\}$ остатка входного кода по модулю K.

Первый блок схемы формирования остатка двоичного кода по модулю K, синтезируемой методом последовательной свертки, выполняется в виде композиции блоков свертки, реализующих оператор $C_{m,}$. Порядок их соединения может быть различным от параллельного до последовательного. При параллельном соединении синтезируемая схема имеет древовидную структуру и на каждом уровне группа блоков производит свертку разрядов, сформированных блоками предыдущих уровней, блоки первого уровня производят свертку входных разрядов. Число блоков свертки на каждом уровне зависит от мощности множества значений остатков весов разрядов входного кода по модулю K, его разрядности и способа свертки. Схемы такого типа имеют наибольшее быстродействие. При последовательном соединении на каждом из уровней схема содержит один блок свертки, производящий свертку разрядов, сформированных блоками предыдущих уровней, а также разрядов входного кода. Возможно большое число промежуточных вариантов соединения блоков свертки, приводящих к получению схем различной глубины и сложности.

Рассмотрим возможные варианты реализации оператора C_m.

2.1. Свертка равновесных разрядов кода блоками сложения по модулю два $\left(C_m^1\right)$

Блок сложения обеспечивает свертку по модулю два разрядов $x_i \in X_j$ с весами, удовлетворяющими условию $(\omega_i) \bmod K = \omega = \text{const}$. Он выполняется [12,25] на сумматорах, причем сумматоры соединяются входами с входами блока и выходами суммы сумматоров предыдущих уровней данного блока. Выход суммы последнего сумматора является выходом блока с весом ω, а выходы переноса всех сумматоров являются выходами блока с весом 2ω. Возможны различные варианты соединения сумматоров в блоке сложения по модулю два. Наибольший интерес представляют два предельных варианта их соединения: последовательный и параллельный.

В первом варианте осуществляется последовательное соединение сумматоров, при котором первый сумматор соединен входами с

входами блока, а каждый последующий сумматор соединен двумя входами с входами блока, а третьим входом - с выходом суммы предыдущего сумматора. При такой реализации блока получаем регулярную структуру в виде последовательно соединенной цепочки сумматоров. Ее недостатком является большая глубина схемы, равная $H=\lceil m/2\rceil h_s^s$, где m - число входов блока, а h_s^s - глубина схемы сумматора по выходу суммы, и, следовательно, низкое быстродействие.

Во втором варианте осуществляется параллельное соединение сумматоров в виде древовидной структуры, при котором сумматоры первого уровня соединены входами с входами блока, а сумматоры каждого из последующих уровней - с выходами суммы сумматоров предыдущих уровней. При таком соединении глубина блока сложения по модулю два оказывается минимальной, равной $H=\rceil\log_8 m\lceil h_s^s$.

Возможно также большое число промежуточных вариантов реализации блока сложения по модулю два, отличающихся глубиной схемы.

Рассматриваемый вариант реализации оператора C_m приводит к формированию из входного m-разрядного набора $X_j \subset X$ выходного $k=[m/2]+1$ - разрядного кода $X'=\{x'_1, x'_2,..., x'_k\}$, удовлетворяющего условию (1), причем

$$(2) \qquad \begin{cases} x'_1 = \bigoplus\limits_{\{i\}_j} x_i, \\ x'_i = p_{i-1} \quad (i = 2,3,...,k), \end{cases}$$

где $x_i \in X_j$, а p_{i-1} - функция, реализуемая на выходе переноса $(i-1)$-го сумматора блока сложения по модулю два, $\omega'_1=(\omega_i)\mathrm{mod}K=\omega$, а $\omega'_i=(2\omega)\mathrm{mod}K$.

2.2. Свертка равновесных разрядов кода с использованием параллельного счетчика $\left(C_m^2\right)$

В данном случае блок свертки выполняется [12] в виде композиции параллельных счетчиков, реализующих оператор $\mathcal{E}_n$.

Оператор $\mathcal{E}_n$ представляет собой (m,k)-оператор, где $k=]\log_2(m+1)[$, который по набору $X=\{x_1,x_2,..., x_m\}$ формирует набор $Z=\{ z_1,z_2,..., z_k\}$, являющийся двоичной записью числа единиц в X. Вопросы

реализации таких операторов подробно рассмотрены в [13], где приведены оценки сложности и глубины их схем.

Данный вариант реализации оператора C_m приводит к формированию из входного m-разрядного набора $X_j \subset X$, веса разрядов x_i которого удовлетворяют условию $(\omega_i) \bmod K = \omega = \text{const}$, выходного $k =]\log_2(m+1)[$ - разрядного кода $X' = \{x'_1, x'_2, ..., x'_k\}$, являющегося двоичной записью числа единиц в X_j, причем сформированный код удовлетворяет условию (1), а его i-й разряд x'_i имеет вес $\omega'_i = (2^{i-1}\omega) \bmod K$.

Для данного варианта свертки наиболее удобны структуры с параллельным соединением блоков свертки.

2.3. Свертка разрядов кода многоразрядным сумматором $\left(C_m^3\right)$.

В данном случае блок свертки выполняется [12,14] в виде композиции многоразрядных сумматоров двоичных чисел.

Использование для свертки кода многоразрядного сумматора приводит к формированию из входного $m = (2\xi+1)$-разрядного набора $X_j \subset X$, имеющего веса разрядов, равные $\omega_i = \omega_{\xi+i} = (2^{i-1}\omega_1) \bmod K$, где ω_1 -веса разрядов x_1, $x_{\xi+1}$, $x_{2\xi+1}$ входного набора X, подаваемых на входы первого разряда ξ-разрядного сумматора, выходного $k = (\xi+1)$ - разрядного кода $X' = \{x'_1, x'_2, ..., x'_k\}$, удовлетворяющего условию (1), разряд x'_i которого имеет вес $\omega'_i = (2^{i-1}\omega_1) \bmod K$. Схема устройства формирования остатка по модулю $K=7$, синтезированная данным методом приведена на рис.1.

2.4. Свертка разрядов входного кода блоками модульного сложения $\left(C_m^4\right)$.

В данном случае блок свертки выполняется [15,16] в виде композиции устройств сложения по модулю K двух двоичных кодов. Они отличаются от модульных сумматоров более широкими функциональными возможностями, обеспечивая суммирование по модулю K входных кодов X_j , $X_i > K-1$.

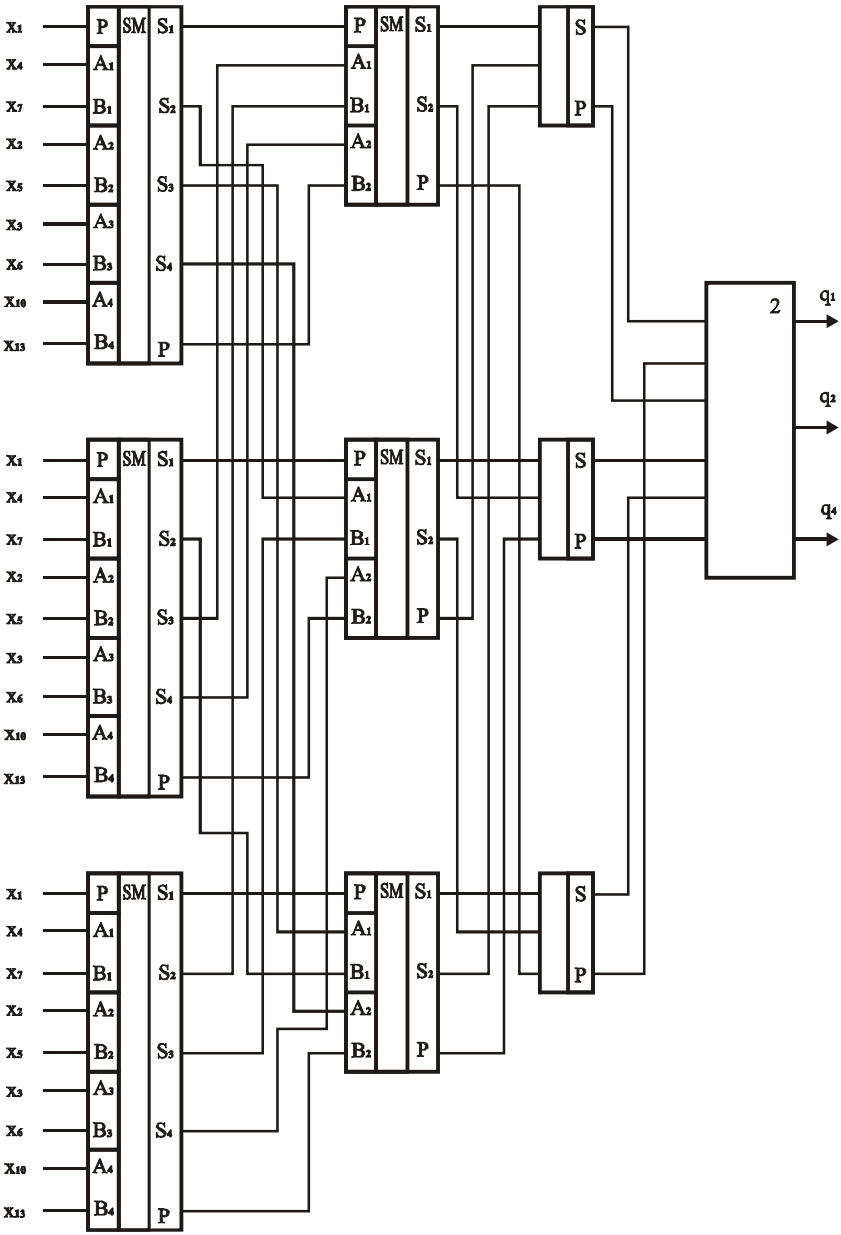


Рис. 1. Схема формирования остатка натурального двоичного 27-разрядного кода по модулю К=7, синтезированная методом свертки многоразрядными сумматорами

Использование для свертки кода устройства модульного сложения приводит к формированию из входного $m=2t$-разрядного набора $X_j \subset X$, где $t \leq]\log_2 K[$, имеющего веса разрядов, равные $(2^{i-1}\omega_1) \bmod K$ ($i= 1,2,...,t$), где ω_1 -веса разрядов входного набора X, подаваемых на входы первого разряда устройства, выходного $k \leq]\log_2 K[$-разрядного набора $X'=\{x'_1, x'_2,..., x'_k\}$, удовлетворяющего условию (1), i-й разряд которого имеет вес $\omega'_i=(2^{i-1}\omega_1) \bmod K$. Если выходные коды блоков удовлетворяют условию $X'<K$, то их дальнейшая свертка может осуществляться модульными сумматорами.

Для данного способа свертки наиболее удобны структуры с параллельным соединением блоков модульного сложения. С целью упрощения синтезируемого устройства могут использоваться блоки модульного сложения, формирующие на ряде входных наборов не приведенный по модулю K код.

Рассмотренный способ свертки наиболее эффективен в случае $K=2^r-1$, при этом в качестве блоков свертки могут использоваться r-разрядные сумматоры с обратной связью с выхода переноса старшего разряда на вход переноса первого разряда. Необходимо отметить, что такие блоки свертки на ряде наборов входного кода X такого, что $(X) \bmod K=K$, дают не приведенный код остатка $X'=K$.

Возможна модификация данного метода свертки [17], заключающаяся в том, что сначала свертка осуществляется по модулю $K''=\beta K$, где $\beta=2,3,...,m$, а затем производится свертка сформированного кода по модулю K. Наиболее удобным является случай, когда $K''=2^\gamma-1$, поскольку в этом случае блок свертки по модулю K может быть выполнен в виде многоразрядного сумматора с обратной связью. Следует отметить, что предварительная свертка по модулю K' практически не влияет на сложность синтезируемой схемы, но может приводить к существенному возрастанию ее глубины.

2.5. Свертка с использованием отрицательных весов разрядов входного кода $\left(C_m^5\right)$

Представим входной натуральный двоичный код $X=x_1 2^0+ x_2 2^1+...+ x_n 2^{n-1}$ в виде

(3) $$X=X_0 W_0+ X_1 W_1+...+ X_\rho W_\rho,$$

где $X_i=x_{il+1}2^0+x_{il+2}2^1+...+ x_{(i+1)l}2^{l-1}$, $\rho=]n/l[-1$, W_i - целое число, равное весу $\omega_{il+1}=2^{il}$ разряда x_{il+1} кода X, а $X=\bigcup_{i=0}^{p} X_i$.

При этом

(4) $$(X)\bmod K=\left(\sum_{i=0}^{\rho} X_i 2^{il}\right)\bmod K .$$

Для натурального двоичного кода X всегда существует такое $l\le [K/2]$, что

(5) $$(W_i)\bmod K=(2^{il})\bmod K=(2^l)^i \bmod K=(-1)^i .$$

С учетом свойства (5) получаем

(6) $$(X)\bmod K=\left(\sum_{i=0}^{\rho} X_i(-1)^i\right)\bmod K=$$
$$(X_0-X_1+X_2-...-X_\rho)\bmod K$$

Наиболее эффективно такое представление при $K=2^r+1$, когда $l=r$.

Представление (6) позволяет реализовать несколько вариантов свертки.

Первый вариант [18] основан на следующем представлении уравнения (6):

(7) $$(X)\bmod K=(S_1-S_2)\bmod K=$$
$$((S_1)\bmod K-(S_2)\bmod K)\bmod K=$$
$$\{(X_0+X_2+...+X_{2]\rho/2[})\bmod K-(X_1+X_3+...+X_{2]\rho/2[-1})\bmod K\}\bmod K$$

Суммирование (свертка) кодов X_i в каждой группе представления (7) может осуществляться любым из рассмотренных выше методов. В результате суммирования формируются два кода остатков сумм S_1 и S_2, после чего вычисляется остаток по модулю K их разности. Вычисление разности целесообразно реализовать с использованием инверсного кода.

Данный вариант свертки можно рассматривать как частный случай рассмотренного ранее варианта свертки C_m^3 при дополнительных ограничениях на порядок свертки.

Второй вариант [19] использует последовательную свертку входного кода X многоразрядными сумматорами в соответствии со следующим представлением:

(8) $$\begin{cases} S_i^j = \left(S_\gamma^{j-1} + S_{\gamma+2}^{j-1} + p_i^j\right) \bmod 2^l, \\ p_i^j = \left[\left(S_\gamma^{j-1} + S_{\gamma+2}^{j-1} + p_i^j - S_i^j\right) / 2^l\right], \end{cases}$$

где j=1,…,]$\log_2\rho$[-1 – номер сумматора; i=1,…,]$n/2l$[- номер группы выходов j-го сумматора;

$$\gamma = \begin{cases} 2i-1 \quad \text{при} \quad i = 2[i/2], \\ 2i-2 \quad \text{при} \quad i \neq 2[i/2], \end{cases}$$

$S_i^j = \left\{s_{il}^j, ..., s_{(i-1)l+1}^j\right\}$- i-я группа выходов суммы j-го сумматора;

p_i^j - сигнал переноса li-го разряда j-го сумматора, причем $p_0^1 = 0$, а $p_0^i = p_{]n/2l[}^{j-1}$;

$S_i^0 = X_{i-1}$.

При таком порядке свертки вес сигнала переноса $\omega\left(p_i^j\right) = (-1)^j$ равен весу разрядов s_{li+1}^{j-1} и $s_{(l+1)i+1}^{j-1}$, подаваемых на входы i-го разряда сумматора.

В результате свертки получаем (λ+1)-разрядный код $S = \left\{s_{2l+1}, s_{2l}, ..., s_1\right\} = \left\{p_2^t, S_2^t, S_1^t\right\}$, где t=]$\log_2\rho$[-1, а λ=2l, разряды которого имеют веса W={2^{2l},...,2^1,2^0}, причем их остатки по модулю K равны W'={1,-2^l,...,-1,2^l,...,1},. Последующая свертка до получения (l+1)- разрядного кода может быть проведена путем вычитания (S_1-S_2+P_2), реализуемого на l-разрядном сумматоре с использованием инверсного кода $\overline{S}_2$.

Данный вариант свертки является частным случаем рассмотренного выше варианта свертки C_m^3.

Третий вариант [20], который можно назвать конвейерной сверткой, реализуется на основе следующего представления

(9) $$(X) \bmod K =$$

$$\left(...\left(\left(\left(X_0 - X_1\right) \bmod 2^l + X_2 + p_1\right) \bmod 2^l\right) - ... - X_\rho + p_{\rho-1}\right) \bmod K \ ,$$

где p_i- сигнал переноса, формируемый при выполнении операции ($X_{i-1} \pm X_i$).

Вычисление разности выполняется с использованием инверсного кода

(10) $$(X) \bmod K =$$

$$\left(... \overline{\left(\left(\overline{(X_0 + X_1) \bmod 2^l} + X_2 + p_1 \right) \bmod 2^l \right)} + ... + X_\rho + p_{\rho-1} \right) \bmod K$$

и реализуется следующим образом

$$\begin{cases} S_j = \left(\overline{S}_{j-1} + X_j + p_{j-1} \right) \bmod 2^l, \\ p_j = \left[\left(\overline{S}_{j-1} + X_j + p_{j-1} \right) / 2^l \right], \end{cases} \quad (11)$$

где j=1,2,...,ρ, S_0=X_0, а p_0=0.

Сигналы переноса p_i имеют вес, равный весу переменных младшего разряда следующего сумматора $|(\omega_i) \bmod K|$=1. Получаемая структура состоит из ρ последовательно соединенных l-разрядных сумматоров. Выходные сигналы всех сумматоров инвертируются. Это позволяет избежать необходимости введения поправок, обусловленных использованием для реализации вычитания инверсного кода, так как поправки компенсируются.

В результате свертки формируется (l+1) - разрядный код Z. При формировании окончательного результата необходимо учитывать его знак, так как при четном ρ результат свертки формируется в инверсном коде. Схема формирования остатка по модулю K=9, синтезированная данным методом, приведена на рис.2.

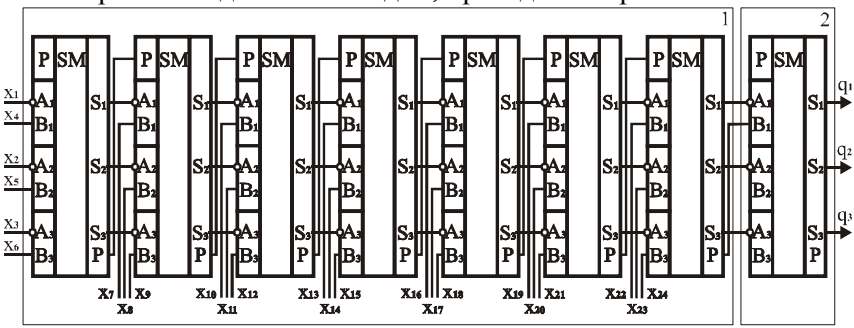


Рис. 2. Схема формирования остатка натурального двоичного 24-разрядного кода по модулю К=9,синтезированная методом конвейерной свертки

Четвертый вариант [21] заключается в последовательной свертке с использованием полных сумматоров разрядов входного кода и сигналов переноса, формируемых при суммировании, имеющих одинаковые значения $|(\omega_i) \bmod K|$, выполняемой в следующем порядке

(12)
$$\begin{cases} s_i^1 = x_i \oplus \overline{x}_{l+i} \oplus x_{2l+1}, \\ p_i^1 = F_3^2\left(x_i, \overline{x}_{l+i}, x_{2l+1}\right), \end{cases}$$

(13)
$$\begin{cases} s_1^j = \overline{s}_1^{j-1} \oplus p_l^{j-1} \oplus x_{(j+1)l+i}, \\ p_1^j = F_3^2\left(\overline{s}_1^{j-1}, p_l^{j-1}, x_{(j+1)l+i}\right), \\ s_i^j = \overline{s}_i^{j-1} \oplus \overline{p}_{i-1}^{j-1} \oplus x_{(j+1)l+i}, \\ p_i^j = F_3^2\left(\overline{s}_i^{j-1}, \overline{p}_{i-1}^{j-1}, x_{(j+1)l+i}\right), \end{cases}$$

где j=2,…,ρ-2, i=1,...,r,

s_i^j, p_i^j - сигналы на выходах суммы и переноса i-го сумматора j-й группы,

$F_3^2\left(z_1, z_2, z_3\right)$ - пороговая равновесная функция трех переменных с порогом a=2.

В результате свертки получаем l=2r-разрядный код Z. При формировании кода остатка необходимо учитывать в прямом или инверсном виде он получен. Схема формирования остатка по модулю K=9, синтезированная данным методом, приведена на рис.3.

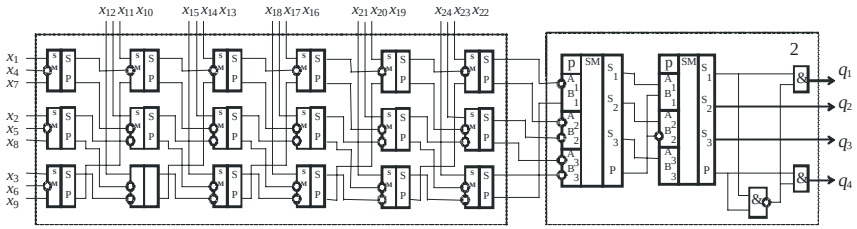


Рис. 3. Схема формирования остатка натурального двоичного 24-разрядного кода по модулю К=9, синтезированная методом свертки полными сумматорами

Возможно одновременное использование рассмотренных вариантов свертки с целью минимизации объема оборудования или глубины схемы за счет оптимального использования номенклатуры имеющихся многоразрядных сумматоров двоичных чисел, приводящее к получению комбинированных структур.

3. Синтез схем формирования кода остатка по модулю К

В том случае, когда входной код X является натуральным двоичным кодом, а $K=2^r-1$, любой из рассмотренных вариантов свертки $C_m^1, C_m^2, C_m^3, C_m^4$ приводит к получению двух r-разрядных кодов Z_1 и Z_2 ($Z= Z_1 \cup Z_2$) из которых суммированием по модулю K формируется код $Q=\{q_1, q_2,..., q_r\}$ остатка входного кода X по модулю K. Т.е. в этом случае второй блок осуществляет суммирование по модулю K двух r-разрядных кодов и при $Z_1,Z_2<K$ выполняется в виде модульного сумматора.

Если $Z_1,Z_2 \geq K$, либо входной код X не является натуральным двоичным кодом, либо $K \neq 2^r-1$, то в результате свертки входного кода получаем $p \leq (\lambda+1)$-разрядный код Z, дальнейшая свертка которого с использованием рассмотренных выше способов свертки оказывается невозможной. Для решения последней задачи требуется использованием специальных способов свертки.

3.1. Синтез блока формирования кода остатка по модулю К методом последовательного расширения и свертки.

Процесс синтеза данным методом [22] заключается в последовательном увеличении разрядности кода $Z=\{z_1,z_2,...,z_{\lambda+1}\}$ за счет представления его разрядов z_i с весами ω_i , удовлетворяющими условию $(\omega_i) \bmod K \neq 2^e$, в виде $z_i = \{z_i^1, z_i^2, ..., z_i^\rho\}$, где ρ - число единичных разрядов в двоичном представлении остатка по модулю K веса ω_i причем $\omega_i^j = 2^{e_j} < 2^t$, где $t = \lceil \log_2 K \rceil$, а $\sum_{j=1}^{\rho} \omega_i^j = (\omega_i) \bmod K$, и последующей свертки сформированного кода с использованием C_m оператора. Проведение операций последовательного расширения кода Z и его свертки позволяет каждый раз сокращать разрядность кода и мощность множества значений остатков весов разрядов по модулю K и получить $\xi \leq 2t$ - разрядный код Z', который можно представить в виде $Z'= Z_1 \cup Z_2$, где $Z_j = \{z_1^j, z_2^j, \ldots, z_t^j\}$ с весами разрядов $\omega_i=2^{i-1}$.

Возможен различный порядок проведения свертки. Наибольший интерес представляют варианты последовательной и параллельной свертки. В первом случае свертка осуществляется

многоразрядными сумматорами, соединенными последовательно. При этом расширение кода производится перед каждым суммированием в объеме, обеспечивающем использование всех входов очередного сумматора.

Во втором случае все разряды z_i кода Z с весами $\omega_i \neq 2^e < 2^t$ представляются набором $\{z_i^1, z_i^2, ..., z_i^\rho\}$ разрядов с весами $\omega_i^j = 2^{e_j}$, после чего производится их свертка любым из рассмотренных способов. При получении в результате суммирования разрядов с весом $\omega_i > 2^t$, для которых $(\omega_i) \bmod K \neq 2^e < 2^t$, производится их расширение, а затем свертка сформированного кода.

Свертка производится до получения кода $Z' \leq 2K-2$ разрядностью не более $2t$ из которого сумматором по модулю K формируется код $Q=\{q_1,q_2,...,q_t\}$ остатка входного кода X.

3.2. Синтез блока формирования кода остатка по модулю К методом прямого формирования вычетов

Пусть Z -натуральный двоичный код, формируемый в результате свертки. При этом

(14) $$Q(Z) = \left(\Psi(Z) + W(Z)\right) \bmod 2^{]\log_2 K[},$$

где $Z = \{z_1, z_2, ..., z_p\}$, $\Psi(Z) = \{z_1, z_2, ..., z_t\} = (Z) \bmod 2^{]\log_2 K[}$, $W(Z) = \{w_1, w_2, ..., w_t\}$, причем

$W(Z) = \left(2^p - \left[Z / K\right]K\right) \bmod 2^{]\log_2 K[}$, а $t =]\log_2 K[$.

Код $\Psi(Z)$ может быть сформирован схемой, состоящей из трех последовательно соединенных блоков, первый из которых реализует монотонный (p,ρ)-оператор $M(p,\{K,2K,...,\rho K\})$ для кода Z ($\rho = [Z_{max}/K] = [(2^p-1)/K]$), а второй реализует систему функций $E=\{E_i\}$, где $E_i = M(p,iK)\overline{M(p,(i+1)K)}$ при $i=K,2K,...,\rho K$. Третий блок является шифратором $P(E)$, осуществляющим преобразование $E \rightarrow W$ вида $w_j = \bigvee\limits_{\{i\}_j} E_i$, где $\{i\}_j$ - множество индексов функций E_i, для которых $E_i = 1$ при $w_i = 1$, и выполняется в общем случае в виде t элементов ИЛИ. Монотонный оператор $M(p,\{K,2K,...,\rho K\})$

реализует систему p пороговых функций с порогами $a= K,2K,...,\rho K$ и весами входных переменных $\omega_i=2^i$. Вопросы синтеза таких схем подробно рассмотрены в [23].

Реализация устройства может быть несколько упрощена, если формировать код $W(Z)$ непосредственно по коду, формируемому монотонным оператором $M(p,\{jK\})$, на основе следующего преобразования

$$\bigvee_{i=\alpha}^{\beta} E_{iK} = M\left(p,\alpha K\right)\overline{M\left(p,(\beta+1)K\right)}. \quad (15)$$

Упрощение достигается за счет исключения формирования ряда функций E_i, которые становятся избыточными.

Пример такой реализации блока формирования остатка приведен на рис. 4 для модуля K=5.

Сложность рассмотренного преобразователя быстро возрастает с увеличением разрядности p кода Z, в основном из-за сложности реализации монотонного оператора $M(p,\{jK\})$.

3.3. Синтез блока формирования кода остатка по модулю K с использованием отрицательных весов разрядов

Синтез блока формирования кода остатка данным методом возможен при значении модуля $K=2^r+1$, когда получаемый в результате свертки (λ+1)-разрядный код $Z=\{z_1,z_2,...,z_{\lambda+1}\}$, где $\lambda=2r$, может быть преобразован в код остатка Q следующим образом

$$Q = \left(Z_1 + 2^r Z_2 + 2^{2r} z_{\lambda+1}\right) \bmod K = \left(Z_1 - Z_2 + z_{\lambda+1}\right) \bmod K, \quad (16)$$

где $Z=Z_1\cup Z_2\cup z_{\lambda+1}$, $Z_1=\{z_1,z_2,...,z_r\}$, а $Z_2=\{z_{r+1},z_{r+2},...,z_\lambda\}$.

Операция вычитания выполняется с использованием инверсного кода, при этом

$$Q = \left(Z_1 - Z_2 + z_{\lambda+1}\right) \bmod K = \left(Z_1 + \overline{Z}_2 - 2^r + 1 + z_{\lambda+1}\right) \bmod K = \quad (17)$$
$$\left\{\left(Z_1 + \overline{Z}_2 + z_{\lambda+1}\right) \bmod K - \left(2^r - 1\right) \bmod K\right\} \bmod K.$$

Откуда для $K=2^r+1$ получаем

$$Q = \left\{\left(Z_1 + \overline{Z}_2 + z_{\lambda+1}\right) \bmod K + 2\right\} \bmod K. \quad (18)$$

Следовательно, при $Z_1 \geq Z_2$

$$\left\{\left(Z_1 + \overline{Z}_2 + z_{\lambda+1}\right) \bmod K + 2\right\} \bmod K = \left(Z_1 + \overline{Z}_2 + z_{\lambda+1}\right) \bmod 2^r + 1, \quad (19)$$

а при $Z_1 < Z_2$

$$\left\{\left(Z_1 + \overline{Z}_2 + z_{\lambda+1}\right) \bmod K + 2\right\} \bmod K = \left(Z_1 + \overline{Z}_2 + z_{\lambda+1}\right) \bmod 2^r + 2. \quad (20)$$

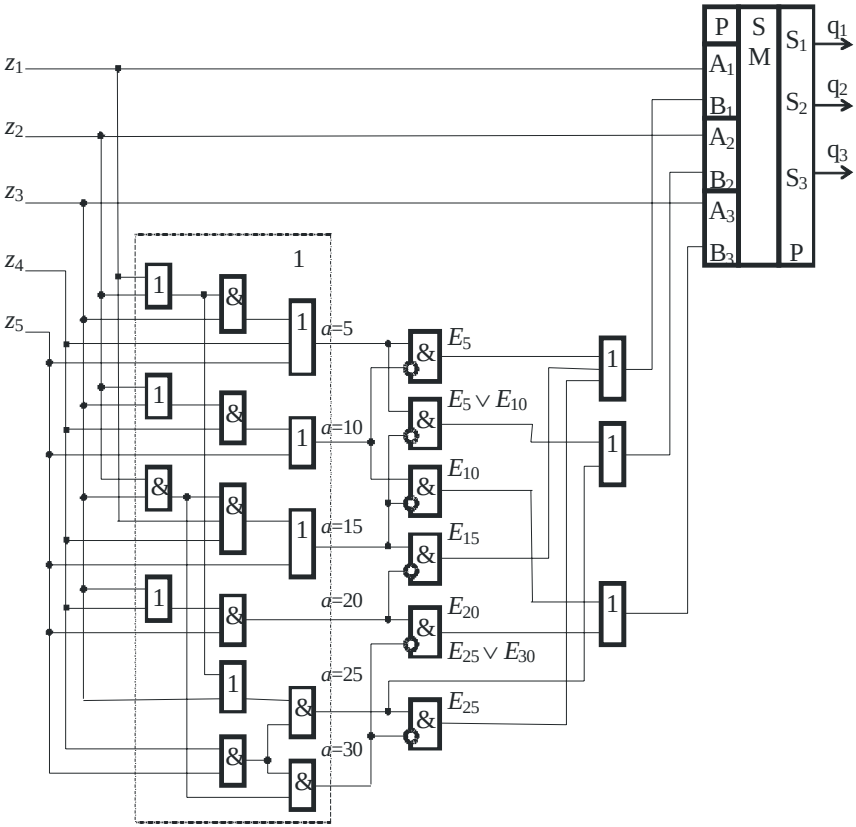


Рис. 4. Схема свертки пятиразрядного двоичного кода по модулю пять, синтезированная методом прямого формирования вычетов

Описанная процедура формирования кода остатка может быть реализована на двух последовательно соединенных *r*-разрядных сумматорах со схемой коррекции. Сумматоры выполняют операцию

(21) $$Q' = \left(Z_1 + \overline{Z}_2 + z_{\lambda+1}\right) \bmod 2^r + p + 2\overline{p} \ ,$$

где p - сигнал на выходе переноса первого сумматора.

Схема коррекции инвертирует единичные разряды результата суммирования в случае его равенства модулю K:

(22) $$q_i = q'_i \overline{\left(\underset{\{j\}}{\&} q'_j \right)},$$

где $\{j\}$ - множество номеров единичных разрядов в двоичном представлении модуля K. Такое инвертирование необходимо только при нулевом входном коде Z.

При отсутствии схемы коррекции в ряде случаев будет формироваться не приведенный по модулю K результат суммирования, если он равен K.

3.4. Синтез блока формирования кода остатка по модулю K комбинированными методами

На основе рассмотренных выше могут быть сформированы [24] два комбинированных метода преобразования кода Z в код остатка Q.

Первый использует комбинацию метода последовательного расширения и свертки и метода формирования вычетов и заключается в следующем. Сначала первым из методов осуществляется свертка $p \le \lambda+1$ - разрядного кода Z в $(t+1)$-разрядный код Z', а затем он преобразуется в код остатка Q с использованием метода прямого формирования вычета в описанном выше порядке.

Экономичные схемы преобразования кода Z в код остатка Q синтезируются при одновременном использовании всех трех рассмотренных методов свертки.

$(\lambda+1)$-разрядный код Z может быть представлен в виде

$$Z = \{Z_1; Z_2; z_{\lambda+1}\} = \{(z_1, z_2, \ldots, z_l); (z_{l+1}, z_{l+2}, \ldots, z_{2l}); z_{\lambda+1}\}, \quad (23)$$

где $l=\lambda/2$, а

$$\omega_i = \begin{cases} \omega_i = (2^{i-1}) \bmod K & \text{при } i=1,2,\ldots,l,\lambda+1, \\ \omega_i - K = -(2^{i-l-1}) \bmod K & \text{при } i=l+1,\ldots,\lambda. \end{cases} \quad (24)$$

При этом на первом шаге свертки кода Z формируется код

$$Z' = Z_1 - Z_2 = Z_1 + \overline{Z}_2 + 1 = \{z'_1, z'_2 \ldots, z'_{l+1}\}, \quad (25)$$

где $z'_{l+1} = \overline{p}$, $z'_i = s_i$, а s_i, p - сигналы, формируемые на i-м выходе суммы и выходе переноса сумматора.

Далее сформированный код Z' и разряд $z_{\lambda+1}$ методом последовательного расширения и свертки преобразуются в $(t+1)$-разрядный код Z'', из которого методом прямого формирования вычетов получается код остатка Q. Схема такого преобразователя для случая $K=11$, приведена на рис. 5.

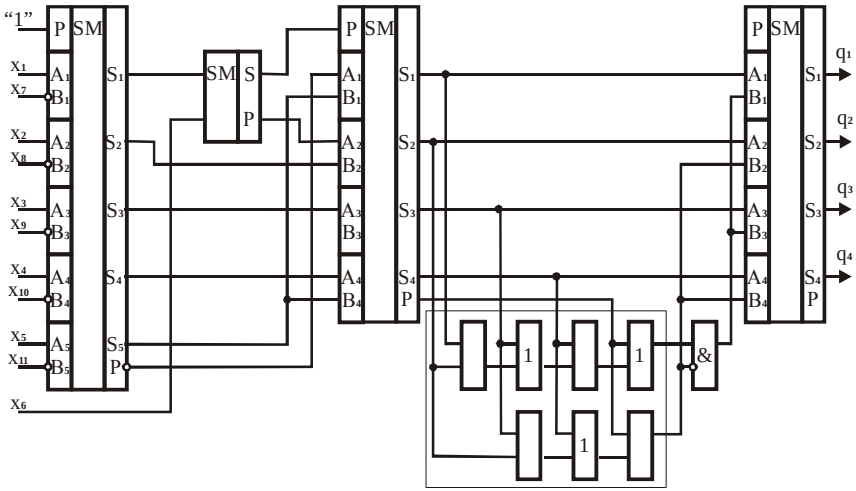


Рис. 5. Схема блока формирования кода остатка по модулю K=11, синтезированная комбинированным методом

4. Синтез логических схем подсчета количества единиц двоичного кода по модулю К методом промежуточного преобразования

Схема, формирующая натуральный двоичный код G остатка по модулю K числа единиц входного кода X, может быть синтезирована путем промежуточного преобразования числа единиц кода X в $\tau \geq 1$ натуральных двоичных кода Y_i числа единиц в X_i, где i=1,2,..., τ, а $X=X_1 \cup X_2 \cup ... \cup X_\tau$ и $X_i \cap X_j = \varnothing$ при $i \neq j$. Каждый из преобразователей кода X_i в код Y_i выполняется в виде (n_i,k_i)-оператора ε_{n_i}, где $k_i =]log_2(n_i+1)[$, который по набору X_i формирует набор Y_i, являющийся двоичной записью числа единиц в X_i. В результате такого преобразования на выходах параллельных счетчиков, реализующих операторы ε_{n_i}, получаем параллельный двоичный m-разрядный код $Y = Y_1 \cup Y_2 \cup ... \cup Y_r = \{y_1, y_2, ..., y_m\}$, где $m = k_1 + k_2 + ... + k_\tau$. Далее для кода Y в описанном выше порядке осуществляется синтез схемы формирования остатка Q по модулю K.

Сложность синтезируемой таким способом схемы составляет

(26) $$L(G(n)) = \sum_{i=1}^{\tau} L\left(\varepsilon_{n_i}\right) + L(D(Y)),$$

где $L\left(\varepsilon_{n_i}\right)$-сложность реализации оператора ε_{n_i}, а $L(D(Y))$-сложность схемы формирования остатка двоичного кода Y по модулю K.

Выбор параметра τ, а также мощности подмножеств переменных X_i позволяет оптимизировать синтезируемую схему по сложности и быстродействию. Решение задачи оптимизации в аналитической форме затруднено из-за рекуррентности оценок сложности реализации операторов ε_{n_i}. Результаты проектирования и оптимизации схем на ЭВМ путем перебора возможных значений параметра τ и мощности подмножеств переменных X_i показали, что при формировании результата в натуральном двоичном коде выбор τ=1, 2 позволяет получать наиболее экономичные схемы.

5. Оценки сложности и быстродействия схем модульного контроля в натуральных двоичных кодах

Оценку сложности схем модульного контроля будем проводить для классического базиса И, ИЛИ, НЕ, а оценку быстродействия – по глубине схемы (максимальному числу последовательно соединенных элементов от входа к выходу).

Сложность и глубина схем числового модульного контроля, синтезируемых рассмотренными методами свертки, составляют

(27) $$L(Q) = L(X) + L(Z),$$

(28) $$H(Q) = H(X) + H(Z),$$

где $L(X)$,$H(X)$- сложность и глубина блока свертки осуществляющего преобразование $X \rightarrow Z$, а $L(Z)$,$H(Z)$- сложность и глубина блока формирования кода остатка, осуществляющего преобразование $Z \rightarrow Q$.

В случае натурального двоичного кода и модуля $K=2^r-1$ блок формирования кода остатка выполняется в виде устройства суммирования по модулю K двух r-разрядных кодов. В настоящее время известен ряд вариантов реализации таких устройств. Одним из простейших является реализация в виде r-разрядного сумматора двоичных чисел с обратной связью с выхода переноса r-го разряда на вход переноса и схемой коррекции, инвертирующей выходной

код в случае его равенства модулю K. Сложность такой реализации блока равна

(29) $$L\ (s_K) = rl_s + 2r\ ,$$

а глубина

(30) $$H\ (s_K) = rh_s^p + h_s^s + 3\ .$$

где h_s^s, h_s^p - глубина полного сумматора по выходам суммы и переноса.

Оценим сложность и быстродействие блоков свертки для рассмотренных в разделе 2 вариантов его реализации. Приводимые оценки глубины схем соответствуют вариантам реализации блока с максимальным быстродействием.

При свертке с использованием сумматоров по модулю два

(31) $$L_1 \le rl_S \sum_{i=1}^{\gamma} \left\lceil n_{i-1}/2 \right\rceil,$$

(32) $$H_1 \le h_S^S \sum_{i=1}^{\gamma} \left\lceil \log_3 n_{i-1} \right\rceil,$$

где $n_0 = \left\rceil n/r \right\lceil$, $n_i = \left\lceil n_{i-1}/2 \right\rceil + 1$, l_s – сложность полного сумматора, h_s^s, h_s^p - глубина полного сумматора по выходам суммы и переноса, S_i, p ,

$$\gamma \le \sum_{i=1}^{\alpha+1} Sgn\left(n_{i-1} - 2\right), \qquad Sgn(x) = \begin{cases} 1 & \text{при } x > 0, \\ 0 & \text{при } x \le 0. \end{cases}$$

При свертке с использованием параллельных счетчиков

(33) $$L_2 \approx r \sum_{i=1}^{\gamma} L\ (\varepsilon_{n_i}) = rl_s \sum_{i=1}^{\gamma} \sum_{j=0}^{\lceil \log_2 m_i \rceil} \left\lceil n_i / 2^{j+1} \right\rceil,$$

(34) $$H_2 \approx \sum_{i=1}^{\gamma} \left(\left\lceil \log_2 n_{i-1} \right\rceil \left(h_S^S + h_S^P\right) - h_S^P \right) - \left(\gamma - 1\right)\left(h_S^S - h_S^P\right) Sgn\ \left(\gamma - 2\right),$$

где $n_0 = \left\rceil n/r \right\lceil$, $n_i = \log_2 \left\lceil n_{i-1} \right\rceil + 1$,

$$\alpha = \left\rceil \log_2 \right\rceil n/r \left\lceil \right[, \gamma \le \sum_{i=1}^{\alpha} Sgn\left(n_{i-1} - 2\right).$$

При свертке с использованием многоразрядных сумматоров

(35) $$L_3 \approx rl_S \sum_{i=1}^{\gamma} \left\lceil n_{i-1}/2r \right\rceil,$$

(36) $$H_3 \approx h_S^P \left(\left[n / r \right] - 1 \right) + \gamma h_S^S ,$$

где $n_0 = n$, $n_i = n_{i-1} - (r-1)\left[n_{i-1}/2r \right] + 1 - \left] n_{i-1}/2r \right[$,

$$\gamma = \sum_{i=1}^{\alpha} Sgn\left(n_{i-1} - 2r \right),\ \alpha = \left] \log_2 [n / r] \right[.$$

При свертке с использованием блоков модульного сложения, выполняемых на r-разрядном сумматоре с обратной связью

(37) $$L_4 \le r l_S \left(\left] n / r \right[- 2 \right),$$

(38) $$H_4 \le \left(r h_S^P + h_S^S \right) \left(\left[\log_2 \right] n / r \left[\right] - 1 \right).$$

В таблице 1 приведены оценки сложности и глубины схем числового модульного контроля, синтезируемых рассмотренными методами свертки, полученные по результатам проектирования.

Анализ полученных оценок а также результатов оптимизации на ЭВМ синтезируемых схем показывает, что сложность рассмотренных вариантов реализации схем числового модульного контроля отличается незначительно и весьма точно определяется оценкой (37). Глубина схем для первых трех вариантов реализации блока свертки в случае древовидных структур при параллельном соединении блоков свертки также отличается незначительно и меньше глубины схем, синтезируемых с использованием свертки блоками модульного сложения, что обусловлено наличием в них обратной связи. При использовании более быстродействующих вариантов реализации блоков модульного сложения, например, предложенных в [28, 29], быстродействие схем модульного контроля может быть существенно повышено. Особо необходимо отметить, что сложность схем числового модульного контроля практически не зависит от значения модуля $K=2^r-1$.

Для случая $K=2^r+1$ сложность и глубина схем формирования остатка для рассмотренных методов свертки C_m^5 с использованием отрицательных весов разрядов кода определяются следующими оценками.

Для варианта конвейерной свертки

(39) $$L_5^3 \le \mu r \left(l_S + 1 \right) + r \left(\left] \mu / 2 \right[- \left\lfloor \mu / 2 \right\rfloor \right) \left(l_S + 1 \right),$$

(40) $$H_5^3 \le 3 h_S^P \left(\mu + \left] \mu / 2 \right[- \left\lfloor \mu / 2 \right\rfloor \right) + h_S^S - h_S^P ,$$

где $\mu =]n/r[$.

Таблица 1

Зависимость сложности схем числового модульного контроля от разрядности входного кода ***n***

n	**Сложность *L* / Глубина *H***															
	***K*=7**								***K*=15**							
	C_n^1		C_n^2		C_n^3		C_n^4		C_n^1		C_n^2		C_n^3		C_n^4	
15	114	234	126	246	114	234	114	234	111	221	111	221	123	223	111	221
	36	21	36	23	33	20	48	30	33	20	33	20	34	21	39	25
21	168	348	168	349	168	348	168	348	173	343	177	347	173	343	173	343
	39	23	39	23	39	24	48	30	42	27	40	24	39	26	57	36
27	222	462	222	462	222	462	222	462	219	449	219	449	219	449	219	449
	42	24	42	25	45	27	63	39	42	25	42	25	49	33	57	36
33	276	576	324	624	276	576	276	576	281	571	273	563	281	571	281	571
	48	27	49	29	48	30	63	39	45	26	48	30	48	30	69	44
45	384	804	384	804	384	804	384	804	389	799	389	799	389	799	389	799
	54	30	54	31	60	37	63	37	57	33	57	33	57	36	105	66

Примечания:

1. В числителе дроби приведены оценки сложности схем, а в знаменателе - их глубины.
2. Оценки приведены для случаев реализации схем на сумматорах минимальной сложности l_s=9, имеющих глубину по выходу суммы $h_s^s = 6$ и по выходу переноса $h_s^p = 3$, и реализации схем на сумматорах минимальной глубины ($h_s^s = 3$, $h_s^p = 2$), имеющих сложность l_s=19.

Для варианта последовательной свертки с использованием полных сумматоров

(41) $$L_5^4 \le (\mu-2)rl_S + (\mu-2)(2r-1_s) - (r-1) + L_K \ ,$$

(42) $$H_5^4 \le (h_S^S+1)(\mu-1) + H_K \ ,$$

где L_K , H_K - сложность и глубина блока сложения по модулю К, определяемые следующими оценками

(43) $$L_K = \begin{cases} r(l_S + l_P) + 2r + 4 & \text{при } \mu = 2[\mu/2], \\ 2(r-1)l_S + (r+4)l_P + 4r & \text{при } \mu \ne 2[\mu/2], \end{cases}$$

(44) $$H_K = \begin{cases} r(h_S^P + h_P^p) + 3 & \text{при } \mu = 2[\mu/2], \\ 4h_S^P + 5h_P^P + 2h_P^S + 2 & \text{при } \mu \ne 2[\mu/2], \end{cases}$$

Для варианта свертки на многоразрядных сумматорах

(45) $$L_5^2 \le l_S \sum_{i=1}^{\gamma} \left(2ra_i + 2rb_i + \left(1 - b_i\right)c_i\right) + L_K \ ,$$

(46) $$H_5^2 \le h_S^p \sum_{i=1}^{\gamma} \left(2ra_i + 2rb_i + \left(1 - b_i\right)c_i\right) + H_K \ ,$$

где $\gamma = \left\rceil \log_2 \right\rceil (n-1) / 2r \left[\right[\ , \qquad \alpha_i = \left[\left(n_i - 1\right) / 4r \right] ,$

$$b_i = \left[\left(n_i - 4ra_i - 1\right) / 2r \right], \quad c_i = n_i - 4r\alpha_i - 1 - 2rb_i \ ,$$

$$n_1 = n \ , \qquad n_{i+1} = 2ra_i + 2rb_i + (1-b)c_i + 1 \ ,$$

L_K , H_K - сложность и глубина блока сложения по модулю K, выполняемого в соответствии с (21), определяемые следующими оценками

(47) $$L_K = \left(r + 1\right) l_S + \left(r - 1\right) l_P + r + 5 \ ,$$

(48) $$H_K = \left(r + 1\right) h_S^p + \left(r - 1\right) h_p^p + 3 \ .$$

Схемы, синтезированные на основе представления (7), имеют большую сложность и глубину, чем у рассмотренных выше вариантов свертки C_m^5.

Сложность и глубина схем числового модульного контроля при $K \neq 2^r \pm 1$ определяются прежде всего характеристиками блока формирования кода остатка, поскольку сложность блоков свертки, синтезируемых с использованием рассмотренных методов свертки, отличается незначительно, в особенности в представляющем практический интерес диапазоне числа переменных n. Получение аналитических оценок сложности и глубины блоков формирования кода остатка для произвольного K в этом случае не представляется возможным.

В таблице 2 приведены оценки сложности и глубины блоков формирования кода остатка, синтезированных рассмотренными выше методами.

Из приведенных оценок видно, что наиболее экономичные схемы блоков формирования остатка для $K=2^r+1$ получаются при синтезе методом свертки, использующим представление разрядов кода Z с отрицательными весами. При $K \neq 2^r \pm 1$ наилучшие характеристики имеют схемы, синтезированные комбинированными методами. Следует отметить, что в большинстве случаев минимум сложности и глубины достигается для различных схемных реализаций блока.

Таблица 2

Оценки сложности блоков формирования остатка входного $p=(\lambda+1)$-разрядного код Z по модулю K

Метод синтеза	K=5		K=9		K=11		K=13		K=17	
Последовательное расширение и свертка	59 23	99 15	134 42	244 31	194 52	374 36	134 42	244 31	134 42	244 31
Прямое формирование вычета	55 13	70 12	146 22	171 17	*	*	*	*	*	*
Комбинированный 1	51 23	86 17	100 35	175 23	186 45	321 28	228 50	458 31	164 47	309 30
Использование отрицательных весов разрядов	38 14	63 11	48 18	88 14	-	-	-	-	66 24	116 19
Комбинированный 2	-	-	-	-	126 48	231 33	144 35	259 26	-	-

Примечания:

1. В числителе дроби приведены оценки сложности, а в знаменателе - глубины схем.
2. Оценки приведены для случаев реализации схем на сумматорах минимальной сложности l_s=9, имеющих глубину по выходу суммы $h_s^s = 6$, а по выходу переноса $h_s^p = 3$, и сумматорах минимальной глубины, имеющих сложность l_s=19, $h_s^s = 3$ и $h_s^p = 2$.
3. «*» - реализация не целесообразна из-за большой сложности схемы.
4. «-» - данный вариант реализации невозможен.
5. Метод «Комбинированный 1» использует методы последовательного расширения и свертки и метод прямого формирования вычета.
6. Метод «Комбинированный 2» использует представление разрядов входного кода с отрицательными весами, метод последовательного расширения и свертки и метод прямого формирования вычета.

6. Заключение

В статье исследованы вопросы синтеза логических схем модульного контроля в натуральных двоичных кодах. Показано,

что большинство из них может быть сведено к обобщенной структуре, состоящей из двух последовательно соединенных блоков. Первый блок осуществляет операцию свертки входного кода, а второй формирует натуральный двоичный код остатка. Рассмотрены возможные варианты реализации обоих блоков. Получены оценки сложности и быстродействия синтезируемых рассмотренными методами схем. Они показывают, что характеристики синтезируемых схем определяются прежде всего сложностью и глубиной используемых при их реализации сумматоров. Использование сумматоров минимальной сложности позволяет получать более экономичные схемы по сравнению со схемами модульного контроля в унитарных параллельных позиционных и непозиционных кодах, причем по мере увеличения модуля K выигрыш в сложности возрастает. Вместе с тем их быстродействие оказывается значительно меньшим.

Приведенные в данной статье результаты формируют достаточно полный теоретический подход к проблеме синтеза логических схем модульного контроля в натуральных двоичных кодах, позволяют определить области их возможного использования, а также точно оценить на ранних стадиях проектирования параметры синтезируемых схем и решить вопрос о целесообразности их применения.

ЛИТЕРАТУРА

1. *Путинцев Н.Д.* Аппаратный контроль управляющих цифровых вычислительных машин. М.: Сов. Радио, 1966.
2. *Гаврилов Ю.В., Пучко А.М.* Арифметические устройства быстродействующих ЭВМ.-М.: Сов. Радио, 1970.
3. *Журавлев Ю.П., Кателюк Л.А., Циклинский Н.И.* Надежность и контроль ЭВМ. М.: Радио и связь, 1978.
4. *Долгов А.И.* Диагностика устройств, функционирующих в системе остаточных классов. М.: Сов. Радио, 1982.
5. *Селлерс Ф.* Методы обнаружения ошибок в работе ЭЦВМ. М.: Мир, 1972.
6. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. М.: Мир, 1976.
7. *Касами Т., Токура Н., Ивадари Ё., Инагаки Я.* Теория кодирования. М.: Мир, 1978.
8. *Дадаев Ю.Г.* Теория арифметических кодов. М.: Радио и связь, 1981.
9. *Музыченко О.Н.* Синтез логических схем модульного контроля в унитарных позиционных двоичных кодах. // Автоматика и телемеханика. 2001. № 3. С. 158 - 173.

10. *Музыченко О.Н.* Устройство для контроля параллельного двоичного кода по модулю К. А.с. 1361557 СССР // Б. И. 1987. № 47. С. 202.
11. *Музыченко О.Н.* Устройство для контроля параллельного двоичного кода по модулю К. А.с. 1425676 СССР // Б. И. 1988. № 35. С. 210.
12. *Музыченко О.Н.* Преобразователь двоичного кода. А.с. 1476614 СССР // Б. И. 1989. № 16. С. 250.
13. *Музыченко О.Н.* Упрощение пороговых схем, синтезируемых методом промежуточного преобразования. // Автоматика и телемеханика. 1990. № 12. С. 164 - 170.
14. *Черкасский Н.В.* Устройство для формирования остатков по модулю. А.с. 1305684 СССР// Б. И. 1987. № 15. С. 250.
15. *Черкасский Н.В., Митьков В.С., Аксарин Л.Л.* Устройство для формирования остатка по модулю три. А.с. 1084799 СССР// Б. И. 1984. № 13. С. 177.
16. *Gajski D.D.* Pat. 4187549 USA Modular modulo 3 module / Filed 17.11.77.
17. *Самойлов А.Л.* Пирамидальная свертка по модулю три. А.с. 1105896 СССР// Б. И. 1984. № 28. С. 145.
18. *Черкасский Н.В.* Устройство для формирования остатков по модулю. А.с. 1736006 СССР// Б. И. 1992. № 19. С. 228-229.
19. *Черкасский Н.В., Титков В.Н.* Устройство для формирования остатка по модулю $m=2^K+1$. А.с. 1339566 СССР// Б. И. 1987. № 35. С. 173-174.
20. *Черкасский Н.В.* Устройство для формирования остатков по модулю. А.с. 1387201 СССР// Б. И. 1988. № 13. С. 259-260.
21. *Черкасский Н.В.* Устройство для формирования остатков по модулю. А.с. 1401610 СССР// Б. И. 1988. № 21. С. 256-257.
22. *Музыченко О.Н.* Устройство для свертки двоичного кода в код по модулю К. А.с. 1425845 СССР // Б. И. 1988. № 35. С. 268.
23. *Музыченко О.Н.* Преобразователь двоичного кода в код по модулю К. А.с. 1429322 СССР // Б. И. 1988. № 37. С. 245.
24. *Музыченко О.Н.* Преобразователь двоичного кода в код по модулю К. А.с. 1732472 СССР // Б. И. 1992. № 17. С. 227.
25. *Черкасский Н.В.* Устройство для формирования остатков по модулю. А.с. 1449986 СССР// Б. И. 1989. № 1. С. 211.
26. *Черкасский Н.В.* Устройство для формирования остатков по модулю. А.с. 1444774 СССР// Б. И. 1988. № 46. С. 220.
27. *Балюк В.В., Выжиковски Р., Каневский Ю.С.* Преобразователь n-разрядного двоичного кода в его представление по модулю М. А.с. 1076899 СССР// Б.И. 1984, № 8. С. 159.
28. *Музыченко О.Н.* Сумматор по модулю семь. А.с. 1603371 СССР // Б. И. 1990. № 40. С. 210.
29. *Музыченко О.Н.* Сумматор по модулю пятнадцать. А.с. 1603375 СССР // Б. И. 1990. № 40. С. 212-215.

УДК 62-507.15

О.Н. Музыченко

Методы синтеза логических схем модульного контроля в унитарных непозиционных двоичных кодах

(*Балтийский государственный технический университет «Военмех» им. Д.Ф. Устинова, Санкт-Петербург*)

Рассмотрены методы синтеза логических схем модульного контроля в унитарном параллельном непозиционном двоичном коде. Получены оценки сложности и быстродействия схем, синтезируемых различными методами.

1. Введение

Устройства модульного контроля параллельных двоичных кодов находят достаточно широкое применение [1-5] в современных цифровых системах управления, передачи и переработки дискретной информации, и, прежде всего, в системах специального назначения, к надежности, достоверности функционирования и контролепригодности которых предъявляются высокие требования. Они применяются в качестве средств аппаратного контроля в системах, использующих контроль по модулю [1,3], арифметические [4,5] и ряд других кодов, в устройствах, работающих в системе остаточных классов [2], в аппаратуре

кодирования и декодирования помехозащищенных кодов в системах передачи информации.

Известны устройства модульного контроля, работающие в унитарных позиционных двоичных кодах [3]. В работе [6] исследованы методы их синтеза, получены оценки сложности и быстродействия схем, синтезируемых различными методами. Достоинством таких устройств является высокое быстродействие, а основным недостатком - большая сложность, что ограничивает возможности их использования, в особенности при значениях модуля $K>5$. В статье разрабатываются и исследуются методы синтеза логических схем модульного контроля, функционирующих в унитарных параллельных непозиционных двоичных кодах, получаются оценки сложности и быстродействия синтезируемых схем, позволяющие оценить целесообразность использования таких устройств при проектировании цифровых систем.

2. Основные понятия и определения

В современных цифровых системах используется как числовой, так и кодовый модульный контроль. В случае числового модульного контроля синтезируемая схема должна формировать остаток параллельного двоичного кода по модулю К с учетом весов его разрядов. При кодовом модульном контроле схема формирует остаток по модулю К числа единичных разрядов контролируемого кода, при этом веса всех его разрядов считаются единичными.

Определение 1. Унитарным параллельным непозиционным кодом остатка по модулю K количества единиц двоичного кода $X=\{x_1,x_2,x_3,...,x_n\}$ будем называть код $G=\{g_n^1 g_n^2 \dots g_n^{K-1}\}$, удовлетворяющий условию:

$$g_n^a(X)=\begin{cases}1 \text{ при } \left(\sum_{i=1}^{n} x_i\right) \bmod K \geq a, \\ 0 \text{ при } \left(\sum_{i=1}^{n} x_i\right) \bmod K < a.\end{cases} \quad (1)$$

Определение 2. Унитарным параллельным непозиционным кодом остатка по модулю K двоичного кода $X=\{x_1,x_2,x_3,...,x_n\}$ будем называть код $U=\{u_n^1 u_n^2 \dots u_n^{K-1}\}$, удовлетворяющий условию:

(2) $$u_n^a(X)=\begin{cases}1 \text{ при } \left(\sum_{i=1}^{n}\omega_i x_i\right)\bmod K \geq a\ , \\ 0 \text{ при } \left(\sum_{i=1}^{n}\omega_i x_i\right)\bmod K < a\ .\end{cases}$$

Унитарный параллельный непозиционный код $G(X)$ может быть представлен композицией пороговых равновесных функций [7, 8] вида:

(3) $$g_n^a(X)=\bigvee_{j=0}^{l} F_n^{a+jK}(X)\overline{F_n^{(j+1)K}(X)}, \qquad a=1,2,3,...,K-1,$$

где $l=](n-a+1)/K[$ -1, а] [- округление в большую сторону до ближайшего целого.

Унитарный параллельный непозиционный код $U(X)$ может быть представлен композицией пороговых неравновесных функций с весами входных переменных, равными весам разрядов входного кода, вида:

(4) $$u_n^a(X)=\bigvee_{j=0}^{l'} F_n^{a+jK}(X)\overline{F_n^{(j+1)K}(X)}, \qquad a=1,2,3,...,K-1,$$

где $l'=\left]\sum_{i=1}^{n}(\omega_i-a+1)/K\right[-1$, а ω_i - вес i-го разряда входного кода X.

Представления (3) и (4) следуют из приведенных выше определений кодов $G(X)$ и $U(X)$, а также свойств пороговых функций [7, 9, 10].

3. Методы синтеза логических схем подсчета количества единиц двоичного кода по модулю К в унитарном параллельном непозиционном коде

Функции $g_n^a(X)$ унитарного параллельного непозиционного кода $G(X)$ обладают свойствами, аналогичными свойствам пороговых равновесных функций [11, 12], в том числе следующим свойством

(5) $$g_n^a(X)g_n^b(X)=g_n^a(X) \quad \text{при} \quad a>b\ .$$

Это позволяет адаптировать разработанные для синтеза пороговых схем методы декомпозиции [11] и факторизации [12, 13] при проектировании схем модульного контроля.

3.1. Декомпозиционный метод

Функции $g_n^a(X)$ могут быть представлены в виде следующей композиции функций разложения

(6)

$$g_n^a(X)=\bigvee_{j=0}^{l}\left(\left\{\bigvee_{\{A_r^j\}} g_{m_1}^{a_1}(X_1)g_{m_2}^{a_2}(X_2)\ldots g_{m_r}^{a_r}(X_r)\right\}\&\overline{\left\{\bigvee_{\{B_r^j\}} g_{m_1}^{b_1}(X_1)g_{m_2}^{b_2}(X_2)\ldots g_{m_r}^{b_r}(X_r)\right\}}\right)=$$

$$\bigvee_{j=0}^{l}\Phi_n^{a+jK}(X)\&\overline{\Phi}_n^{(j+1)K}(X),$$

где $l=\left]\sum_{i=1}^{r}(\eta_i-a+1)/K\right[-1$, а $\eta_i=\min(K-1,m_i)$;

r - параметр разложения;

$g_{m_i}^{a_i}(X_i)$, $g_{m_i}^{b_i}(X_i)$- функции разложения;

$\{A_r^j\},\{B_r^j\}$ - множества всех r-мерных векторов, элементы которых удовлетворяют условиям:

(7) $$\sum_{i=1}^{r}a_i=a+jK\ ,$$

(8) $$\sum_{i=1}^{r}b_i=(j+1)K\ .$$

$A_r=\{a_1,\ldots,a_r\}$ - r - мерный вектор;

a_i - элемент r-мерного вектора A_r, являющийся индексом функции $g_{m_i}^{a_i}(X_i)$;

$B_r=\{b_1,\ldots,b_r\}$ - r - мерный вектор;

b_i - элемент r-мерного вектора B_r, являющийся индексом функции $g_{m_i}^{b_i}(X_i)$;

X_i - множества переменных функций разложения $g_{m_i}^{a_i}(X_i)$ и $g_{m_i}^{b_i}(X_i)$, удовлетворяющие условиям:

(9) $$X_1\cup X_2\cup\ldots\cup X_r=X\ ,$$

(10) $$X_i\cap X_j=\varnothing \quad \text{при} \quad i\neq j.$$

Справедливость разложения (6) вытекает из следующего. В [9] доказано, что любая пороговая равновесная функция алгебры логики может быть представлена композицией функций разложения, зависящих от меньшего числа переменных, вида

(11) $$F_n^a(X) = \bigvee_{\{A_r\}} F_{m_1}^{a_1}(X_1) F_{m_2}^{a_2}(X_2) \ldots F_{m_r}^{a_r}(X_r),$$

где $\{A_r\}$- множество всех r -мерных векторов, удовлетворяющих условию

(12) $$\sum_{i=1}^{r} a_i = a \qquad (a_i \geq 0).$$

С учетом (11) справедливость разложения (6) при $\sigma(X_i) < K$, следует из определения (1) унитарного параллельного непозиционного кода остатка и представления (3), так как в этом случае $g_{m_i}^{a_i}(X_i) = F_{m_i}^{a_i}(X_i)$. При $\sigma(X_i) > K$ разложение (6) также остается справедливым, поскольку свойства системы функций $g_m^a(X_i)$ полностью аналогичны свойствам системы функций $F_m^a(X_i)$. При этом логическая схема, работа которой описывается системой уравнений вида (6) при a=1,2,...,К-1, представляет собой два последовательно соединенных блока, первый из которых выполняет операцию суммирования r унитарных параллельных непозиционных кодов, а второй-свертку кода суммы по модулю K. Возможно эквивалентное (6) представление вида:

(13) $$g_n^a(X) = \bigoplus_{j=0}^{l} \left(\Phi_n^{a+jK}(X) \oplus \overline{\Phi}_n^{(j+1)K}(X) \right).$$

Из изложенного следует, что схема формирования остатка количества единиц двоичного кода X по модулю K может быть синтезирована [14] в порядке, описанном в [12] для случая пороговых равновесных функций, т.е. путем последовательной совместной декомпозиции системы функций $G(X)$, описывающей работу схемы, на основе обобщенного разложения (6) или (13). Декомпозиция осуществляется до получения $\sigma(X_t) < K$, при этом синтезируемый блок представляет собой фундаментальный многопороговый элемент [11], реализующий систему из $\sigma(X_t)$ пороговых равновесных функций с порогами от a_{min}=1 до $a_{max}=\sigma(X_t)$.

Очевидно, что глубина и сложность синтезируемых схем зависят от выбора параметров разложения r и мощности подмножеств переменных X_i на каждом шаге декомпозиции, т.е. от порядка декомпозиции, а вывод о минимальной сложности схем, синтезируемых методом декомпозиции с параметром r=2, сделанный в [11] для случая пороговых равновесных и элементарных симметричных функций, справедлив и в данном случае.

Возможны следующие типы декомпозиции. Будем называть декомпозицию детерминированной, если на всех шагах $r=const$, в противном случае будем называть ее вероятностной. Будем называть декомпозицию регулярной, если на каждом шаге множества переменных систем функций разложения удовлетворяют условию $\left|\sigma\left(X_i\right)-\sigma\left(X_j\right)\right|\le 1$, в противном случае будем называть ее нерегулярной.

Все возможные варианты разложения по переменным функций $g_n^a\left(X\right)$ являются частными случаями представления (6), получаемыми при определенных значениях параметра разложения r и мощности множеств переменных X_i. Так при $r\ge 2$ и $\sigma\left(X_i\right)=\sigma\left(X_2\right)=\ldots=\sigma\left(X_{r-1}\right)=1$ из (6) получаем все виды разложения по переменным, в том числе при r =2 - тривиальное разложение.

Выбор вида и порядка декомпозиции позволяет формировать требуемую структуру схемы, регулировать ее сложность и быстродействие. Оценим сложность синтезируемых схем.

Утверждение 1. Логическая схема формирования остатка количества единиц двоичного кода по модулю К в унитарном параллельном непозиционном коде, синтезированная методом разложения по переменным содержит

$$L\left(G(n)\right)=\begin{cases} n^2-n\,, & при \quad n<K,\\ (K-1)^2-(K-1)+(n-K+1)(3K-2), & при \quad n\ge K, \end{cases} \tag{14}$$

элементов И, ИЛИ, НЕ.

Доказательство. При $n<K$ $L(G(n))$ представляет собой фундаментальный многопороговый элемент , сложность которого определяется доказанной в [11] оценкой вида $L(F(n))=n^2-n$, а при n

$\geq K$ $L(G(n))=L(F(K\text{-}1))+(n\text{-}K+1)l_1=(K\text{-}1)^2\text{-}(K\text{-}1)+(n\text{-}K+1)(3K\text{-}2)$, где l_1 - количество логических элементов, необходимых для реализации уравнения (6) при сформированных функциях разложения. Что и требовалось доказать.

Утверждение 2. Логическая схема формирования остатка количества единиц двоичного кода по модулю К в унитарном параллельном непозиционной коде, синтезированная методом детерминированной декомпозиции с параметром r=2, содержит

$$L(G(n)) = \begin{cases} n^2 - n & при \quad n < K, \\ n^2 & при \quad K \le n \le 2K-2, \\ L\left(G\left(m_1\right)\right) + L'\left(G\left(m_2\right)\right) + 2\left(K^2 - K\right) & при \quad n > 2K-2, \end{cases} \tag{15}$$

элементов И, ИЛИ, НЕ, где m_1, $m_2 \geq K$-1.

Доказательство. При $n<K$ $L(G(n))$ представляет собой фундаментальный многопороговый элемент , сложность которого определяется доказанной в [11] оценкой вида $L(F(n))=n^2\text{-}n$. При $K \leq n \leq 2K$-2 $L(G(n))=L(F(n))+l_2=n^2\text{-}n+n$, где $l_2=(n\text{-}K)+(K\text{-}1)+1=n$ - количество логических элементов, необходимых для реализации уравнения (6) при сформированных функциях разложения. При $n>2K$-2 $L(G(n))=L(G(m_1))+L(G(m_2))+l_3+l_4$, где $l_3=L(F(2K\text{-}2))\text{-}2L(F(K\text{-}1))= 2K^2\text{-}4K+2$ - количество элементов, необходимых для формирования всех конъюнкций функций разложения в (6), а $l_4=2K$-2 - количество элементов, необходимых для реализации уравнения (6) при сформированных конъюнкциях функций разложения. Что и требовалось доказать.

Соответствующие оценкам (14) и (15) графики 1 и 2 представлены на рис. 1 для K=3,5,7 соответственно. На рис. 2 показан пример схемной реализации устройства контроля количества единиц параллельного двоичного кода по модулю K=5, синтезированного методом регулярной детерминированной декомпозиции с параметром r=2.

3.2. Факторизационные методы синтеза

Рассматриваемые функции $g_n^a\left(X\right)$ обладают свойством (5), аналогичным свойству пороговых равновесных функций. Это позволяет адаптировать разработанныå в [12, 13] методы первичной и вторичной факторизации , а также предложенные в [15] однородные и регулярные структуры, для синтеза схем

формирования остатка количества единиц двоичного кода по модулю K.

3.2.1. Первичная факторизация

Уменьшение сложности синтезируемой схемы путем факторизации системы уравнений вида (6) или (13) , полученных в результате декомпозиции, существующими методами [16,17] невозможно из-за отсутствия общих факторов. Однако, с учетом свойства (5) функций $g_n^a(X)$ общие факторы могут быть сформированы искусственно.

Для этого в представлении функций $\Phi_n^a(X)$ из (6) или (13) выделим группу однотипных конъюнкций, для которых все векторы A'_r получаются перестановкой элементов множества $\{\alpha_1,...,\alpha_\rho,\beta_1,...,\beta_l\}$, где $\rho+l\le r,\quad \alpha=const,\quad \alpha<\beta,\quad \beta_{j+1}\ge\beta_j$.

представлена в виде

$$\bigvee_{\{A'_r\}} g_{m_1}^{a_1}(X_1)...g_{m_r}^{a_r}(X_r)=\Big\{g_{m_1}^{\alpha}(X_1)...g_{m_\rho}^{\alpha}(X_\rho)g_{m_{\rho+1}}^{\beta_1}(X_{\rho+1})...g_{m_{\rho+l}}^{\beta_l}(X_{\rho+l})\vee$$

$$...\vee g_{m_1}^{\alpha}(X_1)...g_{m_\rho}^{\alpha}(X_\rho)g_{m_{\rho+1}}^{\beta_l}(X_{\rho+1})...g_{m_{\rho+l}}^{\beta_1}(X_{\rho+l})\vee.....\vee g_{m_{r-\rho-l+1}}^{\beta_l}(X_{r-\rho-l+1})\cdot$$

$$...g_{m_{r-\rho}}^{\beta_1}(X_{r-\rho})g_{m_{r-\rho+1}}^{\alpha}(X_{r-\rho+1})...g_{m_r}^{\alpha}(X_r)\Big\}=\Big\{g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)...g_{m_{\rho+l}}^{\alpha}(X_{\rho+l})\vee... \quad (16)$$

$$\vee g_{m_{r-\rho-l+1}}^{\alpha}(X_{r-\rho-l+1})...g_{m_{r-1}}^{\alpha}(X_{r-1})g_{m_r}^{\alpha}(X_r)\Big\}\&\Big\{g_{m_{\rho+1}}^{\beta_1}(X_{\rho+1})g_{m_{\rho+2}}^{\beta_2}(X_{\rho+2})..$$

$$...g_{m_{\rho+l}}^{\beta_l}(X_{\rho+l})\vee.....\vee g_{m_{\rho+1}}^{\beta_l}(X_{\rho+1})...g_{m_{\rho+l-1+}}^{\beta_2}(X_{\rho+l-1})g_{m_{\rho+l}}^{\beta_1}(X_{\rho+l})\vee...$$

$$..\vee g_{m_{r-\rho-l+1}}^{\beta_l}(X_{r-\rho-l+})...g_{m_{r-\rho-1}}^{\beta_2}(X_{r-\rho-1})g_{m_{r-\rho}}^{\beta_1}(X_{r-\rho})\Big\}$$

Очевидно, что использование преобразования (16) обеспечивает уменьшение сложности реализации функций $\Phi_n^a(X)$ и, следовательно, функций $g_n^a(X)$, поскольку разность числа логических операций И,ИЛИ в его левой и правой частях составляет

$$\Delta = C_r^l(r-l)-r,$$

где l - число функций разложения с порогом β в каждой конъюнкции.

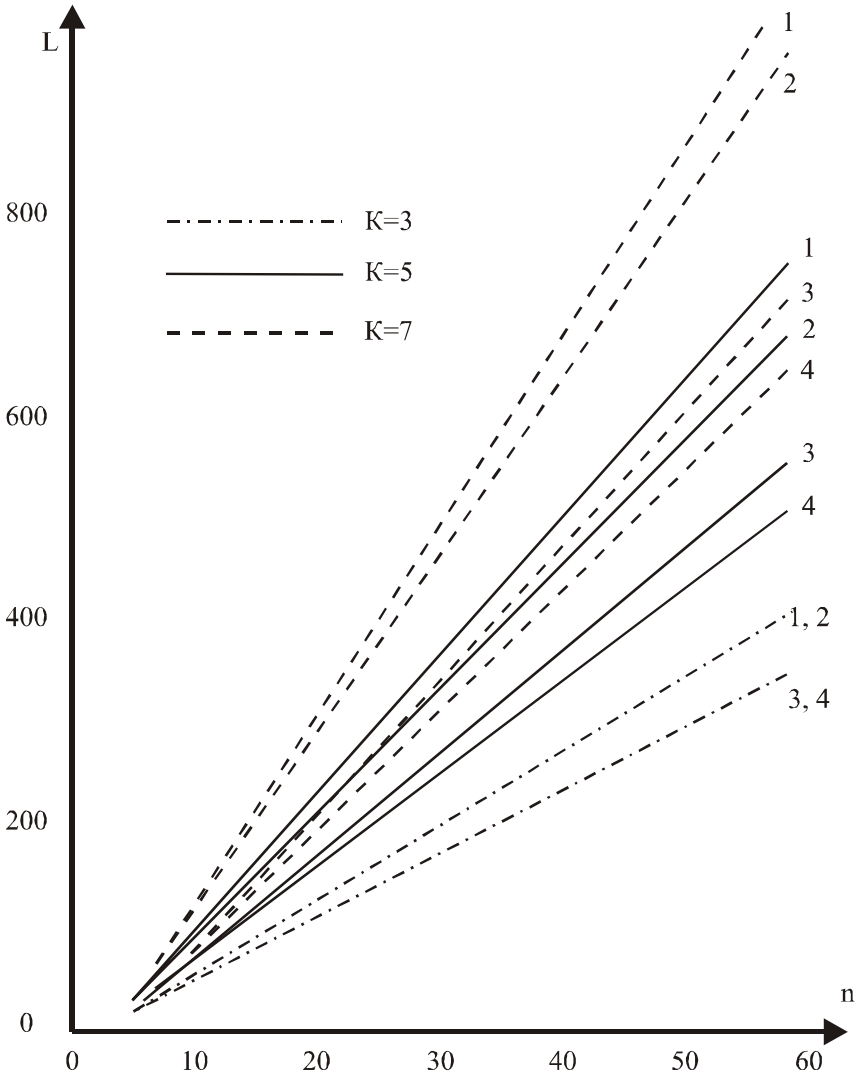


Рис. 1. Зависимость сложности схем подсчета количества единиц параллельного двоичного кода по модулю K от его разрядности n

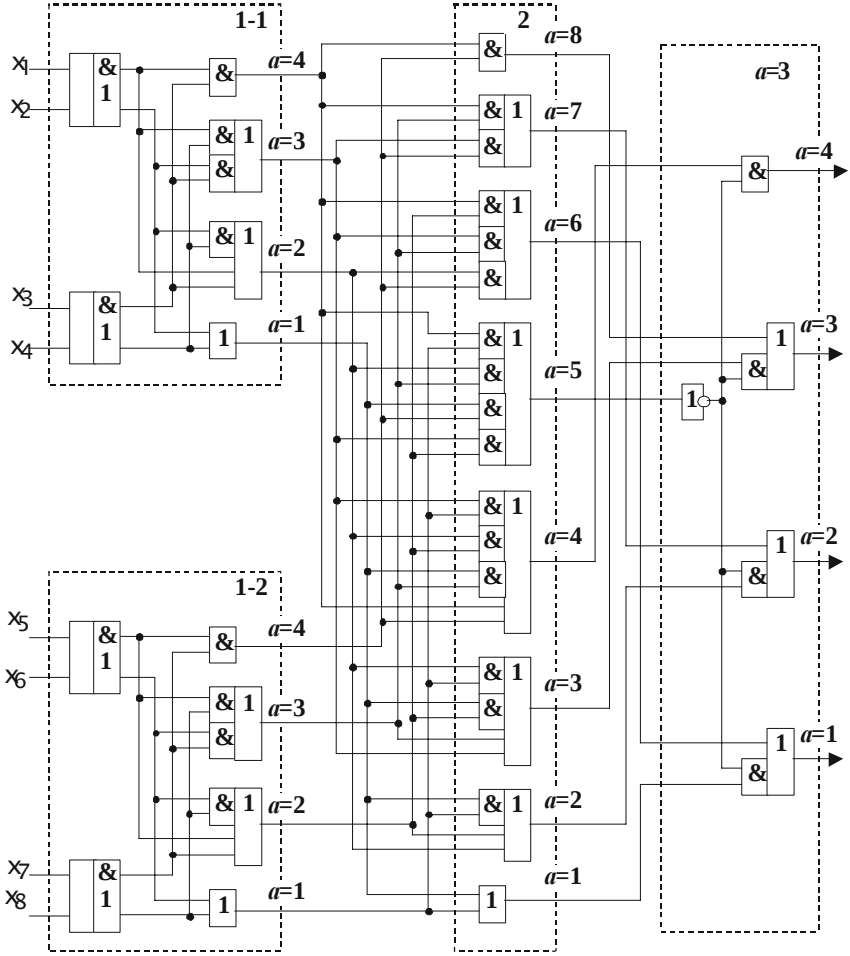


Рис. 2. Устройство подсчета количества единиц 8-разрядного двоичного кода по модулю K=5 в унитарном параллельном непозиционном коде, синтезированное методом регулярной детерминированной декомпозиции (r=2)

Данная группа конъюнкций с учетом (5) может быть (t-1) - кратное выполнение преобразования (16), где t - количество различных значений индексов a_j векторов A'_r, приводит к представлению

$$(17)\quad \bigvee_{\{A'_r\}} g_{m_1}^{a_1}(X_1) g_{m_2}^{a_2}(X_2) g_{m_r}^{a_r}(X_r) = \&_{j=1}^{t} F_r^{p_j}\left(g_{m_1}^{\alpha_j}(X_1), g_{m_2}^{\alpha_j}(X_2), ..., g_{m_r}^{\alpha_j}(X_r)\right)$$

где $\sum_{j=1}^{t}\tilde{p}_j\alpha_j=\sum_{i=1}^{r}a_i=a$, $\tilde{p}_j=p_j-p_{j+1},\quad p_{t+1}=0$,

а $F_r^{p_j}(Y)$ - пороговая равновесная функция r переменных с порогом p_j.

Представления (17) имеют для различных $\Phi_n^a(X)$ общие факторы вида $F_r^{p_j}(Y)$, а при $r>2$ обеспечивают уменьшение сложности реализации отдельной функции $\Phi_n^a(X)$.

При $r=2$ преобразование (17) имеет вид

$$(18)\quad g_{m_1}^{\alpha}(X_1)g_{m_2}^{\beta}(X_2)\vee g_{m_1}^{\beta}(X_1)g_{m_2}^{\alpha}(X_2)=g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)\left\{g_{m_1}^{\beta}(X_1)\vee g_{m_2}^{\beta}(X_2)\right\}.$$

При $r=3$ получаем преобразования следующих четырех видов:

$$(19)\quad \begin{aligned} &g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)g_{m_{13}}^{\beta}(X_3)\vee g_{m_1}^{\alpha}(X_1)g_{m_2}^{\beta}(X_2)g_{m_{13}}^{\alpha}(X_3)\vee g_{m_1}^{\beta}(X_1)g_{m_2}^{\alpha}(X_2)\\ &g_{m_{13}}^{\alpha}(X_3)=g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)g_{m_3}^{\alpha}(X_3)\left\{g_{m_1}^{\beta}(X_1)\vee g_{m_2}^{\beta}(X_2)\vee g_{m_3}^{\beta}(X_3)\right\},\\ &g_{m_1}^{\alpha}(X_1)g_{m_2}^{\beta}(X_2)g_{m_{13}}^{\beta}(X_3)\vee g_{m_1}^{\beta}(X_1)g_{m_2}^{\alpha}(X_2)g_{m_{13}}^{\beta}(X_3)\vee g_{m_1}^{\beta}(X_1)g_{m_2}^{\beta}(X_2) \end{aligned}$$

$$(20)\quad \begin{aligned} &g_{m_{13}}^{\alpha}(X_3)=g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)g_{m_3}^{\alpha}(X_3)\left\{g_{m_1}^{\beta}(X_1)g_{m_2}^{\beta}(X_2)\vee g_{m_1}^{\beta}(X_1)g_{m_3}^{\beta}(X_3)\vee\right.\\ &\left.\vee\, g_{m_2}^{\beta}(X_2)g_{m_3}^{\beta}(X_3)\right\},\\ &g_{m_1}^{\alpha}(X_1)g_{m_2}^{\beta}(X_2)\vee g_{m_1}^{\alpha}(X_1)g_{m_3}^{\beta}(X_3)\vee g_{m_2}^{\alpha}(X_2)g_{m_{13}}^{\beta}(X_3)\vee g_{m_1}^{\beta}(X_1)g_{m_2}^{\alpha}(X_2) \end{aligned}$$

$$(21)\quad \begin{aligned} &\vee\, g_{m_1}^{\beta}(X_1)g_{m_{13}}^{\alpha}(X_3)\vee g_{m_2}^{\beta}(X_2)g_{m_{13}}^{\alpha}(X_3)=\left\{g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)\vee\right.\\ &\left.\vee\, g_{m_1}^{\alpha}(X_1)g_{m_3}^{\alpha}(X_3)\vee g_{m_2}^{\alpha}(X_2)g_{m_3}^{\alpha}(X_3)\right\}\left\{g_{m_1}^{\beta}(X_1)\vee g_{m_2}^{\beta}(X_2)\vee g_{m_3}^{\beta}(X_3)\right\}, \end{aligned}$$

$$(22)\quad \begin{aligned} &g_{m_1}^{\alpha}(X_1)g_{m_2}^{\beta}(X_2)g_{m_{13}}^{\gamma}(X_3)\vee g_{m_1}^{\alpha}(X_1)g_{m_2}^{\gamma}(X_2)g_{m_{13}}^{\beta}(X_3)\vee g_{m_1}^{\beta}(X_1)g_{m_2}^{\alpha}(X_2)g_{m_{13}}^{\gamma}(X_3)\vee\\ &g_{m_1}^{\gamma}(X_1)g_{m_2}^{\alpha}(X_2)g_{m_{13}}^{\beta}(X_3)\vee g_{m_1}^{\beta}(X_1)g_{m_2}^{\gamma}(X_2)g_{m_{13}}^{\alpha}(X_3)\vee g_{m_1}^{\gamma}(X_1)g_{m_2}^{\beta}(X_2)g_{m_{13}}^{\alpha}(X_3)=\\ &g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)g_{m_3}^{\alpha}(X_3)\left\{g_{m_1}^{\beta}(X_1)g_{m_2}^{\beta}(X_2)\vee g_{m_1}^{\beta}(X_1)g_{m_3}^{\beta}(X_3)\vee g_{m_2}^{\beta}(X_2)g_{m_3}^{\beta}(X_3)\right\}\\ &\left\{g_{m_1}^{\gamma}(X_1)\vee g_{m_2}^{\gamma}(X_2)\vee g_{m_3}^{\gamma}(X_3)\right\}\qquad(\alpha<\beta<\gamma). \end{aligned}$$

Аналогичные представления могут быть получены для любого r.

С учетом (17) функции $\Phi_n^a(X)$ из (6) и (13) могут быть представлены в виде

(23) $$\Phi_n^a(X)=\bigvee_{\{A_r^j\}} g_{m_1}^{a_1}(X_1)g_{m_2}^{a_2}(X_2)\ldots g_{m_r}^{a_r}(X_r)=$$

$$\bigvee_{\{A_r^j\}}\ \&_{j=1}^{t} F_r^{p_j}\left(g_{m_1}^{\alpha_j}(X_1), g_{m_2}^{\alpha_j}(X_2),\ldots, g_{m_r}^{\alpha_j}(X_r)\right)$$

где $\sum_{j=1}^{t}\tilde{p}_j\alpha_j=\sum_{i=1}^{r}a_i=a,\ \tilde{p}_j=p_j-p_{j+1},\ p_{t+1}=0,\ p_{j+1}<p_j, a_{j+1}>a_j,$

$F_r^{p_j}(Y)$ - пороговая равновесная функция *r* переменных с порогом p_j;;

$\{A_r^j\}'$ - множества типов r-мерных векторов $A_r^j=\{a_1,\ldots,a_r\}$, элементы которых удовлетворяют условию (7);

$A_r^j=\{a_1,\ldots,a_r\}$ - *r* - мерный вектор;

a_i - элемент *r*-мерного вектора A_r^j, являющийся индексом функции $g_{m_i}^{a_i}(X_i)$;

t - число ненулевых элементов вектора A_r^j ($t\le r$).

Множество $\{A_r^j\}$ может быть получено из $\{A_r^j\}'$ путем замены каждого из векторов $A_r^j\in\{A_r^j\}'$ множеством векторов, получаемых из него перестановкой элементов, т.е. $\{A_r^j\}'\subset\{A_r^j\}$.

При *r* =2 из (23) получаем следующее представление функции $\Phi_n^a(X)$:

(24) $$\Phi_n^a(X)=\bigvee_{\{A_2\}'} g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)\left\{g_{m_1}^{a-\alpha}(X_1)\vee g_{m_2}^{a-\alpha}(X_2)\right\}=$$

$$=\bigvee_{\{i\}} F_2^2\left(g_{m_1}^{i}(X_1), g_{m_{12}}^{i}(X_2)\right)F_2^1\left(g_{m_1}^{a-i}(X_1), g_{m_{12}}^{a-i}(X_2)\right)\vee f_{\Delta},$$

где : $\{A_2\}'$ - множество типов двумерных векторов A_2 ;

$i\in\{1,2,\ldots,\mu\}$;

$\mu=\min\{]a/2[-1, K-1,]n/2[-1\}$;

$$(25)\quad f_{\Delta}=\begin{cases} F_2^2\left(g_{m_1}^{a/2}(X_1),g_{m_2}^{a/2}(X_2)\right)\vee F_2^1\left(g_{m_1}^{a}(X_1),g_{m_2}^{a}(X_2)\right) & \\ \qquad\qquad\qquad\qquad \text{при} \quad a\le m_1,m_2 \quad \text{и} \quad a=2]a/2[, & \\ F_2^2\left(g_{m_1}^{a/2}(X_1),g_{m_2}^{a/2}(X_2)\right) \text{ при} \quad a>m_1,m_2 \quad \text{и} \quad a=2]a/2[, & \\ F_2^1\left(g_{m_1}^{a}(X_1),g_{m_2}^{a}(X_2)\right) \text{ при} \quad a\le m_1,m_2 \quad \text{и} \quad a\ne 2]a/2[, & \\ 0 \qquad\qquad\qquad \text{при} \quad a>m_1,m_2 \quad \text{и} \quad a\ne 2]a/2[. & \end{cases}$$

При μ=0 в представлении (24) $\Phi_n^a(X)=f_{\Delta}$.

Наибольшее уменьшение сложности схем достигается при факторизации логических уравнений, полученных в результате регулярной декомпозиции. В частном случае разложения по переменным факторизация, обеспечивающая уменьшение сложности схем, невозможна.

Использование первичной факторизации уравнений (6) приводит к получению системы логических уравнений вида:

$$(26)\quad g_n^a(X)=\bigvee_{j=0}^{l}\left(\left\{\bigvee_{\{A_i^j\}'}\left(\&_{\rho=1}^{t}F_r^{p_\rho}\left(g_{m_{11}}^{\alpha_\rho}(X_1)g_{m_2}^{\alpha_\rho}(X_2)\dots g_{m_r}^{\alpha_\rho}(X_1)\right)\right)\right\}\&\right.$$

$$\left.\overline{\left\{\bigvee_{\{B_i^j\}'}\left(\&_{\rho=1}^{t}F_r^{p_\rho}\left(g_{m_{11}}^{\beta_\rho}(X_1)g_{m_2}^{\beta_\rho}(X_2)\dots g_{m_r}^{\beta_\rho}(X_1)\right)\right)\right\}}\right)$$

$$=\bigvee_{j=0}^{l}\Phi_n^{a+jK}(X)\overline{\Phi}_n^{(j+1)K}(X)$$

или эквивалентной ей системы уравнений вида (13),

где $t\le r$, $p_{j+1}<p_j$, $\alpha_\rho\ge 0$, $\alpha_{\rho+1}>\alpha_\rho$, $\beta_\rho\ge 0$, $\beta_{\rho+1}>\beta_\rho$,

$\sum_{\rho=1}^{t}\tilde{p}_\rho\alpha_\rho=\sum_{i=1}^{r}a_i=a+jK;$

$\sum_{\rho=1}^{t}\tilde{p}_\rho\beta_\rho=\sum_{i=1}^{r}b_i=(1+j)K;\ \ \tilde{p}_\rho=p_\rho-p_{\rho+1},\quad p_{t+1}=0$,

$\{A_r^j\}'$, $\{B_r^j\}'$ множества типов r-мерных векторов, удовлетворяющих условиям (7) и (8) соответственно.

При r=2 уравнение (26) имеет вид:

$$g_n^a(X)=\left(\bigvee_{\{A_r^0\}}\left\{g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)\right\}\left\{g_{m_1}^{a-\alpha}(X_1)\vee g_{m_2}^{a-\alpha}(X_2)\right\}\right)\&$$

(27)
$$\overline{\left(\bigvee_{\{B_i^0\}}\left\{g_{m_1}^{\beta}(X_1)g_{m_2}^{\beta}(X_2)\right\}\left\{g_{m_1}^{K-\beta}(X_1)\vee g_{m_2}^{K-\beta}(X_2)\right\}\right)}\vee$$
$$\left(\bigvee_{\{A_r^1\}}\left\{g_{m_1}^{\alpha}(X_1)g_{m_2}^{\alpha}(X_2)\right\}\left\{g_{m_1}^{a+K-\alpha}(X_1)\vee g_{m_2}^{a+K-\alpha}(X_2)\right\}\right)=$$
$$=\left(\bigvee_{\{i\}_a}F_2^2\left\{g_{m_1}^{i}(X_1)g_{m_2}^{i}(X_2)\right\}F_2^1\left\{g_{m_1}^{a-i}(X_1)g_{m_2}^{a-i}(X_2)\right\}\right)\&$$
$$\overline{\left(\bigvee_{\{i\}_K}F_2^2\left\{g_{m_1}^{i}(X_1)g_{m_2}^{i}(X_2)\right\}F_2^1\left\{g_{m_1}^{K-i}(X_1)g_{m_2}^{K-i}(X_2)\right\}\right)}\vee$$
$$\left(\bigvee_{\{i\}_{a+K}}F_2^2\left\{g_{m_1}^{i}(X_1)g_{m_2}^{i}(X_2)\right\}F_2^1\left\{g_{m_1}^{a+k-i}(X_1)g_{m_2}^{a+k-i}(X_2)\right\}\right)=$$
$$\Phi_n^a(X)\&\ \overline{\Phi}_n^K(X)\vee\ \Phi_n^{a+K}(X),$$

где:

$\{i\}_a \in \{0,1,2,\ \dots\ ,\ \min([n/2],[a/2])\}$;

$\{i\}_K \in \{1,2,\ \dots\ ,\ \min([n/2],[K/2])\}$;

$\{i\}_{a+K} \in \{a+1,\dots,\min([(a+K)/2],[n/2])\}$.

Схема устройства подсчета единиц параллельного двоичного 24-х разрядного кода по модулю K=5, синтезированная методом первичной факторизации (r=3), представлена на рис. 3.

Оценим сложность схем, синтезируемых рассмотренным методом первичной факторизации.

Утверждение 4. Логическая схема формирования остатка количества единиц двоичного кода по модулю K в унитарном параллельном непозиционном коде, синтезированная методом факторизации (r=2) содержит

(28)
$$L\left(G(n)\right)=\begin{cases}L(F(n)) & при\quad n<K,\\ L(F(n))+n & при\quad K\le n\le 2K-2,\\ L(G(m_1))+L(G(m_2))+L(F(2K-2))-2L(F(K-1)+2K-2 & при\quad m_1,m_2>K-1\end{cases}$$

элементов И, ИЛИ, НЕ, где $L(F(n))$ - сложность реализации фундаментального многопорогового элемента $F(n)$, определенная

для данного метода факторизации (соответствующие оценки приведены в [12,13]) .

Доказательство. Случай $n<K$ очевиден, поскольку синтезируемая схема представляет собой фундаментальный многопороговый логический элемент. При $K \le n \le 2K$-2 сложность синтезируемой схемы составляет $L(G(n))=L(F(n))+l_1$, где $l_1=n$ - количество элементов, необходимых для реализации системы уравнений (27) при сформированных функциях $\Phi_n^a(X)$. При $n>2K$-2 сложность синтезируемой схемы составляет $L(G(n))=L(G(m_1))+L(G(m_2))+l_1+l_2$, где $l_1=L(F(2K\text{-}2))\text{-}2L(F(K\text{-}1))$ - количество элементов, необходимых для реализации блока сложения унитарных двоичных непозиционных кодов, в случае синтеза используемым методом факторизации системы уравнений функций $\Phi_n^a(X)$ при сформированных функциях разложения $g_{m_i}^{\alpha_i}(X_i)$, а $l_2=2K$-2 - количество элементов, необходимых для реализации системы уравнений (27) при сформированных функциях $\Phi_n^a(X)$.

Следствие 1. Логическая схема формирования остатка количества единиц двоичного кода по модулю K в унитарном параллельном непозиционном коде, синтезированная методом первичной факторизации (r=2) содержит:

(29) $$L(G(n)) = L(G(m_1)) + L(G(m_2)) + \Delta L_1$$

элементов И, ИЛИ, НЕ, где m_1, $m_2>K$-1 или $m_1=]n/2[$,

(30) $$\Delta L_1 = \begin{cases} \frac{n}{2}\left(\frac{n}{2}+1\right) & при \quad n=2]n/2[\quad и \quad n<K, \\ \frac{(n+1)}{2}\left(\frac{(n+1)}{2}+1\right)-2 & при \quad n\neq 2]n/2[\quad и \quad n<K, \\ \frac{n}{2}\left(\frac{n}{2}+1\right)+n & при \quad n=2]n/2[\quad и \quad K\le n\le 2K-2, \\ \frac{(n+1)}{2}\left(\frac{(n+1)}{2}+1\right)+n-2 & при \quad n\neq 2]n/2[\quad и \quad K\le n\le 2K-2, \\ K^2+K-2 & при \quad n>2K-2.. \end{cases}$$

Оценка (30) получается из (28) с учетом представления (27).

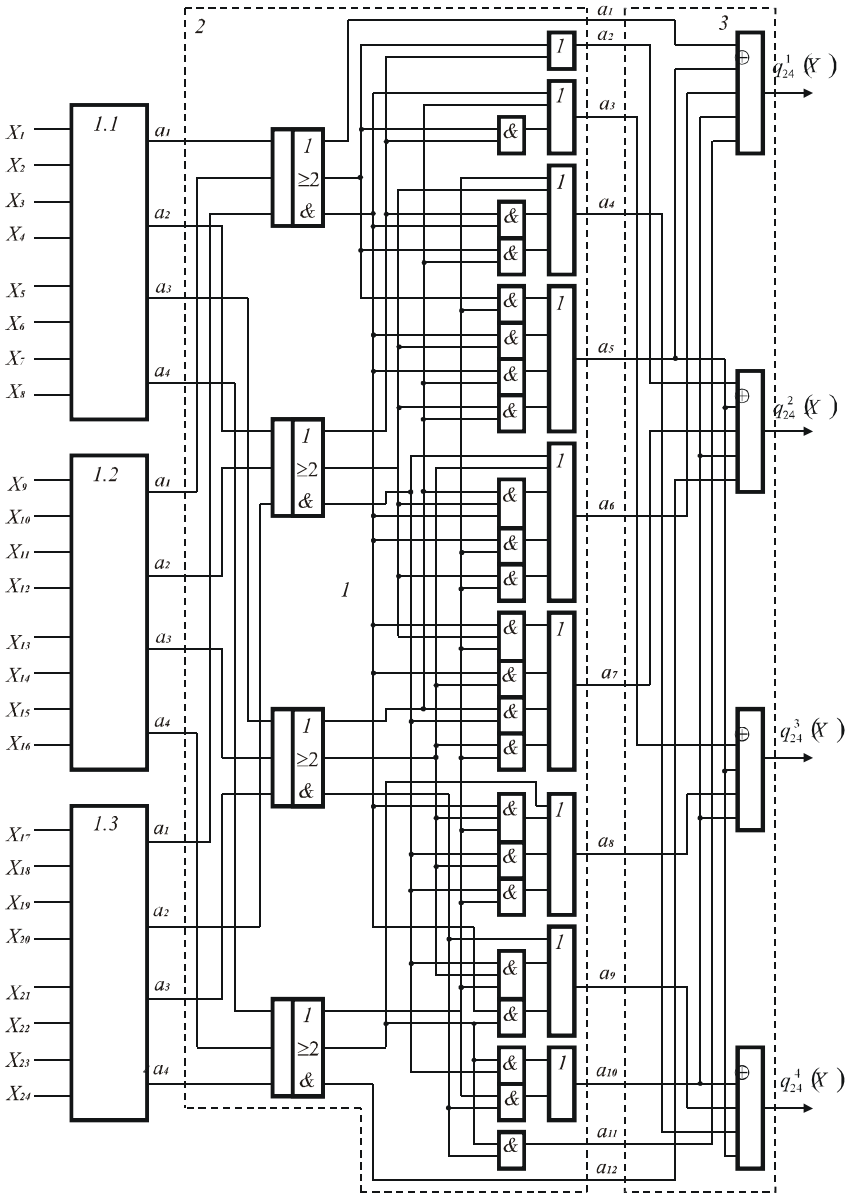


Рис. 3. Схема подсчета количества единиц 24-разрядного двоичного кода по модулю K=5 в унитарном непозиционном коде, синтезированная методом первичной факторизации (r=3)

3.2.2. Вторичная факторизация

Дальнейшее уменьшение сложности схем модульного контроля возможно [18] за счет совместной реализации систем функций $\Phi_n^a(X)$. Для этого представим функцию $\Phi_n^a(X)$ в виде

$$\Phi_n^a(X) = \bigvee_{i=1}^{\tau+1} D_i = F_\tau^1(D_1, D_2, ..., D_\tau) \vee D_{\tau+1}. \quad (31)$$

В этом случае система функций $\Phi_n^a(X)$ при $a = \{a, a+1, ..., a+\tau-1\}$ может быть реализована в виде

$$\begin{cases} \Phi_n^a(X) = F_\tau^1(D_1, D_2, ..., D_\tau) \vee D_{\tau+1} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \Phi_n^{a+i}(X) = F_\tau^{i+1}(D_1, D_2, ..., D_\tau) \vee \Delta F_i \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \Phi_n^{a+\tau-1}(X) = F_\tau^{\tau}(D_1, D_2, ..., D_\tau) \vee \Delta F_{\tau-1} \end{cases} \quad (32)$$

где $\Delta F_i = \Phi_n^{a+i}(X) \& \overline{F}_\tau^{i+1}(D_1, ..., D_\tau)$.

Проблема заключается в определении оптимального состава дизъюнкций $D_1, ..., D_r$, обеспечивающих представление функций системы $\{\Phi_n^{\epsilon}(X)\}$ наименьшей сложности.

Результаты исследований показывают, что при r=2 для любых n и K, а при r=3 по крайней мере для K<17, могут быть сформированы представления (31), для которых $D_{\tau+1} = 0$ при любых a, τ. В случае r=2 оптимальным является представление уравнения $\Phi_n^a(X)$ системы (32) вида

$$\Phi_n^a(X) = \bigvee_{i=1}^{\tau} \left\{ \bigvee_{j=j_{\min}}^{\xi} g_{m_1}^{\tau j+i-1}(X_1) g_{m_2}^{\tau j+i-1}(X_2) \left(g_{m_1}^{a-\tau j-i+1}(X_1) \vee g_{m_1}^{a-\tau j-i+1}(X_1) \right) \right\}, \quad (33)$$

где $\xi = \left[\dfrac{a-2i+2}{2\tau} \right]$, $\mu = \min\{m_1, K-1\}$, $m_1 =]n/2[$, а

$$j_{\min} = \begin{cases} [(a-\mu)/\tau] & \text{при} \quad \mu < a, \\ 0 & \text{при} \quad \mu \geq a. \end{cases}$$

При этом в (32) $D_{\tau+1} = 0$, а $\Delta F_i = \Delta F_i^1 \vee \Delta F_i^2$, где

(34) $$\Delta F_i^1 = \begin{cases} 0 & \text{при } a+i > \mu, \\ \bigvee\limits_{j=1}^{\rho_1} g_{m_1}^{i-j}(X_1) g_{m_2}^{i-j}(X_2) \left\{ g_{m_1}^{a+j}(X_1) \vee g_{m_2}^{a+j}(X_2) \right\} & \text{при } a+i \le \mu, \end{cases}$$

$$\rho_1 = \min\{i, \mu - a\},$$

(35) $$\Delta F_i^2 = \begin{cases} 0 & \text{при } \rho_2 > i \text{ или } \tau = 2 \text{ и } a = 2]a/2[, \\ \bigvee\limits_{j=\rho_2}^{i} g_{m_1}^{b}(X_1) g_{m_2}^{b}(X_2) \left\{ g_{m_1}^{a+i-b}(X_1) \vee g_{m_2}^{a+i-b}(X_2) \right\} & \text{при } \tau > 2 \text{ или } a \ne 2]a/2[, \end{cases}$$

$$b = \left[\frac{a+j}{2} \right], \qquad \rho_2 = \begin{cases} 1 & \text{при } a \ne 2]a/2[, \\ 2 & \text{при } a = 2]a/2[. \end{cases}$$

Возможен другой вариант вторичной факторизации систем функций $\Phi_n^a(X)$, использующий их представление вида

(36) $$\begin{cases} \Phi_n^a(X) = F_\tau^1(D_1 \vee \Delta D_1, D_2 \vee \Delta D_2, \ldots, D_\tau \vee \Delta D_\tau) \vee D_{\tau+1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Phi_n^{a+i}(X) = F_\tau^{i+1}(D_1 \vee \Delta D_1, D_2 \vee \Delta D_2, \ldots, D_\tau \vee \Delta D_\tau) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \Phi_n^{a+\tau-1}(X) = F_\tau^{\tau}(D_1 \vee \Delta D_1, D_2 \vee \Delta D_2, \ldots, D_\tau \vee \Delta D_\tau) \end{cases}$$

где $$\Delta D_j = \bigvee_{i=1}^{\tau-1} \Delta d_j^i \quad (j = 1,2,\ldots,\tau),$$

(37) $$\bigvee_{j=1}^{\tau} \Delta d_j^i = \Phi_n^{a+i}(X) \,\&\, \overline{F}_\tau^{i+1}\left(D_1 \vee \left(\bigvee_{e=1}^{i-1} \Delta d_1^e \right), D_2 \vee \left(\bigvee_{e=1}^{i-1} \Delta d_2^e \right), \ldots, D_\tau \vee \left(\bigvee_{e=1}^{i-1} \Delta d_\tau^e \right) \right).$$

Очевидно, что

$$\left(\bigvee_{i=1}^{\tau-1} \Delta F_i \right) \& \left(\bigvee_{i=1}^{\tau} \Delta D_i \right) = \left(\bigvee_{i=1}^{\tau} \Delta D_i \right)$$

и реализация на основе системы уравнений (36) не сложнее, чем на основе системы уравнений (32).

Результаты исследований показывают, что при r=2 для любых n могут быть сформированы $D_i, \Delta D_i$ такие, что в (36) $D_{\tau+1} = 0$, а

(38) $$\bigvee_{i=1}^{\tau-1} \Delta D_i = \left\{ \bigvee_{j=1}^{\tau-1} \left\lfloor g_{m_1}^{a+j}(X_1) \vee g_{m_2}^{a+j}(X_2) \right\rfloor \right\} \vee \left\{ \bigvee_{j=2-(a) \bmod 2}^{\tau-1} g_{m_1}^{b}(X_1) g_{m_{12}}^{b}(X_2) \right\},$$

где $b = [(a+j)/2]$.

Оптимальным является представление

(39) $$D_i \vee \Delta D_i = \bigvee_{j=j_{\min}}^{\xi} g_{m_1}^{\beta}(X_1) g_{m_{12}}^{\beta}(X_2) \left\{ g_{m_1}^{a-\beta}(X_1) \vee g_{m_2}^{a-\beta}(X_2) \right\} \vee \left(\Delta D_i^1 \vee \Delta D_i^2 \right),$$

где $\beta = \tau j + i - 1$, $\xi = \left[(a - 2i + 2)/2\tau \right]$,

$$\vartheta = a + \tau - i + 1, \varpi = \tau \left[(a - 2i + 2)/2\tau \right] + \tau + i - 1,$$

$$j_{\min} = \begin{cases} \left[(a - \mu) / \tau \right] & \text{при} \quad \mu < a, \\ 0 & \text{при} \quad \mu \geq a, \end{cases}$$

(40) $$\Delta D_i^1 = \begin{cases} 0 & \text{при} \quad i = 1 \quad \text{или} \quad a + \tau - i + 1 > \mu, \\ g_{m_1}^{\vartheta}(X_1) \vee g_{m_2}^{\vartheta}(X_2) & \text{при} \quad i > 1 \quad \text{и} \quad a + \tau - i + 1 \leq \mu, \end{cases}$$

(41) $$\Delta D_i^2 = \begin{cases} 0 & \text{при} \quad a + \tau \leq 2\varpi, \\ g_{m_1}^{\varpi}(X_1) g_{m_2}^{\varpi}(X_2) & \text{при} \quad a + \tau > 2\varpi, \end{cases}$$

причем $D_{\tau+1} = 0$. В этом случае система уравнений (36) имеет минимальную сложность реализации. При r=2 сложность схем, синтезированных на основе представлений (32) и (36), одинакова, однако последнее из них приводит к схемам меньшей глубины. При r>2 представление (36) обеспечивает получение более экономичных схем.

Следствие 2. Логическая схема формирования остатка количества единиц двоичного кода по модулю K в унитарном параллельном непозиционном коде, синтезированная методом вторичной факторизации ($r = \tau = 2$) содержит:

(42) $$L(G(n)) = L(G(m_1)) + L(G(m_2)) + \Delta L_2,$$

элементов И, ИЛИ, НЕ, где $m_1, m_2 > K$-1 или $m_1 =]n/2[$,

(43) $$\Delta L_2 = \begin{cases} \left[(n^2 + 4)/8 \right] + 3n/2 - 2 & \text{при} \quad n = 2]n/2[\quad \text{и} \quad n < K, \\ \left[(n^2 + 1)/8 + 7n/4 \right] - 2 & \text{при} \quad n = 2]n/2[\quad \text{и} \quad n < K, \\ \left[(n^2 + 4)/8 \right] + 5n/2 - 2 & \text{при} \quad n = 2]n/2[\quad \text{и} \quad K \leq n \leq 2K - 2, \\ \left[(n^2 + 1)/8 + 7n/4 \right] + n - 2 & \text{при} \quad n = 2]n/2[\quad \text{и} \quad K \leq n \leq 2K - 2, \\ \left[((K-1)^2 + 1)/2 \right] + 5(K-1) - 2 & \text{при} \quad n > 2K - 2. \end{cases}$$

Использование вторичной факторизации с параметром $\tau > 2$ позволяет в ряде случаев при $K \geq 7$ уменьшить сложность схемы по сравнению с определяемой оценкой (42) .

Следует отметить, что при $r > 2$ получение представлений вида (36) возможно только путем перебора различных вариантов и проверки правильности реализации функций $\Phi_n^a(X)$.

На рис. 1 приведены графики зависимости сложности схем, синтезированных методом первичной факторизации (график 2) и вторичной факторизации при $r = \tau = 2$ (график 3). На рис. 4 приведена схема устройства подсчета количества единиц 24-разрядного двоичного кода, синтезированная методом вторичной факторизации (r=3, τ=2,3) с использованием представления (36).

3.3. Однородные и регулярные структуры

В [15] исследованы однородные и регулярные структуры для реализации пороговых равновесных функций и , в частности, симметричные древовидные структуры, в которых реализация функций $F_n^a(X)$ осуществляется блоком, представляющим собой сумматор двух унитарных параллельных непозиционных кодов, по функциям разложения $F_{m_1}^{a_1}(X_1)$ и $F_{m_2}^{a_2}(X_2)$. Данный подход применим и для синтеза схем модульного контроля, которые могут быть выполнены в виде древовидной структуры, каждый из узлов которой при $n > K-1$ содержит сумматор двух унитарных параллельных непозиционных кодов и блок свертки результата суммирования по модулю К в соответствии с (6) или (13).

Блок суммирования может быть выполнен одним из трех способов. В первом случае сумматор двух унитарных параллельных непозиционных кодов, формирующий систему функций $g_n^a(X)$, где $\{a\}=\{1,2,...,m\}$, содержит $(m^2+2m)/8$ ячеек И/ИЛИ ($m = \min\{2K-2,\ 2\lceil n/2\rceil\}$), соединенных пирамидально, причем

$$x_1(i,j)=\begin{cases} g_{m_1}^{i}(X_1) & \text{при} \quad j=1; \quad i=1,...,m/2, \\ z(i,j-1) & \text{при} \quad j=2,...,m/2; \quad i=1,...,m/2-j+1, \end{cases} \tag{44}$$

$$x_2(i,j)=\begin{cases} g_{m_2}^{i}(X_2) & \text{при} \quad j=1; \quad i=1,...,m/2, \\ y(i+1,j-1) & \text{при} \quad j=2,...,m/2; \quad i=1,...,m/2-j+1, \end{cases} \tag{45}$$

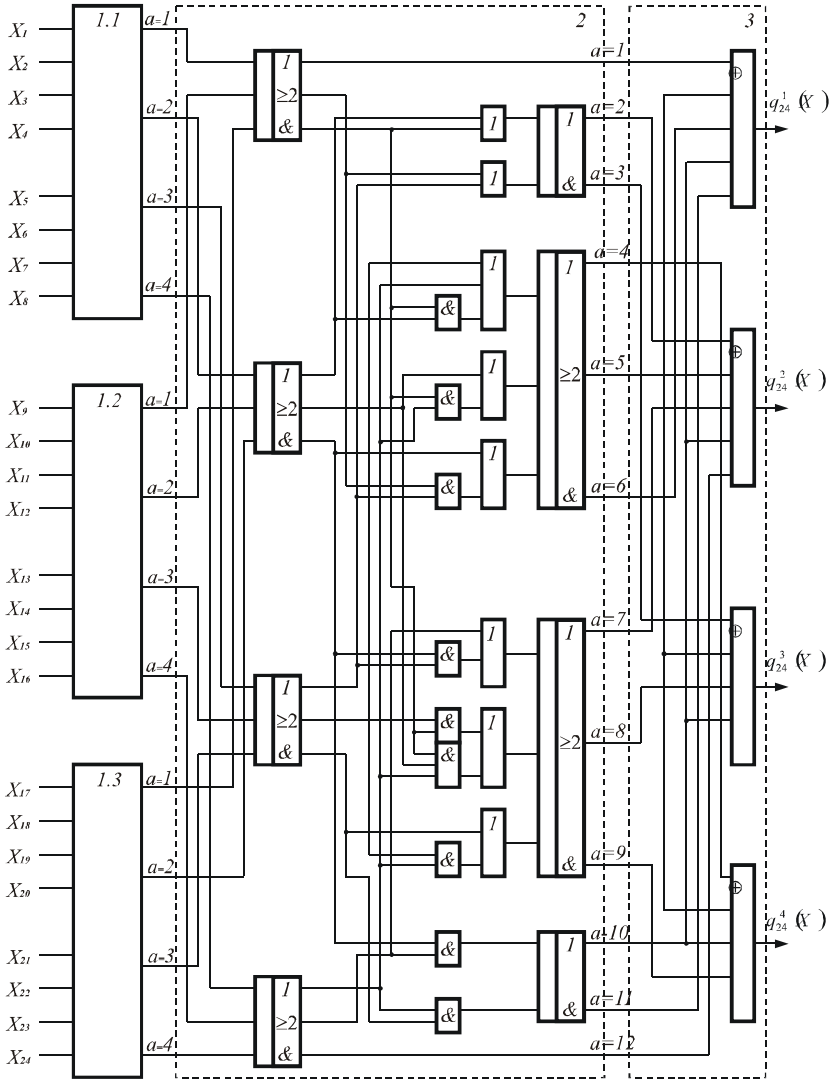


Рис.4. Схема подсчета количества единиц 24-разрядного двоичного кода по модулю K=5 в унитарном непозиционном коде, синтезированная методом вторичной факторизации (r=3, τ=2,3)

где $x_1(i,j)$ и $x_1(i,j)$ сигналы на входах i-й ячейки j-го столбца, а $z(i,j)$ и $y(i,j)$ - сигналы на ее выходах ИЛИ и И соответственно, причем функции $\Phi_n^a(X)$ реализуются на следующих выходах блока

$$\Phi_n^a(X) = \begin{cases} y(1,a) & \text{при } a = 1,2,\ldots,]m/2[, \\ z(a-]m/2[, m-a+1) & \text{при } a =]m/2[+1,\ldots,m. \end{cases} \quad (46)$$

Синтезируемая таким способом схема устройства подсчета количества единиц параллельного двоичного кода по модулю K=5 приведена на рис. 5.

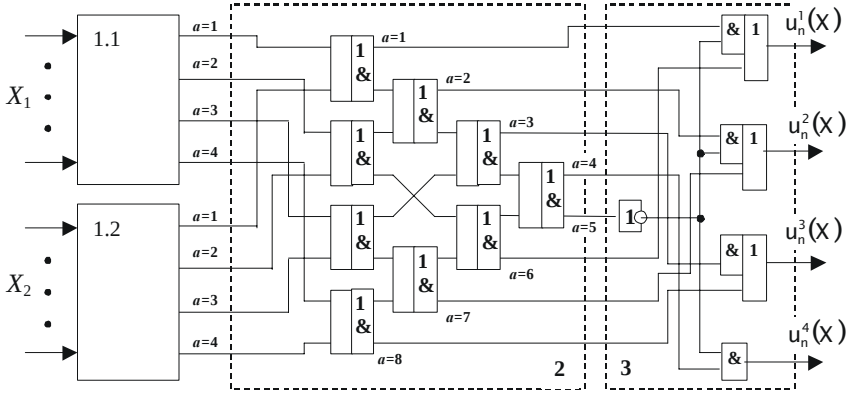


Рис. 5. Устройство подсчета количества единиц двоичного n-разрядного кода по модулю K=5, выполненное в виде композиции однородных структур, синтезированных методом симметричной факторизации

Сумматор унитарных параллельных непозиционных кодов для нечетного $n<2K-2$ получается из сумматора на $(n+1)$ вход путем исключения последней $(n+1)/2$-й ячейки первого столбца.

Сложность синтезированной таким способом схемы определяется следующей оценкой, получаемой из (28) с учетом сложности блока суммирования.

Следствие 3. Логическая схема формирования остатка количества единиц двоичного кода по модулю K в унитарном параллельном непозиционном коде, выполненная в виде композиции пирамидальных структур из однотипных ячеек И/ИЛИ содержит:

$$L(G(n)) = L(G(m_1)) + L(G(m_2)) + \Delta L_3 \quad (47)$$

элементов И, ИЛИ, НЕ, где где m_1, m_2>K-1 или m_1=]n/2[,

$$(48)\quad \Delta L_3 = \begin{cases} \left[\left(n^2+2n\right)/4\right] & при \quad n=2]n/2[\quad и \quad n<K, \\ \left[\left(n^2+4n-5\right)/4\right] & при \quad n\neq 2]n/2[\quad и \quad n<K, \\ \left[\left(n^2+2n\right)/4\right]+n & при \quad n=2]n/2[\quad и, \ K\le n\le 2K-2, \\ \left[\left(n^2+4n-5\right)/4\right]+n & при \quad n\neq 2]n/2[\quad и \quad K\le n\le 2K-2, \\ (K-1)^2+3(K-1) & при \quad n>2k-2.. \end{cases}$$

Из полученной оценки видно, что сложность схем, выполненных в виде композиции пирамидальных структур из однотипных ячеек И/ИЛИ, равна сложности схем, синтезированных методом первичной факторизации.

Более быстродействующие и экономичные однородные симметричные структуры сумматоров унитарных параллельных непозиционных кодов могут быть синтезированы с использованием метода симметричной факторизации систем пороговых равновесных функций, предложенного в [15]. Получаемые при его использовании схемы также представляют собой пирамидальную структуру из однотипных ячеек И/ИЛИ, однако число ячеек уменьшено за счет реконфигурации связей внутри пирамидальной структуры. Получение аналитических оценок сложности и быстродействия для схем, синтезированных методом симметричной факторизации не представляется возможным. Для фундаментальных симметричных многополюсников, реализующих систему всех пороговых равновесных функций n переменных, при $n \le 32$ соответствующие оценки получены [15] путем экспериментального проектирования и последующего анализа разработанных технических решений. Они позволяют получить оценки сложности устройств модульного контроля при $K \le 16$. Как показали результаты оптимизации схем, синтезируемых методом симметричной факторизации, выполненной путем перебора на ЦВМ всех возможных значений m_i для K=5,7, их минимальная сложность равна сложности схем, синтезируемых методом вторичной факторизации, определяемой оценкой (42).

Синтезируемая с использованием метода симметричной факторизации схема устройства подсчета количества единиц параллельного двоичного кода по модулю K=5 приведена на рис. 5.

Дальнейшее повышение быстродействия и уменьшение сложности схем модульного контроля, выполняемых в виде композиции

пирамидальных структур может быть достигнуто [15] при использовании сумматора унитарных параллельных непозиционных кодов, реализуемого в виде композиции двух блоков суммирования унитарных параллельных непозиционных кодов на m_1 и $m_2 = n - m_1$ входов и группы ячеек И/ИЛИ, формирующих выходные сигналы. Входы первого и второго блоков суммирования соединяются с входами первого $A_1 = \{a_1^1, a_2^1, ..., a_{]m/2[}^1\}$ и второго $A_2 = \{a_1^2, a_2^2, ..., a_{[m/2]}^2\}$ слагаемых сумматора унитарных параллельных непозиционных кодов в следующем порядке

(49) $b_i^1 = a_{2i-1}^1$ при $i = 1,...,]m_1/2[$,

(50) $b_i^2 = a_{2i-1}^2$ при $i = 1,..., [m_1/2]$,

(51) $d_i^1 = a_{2i}^1$ при $i = 1,...,]m_2/2[$,

(52) $d_i^2 = a_{2i}^2$ при $i = 1,..., [m_2/2]$,

где $a_i^1 = g_{n_1}^i(X_1)$, $a_i^2 = g_{n_2}^i(X_2)$,

$B_1 = \{b_1^1, b_2^1, ..., b_{]m_1/2[}^1\}$ и $B_2 = \{b_1^2, b_2^2, ..., b_{[m_1/2]}^2\}$ - входы первого и второго слагаемых первого блока суммирования;

$D_1 = \{d_1^1, d_2^1, ..., d_{]m_2/2[}^1\}$ и $D_2 = \{d_1^2, d_2^2, ..., d_{[m_2/2]}^2\}$ входы первого и второго слагаемых второго блока суммирования.

Порядок соединения выходов блоков суммирования со входами выходных ячеек И/ИЛИ $D_1 = \{d_1^1, d_2^1, ..., d_{]m_2/2[}^1\}$ определяется следующими соотношениями

(53) $x_1(i) = p_{i+1}$,

(54) $x_2(i) = q_i$,

где i=1,2,...,m/2-1, $x_1(i)$ и $x_2(i)$ - первый и второй входы i-й ячейки И/ИЛИ, а p_j и q_j - j-е выходы первого и второго блоков суммирования.

На выходах ячеек И/ИЛИ реализуется система функций $\Phi_n^i(X)$, причем

(55) $$\Phi_n^i(X) = \begin{cases} p_1 & \text{при } i = 1, \\ y_{[i/2]} & \text{при } i = 2,4,\ldots,m-2, \\ z_{[i/2]} & \text{при } i = 3,5,\ldots,m-1, \\ q_{m/2} & \text{при } i = m, \end{cases}$$

где y_i и z_i - выходы i-й ячейки И/ИЛИ.

Каждый из блоков суммирования унитарных параллельных непозиционных кодов, вплоть до блоков на два входа, представляющих собой ячейку И/ИЛИ, выполняется аналогичным образом.

Сложность таких схем определяется соотношением

(56) $$L(G(n)) = L(G(m_1)) + L(G(m_2)) + l(s(\mu)),$$

где $\mu=\mu_1+\mu_2$, $\mu_i=\min\{m_i, K-1\}$

(57) $$l(s(\mu)) = l(s(\mu_1)) + l(s(\mu_2)) + 2](\mu-2)/2[,$$

$l(s(\mu_i))$- сложность сумматора унитарных параллельных непозиционных кодов на μ_i входов.

Последняя из рассмотренных однородных древовидных симметричных структур обладает наибольшим быстродействием и наименьшей сложностью, однако, регулярность связей между ячейками практически отсутствует.

4. Методы синтеза логических схем формирования остатка двоичного кода по модулю К в унитарном параллельном непозиционном коде

Из введенного выше понятия унитарного параллельного непозиционного кода $U(X)$ остатка по модулю K двоичного кода $X=\{x_1, x_2,\ldots, x_n\}$ следует, что он обладает свойствами, аналогичными свойствам кода $G(X)$.

С учетом этого легко доказывается справедливость (по аналогии с приведенными в разделе 3.1 доказательствами) следующего представления

$$u_n^a(X)=\bigvee_{j=0}^{i'}\left(\left\{\bigvee_{\{A_i^j\}}u_{m_1}^{a_1}(X_1)u_{m_2}^{a_2}(X_2)\ldots u_{m_r}^{a_r}(X_r)\right\}\&\overline{\left\{\bigvee_{\{B_i^j\}}u_{m_1}^{b_1}(X_1)u_{m_2}^{b_2}(X_2)\ldots u_{m_r}^{b_r}(X_r)\right\}}\right)=$$

$$\bigvee_{j=0}^{l'}\hat{\Phi}_n^{a+jK}(X)\&\overline{\hat{\Phi}_n^{(j+1)K}(X)}=\bigoplus_{j=0}^{l'}\left(\hat{\Phi}_n^{a+jK}(X)\oplus\hat{\Phi}_n^{(j+1)K}(X)\right), \quad (58)$$

являющегося результатом декомпозиции с параметром *r*. Т.е. рассмотренный выше метод декомпозиции может быть использован для синтеза схем формирования остатка параллельного двоичного кода по модулю *K* без существенных изменений.

Используя свойства функций $u_n^a(X)$ из (58) получаем представления, являющиеся результатом первичной и вторичной факторизации, полностью аналогичные (26) или (32) и (36). Для синтеза схем формирования остатка двоичного кода по модулю К могут также быть использованы однородные и регулярные структуры, описанные выше.

Рассмотренные методы синтеза могут использоваться при произвольном порядке декомпозиции. Под порядком декомпозиции понимается порядок выбора значений параметра разложения *r* и мощности подмножеств X_1, X_2,..., X_n на каждом шаге декомпозиции. Однако, в данном случае сложность синтезируемой схемы зависит не только от выбора мощности подмножеств, но и от значений весов ω_j переменных $x_j \in X_i$. Оптимальным является порядок декомпозиции, при котором минимизируется мощность множества функций разложения $u_{m_i}^{a_i}(X_i)$ на всех шагах декомпозиции. Очевидно, что это достигается, когда в процессе декомпозиции при мощности подмножеств $\sigma(X_i)<K$ для всех $x_j \in X_i$ имеем $(\omega_j)\mathrm{mod}K=\mathrm{const}$. В этом случае сложность схемы формирования остатка параллельного двоичного кода по модулю *K*, синтезируемой методами декомпозиции и факторизации, а также выполняемых в виде композиции однородных структур, незначительно отличается от определяемой оценками (15), (29), (42), (47), (56).

В целом процесс синтеза схемы формирования остатка двоичного кода по модулю *K* отличается от процесса синтеза схемы подсчета количества единиц по модулю *K* только на этапе, когда перестают совпадать мощности множеств значений индексов функций

разложения, т.е. на конечных циклах процесса синтеза. Различия в процессе синтеза и сложности синтезируемых схем можно минимизировать, если в случае, когда для всех $x_j \in X_i$ имеем $(\omega_j) \bmod K = \text{const}$, формировать систему функций $u_{m_i}^{a_i}(X_i)$ на основе системы функций $g_{m_i}^{a_i}(X_i)$ в соответствии с представлением

$$u_{m_i}^{a_i}(X_i) = \bigvee_{j=a_i}^{K-1} \gamma_{m_i}^{j}(X_i) = \gamma_{m_i}^{a_i}(X_i) \vee u_{m_i}^{a_i+1}(X_i), \tag{59}$$

где

$$\gamma_{m_i}^{j}(X_i) = \begin{cases} g_{m_i}^{b}(X_i) & \text{при } j = (b\omega) \bmod K \text{ и } [b\omega/K] \geq [\omega(K-1)/K], \\ g_{m_i}^{b}(X_i)\overline{g_{m_i}^{d}(X_i)} & \text{при } j = (b\omega) \bmod K \text{ и } [b\omega/K] < [\omega(K-1)/K], \end{cases} \tag{60}$$

или

$$\gamma_{m_i}^{j}(X_i) = \bigvee_{\{B\}_j} \Phi_{m_i}^{b}(X_i)\overline{\Phi_{m_i}^{d}(X_i)}, \tag{61}$$

$\{B_j\}$ - множество индексов функций разложения $\Phi_{m_i}^{b}(X_i)$, удовлетворяющих условию $j=(b\omega) mod K$, а $d=]([b\omega/K]+1)K/\omega[$.

Поскольку сложность схем, синтезируемых рассмотренным методом, зависит от порядка декомпозиции, получить оценку сложности в общем виде не представляется возможным. Однако в том случае, когда выполняется указанное выше условие, сложность схемы составляет

$$L\left(U(n)\right) = L(G(n)) + \Delta L \tag{62}$$

где ΔL - количество логических элементов, необходимых для реализации преобразования (59) или (61), которое может быть оценено следующим образом

$$\Delta L \approx \begin{cases} 3 & \text{при } K = 3, \\ 23 & \text{при } K = 5, \\ 22 & \text{при } K = 7. \end{cases} \tag{63}$$

5. Быстродействие логических схем модульного контроля в унитарных параллельных позиционных кодах

Важным параметром логических схем, кроме их сложности является также быстродействие. Как правило, его определяют глубиной схемы, т.е. максимальным числом последовательно

соединенных логических элементов от входа схемы к ее выходу. Используем данный критерий для оценки быстродействия логических схем модульного контроля, синтезируемых рассмотренными выше методами.

Схемы максимальной глубины получаются в частном случае декомпозиции, представляющем собой разложение по переменным. Их глубина зависит от числа переменных n и параметра разложения r и для классического базиса И, ИЛИ, НЕ определяется следующей оценкой

(64) $$H\left(G\left(n\right)\right) = \begin{cases} 3n - K - 2 & при\ \ r = 2, \\ H\left(F\left(n - t\left(r-1\right)\right)\right) + \Delta ht - \Delta H & при\ \ r>2, \end{cases}$$

где $\Delta h = 5$, t=](*n*-*K*+1)/(*r*-1)[, а

$$\Delta H = \begin{cases} 2 & при\ \ n - t(r-1) = K - r + 1, \\ 0 & при\ \ n - t(r-1) \neq K - r + 1, \end{cases}$$

$$H\left(F\left(n\right)\right) = \begin{cases} 2\left(n-1\right)-1 & при\ \ r = 2, \\ 2k & при\ \ n = \left(r\text{-}1\right)k + 1\ u\ \ r > 2, \\ 2k+1 & при\ \ n = \left(r\text{-}1\right)k + 2\ u\ \ r > 2, \\ 2k+2 & при\ \left(r\text{-}1\right)k + r\text{-}1 \geq n > \left(r\text{-}1\right)k + 2\ u\ \ r > 3. \end{cases}$$

Схемы минимальной глубины получаются при регулярной детерминированной декомпозиции. Из уравнения (6) следует, что каждый шаг декомпозиции увеличивает глубину схема на Δh_1 элементов, при этом глубина синтезируемых схем определяется следующим рекуррентным соотношением

(65) $$H\left(G\left(n\right)\right) = H\left(G\left(\right]n/r\left[\right)\right) + \Delta h_1 < \Delta h_1 \log_r n\ ,$$ где $n>K$.

В классическом базисе Δh_1=5, так как для реализация функции $g_n^a\left(X\right)$ или $u_n^a\left(X\right)$ по их функциям разложения $g_{m_i}^{a_i}\left(X_i\right)$ и $u_{m_i}^{a_i}\left(X_i\right)$ требуется структура типа И-ИЛИ, имеющая глубину 2, и схема свертки сформированных функций $\Phi_{m_i}^{a_i}\left(X_i\right)$ и $\Phi_{m_i}^{a_i}\left(X_i\right)$ по модулю К, также имеющая глубину 3. При использовании базиса, включающего операцию суммирования по модулю два Δh_1=3.

При синтезе методом первичной факторизации на основе представлений (26) глубина схем возрастает, по сравнению с синтезированными методом декомпозиции. Это обусловлено

необходимостью реализации системы пороговых равновесных функций $F_r^{\alpha_i}(Y_i)$. Глубина синтезируемой схемы при r=2, 3 определяется следующим рекуррентным соотношением

(66) $$H(G(n)) = H(G(\lfloor n/r \rfloor)) + r + \Delta h_1 - 1 < (r + \Delta h_1 - 1)\log_r n \ .$$

Использование вторичной факторизации на основе представления (36) ($r=\tau=2$) приводит к схемам глубины

(67) $$H(G(n)) = H(G(\lfloor n/2 \rfloor)) + \Delta h_1 + 2 < 7\log_2 n \ .$$

Реализация в виде композиции пирамидальных структур приводит к схемам глубины

(68) $$H(G(n)) = H(G(\lfloor n/2 \rfloor)) + K + 2 \ , \quad \text{где } n \geq 2K\text{-}2 \ .$$

В случае симметричной факторизации

(69) $$H(G(n)) = H(G(\lfloor n/2 \rfloor)) + K + 2 - \Delta H \ ,$$

где $n \geq 2K$-2, а $\Delta H \approx \rfloor (K\text{-}5)/4 [$.

Глубина схем формирования остатка двоичного кода по модулю К в унитарном параллельном непозиционном коде, синтезированных рассмотренными методами при $n \geq 2K$-2 определяется оценками (59)-(62). При $n < 2K$-2 их глубина, в зависимости от порядка декомпозиции, может превышать определяемую указанными оценками на величину $\Delta h_2 \leq 3$.

Приведенные оценки показывают, что схемы модульного контроля в унитарных параллельных непозиционных кодах имеют на 20-25% меньшую сложность, но в 1,5- 2 раза большую глубину, чем схемы модульного контроля в унитарных параллельных позиционных кодах [6]. Увеличение параметра разложения r приводит к уменьшению глубины схем, однако, при этом быстро возрастает их сложность.

6. Заключение

В статье исследованы вопросы синтеза логических схем модульного контроля в унитарных параллельных непозиционных кодах, предложены декомпозиционный метод синтеза и методы факторизации. Получены оценки сложности и быстродействия синтезируемых рассмотренными методами схем. Они показывают, что синтезируемые схемы имеют меньшую сложность и меньшее

быстродействие, по сравнению со схемами модульного контроля в унитарных параллельных позиционных кодах. Характеристики синтезируемых схем существенно зависят от выбора значения параметра разложения *r*. Его увеличение приводит к повышению быстродействия синтезируемой схемы, но увеличивает ее сложность.

Полученные оценки сложности и быстродействия логических схем модульного контроля в унитарных параллельных непозиционных кодах показывают, что их использование целесообразно при высоких требованиях к быстродействию схемы и , в основном, при малых значениях модуля К. Приведенные в данной статье результаты формируют достаточно полный теоретический подход к проблеме синтеза логических схем модульного контроля в унитарных параллельных непозиционных кодах, позволяют определить области их возможного использования, а также точно оценить на ранних стадиях проектирования параметры синтезируемых схем и решить вопрос о целесообразности их использования.

Литература

1. *Журавлев Ю.П., Кателюк Л.А., Циклинский Н.И.* Надежность и контроль ЭВМ. М.: Радио и связь, 1978.
2. *Долгов А.И.* Диагностика устройств, функционирующих в системе остаточных классов. М.: Сов. Радио, 1982.
3. *Селлерс Ф.* Методы обнаружения ошибок в работе ЭЦВМ. М.: Мир, 1972.
4. *Дадаев Ю.Г.* Теория арифметических кодов. М.: Радио и связь, 1981.
5. *Касами Т., Токура Н., Ивадари Ё., Инагаки Я.* Теория кодирования. М.: Мир, 1978.
6. *Музыченко О.Н.* Синтез логических схем модульного контроля в унитарных позиционных двоичных кодах. // Автоматика и телемеханика. 2001. № 3. С. 158 - 173.
7. *Фридман А., Менон П.* Теория и проектирование переключательных схем. М.: Мир, 1978.
8. *Поляков В.Е., Проскурин Г.М., Федотов В.П., Шарнин Ю.К.* Преобразование симметричных булевых функций.// Материалы семинара по кибернетике. Кишенев, 1972. Вып. 47. С. 16-30.
9. *Дертоузос М.* Пороговая логика. М.: Мир, 1967.
10. *Вавилов Е.Н., Егоров Б.М., Ланцев В.С., Тоценко В.Г.* Синтез схем на пороговых элементах. М.: Сов. Радио, 1970.

11. *Музыченко О.Н., Лукоянов В.П.* Быстродействующий алгоритм синтеза схем симметричных функций алгебры логики для систем автоматизированного проектирования. // Вопросы радиоэлектроники. Сер. Общие вопросы радиоэлектроники. 1984. Вып.12. С. 120-128.
12. *Музыченко О.Н.* Факторизационный метод синтеза пороговых схем. // Автоматика и телемеханика. 1986. № 8. С. 166-170.
13. *Музыченко О.Н.* Факторизация систем пороговых равновесных функций. // Кибернетика. 1989. № 4. С. 124-127.
14. *Музыченко О.Н.* Устройство для контроля параллельного двоичного кода по модулю К. А.с. 1361557 СССР// Б. И. 1987. № 47.
15. *Музыченко О.Н.* Однородные и регулярные структуры для реализации симметричных функций алгебры логики// Автоматика и телемеханика. 1998. № 4. С. 152-164.
16. *Пихтова Л.Д.* К факторизационному методу синтеза комбинационных схем.// Вычислительная техника в машиностроении. Минск: Ин-т техн. кибернетики АН БССР. 1973. С. 122-128.
17. *Маркаускас Р.К.* Алгоритм факторизации булевых функций. В кн.: Вычислительная техника. Каунас: Каунасский политехн. ин-т, 1972, с. 122-128.
18. *Музыченко О.Н.* Устройство для контроля параллельного двоичного кода по модулю К. А.с. 1425676 СССР// Б. И. 1988. № 35.

УДК. 681.3

Н.И. Червяков., И.В. Дьяченко

Принципы построения модулярных сумматоров и умножителей

(Ставропольский государственный университет)

В данной работе система остаточных классов и модулярная арифметика рассматриваются как средства повышения производительности и надёжности вычислений. Обладая возможностью распараллеливания, модулярные вычисления для повышения эффективности требуют разработки специализированных схем выполнения арифметических операций в модулярных каналах. Рассмотрены базовые принципы и преимущества вычислений в системе остаточных классов, обозреваются основные результаты, полученные в этом перспективном направлении.

Abstract. Modular arithmetic and Residue Number System are considered in this article as means of increase of productivity and reliability of computations. Possessing its parallel computing opportunity, modular calculations demand development of specialized arithmetic operations schemes for increase of computation efficiency in modular channels. Base principles and advantages of calculations in system of residual classes are considered, the basic results received in this perspective direction are surveyed

Задачу повышения скорости и надёжности вычислений можно рассматривать с двух сторон. С одной стороны это аппаратный уровень, фундаментальными ограничениями на котором являются технические возможности создания элементной базы – уменьшение размеров кристаллов, увеличение частоты синхронизации (тактовой частоты), решение проблем теплоотвода и др. Во многом этот уровень определяется современным состоянием фундаментальных наук, прежде всего, физики. С другой стороны это – математико-алгоритмический уровень вычислений, и фундаментальными ограничивающими факторами здесь выступают, в числе прочих, необходимость последовательного вычисления, когда следующий этап (шаг) частично или полностью зависит от предыдущих шагов. Даже простейшие арифметические операции сложения и умножения (не говоря уже о делении) при реализации их вычислителями с архитектурой фон-Неймана осуществляются побитово, и вычисление каждого последующего бита зависит от результата операции над предыдущими битами (в данном случае это знак переноса – carry sign). Существуют и другие вычислительные архитектуры, в которых акцент сделан на параллельность и массовость вычислений. Большую популярность сейчас имеют нейронные сети, которые, обладая алгоритмической универсальностью машины Тьюринга [1], уже доказали своё преимущество в слабо формализованных задачах, связанных с необходимостью обучения.

Использование системы остаточных классов (СОК) и модулярных вычислений позволяет существенно увеличить скорость арифметических вычислений за счёт параллельного выполнения операций над остатками. Современная аппаратная база позволяет также заменять арифметические операции над остатками однотактными табличными выборками.

В работах [2, 3] разработан вычислительный объект «нейронная сеть конечного кольца», сочетающий преимущества модулярного представления информации и параллельность и помехоустойчивость нейронных сетей.

Долгое время модулярная арифметика рассматривалась как интересный сугубо теоретический вопрос из-за сложности производства вычислительных структур для её реализации. Современное развитие технологии интегральных схем сделало

возможным использование модулярной арифметики для многих областей цифровой обработки сигналов, распознавания образов и других задач, требующих интенсивных вычислений.

Система остаточных классов – непозиционная система счисления

Пусть заданы взаимно простые положительные числа m_1, m_2, … , m_L, которые называют основаниями или модулями системы ($\text{НОД}(m_i, m_j) = 1$ для $i \neq j$). Число $M = \prod_{i=1}^{L} m_i$ называют вычислительным диапазоном получившейся числовой системы. Любое число X из кольца целых чисел по модулю M, $X \in Z(M)$, имеет уникальное представление в заданной СОК [2-4]:

$$X \xrightarrow{\text{СОК}} \left(|X|_{m_1}, |X|_{m_2}, \ldots, |X|_{m_L} \right) \qquad (1)$$

В СОК могут быть также представлены и отрицательные числа из диапазона $\{-(M-1)/2, \ldots, -1, 0, 1, \ldots, (M-1)/2\}$ для нечётного M и $\{-M/2, \ldots, -1, 0, 1, \ldots, (M/2)-1\}$ для чётного M. При этом, если $X < 0$, то:

$$X < 0 \xrightarrow{\text{СОК}} \left(|M - |X||_{m_1}, |M - |X||_{m_2}, \ldots, |M - |X||_{m_L} \right) \qquad (2)$$

Основным достоинством СОК является то, что арифметические операции производятся в ней независимо по каждому из модулей, следовательно, они могут выполняться параллельно по L вычислительным каналам:

$$X * Y \xrightarrow{\text{СОК}} \left(\underbrace{\left| |X|_{m_1} * |Y|_{m_1} \right|_{m_1}}_{\text{канал } m_1}, \underbrace{\left| |X|_{m_2} * |Y|_{m_2} \right|_{m_2}}_{\text{канал } m_2}, \ldots, \underbrace{\left| |X|_{m_L} * |Y|_{m_L} \right|_{m_L}}_{\text{канал } m_L} \right), \qquad (3)$$

$$\forall X, Y \in Z(M), \quad * \in \{\oplus, \otimes\}.$$

Малоразрядность обрабатываемых остатков позволяет для повышения быстродействия арифметических операций в вычислительных каналах применять методы табличной подстановки (LUT). Более подробно модулярная арифметика в вычислительных каналах будет рассмотрена ниже, а пока же

обратим внимание на следующее фундаментальное положение, лежащее в основе модулярных вычислений.

Теорема (Китайская Теорема об Остатках, КТО). *Пусть даны попарно взаимно простые модули* $\{m_1, m_2, \ldots, m_L\}$ *и число* $X \in Z(M)$, *СОК-представление которого* $[x_1, x_2, \ldots, x_L]$ *определяется выражением:*

$$x_i = |X|_{m_i} \quad i = 1, 2, \ldots, L \tag{4}$$

Тогда:

$$Z(M) \cong Z(m_1) \oplus Z(m_2) \oplus \ldots \oplus Z(m_L) \tag{5}$$

Иными словами, кольцо целых чисел по модулю M, $Z(M)$, *изоморфно прямой сумме колец целых чисел по модулям* $m_1, m_2, \ldots, m_L$ *и существует единственное* $X \in Z(M)$, $M = \prod_{i=1}^{L} m_i$, *восстанавливаемое по остаткам из выражения (4) по формуле*:

$$X = \left| \sum_{i=1}^{L} \hat{m}_i \left| x_i \left| \hat{m}_i \right|_{m_i}^{-1} \right|_{m_i} \right|_M \tag{6}$$

где

$$\hat{m}_i = \frac{M}{m_i} \tag{7}$$

и $\left| \hat{m}_i \right|_{m_i}^{-1}$ *– обратный по умножению в кольце* $Z(m_i)$ *элемент для* $\hat{m}_i$*:*

$$\left| \hat{m}_i \left| \hat{m}_i \right|_{m_i}^{-1} \right|_{m_i} = 1 \tag{8}$$

Конечно, в Древнем Китае эта закономерность была сформулирована не в таком виде, а, вероятнее всего, в виде алгорифма решения математических загадок типа «необходимо определить число, имеющее остатками 2, 3, 2, соответственно, при делении на 3, 5, 7» (речь идёт о числе 23) [5]. В XVIII веке Эйлер продемонстрировал одно из возможных применений КТО для

модулярных вычислений в СОК, а в XIX веке К.Ф.Гаусс доказал эту теорему, заложив основы современной теории СОК [6].

Таким образом, КТО предоставляет фундаментальный метод для замены операций в кольце $Z(M)$ на операции в нескольких независимых кольцах $Z(m_i)$, осуществляемые параллельно (5).

Модулярные вычисления

Как было отмечено выше (3), операции сложения и умножения в СОК осуществляются параллельно по L вычислительным каналам. Обобщенная структура устройств цифровой обработки сигналов в модулярной арифметике представлена на рис. 1 [7]. Число X на входе преобразовываются из позиционной системы счисления (ПСС) в модулярное представление в СОК в базисе модулей $\{m_1, m_2, \ldots m_L\}$, после чего выполняются независимые вычисления для каждого модуля m_i. На выходе происходит обратное преобразование из СОК в ПСС.

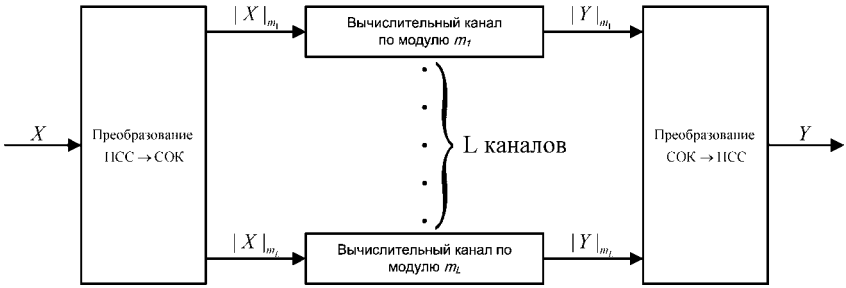


Рис. 1. Общая структура устройств цифровой обработки сигналов в СОК.

Как отмечено в работе [7], структура, изображённая на рис. 1, имеет ряд неоспоримых преимуществ при её реализации на интегральных схемах:

1. Независимость каждого канала по отдельному модулю обеспечивает значительную гибкость при планировке и топологическом проектировании кристалла.

2. Реализация таких устройств на основе ПЛИС, обладающих меньшими вентильными ресурсами, может быть легко перепланирована и размещена в несколько кристаллов.

3. Трассировочные межсоединения распространяются только внутри отдельного вычислительного канала, что исключает наличие длинных трасс и, как следствие, обеспечивает некоторое уменьшение потребляемой мощности и уменьшение задержек по критическим путям.

4. Отсутствие специальных требований по синхронизации между отдельными каналами (за исключением синхронизации на входе и выходе) значительно облегчает трассировку цепей тактовых частот, которые будут иметь меньшую расфазировку. Это, в свою очередь, приводит к уменьшению пиковых выбросов по цепям синхронизации.

5. При необходимости введение дополнительных избыточных каналов обеспечивает возможность построения отказоустойчивых систем.

Приведённые факторы, наряду с преимуществами модулярных вычислителей в быстродействии и занимаемой площади, позволяют говорить о вычислениях в СОК как о перспективной технологии разработки высокопроизводительных систем, функционирующих в реальном времени [7].

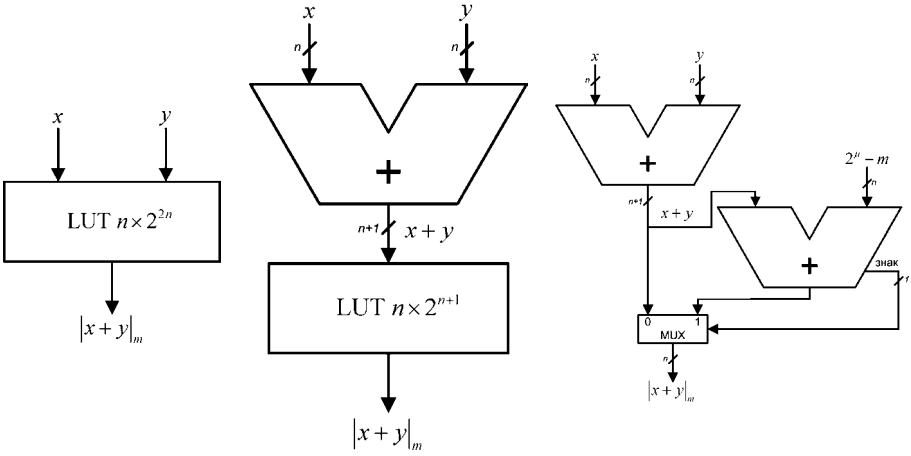
Поскольку в СОК используются модулярные операции, для высокой эффективности необходимо использовать специально спроектированные для СОК сумматоры и умножители.

Существует достаточно большое количество подходов к реализации сумматоров по модулю m [8-10]. Далее будут рассмотрены наиболее типичные и простые схемы модулярного суммирования (рис. 2).

Первая из них вычисляет модульную сумму $\left|x+y\right|_m$ с помощью таблицы размером $n \times 2^{2n}$, $n = \lceil \log_2(m) \rceil$. Для двух соответствующих элементов просто выбирается ответ из большой таблицы. Это решение очень хорошо подходит для случаев, когда длина слова мала, например, $n \le 4$.

Для больших модулей, память LUT была бы значительного размера и другие схемы для суммирования оказываются в этом случае более предпочтительными. Следующее предложение основывается на обычном суммировании $x+y$ и одной таблицы, содержащей все

возможные значения для $|x+y|_m$. При этом существенно сокращается размер подстановочной таблицы с $n\times 2^{2n}$ до $n\times 2^{n+1}$, что даёт возможность расширять набор модулей в случае необходимости большего динамического диапазона или избыточных модульных каналов для коррекции ошибок.



(а) (б) (в)

Рис. 2. Модулярное суммирование. (а) – с помощью большой LUT-таблицы; (б) – с предварительным обычным суммированием; (в) – без использования LUT-таблиц, μ – аппаратная разрядность сумматора.

Третья схема суммирования является самой распространённой и наиболее предпочтительной в большинстве случаев. В этой схеме используется два сумматора и мультиплексор для выбора результата в соответствии с выражением:

$$|x+y|_m=\begin{cases}x+y & 0\le x+y<m\\ x+y-m & m\le x+y\end{cases}. \qquad (9)$$

Умножители на основе закона квадратов (рис. 3) вычисляют модулярное произведение $|x\cdot y|_m$ с помощью следующего равенства (закон квадратов):

$$xy=\left(\frac{x+y}{2}\right)^2-\left(\frac{x-y}{2}\right)^2, \qquad (10)$$

где $0 \le x,\ y < m$. Модулярное умножение на основе (10) можно записать следующим образом:

$$|xy|_m = \left|\Phi(s^+) - \Phi(s^-)\right|_m$$
$$\Phi(s) = \left|s^2\right|_m \qquad (11)$$
$$s^+ = \frac{x+y}{2} \quad s^- = \frac{x-y}{2},$$

и произведение $|xy|_m$ можно вычислять по формуле:

$$|xy|_m = \left|\left|\frac{1}{4}\right|_m \left|(x+y)^2 - (x-y)^2\right|_m\right|_m . \qquad (12)$$

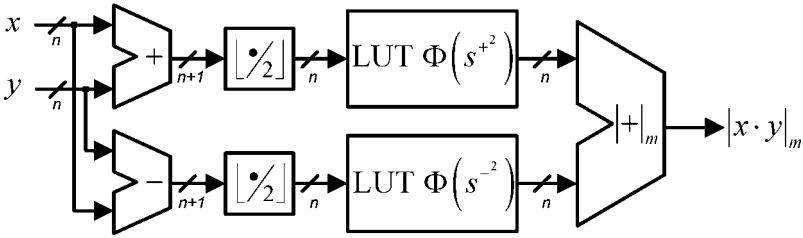


Рис. 3. Схема модулярного умножителя по модулю *m* на основе закона квадратов.

Существование операции деления на 2 ставит под угрозу целочисленность промежуточных вычислений и, соответственно, правильность результата после использования таблиц подстановок. Более того, существование обратного по умножению по модулю *m* для 2 элемента, $|2|_m^{-1}$, гарантируется только в случае, если 2 не делит *m* (т.е. *m* – нечётно). Тэйлор в работе [11] привёл доказательство теоремы, показав, что даже если при вычислении (11) будут промежуточные дроби, они взаимно уничтожатся.

Умножители, основанный на арифметике указателей [12, 13] является сравнимой альтернативой по сложности и скорости умножителям, основанным на законе квадратов. Их использование ограничено простыми модулями и основывается на осуществлении преобразования в степенную форму (так называемое степенное исчисление), в котором умножение может более быстро осуществляться посредством операции суммирования.

Метод работы этого умножителя связан с математическими свойствами полей Галуа [14, 15], обозначаемых $GF(p)$, где p – простое число. Все ненулевые элементы поля Галуа могут быть получены путём многократного возведения в степень примитивного элемента – порождающёго поле $GF(p)$ элемента g_j. Это свойство полей Галуа можно использовать для умножения в $GF(m_j)$ благодаря использованию изоморфизма между мультипликативной по модулю m_j группой $Q = \{1, 2, \ldots, m-1\}$, и аддитивной по модулю $(m_j - 1)$ группой $I = \{0, 1, \ldots, m-2\}$. Этот изоморфизм может быть установлен следующим образом:

$$\forall q_n \in Q \quad \exists i_n \in I \quad : \quad q_n = g^{i_n} \qquad (13)$$

и умножение над полем $GF(m)$ может производиться по формуле:

$$\left| q_j q_k \right|_m = g^{\left| i_j + i_k \right|_{m-1}} \qquad (14)$$

Таким образом, умножение двух чисел q_j и q_k можно производить вычисляя модулярную сумму соответствующих указателей i_j и i_k, а затем проводя обратное преобразование из степенного пространства в исходный вид. Необходимо специально обрабатывать случаи, когда один из операндов на входе умножителя равен нулю и в этом случае назначать нулевой результат произведения. Это происходит потому, что не определён элемент в степенном пространстве, соответствующий нулевому элементу группы Q.

Степени i_j и i_k для q_j и q_k, соответственно, могут быть заранее вычислены и помещены в LUT. Сложение степеней выполняет сумматор по модулю $m_j - 1$. Обратное преобразование из степенного представления i_j и i_k в исходное q_j и q_k также может быть выполнено с помощью предварительно вычисленных LUT.

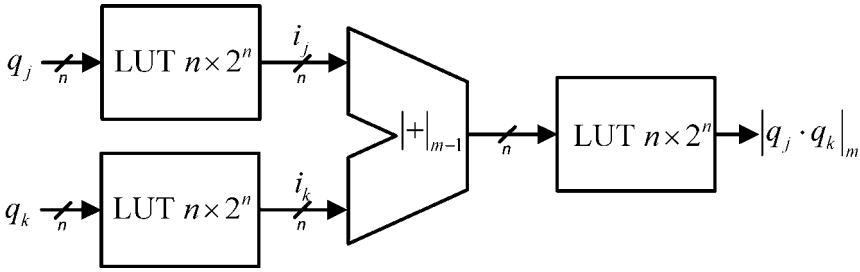


Рис. 4. Схема умножителя, основанного на исчислении степеней (умножитель Галуа).

Рассмотрим в качестве примера работу этого умножителя на примере умножения двух чисел 14 и 28 по модулю 31.

Так как 31 – простое число, существует порождающий элемент g, дающий возможность ассоциировать каждый элемент мультипликативной группы $Q=\{1,2,\ldots,31\}$ с элементом аддитивной группы $I=\{0,1,\ldots,30\}$. Соответствие задаётся выражением $q_n=\left|g^{i_n}\right|_{31}$, где $g=3$, и $q_n\in Q$, $i_n\in I$. В табл. 1 рассчитано соответствие между элементами группы Q и соответствующей степенью из аддитивной группы I. Эта таблица в сущности и представляет собой содержание LUT размером $2^5\cdot 5$ прямого и обратного преобразования в умножителе, изображенном на рис. 4.

Рассмотрим работу умножителя Галуа (рис. 4, рис. 5, табл. 1) на примере умножения $|14\cdot 28|_{31}$. Итак, $q_j=14$ и $q_k=28$, а произведение $|q_jq_k|_{31}$ получается посредством суммирования соответствующих им элементов i_j и i_k, выбранных из табл. 1. Таким образом, указатели оказываются $i_j=22$ и $i_k=16$ и $\left|i_j+i_k\right|_{30}=8$. Элементу $i_n=8$ в табл. 1 соответствует $q_n=20$, следовательно, $\left|14\cdot 28\right|_{31}=20$.

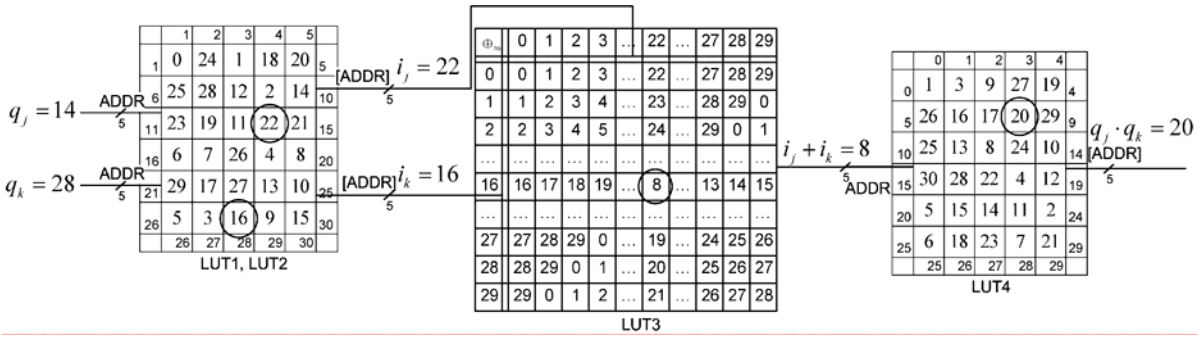
На рис. 5 показано, каким образом происходит умножение чисел 14 и 28 по модулю 31 по схеме, изображённой на рис. 4. Для простоты две таблицы LUT1 и LUT2 объединены в одну и представляют

Таблица 1.

Содержание LUT-таблицы для умножителя Галуа по модулю 31

q_n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i_n	0	24	1	18	20	25	28	12	2	14	23	19	11	22	21
q_n	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
i_n	6	7	26	4	8	29	17	27	13	10	5	3	16	9	15

Рис. 5. Схема работы 5-битного умножителя Галуа



собой таблицы, переводящие умножаемые числа в степенное представление по табл. 1, а в качестве сумматора выступает простой модулярный сумматор, изображённый на рис. 2 (а). LUT3 выполняет сложение по модулю 30, а LUT4 переводит результат из степенного представления обратно в первоначальный. LUT4 представляет собой табл. 1, только отсортированную по i_n. На рис. 5 ADDR на входе таблицы и [ADDR] на выходе показывают, что значение, поступившее на вход таблицы, рассматривается в качестве линейного адреса элемента, который будет выдан на выход таблицы, т.е. [ADDR] – это содержимое ячейки таблицы по адресу ADDR.

Работа выполнена по гранту *A04-2.8-755* Федерального агентства по образованию.

Литература

1. NETO J.P., SIEGELMANN H.T., COSTA J.F., ARAUJO C.P.S. Turing Universality of Neural Nets (revisited). // Lecture Notes in Computer Science – 1333, Springer-Verlag, 1997. pp. 361-366.
2. ЧЕРВЯКОВ Н.И., САХНЮК П.А., ШАПОШНИКОВ А.В., РЯДНОВ С.А. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Под ред. Н.И.Червякова. –М.: Физматлит, 2003. – 288 с.
3. ЧЕРВЯКОВ Н.И., САХНЮК П.А., ШАПОШНИКОВ А.В., МАКОХА А.Н. Нейрокомпьютеры в остаточных классах. Учебное пособие для вузов (научная серия «Нейрокомпьютеры и их применение, ред. А.И.Галушкин, кн. 11). –М.: Радиотехника, 2003. – 272 с.
4. АКУШСКИЙ И.Я., ЮДИЦКИЙ Д.И. Машинная арифметика в остаточных классах. –М.: Советское радио, 1968.
5. НОДЕН П., КИТТЕ К. Алгебраическая алгоритмика (с упражнениями и решениями): Пер. с франц. – М.: Мир, 1999. – 720 с.
6. GAUSS C.F. Disquisitiones Arithmeticae. Yale University Press, New Haven, 1966.
7. СТЕМПКОВСКИЙ А.Л., КОРНИЛОВ А.И., СЕМЁНОВ М.Ю. Особенности реализации устройств цифровой обработки сигналов в интегральном исполнении с применением модулярной арифметики. // *Информационные технологии*, №2, 2004. С. 2–9.
8. BAYOUMI M.A., JULLIEN G.A., MILLER W.C. A VLSI Implementation of Residue Adders, // *IEEE Transactions on Circuits and Systems*, vol. CAS-34, # 3, 1987.
9. LAKHANI G. Some Fast Residual Arithmetic Adders, // *International Journal of Electronics*, vol. 77, #2, 1994. pp. 225–240.

10. DUGDALE M. VLSI Implementation of Residue Adders Based on Binary adders, // *IEEE Trans. on Circuits and Systems II*, vol. 39, #5, 1992. pp. 325–329.
11. TAYLOR F. Large Moduli Multipliers for Signal Processing, // *IEEE Transactions on Circuits and Systems*, vol. CAS-28, #7, 1981. pp. 731–736.
12. JULLIEN G.A. Implementation of Multiplication, Modulo a Prime Number, with Applications to Number Theoretic Transforms, // *IEEE Transactions on Computer*, vol. C-29, #10, 1980. pp. 899–905.
13. RADHAKRISHNAN D., YUAN Y. Fast and Highly Compact RNS Multipliers, // *International Journal of Electronics*, vol. 70, #2, 1991. pp. 281–293.
14. KRISHNA H., KRISHNA B., LIN K.Y., SUN J.D. Computational Number Theory and Digital Signal Processing. Fast Algorithms and Error Control Techniques. – CRC Press, 1994.
15. KRISHNA H. Digital Signal Processing Algorithms, Number Theory, Convolution, Fast Fourier Transforms, and Applications. – CRC Press, 1998.

УДК. 681.3

И. В. Коряков

Защищенная передача сигналов на основе модулярного преобразования

(ООО НВФ «Криптон»)

Рассматривается оригинальный класс устройств защиты речевых и речеподобных сигналов, на основе *модулярного преобразования* сигнала. Такое преобразование совмещает процесс модуляции и шифрования. Модуляционное созвездие формируется путем выполнения операций по модулю над цифровыми отсчетами сигнала и отсчетами гаммы.

Метод

Идея модулярного преобразования достаточно проста: если отсчеты исходного сигнала *s* представлены числами, то эти числа могут быть подвергнуты шифрованию и преобразованы в отсчеты зашифрованного сигнала *v*; восстановление подразумевает взятие отсчетов, расшифрование и преобразование в исходные отсчеты *s^ = s + ds* с некоторой ошибкой *ds*. Степень защиты при этом соответствует уровню алгоритмов шифрования данных.

Трудность прямого решения данной задачи заключается в том, что необходимо при шифровании отсчетов сохранять полосу исходного сигнала без “просачивания” исходной статистики в результат преобразования и, главное, восстанавливать сигнал при

наличии шума и/или необратимых искажений в закрытом сигнале так, чтобы мощность шума существенно не увеличивалась при восстановлении сигнала. Другими словами, преобразование должно обладать свойством непрерывности в следующем смысле: при некотором уменьшающемся приращении δv значения v отсчета закрытого сигнала, восстановленный отсчет $s\hat{}$ также должен иметь уменьшающееся приращение δs.

Этим требованиям удовлетворяет преобразование, в котором каждый комплексный отсчет сигнала **s** суммируется по модулю D с отсчетом псевдослучайной последовательности **g** (гаммы) с равномерным на квадрате $D \times D$ распределением. Статистика результирующего сигнала **v** соответствует статистике гаммы и не содержит статистики исходного сигнала [1].

Формирование комплексного отсчета **v** из сигнала **s** и гаммы **g**, когда каждый из отсчетов представлен двумя целыми d-разрядными числами, осуществляется в соответствии с выражениями:

$$\begin{aligned} v_x &= (s_x + g_x) \bmod 2^d \\ v_y &= (s_y + g_y) \bmod 2^d \end{aligned} \qquad (1)$$

Удобно выбирать d, равное длине машинного слова, например, в конкретной реализации d=32. Тогда сложение по модулю определяется как сложение целых без учета переполнения. Причем, будут числа представлены как целые без знака или целые со знаком в дополнительном коде – не имеет значения, поскольку коды, получаемые в результате операций, идентичны. Восстановление сигнала **s^** производится сложением **v** с дополнением **g** до 2^d по модулю 2^d:

$$\begin{aligned} s_x^{\wedge} &= (v_x + (2^d - g_x)) \bmod 2^d \\ s_y^{\wedge} &= (v_y + (2^d - g_y)) \bmod 2^d \end{aligned} \qquad (2)$$

Свойства последовательности бит b_i, из которой формируются отсчеты g_k гаммы как

$$g_k = \sum_{j=0}^{d-1} b_{kd+j} 2^{d-j-1}$$

должны соответствовать свойствам случайной равновероятной последовательности нулей и единиц. Тогда статистика последовательности **v** будет неотличима от статистики **g**, а проникновение статистики **s** в **v** полностью исключено. Отсчеты **g** и **v** имеют равномерное распределение плотности вероятности на квадрате $\{[0, 2^d\text{-}1], [0, 2^d\text{-}1]\}$ или для целых со знаком на квадрате $\{[-2^{d-1}, 2^{d-1}\text{-}1], [-2^{d-1}, 2^{d-1}\text{-}1]\}$.

Сигнал модулярного преобразователя в комплексном виде представляет собой квадратное созвездие с большим числом равноотстоящих точек (2^{2d}) с равномерной плотностью распределения вероятности, обеспечиваемой генератором гаммы. Для генерации g_k возможно применение любой *работоспособной* криптографической функции [3].

Структура

На рисунке 1 представлена упрощенная структура передающей и приемной части модулярного преобразователя. Отсчеты *s(n)* исходного сигнала *s(t)* при помощи преобразователя Гильберта H переводятся в комплексные отсчеты и переносятся к нулю частот (обозначим $\mathbf{s}_0(n)$) умножением на сигнал условной несущей частоты $\boldsymbol{f}_{cs}$ такой, чтобы полоса сигнала была симметричной относительно нуля частоты и, естественно, меньше половины частоты дискретизации $\boldsymbol{f}_s$.

Отсчет $\mathbf{s}_0(n)$ выражается в виде пары целых чисел длиной *d* двоичных разрядов. Числа шифруются сложением в соответствие с (1) с элементами гаммы **g***(n)* также длиной *d* двоичных разрядов, выражаются в виде отсчетов, пропущенных через формирующий фильтр Найквиста F, и переносятся на несущую частоту $\boldsymbol{f}_v$ (не обязательно $\boldsymbol{f}_{cs} = \boldsymbol{f}_v$) для получения действительных отсчетов *v(n)* и, соответственно, сигнала *v(t)*. На приемной стороне обеспечивается строгая синфазность дискретизации сигнала *v(t)* по отношению к передатчику, перенос на нулевую несущую от частоты $\boldsymbol{f}_v$, компенсация линейных искажений в канале при помощи фазового корректора. Комплексные отсчеты выхода корректора представляются *d*-разрядными числами, суммируются в соответствии с (2) с элементами гаммы приемника **g***(n)*, интерполируются фильтром F и переносятся в действительную область на частоту $\boldsymbol{f}_{cs}$ для получения отсчетов *s^(n)* и восстановления сигнала *s^(t)*. На рисунке 2 приведен вид сигналов

в частотной области на этапах прямого модулярного преобразования.

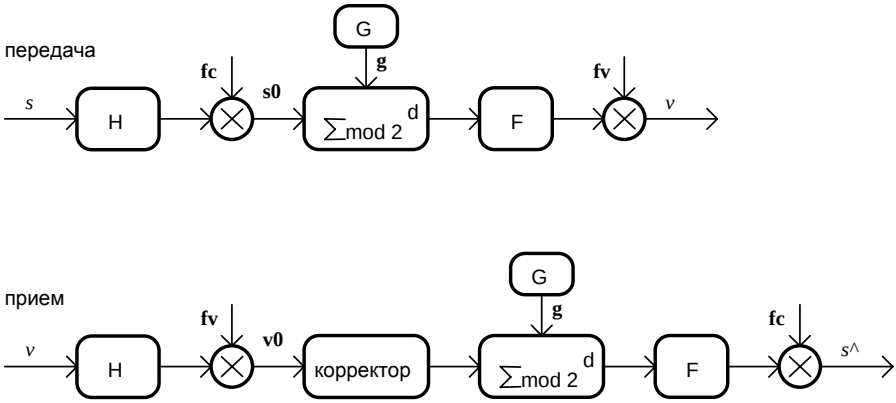


Рис. 1 – Структура модулярного преобразователя

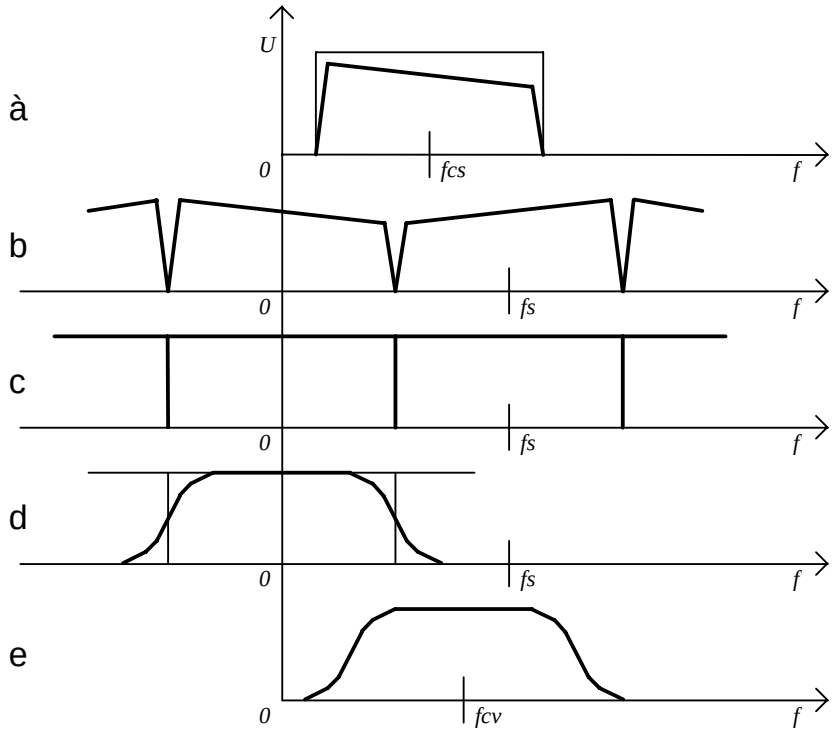


Рис. 2 – Формирование закрытого сигнала

На рисунке 2 этапы формирования сигналов обозначены буквами:

a. Спектр исходного действительного сигнала *s*, ограниченный фильтром с частотной характеристикой, близкой к прямоугольной и шириной полосы, равной частоте дискретизации f_s комплексного сигнала. Условная несущая f_{cs} соответствует середине спектра сигнала *s*.

b. Спектр комплексного сигнала **s**, децимированного к частоте дискретизации f_s на нулевой несущей.

c. Спектр комплексной гаммы **g** (и результата суммирования по модулю).

d. Комплексный сигнал, полученный в результате интерполяции к частоте дискретизации $4f_s$ фильтром Найквиста.

e. Действительный сигнал *v* после переноса на несущую f_{cv}.

На рисунке 3 приведен вид сигналов в частотной области на этапах обратного модулярного преобразования:

a. Спектр принятого действительного сигнала *v*.

b. Спектр комплексного сигнала **v**, децимированного к частоте дискретизации f_s после переноса от несущей f_{cv} к нулю.

c. Спектр комплексной гаммы **g** (и результата суммирования по модулю).

d. Комплексный сигнал **s^**, полученный в результате интерполяции к частоте дискретизации $4f_s$ фильтром с частотной характеристикой, близкой к прямоугольной.

e. Восстановленный действительный сигнал *s^* после переноса на условную несущую f_{cs}.

Особенности реализации

В практической реализации устройства защиты на основе модулярного преобразования для передачи сигналов в стандартном телефонном канале структура рисунка 1 дополняется адаптивным подавителем сигналов эхо и системой синхронизации по несущей f_{cv} и по тактовой частоте f_s.

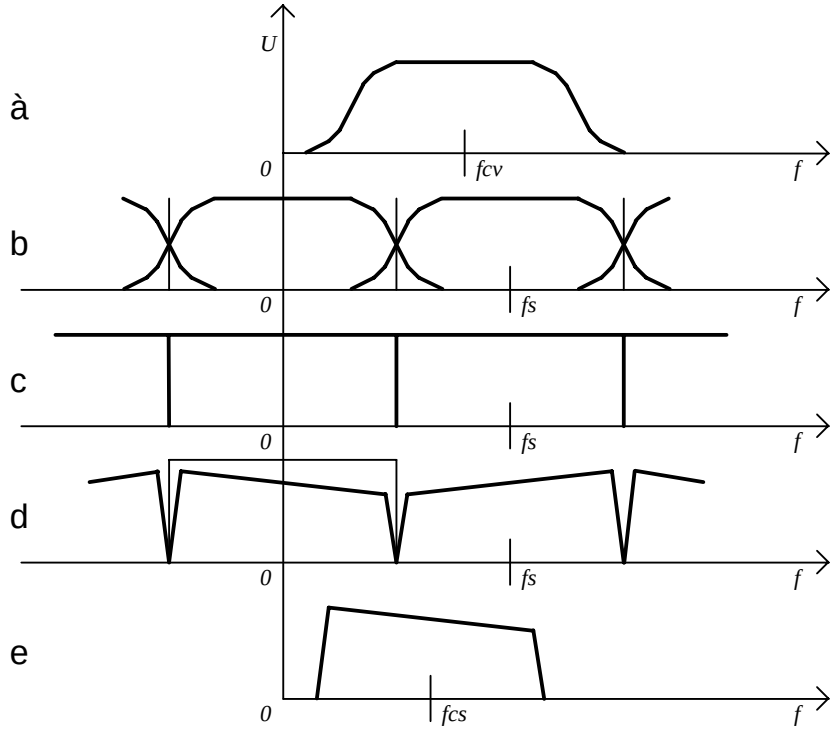


Рис. 3 – Восстановление исходного сигнала

Для реализации функций адаптации и синхронизации в структуру сигнала модулярного преобразователя включаются так называемые реперные точки, представляющие собой элементы созвездия (например CCITT-81380) в последовательности, известной обеим сторонам. В конкретной реализации одна реперная точка включается после каждых 18 отсчетов сигнала **v**. На приемной стороне реперные точки исключаются.

Фазовый корректор в составе модулярного приемника может быть построен так же, как и в обычных модемах передачи данных, за исключением одной уникальной особенности. Типовой корректор модема с решающей обратной связью [2] содержит инверсный фильтр ИФ, на вход которого поступает принимаемый сигнал, а из выходного сигнала вычитается реакция прямого фильтра ПФ, на вход которого поступает решение (идеальные координаты точки

созвездия, по которой было принято решение на предыдущем шаге).

На вход прямого фильтра корректора модулярного приемника поступают комплексные отсчеты гаммы и, если в модеме текущее и все будущие решения неизвестны, то в модулярном приемнике гамма известна в отрицательном и в положительном времени. Прямой фильтр модема, являясь каузальной системой, не может полностью компенсировать искажения, связанные с потерями в спектре принимаемого сигнала, в модулярном приемнике такие потери могут быть скомпенсированы полностью.

На рисунке 4 приведена упрощенная структура корректора модулярного приемника с предсказанием ошибки по гамме.

Модулярный приемник, помимо обычного дробно-интервального адаптивного корректора ИФ, содержит предсказатель ПФ, на вход которого поступают отсчеты гаммы **g**. Сигнал ошибки **e**, по которой производится адаптация, определяется как модулярная разность выхода корректора **y** и предсказателя **eg** в соответствие с (2).

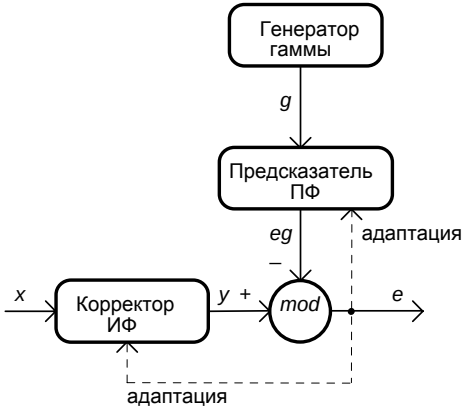


Рис. 4 – Упрощенная структура корректора с предсказанием по гамме

Корректор ИФ определяется фильтром:

$$y(n) = \sum_{i=0}^{N-1} w(i)x(n-i)\,,$$

коэффициенты $\mathbf{w}_i$ которого адаптируются на m-том шаге в соответствии с формулой:

$$w_i = w_i + u\_e^* x(m-i),$$

где u_c - коэффициент адаптации корректора; **e*** означает комплексно сопряженный сигнал ошибки. Для конкретной реализации дробно-интервального корректора $m=2n$.

Предсказатель ПФ определяется фильтром:

$$eg(m) = \sum_{i=0}^{M-1} b(i)e(m-i),$$

коэффициенты b_i которого адаптируются на m-том шаге в соответствии с формулой:

$$b_i = b_i + u_p e^*(m-i),$$

где u_p – коэффициент адаптации предсказателя.

Реальные значения длины фильтров N и M определяются требованиями к допустимой величине ошибки и производительностью процессора DSP. Ориентировочно, для дробно-интервального корректора с кратностью 2 и частотой дискретизации 4800 Гц N может составлять 64 комплексных отсчета, а M – 16 комплексных отсчетов с частотой дискретизации 2400 Гц.

Для обеспечения устойчивости и заданной скорости сходимости адаптивных фильтров ИФ и ПФ требуется нормирование энергии сигналов на входах фильтров.

Коэффициенты адаптации u_c и u_p могут быть неодинаковыми и изменяться во времени в зависимости от режима приемника. Например, в процессе приема сигнала настройки корректора u_c = 0.01 и u_p = 0.01, в процессе приема данных – u_c = 0.005 и u_p = 0.001.

Перед периодом адаптации по настроечной последовательности (train) производится оценка мощности p_c входного сигнала несущей, которая затем используется для вычисления коэффициента адаптации, и отклонения нормирующего коэффициента от единицы. На последующих этапах обработки входной сигнал умножается на множитель k_0 :

$$k_0 = \frac{1}{\sqrt{p_c}}$$

Оценка p_c считается единичной мощностью, относительно которой рассчитывается u_c как

$$u_c = \frac{u_{c0}}{pk_0^2},$$

где u_{c0} имеет типовое значение 0.003.

В рабочем режиме сигнал ошибки **e** представляет собой не что иное, как комплексный восстановленный сигнал **s^**.

Для преобразования **s^** в действительный сигнал осуществляется кратное повышение частоты дискретизации, интерполяция фильтром НЧ и возврат спектра сигнала на условную несущую f_{cs}:

Для предотвращения “перескоков” отсчетов восстановленного сигнала на величину порядка 2^d при определенных сочетаниях значений отсчетов исходного сигнала, гаммы и ошибки амплитуда исходного сигнала должна быть ограничена так, чтобы сумма исходного сигнала и максимального значения ошибки не превышала 2^d. Это достигается умножением отсчетов исходного сигнала на множитель $q<1$ и умножением восстановленных отсчетов на величину $1/q$. Типовое значение q составляет 0.8 для уровня ошибки –24 дБ.

Существенное уменьшение шума в паузах достигается применением кодирования отсчетов исходного сигнала по логарифмическому закону перед выполнением суммирования (1) и потенцирования восстановленных отсчетов после вычитания (2). В практической реализации выполнялось преобразование отсчетов в соответствии с μ-законом компандирования при μ=64. Такое преобразование при практически незаметном возрастании ошибки на больших уровнях сигнала уменьшает шум в паузах на 12 - 15 дБ.

Помимо описанных функций, модулярный преобразователь должен исполнять некоторый протокол начала и окончания сеанса связи, поддержания неразрывности сеанса связи, ретренинг при существенном изменении характеристик канала, передачу служебной информации, в общем, ряд обычных функций модема,

требования к реализации которых могут быть заимствованы из соответствующих рекомендаций ITU-T или разработаны специально для конкретных каналов связи.

Выводы

Особым свойством модулярного преобразования является трансформация искажений любого рода в канале передачи в шум, близкий к нормальному. Уровень шума не зависит от амплитуды передаваемого речевого сигнала и пропорционален приведенной ошибке (в модемах - ошибка принятия решения). Ошибка при восстановлении сигнала модулярного преобразования определяется продуктами нелинейных преобразований в тракте передачи, энергетическими потерями в полосе сигнала, ошибкой корректора из-за ограниченной длины импульсной характеристики фильтра, ошибкой подавителя сигнала эхо, вычислительными ошибками и, наконец, уровнем мощности шума в канале.

Достоинством модулярного преобразования, по сравнению с вокодерными системами, является передача “чистой” речи, т. е. не подвергнутой сжатию. Показатели разборчивости, естественности и узнаваемости у модулярных аппаратов защищенной связи наивысшие среди рассматриваемых классов. Характерные вокодерные искажения речевого сигнала заменяются при модулярном преобразовании шумовым сигналом, к которому в наилучшей степени адаптируется слух человека. Сигнал *s(t)* может иметь произвольную природу, в частности, это может быть факсимильный или модемный сигнал.

Метод защиты сигналов на основе модулярного преобразования реализован как один из режимов в изделиях «КРИПТОН-4М7» и «СЕКМОД-К» предприятия ООО НВФ «Криптон». Реализация метода требует около 10 MIPS для 32-х разрядного процессора цифровой обработки сигналов. Испытания изделий показали высокую степень защиты и качество передачи речевых сигналов, а также сигналов передачи данных и факсимильных сообщений, для скоростей передачи до 9600 бит/с и каналов связи с приведенной ошибкой до -24 дБ (ошибка принятия решения, вклад в которую вносит шум, нелинейность и дисперсия всего тракта передачи сигнала, а также ошибки метода и конечной длины и разрядности вычислений). Требования к каналу для передачи модулярного

сигнала примерно соответствуют требованиям к модемной передаче данных со скоростью 9600 - 12000 бит/с (хорошее качество достигается на подавляющем большинстве соединений городских АТС).

Аудиторные испытания, которым подвергались изделия, проводились по методу мнений [4] (методика, близкая к определению показателя MOS) и подтвердили высокое качество речевой связи с использованием защиты на основе модулярного преобразования. Показатель был равен 3.9 для открытого канала связи и 3.7 для закрытого модулярным преобразованием. Для сравнения укажем, что для защищенного вокодерного канала с алгоритмом сжатия речи класса CELP, реализованным в составе этих же изделий, со скоростью передачи 9600, 4800 и 2400 бит/с значения показателя MOS составили соответственно 3.3, 2.4 и 2.1.

Наиболее эффективное ожидаемое применение криптографически защищенной передачи сигналов на основе модулярного преобразования предполагается на цифровых каналах связи при повышенных требованиях абонентов к качественным характеристикам передачи речи, таким, как естественность звучания речи и узнаваемость диктора.

Литература

1. Кнут. Д. Искусство программирования для ЭВМ. Т.2. Получисленные алгоритмы. Пер. с англ. / Под ред. К.И. Бабенко. - М.: Мир, 1977.
2. Куреши Ш. Адаптивная коррекция. ТИИЭР, 1985, т. 73, №5, с. 5-49.
3. Feistel H. Cryptography and computer privacy. Sci. Amer. 228, 5 (May 1973), 15-23.
4. В. Вемян. Передача речи по сетям электросвязи. М: Радио и связь, 1985.

УДК 681.391

И. В. Коряков

Метод измерения частоты сигнала на основе системы остаточных классов

(ООО НВФ «Криптон»)

Предложен метод измерения частоты, основанный на системе остаточных классов (СОК). Определено представление в СОК частоты как комплексных, так и действительных сигналов. Предложенный метод обработки сигналов в СОК позволяет уменьшить число каналов обнаружения и производить обработку в низкочастотной области спектра, что существенно снижает требования к быстродействию элементов и уменьшает вычислительные затраты.

Приведен пример реализации приемника импульсных сигналов, формируемых многими источниками с неизвестными частотами

Введение

При анализе сигналов со скачкообразным изменением частоты, а также импульсных сигналов, формируемых несколькими источниками с неизвестными частотами, требуется измерять частоту относительно кратковременных узкополосных сигналов в широком диапазоне частот.

Обычно, при элементе разрешения по частоте Δf и полосе анализа $n\Delta f$ приходится использовать n одновременно работающих каналов

обнаружения, что обуславливает высокую сложность и стоимость аппаратуры. Любые методы последовательного анализа, уменьшающие число каналов измерения, в данной постановке задачи неприемлемы, так как значительно снижают вероятность обнаружения сигнала.

Предлагается метод измерения частоты, основанный на системе остаточных классов, позволяющий для n, представленного произведением ряда сомножителей, уменьшить число каналов обнаружения до значения суммы этих сомножителей.

Метод

В системе остаточных классов (СОК) числа представляются остатками от деления на взаимно простые числа $n_1,n_2,\ldots,n_r$ (основания СОК). Диапазон представимых чисел равен произведению оснований $n = n_1n_2\ldots n_r$ [1, 2].

СОК-представление $\{a_i\}$ числа A определяется как

$$a_i = A - \left\lfloor \frac{A}{n_i} \right\rfloor \cdot n_i, \quad i = 1,2,\ldots, r \qquad (1)$$

При этом ряд арифметических операций над A заменяется эквивалентными операциями над $\{a_i\}$. Обычно наиболее важным свойством СОК считают независимость операций с остатками для каждого значения i (отсутствие переносов), что позволяет достичь высокого быстродействия за счет распараллеливания вычислений.

Однако, это не единственное положительное свойство СОК. Для измерительных систем решающим фактором может стать то, что измеряемая величина с большим диапазоном значений порядка n может быть однозначно представлена рядом величин с существенно малыми значениями порядка n_i .

Рассмотрим преобразование сигнала, которое эквивалентно вычислению остатка в частотной области. Пусть сигнал $x(t)$ представляет собой комплексное гармоническое колебание с амплитудой U. Тогда при дискретизации $x(t)$ с периодом T_s будет сформирован сигнал $x_s(t)$:

$$x_s(t) = \sum_{k=-\infty}^{\infty} U e^{j2\pi f k T_s} \delta(t - kT_s), \qquad (2)$$

для которого Фурье-образ представляет периодическую последовательность:

$$X_s(f) = \frac{U}{T_s} \sum_{k=-\infty}^{\infty} \delta(f - kf_s). \qquad (3)$$

При любом значении f всегда будет существовать составляющая (часто называемая «отражением») в диапазоне частот [0, f_s) с частотой f_a, определяемой как

$$f_a = f - \left\lfloor \frac{f}{f_s} \right\rfloor f_s. \qquad (4)$$

Нетрудно видеть, что частота «отражения» является остатком от деления частоты исходного сигнала на частоту дискретизации. На рисунке 1 показан вид сигналов $x(t)$ и $x_s(t)$ в частотной области.

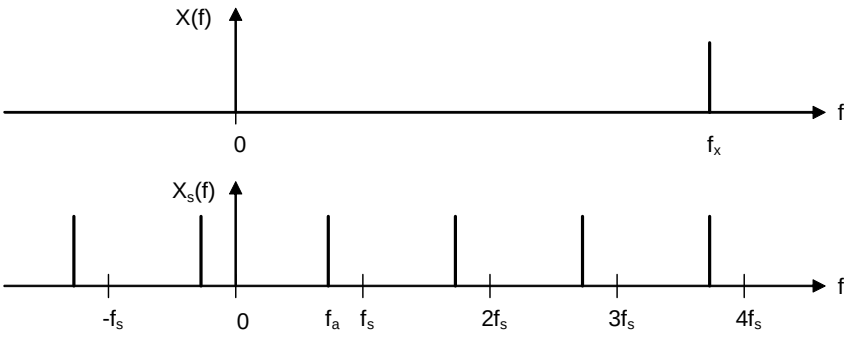


Рис. 1 – Вид сигнала до и после дискретизации

Заметим, что дискретизация производится с нарушением теоремы отсчетов (без антиалиазального фильтра), что, собственно, и является условием получения f_a как остатка от деления f на f_s.

Таким образом, для того, чтобы представить значение частоты f комплексного сигнала $x(t)$ в диапазоне значений $n\Delta f$ с разрешением Δf требуется выполнить дискретизацию r раз с частотами $n_i\Delta f$, за-

тем в диапазоне частот [0, f_s) вычислить модули коэффициентов Фурье и определить номера максимальных модулей в каждом канале. Номера позиций дают СОК-представление $\{a_i\}$ значения частоты f сигнала $x(t)$.

Ниже приведен простейший пример соответствия частоты f и СОК-представления $\{a_i\}$ для Δf =1, n_1 =3, n_2 =5:

f	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a_1	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
a_2	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4

Применение БПФ для вычисления модулей коэффициентов Фурье неэффективно из-за относительно малых и некратных степени 2 значений n_i. Отсчеты модуля $Z_i(k)$ удобно вычисляются через ДПФ, причем, для достижения компромисса между разрешением по частоте и разрешением по времени, интервал анализа устанавливается равным $1/\Delta f$ (число отсчетов для каждого i при этом в точности равно n_i):

$$Z_i(k) = \left\| \sum_{q=0}^{n_i-1} a_i(q) e^{-j(2\pi/n_i)qk} \right\|. \qquad (5)$$

Приведенный метод измерения частоты имеет два существенных недостатка:

1. Входной сигнал $x(t)$ должен быть комплексным, что усложняет реализацию метода.

2. Неопределенность, возникающая при более сложном составе сигнала $x(t)$. Например, если сигнал представляет собой сумму M гармонических колебаний, и если их амплитуды разнятся на величину, сравнимую с приведенной ошибкой измерения амплитуд в тракте приема и преобразования сигнала, то в каждом канале дискретизации возникнет M отраженных составляющих и число ложно обнаруженных частот составит $M^r - M$.

Далее будут рассмотрены усовершенствования метода, устраняющие или уменьшающие степень указанных недостатков.

Действительный сигнал

Рассмотрим дискретизацию действительного сигнала $v(t) = U\cos(2\pi ft)$ с периодом T_s:

$$v_s(t) = \sum_{k=-\infty}^{\infty} U\cos(2\pi fkT_s)\delta(t - kT_s), \qquad (6)$$

для которого «отражение» с частотой f_b определяется в половинном диапазоне частот $[0, f_s/2)$ как

$$f_b = \left| f_v - \left\lfloor \frac{f_v + 0.5 f_s}{f_s} \right\rfloor f_s \right|. \qquad (7)$$

На рисунке 2 показан характер отражений для действительного сигнала.

В чистом виде (1) остаток для действительного сигнала непредставим, но может быть определен некий аналог остатка b для числа B (назовем «отраженный остаток») как

$$b_i = \left| B - \left\lfloor \frac{B + n_i + 0.5}{2n_i} \right\rfloor \cdot 2n_i \right|, \quad i = 1,2,...,r \qquad (8)$$

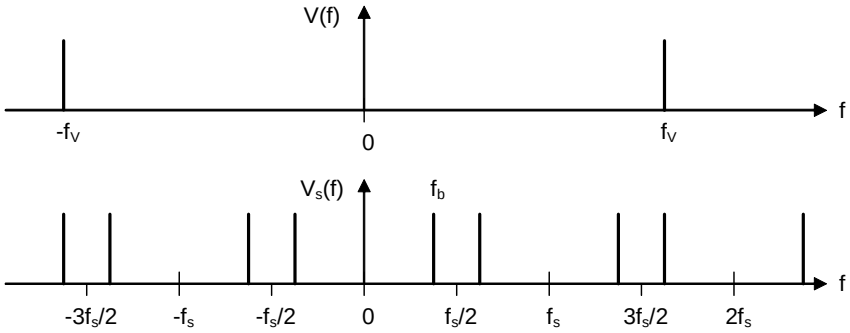


Рис. 2 – Отражения при дискретизации действительного сигнала

На рисунке 3 изображен пример графика функции (7) для $n_i = 5$ с отмеченными отсчетными значениями b, соответствующими формуле (8).

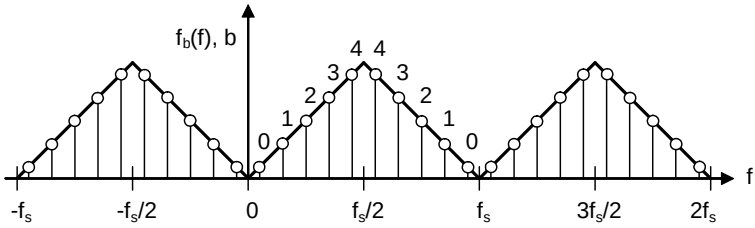


Рис. 3 – «Отражение» по частоте при дискретизации действительного сигнала и «отраженный остаток»

Чтобы представить значение частоты f действительного сигнала $v(t)$ в диапазоне значений $n\Delta f$ с разрешением Δf требуется выполнить дискретизацию r раз с частотами $2n_i\Delta f$, затем в диапазоне частот $[0, f_s/2)$ вычислить модули коэффициентов Фурье и определить номера максимальных модулей в каждом канале. Номера позиций дают представление $\{b_i\}$ значения частоты f.

Ниже приведено, аналогично предыдущему примеру, соответствие частоты f и представления $\{b_i\}$ для Δf =1, n_1 =3, n_2 =5:

f	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
b_1	0	1	2	2	1	0	0	1	2	2	1	0	0	1	2
b_2	0	1	2	3	4	4	3	2	1	0	0	1	2	3	4

Как видим, представление $\{b_i\}$ однозначно, так же, как и представление $\{a_i\}$, изменен лишь порядок следования значений $\{b_i\}$.

Вычисление ДПФ для случая действительного сигнала имеет особенность (см. рисунок 3): отсчеты в частотной области берутся не в точках $(2\pi/n_i)qk$, а в точках, смещенных на $\Delta f/2$:

$$Z_i(k) = \left\| \sum_{q=0}^{n_i-1} a_i(q) e^{-j(\pi/n_i(2qk+1))} \right\|. \qquad (9)$$

Каждый «отраженный остаток» b_i принимает значение индекса k, соответствующего максимальному значению $Z_i(k)$.

Для дальнейшей обработки результатов измерений частоты значения $\{b_i\}$ обычно преобразуют в двоичный код. С целью получения

максимального быстродействия наиболее эффективно выполнять это преобразование табличным способом.

Неопределенность и области применения

Указанный выше, второй недостаток метода может быть устранен путем увеличения числа приемников и принятием совместного решения по результатам измерений каждым из них. Если приемники будут иметь одинаковые базисы, то выигрыша не будет. Для получения наибольшего выигрыша, то есть, наименьшей вероятности совпадения результатов при различных комбинациях входных частот, требуется, чтобы все основания всех приемников были взаимно простыми. Другими словами, при увеличении r в g раз можно либо в r^g раз увеличить разрешение (уменьшить Δf), либо в g раз увеличить число одновременно принимаемых сигналов с различными частотами.

Рассмотрим иллюстративный пример. Пусть g = 2 и $\{n_{i,g}\}$ = 4, 5; 3,7. Входные сигналы v_1 и v_2 имеют частоты f_1 =5 и f_2 =9, соответственно. Ниже приведены значения $b_{i,g}$ и показаны результаты измерений каждого приемника.

Первый приемник дает, помимо правильного результата 5 и 9, ложные частоты 10 и 14, а второй – 8 и 18. Правильное решение определяется по совпадению измерений в обоих приемниках.

f	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
b 1 1	0	1	2	3	3	2	1	0	0	1	2	3	3	2	1	0	0	1	2	3	
b 1 2	0	1	2	3	4	4	3	2	1	0	0	1	2	3	4	4	3	2	1	0	
b 2 1	0	1	2	2	1	0	0	1	2	2	1	0	0	1	2	2	1	0	0	1	2
b 2 2	0	1	2	3	4	5	6	6	5	4	3	2	1	0	0	1	2	3	4	5	6
b 1 1						2				1	2				1						
b 1 2						4				0	0				4						
b 2 1						0			2	2									0		
b 2 2						5			5	4									4		

Следует указать на то, что в каждом отдельном канале дискретизации отношение сигнал/шум будет уменьшаться из-за n/n_i-кратного

наложения сигналов в частотной области. С другой стороны, возможно, это уменьшение может быть скомпенсировано за счет некогерентности шума и когерентности сигнала, а также за счет совместного принятия решения по данным нескольких каналов. Вопрос требует специального изучения и в данной работе не рассматривается.

Специфика обработки сигналов в приемнике на основе СОК накладывает определенные ограничения на его области применения. Прежде всего, это широкополосные системы измерения частоты быстро перестраивающихся сигналов типа ППРЧ, либо импульсных сигналов от многих источников с различными частотами, но существующих кратковременно.

Пример технической реализации

Исходные данные:

– диапазон частот – 0..3000 МГц;

– разрешение по частоте – 1 МГц;

– разрешение по времени – 1 мкс.

Выбранные параметры:

– два приемника (g = 2) с базисами n_1^1 = 11, n_2^1 = 15, n_3^1 = 19 (n^1 = 3135) и n_1^2 = 13, n_2^2 = 14, n_3^2 = 17 (n^2 = 3094);

– сигнал действительный, частоты дискретизации соответственно для первого приемника f_1^1 = 22 МГц, f_2^1 = 30 МГц, f_3^1 = 38 МГц и для второго f_1^2 = 26 МГц, f_2^2 = 28 МГц, f_3^2 = 34 МГц.

На рисунке 4 приведена структура технического решения для двухканального приемника импульсных сигналов [3].

Приемник содержит антенну, фильтр ФНЧ с частотой среза 3000 МГц, необходимый для предотвращения попадания на вход приемника сигналов с частотами, большими $n\Delta f$, широкополосный малошумящий усилитель МШУ, выход которого поступает на две группы схем выборки/хранения СВХ, выходы которых подключены к входам аналого-цифровых преобразователей АЦП. Два синте-

затора тактовых частот СТЧ генерируют требуемые частоты для СВХ и АЦП соответствующих каналов f_1^1, f_2^1, f_3^1 и f_1^2, f_2^2, f_3^2.

Последовательности выходных кодов АЦП накапливаются и обрабатываются в блоках дискретного преобразования Фурье ДПФ. Коэффициенты Фурье поступают в узлы обнаружителей УО, где преобразуются в набор кодов частоты, из которых в блоке принятия решения ПР формируется решение о частоте одного или двух сигналов.

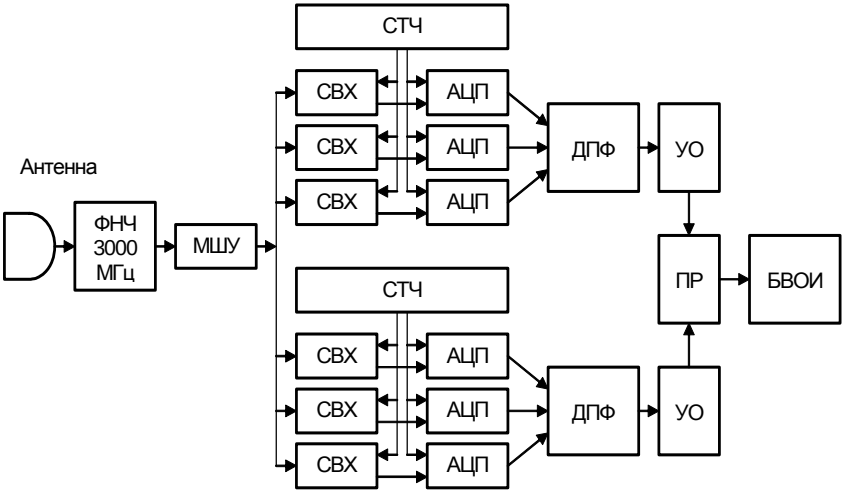


Рис. 4 – Структура двухканального приемника импульсных сигналов

Выходные коды частот от ПР поступают в блок вторичной обработки и отображения информации БВОИ, где преобразуются в отображаемую координатную, параметрическую и признаковую информацию.

Все узлы в приведенном примере могут быть реализованы на современной элементной базе. Процессор ДПФ, узлы УО, ПР удобно реализовать на ПЛИС, содержащих большое количество умножителей с накопителями. Особые требования предъявляются к схемам выборки/хранения, апертурное время которых не должно превышать величину $1/n\Delta f$. Современные решения, применяемые в стробоскопических осциллографах, позволяют получить требуемые параметры СВХ.

Выводы

В остаточных классах могут быть представлены частоты как комплексных, так и действительных сигналов.

Для $n = n_1 n_2 \ldots n_r$ представление частоты в СОК позволяет уменьшить число каналов обнаружения до значения $n_1+n_2+\ldots+n_r$.

Обработка сигналов после дискретизации в представлении СОК производится в низкочастотной области спектра, приблизительно равной $n^{1/r-1}$ части общего диапазона анализа, что при реализации приемника существенно снижает требования к быстродействию элементов и уменьшает вычислительные затраты.

Наиболее подходящей областью применения данного метода является построение широкополосных приемников сигналов с неизвестной частотой, для которых вероятность совпадения во времени более g сигналов достаточно мала.

Литература

1. *Кнут. Д.* Искусство программирования для ЭВМ. Т.2. Получисленные алгоритмы. Пер. с англ. / Под ред. К.И. Бабенко. - М.: Мир, 1977.
2. *Акушский И.Я., Юдицкий Д.И.* Машинная арифметика в остаточных классах. М. Сов. Радио, 1968.
3. А.с. СССР №224930, приоритет от 29 окт. 1984 г./ *В. С. Серегин, А. А. Бондаренко, С. В. Довбешко, И. В. Коряков*.

УДК 519.725; 621.3.037.3

А.А. Смирнов

Корреляционный анализ в системе остаточных классов

(*Ставропольский государственный университет*)

В статье рассматривается автокорреляционный анализ функций, представленных в системе остаточных классов. Показано, что автокорреляционная функция сигнала в системе остаточных классов представляет собой разложение исходной автокорреляционной функции по основаниям системы остаточных классов.

In a paper the autocorrelated analysis of functions represented in a system of residual classes is considered. Is shown, that the autocorrelation function of a signal in a system of residual classes represents expansion of initial autocorrelated function on foundations of a system of residual classes.

1. Введение.

Основу корреляционного анализа составляет интегральное исчисление. Корреляционный анализ позволяет оценивать некоторые характеристики как детерминированных, так и случайных сигналов. Так энергетический спектр сигнала связан с его автокорреляционной функцией (АКФ) $R(\tau)$ преобразованием Хинчена-Винера.

В настоящее время математический аппарат корреляционного анализа хорошо развит для сигналов, представленных в позиционной системе счисления (ПСС), т. е. в такой системе, в которой запись

числа осуществляется последовательностью чисел, значность которых зависит от их позиций. Например в числе 135 единица на третьей позиции означает три сотни, тройка на второй позиции означает три десятка и т. д. В данной системе счисления алгебраические операции сложения и умножения осуществляются последовательно от разряда к разряду. Существуют и непозиционные системы счисления, такие как система остаточных классов (СОК) в которой каждое число представляется набором независимых вычетов (α_i) по взаимно-простым основаниям (p_i). При этом указанные алгебраические операции осуществляются параллельно по всем вычетам без переносов из разряда в разряд.

Известно, что любой сигнал $S(t)$ в ПСС может быть представлен в СОК. В этом случае исходный сигнал представляется совокупностью сигналов $\{S_i(t)\}$ по соответствующим основаниям p_i. Очевидно, что АКФ такого сигнала должна представлять собой совокупность АКФ $R_i(\tau)$, соответствующих каждому основанию p_i.

Целью статьи является определение АКФ функции в СОК при определенной АКФ в ПСС.

2. Решение задачи.

Очевидно, что функция, как отношение на множестве чисел в СОК может быть представлена функцией модулярного аргумента. Другими словами, если функция в ПСС представляется в виде $f(x)$, то в СОК по основаниям $\{p_i\}$ та же функция может быть представлена набором функций по отдельному основанию

$$f_{COK}(x) = \{f(x) \bmod p_i\}_L, \qquad (1)$$

где $i = 1...L$ – i-е основание СОК, L – число оснований СОК.

Исходная автокорреляционная функция для $f(t)$ может быть представлена в виде

$$R(\tau) = \int_{-T/2}^{T/2} f(t) f(t-\tau) dt, \qquad (2)$$

где T – период сигнала. В общем случае непериодического сигнала $T \to \pm\infty$. Пусть функция $f(x)$ имеет такую обратную функцию $\psi(x)$, что для любых $x \in Z$ выполняется $\psi(x) \in Z$, тогда опреде-

ленный интеграл этой функции, взятой по модулю p, сравним с определенным интегралом этой функции по тому же модулю для любых целых значений пределов интегрирования. Подставим выражение (1) в (2) и получим

$$R_i(\tau) = \int_{-T/2}^{T/2} [f(t) \bmod p_i][f(t-\tau) \bmod p_i]dt .$$

Откуда согласно теории сравнений

$$R_i(\tau) \equiv \int_{-T/2}^{T/2} [f(t) f(t-\tau)] \bmod p_i dt. \qquad (3)$$

С учетом теоремы о модулярном интегрировании получим

$$R_i(\tau) \equiv \int_{-T/2}^{T/2} [f(t) f(t-\tau)] \bmod p_i dt \equiv$$

$$\left(\int_{-T/2}^{T/2} f(t) f(t-\tau) dt \right) \bmod p_i = R(\tau) \bmod p_i. \qquad (4)$$

Последнее выражение можно сформулировать в виде теоремы.

Теорема 1.

Автокорреляционная функция сигнала $f(x)$, имеющего такую обратную функцию $\psi(x)$, что для любых $x \in Z$ выполняется $\psi(x) \in Z$, представленного в СОК есть суть перевода самой автокорреляционной функции $R(\tau)$ этого сигнала $f(x)$ из ПСС в СОК $\{R_i(\tau)\}_L$ по взаимно простым основаниям p_i, $i = 1...L$.

Следствием теоремы 1 является тот факт, что в случае представления сигнала в СОК он должен характеризоваться несколькими интервалами корреляции τ_{k_i} по каждому основанию p_i

$$\tau_{k_i} = \frac{\int_0^{\infty} [R(\tau) \bmod p_i] d\tau}{R(0)} = \frac{\int_0^{\infty} R(\tau) d\tau}{R(0)} \bmod p_i = \tau_k \bmod p_i \qquad (5)$$

Последнее выражение справедливо только для случая, если АКФ $R(\tau)$ имеет такую обратную функцию $R^{-1}(\tau)$, что для любых

$\tau \in Z$ выполняется $R^{-1}(\tau) \in Z$. Т. е. целое значение функции соответствует целому значению аргумента. Такими свойствами обладает, например, АКФ дискретного сигнала.

3. Выводы.

На основе проведенного анализа можно сделать следующие выводы.

1. Автокорреляционный анализ функций, представленных в СОК может быть сведен к автокорреляционному анализу отдельных компонент набора сигналов по соответствующим основаниям.

2. АКФ сигнала в СОК есть модулярное представление исходной АКФ сигнала, представленного в ПСС.

3. Интервал корреляции сигнала в ПСС определяется корреляционными интервалами каждого из вычетов в СОК относительно исходного сигнала.

Литература

1. Червяков Н.И. и др. Модулярные параллельные вычислительные структуры нейропроцессорных систем. – М.: Физматлит, 2003. – 288 с.
2. Акушский И.Я., Пак И.Т. Машинная арифметика в остаточных классах. – М.: Сов. радио, 1968. – 440 с.
3. Смирнов А. А. Математическое описание сигналов, используемых для передачи данных в параллельном формате // Вестник Ставропольского государственного университета, 2004. Вып. 38. – С. 40 – 45.
4. Смирнов А.А. Применение чисел Мерсена и Ферма в качестве оснований системы остаточных классов в двоичном канале связи // Инфокоммуникации технологии, 2004. № 2. – 64 с.
5. Бернард Скляр. Цифровая связь. Теоретические основы и практическое применение, 2-е издание. – М.: Вильямс, 2003. –1104 с.

УДК. 681.3

В.В. Бережной

Нейросетевая структура для исправления двукратных ошибок в модулярных нейрокомпьютерах

(Ставропольский государственный университет)

Рассмотрен характер возникновения ошибок в системе остаточных классов. Определена минимальная величина избыточного модуля для коррекции двукратной ошибки. Предложен метод локализации и исправления двукратных ошибок и его нейросетевая реализация.

Examined the character of arising mistakes in the residue number system. Determined the minimal size of the superfluous module for correction of a double mistake. Offered the method of localizating and correcting double mistakes and its neural network realization.

Многочисленные исследования, проведенные за последнее десятилетие, убедительно обосновали возможность построения таких параллельных вычислительных систем, в которых за счет специального кодирования может быть создан иммунитет против самых разнообразных искажений несущих информацию сигналов. Совершенно четко сформировалась точка зрения, что борьба за высокую надежность информации, т. е. за достоверность восстановления информации должна вестись не столько совершенствованием технических средств, где любое возможное повышение надежности достигается дорогой ценой и порой требует разработки сложных

защитных мероприятий, сколько применением таких способов кодирования информации, которые были бы устойчивы по отношению к возможным случайным искажениям информации, понимая под этим способность путем соответствующей обработки информации исключить внесенные в нее ошибки.

Одним из альтернативных направлений создания параллельных вычислительных систем является арифметика в остаточных классах, которая позволяет рассматривать вопросы разработки высокопроизводительных отказоустойчивых модулярных нейрокомпьютеров за счет организации внутренней структуры вычислений в остатках.

Система остаточных классов (СОК) [1] обладает специфическими корректирующими свойствами, проявляющимися при введении в нее контрольных (избыточных) модулей (оснований), величина и количество которых определяют возможности СОК по обнаружению и исправлению ошибок при передачи и обработке информации в модулярных нейрокомпьютерах. При введении избыточных оснований необходимо стремиться к уменьшению их величин, так как это в конечном счете приведет к снижению аппаратурных затрат, а в целом к повышению надежности нейрокомпьютера.

При функционировании модулярных нейрокомпьютеров возможны сбои или отказы отдельных функциональных узлов, которые приводят к искажению результатов обработки данных или, проще говоря, появлению ошибок. Характер проявления ошибок, а также вопросы локализации и исправления однократных ошибок рассмотрены в [2]. В данной статье проведем анализ распределения и исправления двукратных ошибок, а также рассмотрим нейросетевую структуру для обнаружения и исправления такого вида ошибок.

Упорядоченная система оснований. Система оснований $p_1, p_2, \ldots, p_n, p_{n+1}$ является упорядоченной, если выполняется условие

$$p_1 < p_2 < \ldots < p_n < p_{n+1}, \qquad (1)$$

где n-число рабочих модулей; $n+1$-величина контрольного модуля.

Для исследования корректирующих свойств кода упорядоченной системы остаточных классов и минимизации величины контрольного основания используем метод анализа распределения ошибок по интервалам диапазона СОК, рассмотренный в [2].

Пусть заданы основания $p_1, p_2, \ldots, p_n, p_{n+1}$ и число $A=(\alpha_1, \alpha_2, \ldots, \alpha_n, \alpha_{n+1})$ в СОК. Тогда перевод числа в позиционную систему счисления можно осуществить через ортогональные базисы:

$$A = \left[\sum_{i=1}^{n+1} \alpha_i B_i \right] (\mathrm{mod}\, P), \qquad (2)$$

где $P=p_1 \cdot p_2 \cdot \ldots \cdot p_n \cdot p_{n+1}$ – полный диапазон СОК; B_i – ортогональный базис, $B_i = m_i P/p_i$; m_i - целое положительное число, называемое весом базиса.

Появление ошибочных цифр $\tilde{\alpha}_i$ на любых основаниях выводит число $\tilde{A}$ из рабочего диапазона

$$P_p = P \big/ p_{n+1}$$

и переносит его в избыточный диапазон.

Проведем анализ распределения ошибок по двум основаниям. Искажение числа A по двум основаниям p_i и p_j при переводе в позиционную систему счисления приводит к следующему результату

$$\tilde{A} = [A + \Delta\alpha_i B_i + \Delta\alpha_j B_j] (mod\ P), \qquad (3)$$

где $\Delta\alpha_i$ -глубина ошибки по модулю p_i; $\Delta\alpha_j$- глубина ошибки по модулю p_j.

Определенные величины ошибок $\Delta\alpha_i$ и $\Delta\alpha_j$ переносят число $\tilde{A}$ в соответствующий интервал полного диапазона (ошибочный интервал) [1], по абсолютному значению равный рабочему диапазону $K_{i,j}^{r,k} = [\overline{P_p - 1}]$, где i, j - номера оснований, $i=[\overline{1,n}]$, $j=[\overline{1,n}]$, $i \neq j$; r- глубина ошибки по модулю p_i, k- глубина ошибки по модулю p_j, $r=[\overline{1, p_i - 1}]$, $k=[\overline{1, p_j - 1}]$. Границы ошибочных интервалов можно определить из (3) ,если в качестве A использовать числа из рабочего диапазона $P_p=[\overline{0, P_p - 1}]$, при этом значения $\Delta\alpha_i$ изменять от 1 до p_{i-1}, а $\Delta\alpha_j$ – от 1 до p_{j-1} $(i \neq j)$.

Пример. Пусть $p_1=2$, $p_2=3$, $p_3=5$ - рабочие основания, $p_4=17$ - контрольное основание, тогда $P_p=30$, $P=510$.

Ортогональные базисы B_i определяются как $B_1=255$, $B_2=340$, $B_3=306$, $B_4=120$. Ошибочные интервалы находятся по формуле (3):

при основаниях $p_1=2$ и $p_2=3$ $K_{1,2}^{1,1}=[85\text{-}114]$, $K_{1,2}^{1,2}=[425\text{-}454]$;

при основаниях $p_1=2$ и $p_3=5$ $K_{1,3}^{1,1}=[51\text{-}80]$, $K_{1,3}^{1,2}=[357\text{-}386]$, $K_{1,3}^{1,3}=[153\text{-}182]$, $K_{1,3}^{1,4}=[459\text{-}488]$;

при основаниях $p_2=3$ и $p_3=5$ $K_{2,3}^{1,1}=[136\text{-}165]$, $K_{2,3}^{1,2}=[442\text{-}471]$, $K_{2,3}^{1,3}=[238\text{-}267]$, $K_{2,3}^{1,4}=[34\text{-}63]$; $K_{2,3}^{2,1}=[476\text{-}505]$, $K_{2,3}^{2,2}=[272\text{-}301]$, $K_{2,3}^{2,3}=[68\text{-}97]$, $K_{2,3}^{2,4}=[374\text{-}403]$.

Из примера видно, что величина избыточного модуля $p_{n+1}=17$ не позволяет однозначно локализовать ошибки ввиду наличия пересечений $K_{2,3}^{1,4}$ и $K_{1,3}^{1,1}$, $K_{1,3}^{1,1}$ и $K_{2,3}^{2,3}$, $K_{2,3}^{2,3}$ и $K_{1,2}^{1,1}$, $K_{2,3}^{1,1}$ и $K_{1,3}^{1,3}$, $K_{1,3}^{1,2}$ и $K_{2,3}^{2,4}$, $K_{1,2}^{1,2}$ и $K_{2,3}^{1,2}$, $K_{2,3}^{1,2}$ и $K_{1,3}^{1,4}$, $K_{1,3}^{1,4}$ и $K_{2,3}^{2,1}$.

Анализ распределения ошибок по интервалам полного диапазона СОК одним избыточным основанием для различных систем оснований, проведенный на ЭВМ по программе, разработанной в соответствии с изложенным методом, позволяет сделать следующие утверждения:

1) минимальная величина контрольного основания для гарантированного определения двукратной ошибки по рабочим модулям должна быть больше произведения двух наибольших рабочих оснований, оставаясь при этом взаимно-простым числом

$$p_{n+1} > p_n p_{n-1}. \qquad (4)$$

2) распределение ошибок внутри избыточного диапазона упорядоченной системы остаточных классов имеет неравномерный симметричный относительно середины избыточного диапазона характер и является функцией от базисов СОК;

3) для осуществления возможности локализации и исправления двойных ошибок по рабочим основаниям необходимым и достаточным условием является непересечение ошибочных интервалов:

$$\forall A \in [K_{i,j}^{r,k}] \neq \forall C \in [K_{r,k}^{c,s}], \quad (5)$$

где $c=1, 2, \ldots, p_r-1; \; s=1, 2, \ldots, p_k-1; \; r=1, 2, \ldots, n; \; k=1, 2, \ldots, n, \; R \neq k;$

4) с увеличением значения контрольного основания "последнее" пересечение ошибочных интервалов наблюдается при ошибках по основаниям p_{n-2}, p_{n-3} и p_{n-1}, p_n.

Опираясь на перечисленные утверждения, можно определить минимальную величину избыточного основания для однозначной локализации и исправления двукратной ошибки по любым двум модулям.

Так как, число из рабочего диапазона СОК, в котором произошла ошибка по двум основаниям, можно представить формулой (3), то на основании утверждения 3 и 4 и выражения (5) запишем

$$[|P_p| + \Delta\alpha_{n-2}B_{n-2} + \Delta\alpha_{n-3}B_{n-3}](mod P) \neq [|P_p| + \Delta\alpha_{n-1}B_{n-1} + \Delta\alpha_n B_n](mod\, P), \quad (6)$$

где $|P_p|$-числа из рабочего диапазона СОК.

На основании утверждения о симметричности распределения ошибочных интервалов вместо всех чисел рабочего диапазона $|P_p|$ ограничимся его первыми и последними числами $[\overline{0, P_p - 1}]$ и проведем замену знака ≠ на >. Тогда выражение (6) преобразуется к виду

$$[0 + \Delta\alpha_{n-2}B_{n-2} + \Delta\alpha_{n-3}B_{n-3}](mod\, P) \neq [P_p - 1 + \Delta\alpha_{n-1}B_{n-1} + \Delta\alpha_n B_n](mod\, P). \quad (7)$$

Представив $P_p = \frac{P}{p_{n+1}}$ и $B_i = \frac{m_i P}{p_i}$ получим:

$$\frac{P_p}{P_{n+1}} < \left[\frac{\Delta\alpha_{n-2}B_{n-2}P}{p_{n-2}} - \frac{\Delta\alpha_{n-3}B_{n-3}P}{p_{n-3}} - \frac{\Delta\alpha_{n-1}B_{n-1}P}{p_{n-1}} + \frac{\Delta\alpha_n B_n P}{p_n} + 1 \right] (mod P). \quad (8)$$

Сократив на величину полного диапазона P, запишем:

$$\frac{1}{P_{n+1}} < \left[\frac{\Delta\alpha_{n-2} B_{n-2} p_n p_{n-1} p_{n-3} + \Delta\alpha_{n-3} B_{n-3} p_n p_{n-1} p_{n-2}}{p_n p_{n-1} p_{n-2} p_{n-3}} \right. \quad (9)$$

$$\left. - \frac{\Delta\alpha_n B_n p_{n-1} p_{n-1} p_{n-3} + \Delta\alpha_{n-1} B_{n-1} p_n p_{n-2} p_{n-3}}{p_n p_{n-1} p_{n-2} p_{n-3}} + 1 \right] (mod P)$$

Учитывая, что $P=P_p p_{n+1}$ и

$|\Delta\alpha_{n-2} B_{n-2} p_n p_{n-1} p_{n-3} + \Delta\alpha_{n-3} B_{n-3} p_n p_{n-1} p_{n-2} - \Delta\alpha_n B_n p_{n-1} p_{n-2} p_{n-3} - \Delta\alpha_{n-1} B_{n-1} p_n p_{n-2} p_{n-3}| \neq 0$ представляет целое положительное число, получим окончательное выражение:

$$P_{n+1} > \frac{p_n p_{n-1} p_{n-2} p_{n-3} (P_p - 1)}{P_p}. \quad (10)$$

Так как $(P_p-1)/P_p \approx 1$,то:

$$p_{n+1} \geq p_n p_{n-1} p_{n-2} p_{n+3}. \quad (11)$$

Из (11) видно, что величина избыточного модуля p_{n+1} для гарантированного определения неисправных оснований должна быть больше произведения наибольших модулей p_{n-3}, p_{n-2}, p_{n-1} и p_n. Проверка выполнения данного условия с помощью программы подтверждает это.

На рис.1 представлена геометрическая интерпретация распределения ошибочных интервалов по диапазону СОК для рабочих модулей $p_1=2$, $p_2=3$, $p_3=5$ и контрольного модуля $p_4=17$, не удовлетворяющих условию (11).

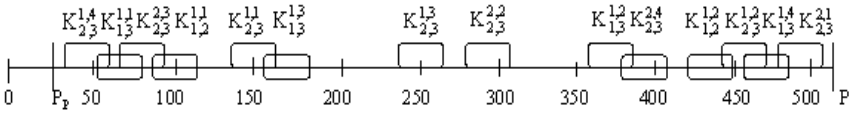


Рис. 1. График распределения ошибок для $p_1=2$, $p_2=3$, $p_3=5$, $p_4=17$

Наличие пересечений ошибочных интервалов свидетельствует о неоднозначности определения искаженных оснований. При увеличении контрольного основания p_4 до значения $31>p_3p_2p_1$ пересечения ошибочных интервалов не наблюдается (рис.2).

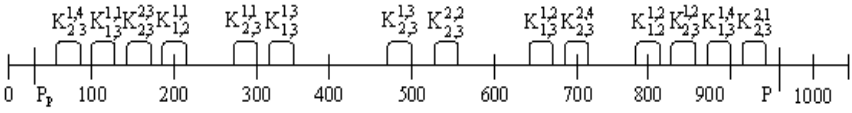


Рис.2. График распределения ошибок для $p_1=2$, $p_2=3$, $p_3=5$, $p_4=31$

Полученное выражение (11) для величины избыточного модуля сохраняет корректирующие способности кода СОК и определяет минимальную его величину. Однако при увеличении значений рабочих оснований согласно выражению (12) значение контрольного основания резко возрастает. При этом увеличиваются и аппаратурные затраты, то есть замедляется скорость работы модулярного нейрокомпьютера. Представив контрольное основание составными взаимно простыми числами, можно несколько уменьшить аппаратурные затраты. Кроме того, за счет допуска небольшого числа ошибок, можно уменьшить величину самого контрольного модуля и также уменьшить аппаратурные затраты.

Результаты моделирования по определению процента необнаруженных ошибок представлены в табл.1.

Таблица №1

Число необнаруживаемых ошибок					
p_{n+1}	$p_i=2,3,5$	p_{n+1}	$p_i=2,3,5,7$	p_{n+1}	$p_i=3,5,7,11$
11	*37,29%*	*47*	*44,17%*	*101*	*51,14%*
13	*29,14%*	*79*	*10,50%*	*211*	*13,73%*
17	*20,02%*	*101*	*2,51%*	*523*	*7,91%*
23	*6,20%*	*179*	*0,34%*	*799*	*1,12%*
29	*1,12%*	*199*	*0,14%*	*1011*	*0,12%*
31	*0%*	*211*	*0%*	*1159*	*0%*

Как видно, число необнаруженных ошибок резко уменьшается с возрастанием контрольного модуля. Так, при величине контрольного модуля $p_{n+1}=799$ при рабочем диапазоне $P_p=1155$ число необнаруженных ошибок около *1%*, а при величине контрольного модуля $p_{n+1}=1159$ - *0%*.Таким образом, допуская одну необнаруженную двукратную ошибку из *100* мы уменьшим величину контрольного модуля на *360* и значительно увеличиваем скорость работы корректирующего устройства. Зависимость необнаруживаеых ошибок (в процентах) от величины контрольного модуля представлена на рис.3.

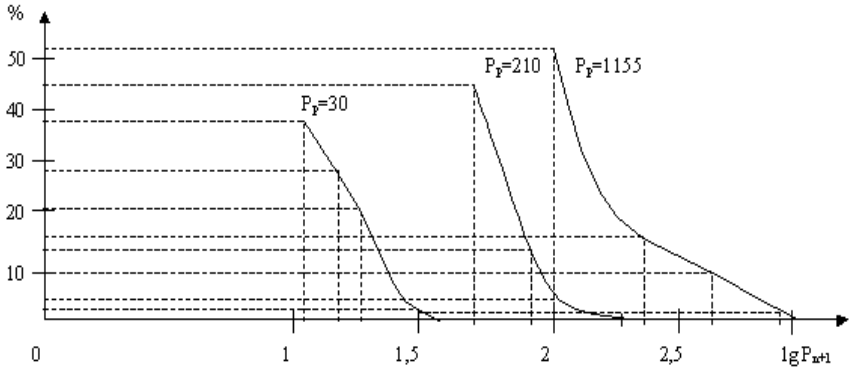


Рис. 3. График зависимости необнаруживаеых ошибок
(в процентах) от величины контрольного модуля

Реализация. Предложенный метод локализации и исправления ошибок достаточно легко реализуется на нейронных сетях конечного кольца (НСКК).

Структура нейронной сети (НС) (рис. 4) будет включать несколько компонентов: преобразователь СОК в ПСС на основе НСКК; НС классификации номера ошибочного интервала; НС исправления ошибки. Рассмотрим подробнее выделенные компоненты.

Преобразователь кода системы остаточных классов в позиционную систему счисления использует выражение (3) и достаточно подробно рассмотрен в [3]. Поэтому здесь приведем только его структуру без описания принципов построения рис. 5.

Рис.4. Структура нейронной сети для определения и исправления ошибок
в модулярных нейрокомпьютерах

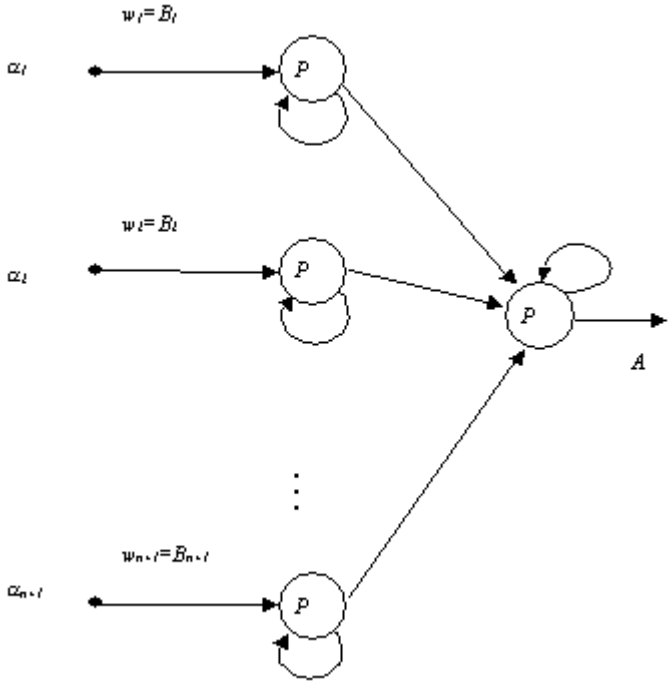


Рис. 5. Преобразователь СОК в ПСС (двоичный код) на основе НСКК

Количество входов преобразователя определяется числом модулей СОК p_i, $i\in[\overline{1,n+1}]$ и их двоичным представлением. Число A на выходе преобразователя будет представлено в двоичной системе счисления. Количество разрядов представления определяет число входов второй компоненты - НС классификации номера ошибочного интервала и зависит от величины полного диапазона СОК. Количество выходов будет равно числу ошибочных интервалов $K_{i,j}^{r,k}$ и определится из выражения $l=2\sum_{i=1}^{n} p_i - 2n$, где n – число рабочих оснований. Для рассмотренного выше примера полный диапазон $P=510$, для представления которого потребуется 9 двоичных разрядов т.к. $2^9=512>510>2^8=256$, число выходов

$$2*(p_1+p_2+p_3)-2*n=2*(2+3+5)-2*3=14.$$

НС классификации номера ошибочного интервала должна по зна-

чению числа A определять в какой ошибочный интервал $K_{i,j}^{r,k}$ оно попадает и на соответствующем выходе формировать *1*, а на остальных *0*. В случае отсутствия ошибки все выходы равны *0* и дальнейшее исправление числа A не требуется.

Решение данной задачи возможно с помощью двухслойной нейронной сети прямого распространения с логарифмическими сигмоидальными функциями активации в каждом слое [3]. Такая функция активации выбрана потому, что диапазон выходных сигналов для этой функции определен от *0* до *1*, и этого достаточно, чтобы сформировать значения выходного вектора. Пример структуры нейронной сети приведен на рис. 6.

Число нейронов первого слоя выбирается на основе разумных предположений исходя из конкретного числа входов и выходов, определяемых для заданной системы оснований СОК. Обучение сети проводится по методу обратного распространения ошибки [3].

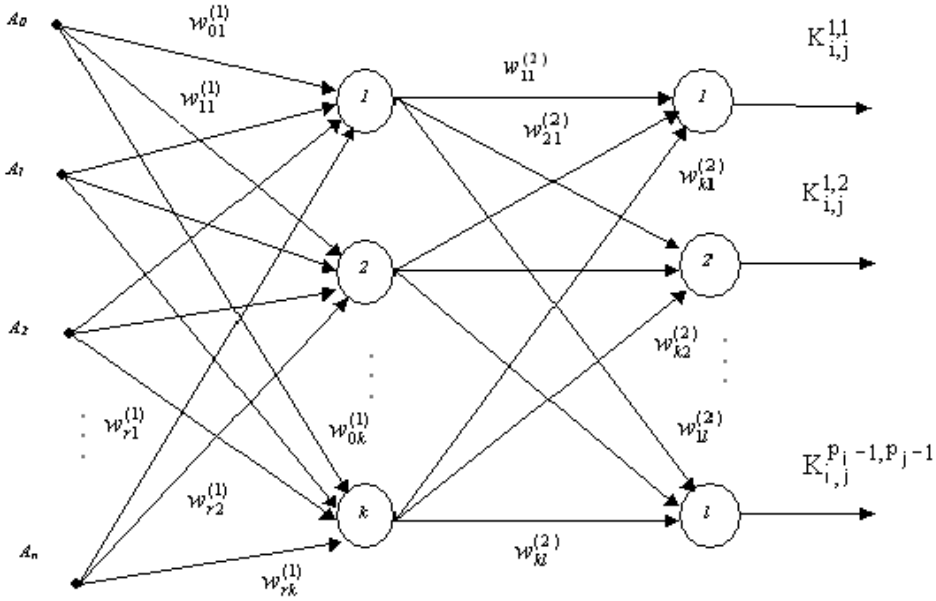


Рис. 6. Нейронная сеть классификации номера ошибочного интервала

Определив ошибочный интервал $K_{i,j}^{r,k}$, становятся известными ошибочные основания p_i и p_j, глубина ошибки $\Delta\alpha_i$ и глубина

ошибки $\Delta\alpha_j$, которые определяют количество входов нейронной сети исправления ошибки (рис. 7), а соответственно и значения синаптических весов первого слоя сети

$$w^{(1)}_{ij} = -\Delta\alpha_i B_i \; -\Delta\alpha_j B_j ; \qquad (14)$$

Значения весовых коэффициентов остальных связей равны $w=1$.

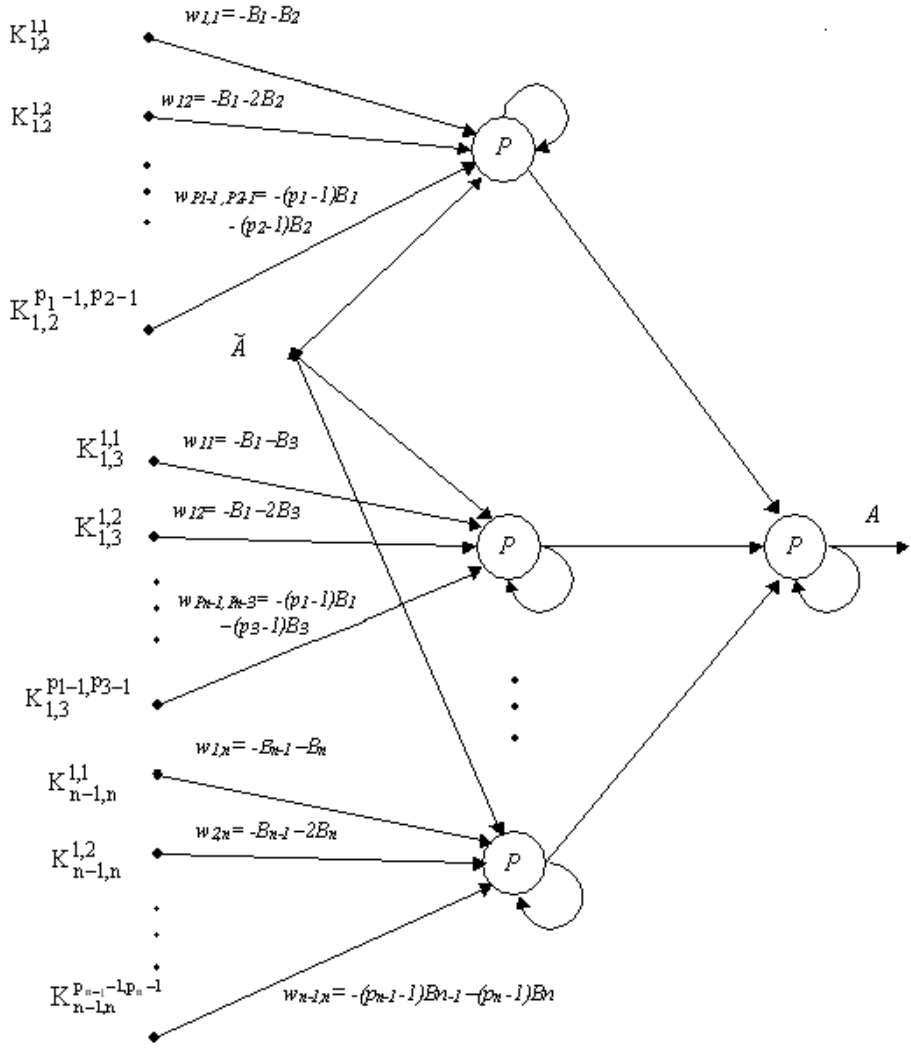


Рис. 7. Нейронная сеть исправления ошибки

На вход каждого сумматора нейронной сети конечного кольца поступают произведения весов (14) на значения *1* или *0* с выходов сети классификации номера ошибочного интервала и исходное число $\tilde{A}$. Так как на выходе сети классификации значение 1 формируется только на одном из выходов, то результатов работы НС исправления ошибки будет правильное число *A*, вычисляемое согласно выражению (11). В случае отсутствия ошибок на выходах сети классификации будут значения *0*, а следовательно все весовые коэффициенты (11) при умножении на *0* примут нулевые значения и на выход нейронной сети пройдет число *A* в позиционной системе счисления.

На выходе сети формируется исправленное число *A* в позиционной системе счисления.

Предложенный метод локализации и исправления и его нейросетевая реализация может быть реализована аппаратным способом на базе программируемых логических интегральных схем (ПЛИС) типа Xilinx [3]. Высокая степень интеграции ПЛИС, а также способность воспроизвести практически любую структуру нейронной сети , в том числе и НСКК позволяет реализовать параллельно работающие нейроны отождествленные с модулями СОК, а также блоки локализации и коррекции, обеспечивающие отказоустойчивость и живучесть модулярного нейрокомпьютера.

Литература

1. *Акушский И.Я., Юдицкий Д.И.* Машинная арифметика в остаточных классах. – М.: Советское радио, 1968, 440 с.
2. *Н. И .Червяков, В. В. Бережной, А .А. Оленев ,И .А. Калмыков* Минимизация избыточности кода системы остаточных классов с одним контрольным основанием – Киев: “Электронное моделирование”,1994 г,Т.16.№1.
3. *Червяков Н.И., Шапошников А.В., Сахнюк П.А.* Модель и структура нейронной сети для реализации арифметики системы остаточных классов // Нейрокомпьютеры: разработка, применение, 2001, №10, с. 6-12.
4. *Галушкин А.И.* Нейрокомпьютеры. – М.: ИПРЖ «Радиотехника», 2000, 526 с

УДК. 681.3

Н.И. Червяков, С.Л. Ремизов

Локализация ошибки на основе метода расширенной проекции

(Ставропольский военный институт связи ракетных войск, Сибирский государственный университет телекоммуникаций и информатики)

Система счисления в остаточных классах (СОК) [1] открывает возможность использования единого помехоустойчивого кода для борьбы с ошибками, возникающими при передаче информации по каналам связи и при ее обработке в цифровых системах.

Рассмотрим систему оснований p_1, p_2,…, p_{n-1} с диапазоном $R = \prod_{i=1}^{n-1} p_i$, который будем называть рабочим. Введем основание p_n, взаимно простое с любым из n-1 оснований, которое назовем контрольным, и будем представлять числа в системе из n оснований. Это означает, что мы будем передавать и обрабатывать числа, принадлежащие диапазону (0 ÷ R), в более широком диапазоне (0 ÷ P), где $P = p_n R$ [1], который будем называть полным. Правильными будем считать числа, принадлежащие диапазону (0 ÷ R), искаженные – диапазону (R ÷ P).

Для обнаружения факта ошибки используется следующее правило

[3].

Если число представлено в обобщенной полиадической системе (ОПС),то

- при $a_n = 0$ число принадлежит рабочему диапазону (правильное);
- при $a_n \neq 0$ число принадлежит диапазону (R ÷ P) (ошибочное).

В [5] показано, что число $A = (\alpha_1,...,\alpha_i,....,\alpha_n)$ преобразованное в число $\tilde{A} = (\alpha_1,...,\tilde{\alpha}_i,...,\alpha_n)$ через ошибку в i-той цифре при умножении каждой цифры α_j на p_i и приведении этого произведения по модулю p_j проецируется на модифицированный правильный диапазон ($0 \div p_iR$) и при умножении на p_j при $i \neq j$ проецируется на модифицированный неправильный диапазон ($p_iR \div P$).

Для определения диапазона, в который попадает расширенная проекция необходимо представление ОПС такое, что произведение оснований при старшем коэффициенте a_n было равно p_iR. Следующая теорема показывает, как осуществить выбор системы оснований, чтобы выполнялось это условие.

Теорема 1

Для того, чтобы при старшем коэффициенте a_n представления числа A в ОПС произведение оснований было равно p_iR необходимо и достаточно в исходной системе основание p_i было заменено на p_i^2.

Доказательство

Докажем достаточность утверждения.

Представление числа A в ОПС имеет вид

$$A = \sum_{k=1}^{n} a_k \prod_{i=1}^{k-1} p_i \ . \qquad (1)$$

Для k = 1: $\prod_{i=1}^{0} p_i = 1$, для k = n $a_n \prod_{i=1}^{n-1} p_i$. Так как $\prod_{i=1}^{n-1} p_i = R$, то для k = n справедлива запись a_nR.

По определению

$$R = \prod_{i=1}^{n-1} p_i = p_1 \prod_{i=2}^{n-1} p_i = p_2 \prod_{\substack{i=1 \\ i \neq 2}}^{n-1} p_i = \ldots = p_j \prod_{\substack{i=1 \\ i \neq j}}^{n-1} p_i = \ldots = p_{n-1} \prod_{i=1}^{n-2} p_i \,. \qquad (2)$$

Умножим обе части этого равенства на p_j и получим

$$p_j R = p_j \prod_{i=1}^{n-1} p_i = p_j p_1 \prod_{i=2}^{n-1} p_i = p_j p_2 \prod_{\substack{i=1 \\ i \neq 2}}^{n-1} p_i = \ldots = p_j p_j \prod_{\substack{i=1 \\ i \neq j}}^{n-1} p_i = \ldots = p_j p_{n-1} \prod_{i=1}^{n-2} p_i \,. \qquad (3)$$

Введем подстановку

$$p'_1 = p_j p_1; p'_2 = p_j p_2; \ldots; p'_j = p_j p_j; \ldots; p'_{n-1} = p_j p_{n-1} \qquad (4)$$

тогда

$$p_j R = p_j \prod_{i=1}^{n-1} p_i = p'_1 \prod_{i=2}^{n-1} p_i = p'_2 \prod_{\substack{i=1 \\ i \neq 2}}^{n-1} p_i = \ldots = p'_j \prod_{\substack{i=1 \\ i \neq j}}^{n-1} p_i = \ldots = p'_{n-1} \prod_{i=1}^{n-2} p_i \,. \qquad (5)$$

Достаточность утверждения доказана. Необходимость докажем следующим образом. При осуществлении перевода числа из СОК в ОПС i-тая цифра будет вычисляться следующим образом

$$a_i = ((((\ldots(\alpha_i - a_1) p_1^{-1} - a_2) p_2^{-1} - a_3) \ldots - a_{i-1}) p_{i-1}^{-1}) \bmod p_i \,. \qquad (6)$$

Введем обозначение

$$c = ((\ldots(\alpha_i - a_1) p_1^{-1} - a_2) p_2^{-1} - a_3) \ldots - a_i \,. \qquad (7)$$

Подставляя (7) в (6), получим

$$a_i \equiv c p_{i-1}^{-1} \bmod p_i \,. \qquad (8)$$

Если положить в p_{i-1} в новой системе оснований равным $p_{i-1}p_i$, то

$$a_i \equiv c(p_{i-1} p_i)^{-1} \bmod p_i \qquad (9)$$

или

$$a_i \frac{p_{i-1}}{c} p_i \equiv 1 \bmod p_i \qquad (10)$$

А сравнение такого вида решений относительно a_i не имеет.

Таким образом, для получения верхней границы модифицированного правильного диапазона при старшем коэффициенте a_n необходимо осуществить преобразование ОПС такое, что в качестве одного из оснований системы взято основание, умноженное само на се-

бя. При этом это должно быть основание, по которому находится расширенная проекция.

Для нахождения представления в ОПС при такой системе оснований необходимо знать остаточную цифру по новому основанию. Ее нахождение можно осуществить способами, изложенными в [2,4].

Рассмотрим пример.

Пусть имеется система оснований $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, где p_1, p_2 – рабочие основания, а p_3 – контрольное, и в этой системе задано число A=3=(1,0,3). Пусть произошло искажение второго символа: (1,0,3) → (1,1,3). Факт искажения будем считать установленным.

Найдем расширенную проекцию по основанию p_1 (1,1,3)·2=(0,2,1). Верхняя граница модифицированного правильного диапазона $p_1R = 2 \cdot 6 = 12$. Заменим основание p_1 на основание $p_1^2 = 4$. Тогда представление расширенной проекции для новой системы оснований СОК ($p_1 = 4$, $p_2 = 3$, $p_3 = 5$) будет иметь вид (2,2,1), а представление в ОПС (2,0,2) = 26. Так как $a_3 \neq 0$, то цифра по основанию p_1 безошибочна ($12 < 26$). Найдем теперь расширенную проекцию по основанию p_2. (1,1,3)·3=(1,0,4). Верхняя граница модифицированного правильного диапазона составит $p_2R = 3 \cdot 6 = 18$.

В новой системе оснований СОК ($p_1 = 2$, $p_2 = 9$, $p_3 = 5$) расширенная проекция будет иметь вид (1,0,3), представление в ОПС (1,4,0) = 9. Так как $a_3=0$, то во втором разряде цифра искажена ($18 > 9$).

Таким образом, с помощью предложенного метода можно осуществить локализацию искаженного разряда в данных, представленных кодом СОК.

Данный способ имеет преимущества перед методом, изложенным в [1] в том, что исчезает необходимость знать величину проекции, достаточно знать только старший коэффициент представления ОПС. Кроме того, в предлагаемом методе сравнение можно реализовать с помощью модульной операции.

Литература

1. *Акушский И.Я., Юдицкий Д.И.*, Машинная арифметика в остаточных классах. – М.: Советское радио, 1968. – 440 с.
2 Справочник по цифровой вычислительной технике/ Под ред. *Б.Н. Малиновского*. – Киев: Техника, 1975. – 512 с.

3. *Торгашев В. А.* Система остаточных классов и надежность ЦВМ. М.: Сов. радио, 1973. – 118 с.
4 *Червяков Н.И., Ряднов С.А., Сахнюк П.А., Шапошников А.В.,* Модулярные параллельные вычислительные структуры нейропроцессорных систем. – М.: ФИЗМАТЛИТ, 2003. – 288 с.
5. *W. Kenneth Jenkins, Edward J Altman.* Self-checking properties of Residue Number error checkers based on Mixed Radix conversion. – IEEE Transactions on circuits and system, vol. 35, No. 2, February.

УДК 511+519.719.2

О.А. Финько

Многоканальные модулярные системы, устойчивые к искажениям криптограмм

(*Краснодарское высшее военное училище*
(*военный институт*))

Предложен принцип построения многоканальных систем защиты информации, устойчивых к искажениям криптограмм. Криптограммы, передаваемые по каналам шифрования, сопоставлены символам модулярного кода. Вычисляются дополнительные криптограммы, которые соответствуют избыточным символам модулярного кода. Таким образом передаваемая система криптограмм является избыточным модулярным кодом. Это обеспечивает обнаружение и/или исправление ошибок различного происхождения (помехи, имитация злоумышленника). В отличие от известных методов индивидуального (одноканального) контроля в масштабе одной криптограммы обеспечивается коррекция ошибки любой кратности. Указано на возможность построения системы групповой цифровой подписи, обладающей новыми полезными свойствами.

Лавинный характер увеличения объемов передаваемой по каналам связи информации и переход к коллективным методам обработки информации на базе локальных вычислительных сетей обуславливает необходимость перехода к многоканальным методам и средст-

вам криптографической защиты информации. Известно сколь негативны последствия, вызываемые ошибками при передаче криптограмм. Поэтому все большее внимание уделяется изучению криптосистем, устойчивых к ошибкам. Если заранее известно, что система является многоканальной, то в ней могут быть использованы особые (групповые) методы контроля ошибок, позволяющие получить преимущества, недоступные обычным методам индивидуального контроля [4].

Будем рассматривать *n*-канальную систему шифрования (например, RSA), правила зашифрования и расшифрования в которой определены формулами:

$$\begin{cases} C^{(1)} = \mathrm{E}_{k^{(1)}}\left(M^{(1)}\right) \pmod{m^{(1)}}, \\ C^{(2)} = \mathrm{E}_{k^{(2)}}\left(M^{(2)}\right) \pmod{m^{(2)}}, \\ \ldots, \\ C^{(n)} = \mathrm{E}_{k^{(1)}}\left(M^{(n)}\right) \pmod{m^{(n)}}; \end{cases} \qquad (1)$$

$$\begin{cases} M^{(1)} = \mathrm{D}_{k^{(1)}}\left(C^{(1)}\right) \pmod{m^{(1)}}, \\ M^{(2)} = \mathrm{D}_{k^{(2)}}\left(C^{(2)}\right) \pmod{m^{(2)}}, \\ \ldots, \\ M^{(n)} = \mathrm{D}_{k^{(n)}}\left(C^{(n)}\right) \pmod{m^{(n)}}; \end{cases} \qquad (2)$$

где

$M^{(1)}, M^{(2)}, \ldots, M^{(n)}$ — открытые тексты,

$C^{(1)}, C^{(2)}, \ldots, C^{(n)}$ — криптограммы,

$k^{(1)}, k^{(2)}, \ldots, k^{(n)}$ — ключи (системы ключей).

Передача криптограмм $C^{(1)}, C^{(2)}, \ldots, C^{(n)}$ по каналу связи приведет к появлению в ней искажений, в результате которых процедуре расшифрования подвергнутся криптограммы $C^{*(1)}, C^{*(2)}, \ldots, C^{*(n)}$. Искажения могут возникнуть в результате как непреднамеренных (помехи, сбои, дефекты), так и преднамеренных воздействий.

Таким образом, процедура расшифрования (2) примет вид:

$$\begin{cases} M^{*(1)} = \mathrm{D}_{k^{(1)}}\left(C^{*(1)}\right) \pmod{m^{(1)}}, \\ M^{*(2)} = \mathrm{D}_{k^{(2)}}\left(C^{*(2)}\right) \pmod{m^{(2)}}, \\ \ldots, \\ M^{*(n)} = \mathrm{D}_{k^{(n)}}\left(C^{*(n)}\right) \pmod{m^{(n)}}; \end{cases} \quad (3)$$

где $M^{*(1)}, M^{*(2)}, \ldots, M^{*(n)}$ — открытые тексты, которые могут содержать ошибки в результате расшифрования искаженной криптограммы.

Введем требование: $\gcd(m_i, m_j) = 1$, где $\gcd(a, b)$ — наибольший общий делитель a и b; $i, j = 1, 2, \ldots, n$. Тогда системе уравнений (3) в соответствии с Китайской теоремой об остатках можно сопоставить единственное решение:

$$C = \mathop{\mathrm{CRT}}_{i=1}^{n} C_i \pmod{m^{(i)}},$$

где CRT — групповой оператор решения системы уравнений по Китайской теореме об остатках (Chinese remainder theorem) [1, 3].

Дополним систему модулей $m^{(1)}, m^{(2)}, \ldots, m^{(n)}$ еще r модулями $m^{(n+1)}, \ldots, m^{(n+r)}$ такими, что $\gcd(m_i, m_j) = 1$, где $i, j = 1, 2, \ldots, n+r$. Потребуем также, чтобы выполнялось условие: $m^{(1)}, m^{(2)}, \ldots, m^{(n)} < m^{(n+1)} < \ldots < m^{(n+r)}$. Тогда можем получить *расширенную* систему криптограмм:

$$C^{(1)}, C^{(2)}, \ldots, C^{(n)}, \ldots, C^{(n+r)},$$

где

$$C^{(n+1)} = C \pmod{m^{(n+1)}}, \ldots, C^{(n+r)} = C \pmod{m^{(n+r)}}.$$

Таким образом, систему уравнений (3) расшифрования перепишем в виде следующей расширенной системы:

$$\begin{cases} M^{*(1)} = \mathrm{D}_{k^{(1)}}\left(C^{*(1)}\right) \quad \left(\mathrm{mod}\, m^{(1)}\right), \\ M^{*(2)} = \mathrm{D}_{k^{(2)}}\left(C^{*(2)}\right) \quad \left(\mathrm{mod}\, m^{(2)}\right), \\ \ldots, \\ M^{*(n)} = \mathrm{D}_{k^{(n)}}\left(C^{*(n)}\right) \quad \left(\mathrm{mod}\, m^{(n)}\right), \\ \ldots, \\ M^{*(n+r)} = \mathrm{D}_{k^{(n+r)}}\left(C^{*(n+r)}\right) \quad (\mathrm{mod}\, m^{(n+r)}). \end{cases}$$

В соответствии с положениями модулярной арифметики расширенная система криптограмм представляет расширенный модулярный код (R-код), обладающий свойствами обнаружения и исправления ошибок [1, 2].

Под одиночной ошибкой будем понимать произвольное искажение одной из криптограмм. t - кратная ошибка — произвольное искажение t криптограмм. Известны следующие положения модулярной арифметики [1, 2]:

1) R-код обнаруживает все одиночные ошибки, если $r \geq 1$;

2) R-код исправляет t или менее ошибок, если $2t \leq r$.

Простейшим признаком обнаруживаемой ошибки является выполнение неравенства [1, 2]:

$$C^* \geq \prod_{i=1}^{n} m^{(i)},$$

где $C^* = \mathrm{CRT}_{i=1}^{n} C_i^* \quad (\mathrm{mod}\, m^{(i)})$.

Исследованиям методов коррекции модулярных кодов посвящено большое количество зарубежной и отечественной литературы, например [1, 2, 5, 6, 7]. Пример структурной схемы n-канальной криптосистемы с одним избыточным каналом и возможностью обнаружения однократных ошибок представлен на рис. 1. В литературе по модулярной арифметике процедура получения избыточных элементов R-кода называется расширением кода. Для обнаружения ошибок на приемной стороне используется устройство, обеспечивающее преобразование кода в соответствии с Китайской теоремой об остатках (CRT) и сравнение полученного результата с по-

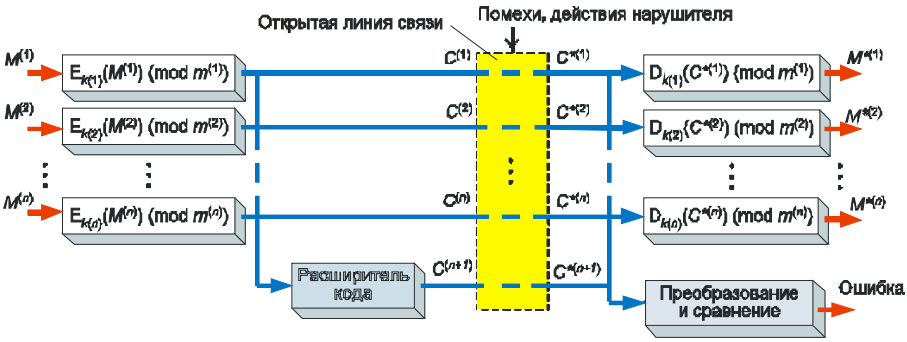


Рис. 1. Структурная схема n-канальной криптосистемы с одним избыточным каналом и возможностью обнаружения однократных ошибок

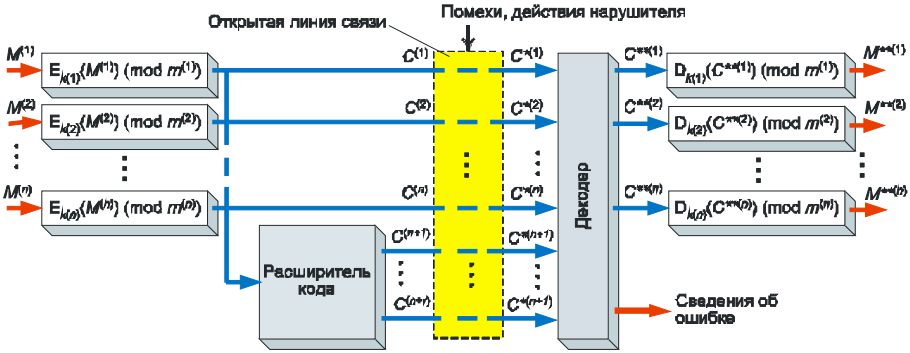


Рис. 2. n-канальная криптосистема с r избыточными каналами и возможностью исправления ошибок

роговым значением, равным произведению рабочих (не избыточных) модулей. Могут быть использованы и другие методы обнаружения ошибок R-кода.

Пример n-канальной криптосистемы с r избыточными каналами и возможностью исправления ошибок представлен на рис. 2. Здесь для исправления ошибок использован декодер, алгоритмы функционирования которого хорошо изучены. Отметим, что после выполнения процедуры исправления ошибок над кодовым словом $C^{*(1)}, C^{*(2)}, \ldots, C^{*(n+r)}$ мы получим криптограммы: $C^{**(1)}, C^{**(2)}, \ldots, C^{**(n)}$ и исправленные открытые тексты $M^{**(1)}, M^{**(2)}, \ldots, M^{**(n)}$. Здесь две звездочки ** указывают на ве-

роятностный характер исправления ошибки. Учитывая большую величину числового диапазона, с которым приходится оперировать, процедуры расширения и декодирования кода будут иметь максимальную эффективность, если они поддержаны аппаратно.

К достоинствам рассмотренного метода контроля ошибок отнесем возможность обнаружения (исправления) искажений в передаваемых криптограммах при *любой* (!) величине ошибки (в масштабе отдельной криптограммы), то есть и в случае стирания криптограмм или обрыве линии, если количество искаженных криптограмм не превышает обнаруживающих (исправляющих) возможностей R-кода.

Для осуществления навязывания ложной информации (имитации криптограммы или цифровой подписи) злоумышленнику придется иметь дело со всей совокупностью информационных криптограмм (для того чтобы получить избыточные криптограммы). При этом для получения гарантированного результата необходимо применить шифрование избыточных криптограмм (избыточных цифровых подписей).

Выбранный вариант помехоустойчивого группового контроля ошибок на основе расширенного модулярного кода основывается на *естественных* свойствах исходных криптосистем (модулярность), что в свою очередь обеспечивает необходимую *совместимость* построенной криптосистемы с другими абонентами, которые не используют рассмотренный метод.

Интересные возможности появляются при использовании данного метода контроля при формировании цифровых подписей для групп документов. В этом случае арбитр сможет восстанавливать любую утраченную (искаженную) цифровую подпись, даже если он хранит только избыточные подписи (избыточные криптограммы), но имеет юридическую возможность доступа к остальным (не утраченным) подписям (например, подписи хранятся у самих пользователей). Для обеспечения этих возможностей необходимо предусмотреть единый центр генерации ключей или обеспечить взаимодействие пользователей.

Устойчивость к криптоанализу обеспечивается правильным выбором точки расширения кода в системе. В данном случае расширению (введению избыточности) подвергаются не открытые тексты, а криптограммы. Решение, основанное на расширении множества

открытых текстов и, как следствие, расширение количества процедур зашифрования привлекательно с позиции открывающихся возможностей контроля ошибок процедур зашифрования-расшифрования. Однако оно не выдерживает атаки, основанной на Китайской теореме об остатках.

Ужесточение требований к модулям может быть компенсировано адекватным расширением множества модулей.

Литература

1. *Амербаев В.М.* Теоретические основы машинной арифметики. — Алма-Ата: Наука, 1976. — 324 с.
2. *Бояринов И.М.* Помехоустойчивое кодирование числовой информации. — М.: Наука, 1983. — 196 с.
3. *Финько О.А.* Восстановление числа в системе остаточных классов с минимальным количеством оснований // Электронное моделирование. — 1998. — Т. 20, № 3. — С. 56–61.
4. *Финько О.А.* Групповой контроль ассиметричных криптосистем методами модулярной арифметики // XIV Междунар. школа-семинар «Синтез и сложность управляющих систем». Н. Новгород, 27 окт. — 2 нояб. 2003. Сб. тр. / Под ред. акад. РАН О.Б. Лупанова. — Н. Новгород: Изд-во Нижегород. пед. ун-та, 2003. — С. 85–86.
5. *Mandelbaum D.M.* Error correction in residue arithmetic // IEEE Trans. Comput. — 1972. — Vol. 21, № 6. — P. 538–545.
6. *Mandelbaum D.M.* On a class of arithmetic codes and decoding algorithm // IEEE Trans. On Information Theory. — 1976. — № 21. — P. 85–88.
7. *Mandelbaum D.M.* Further results on decoding arithmetic residue codes // IEEE Trans. On Information Theory. — 1978. — № 24. — P. 643–644.

УДК 681. 3

O.A. Finko

Algorithms and Devices for *N*-ary Finite Ring Computations

(Krasnodar high military school (military institute), Russia)

Methods of computer algebra connected with execution N-ary (N > 2) operations in a finite ring for specialized computing devices which operate in the modular arithmetics is developed. Is displayed that traditional methods of execution of N-ary operations which grounded on a "horizontal" method are oriented on a small amount of operands. Higher outcomes can be obtained with help parallel and sequential are N-ary of arithmetic devices which have grounded on a "vertical" method.

1. Introduction

The specialized computing devices (SCD) are applied to implementation of algorithms of digital signals processing (number-theoretic conversions, of cyclic convolution) [1—4], cryptography problems [5], control and simulations problems [6, 7], neural-like networks problems [8]. The important stage of development of SCD is multisequencing computations with the help of the modular arithmetics (MA) [10—12].

Today main attention of the experts is directed on increase of efficiency of execution of not modular operations [13, 14] and development of

methods of a check and reconfiguration of SCD by redundant MA [15]. The following stage of development of SCD, which operate in the MA, can be integration of operations, which are realized in a parallel way. The integration of operations is supposed to be realized by transition from binary to *N*-ary modular operations. However now methods of implementation of *N*-ary operations on the given the modulo are advanced insufficiently. It is an obstacle in paths of implementation of the second stage of development of SCD. Therefore purpose of paper is to develop the effectiveness of algorithms and devices which are intended for implementation of *N*-ary operations of MA.

On the basis of Chinese remainder theorem [10—12] any integer $X \geq 0$ can be represented by a sequence of bit digits

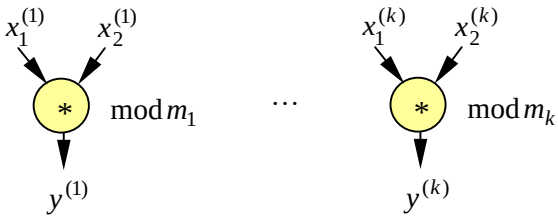
$$\{X\} = \left(x^{(1)}, x^{(2)}, \ldots, x^{(k)}\right),$$

where $x^{(i)} = |X|_{m_i}$, $i = 1, 2, \ldots, k$; $m_1, m_2, \ldots, m_k$ $\left(m_1 < m_2 < \ldots < m_k\right)$ — basis.

The representation of the MA is unique, if $0 \leq X < M = m_1 m_2 \ldots m_k$ and $\gcd\left(m_i, m_j\right) = 1$ for $\forall i \neq j, \quad i, j = 1, 2, \ldots, k$.

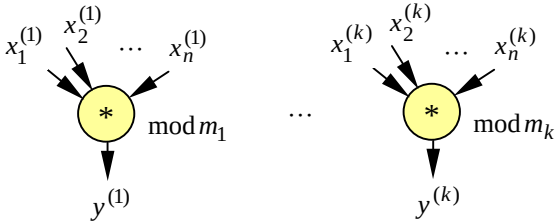
The operations of arithmetics MA are fulfilled by a parallel way and separately for each unit MA and therefore faster, than in traditional arithmetics, in which there are interbit links.

The existing algorithms MA use binary operations $x^{(i)} = \left|x_1^{(i)} * x_2^{(i)}\right|_{m_i}$ $\left(i = 1, 2, \ldots, k\right)$:



Here symbol * means one of modular operations (addition or multiplying modulo m_i).

The N-ary operation modulo m_i we shall name the operation $y^{(i)} = \left| \underset{j=1}{\overset{n}{*}} x_j^{(i)} \right|_{m_i}$:



The residuals $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$ are represented with the help of a binary number system or unitary code. Let's consider a case of usage of a binary number system:

$$x_j^{(i)} = a_{d-1,\,j}^{(i)} 2^{k-1} + \ldots + a_{1,\,j}^{(i)} 2 + a_{0,\,j}^{(i)},$$

where $a_{r,\,j}^{(i)} \in \{0, 1\}$, $i = 1, 2, \ldots, k$; $r = 0, 1, \ldots, d-1$; $j = 1, 2, \ldots, n$.
Here the value of the module m_i and bit grid d are connected as $d = \lceil \log_2 (m_i - 1) \rceil$, where $\lceil A \rceil$ means the least integer $> A$.

2. *N*-ary operator of addition modulo m_i

2.1. Traditional architectures of *N*-ary summators modulo m_i

2.1.1. The "horizontal" *N*-ary summator modulo m_i of a parallel type. The structure of the traditional N-ary summator modulo m_i of a parallel type has a tree structure with $\lceil \log_2 (n-1) \rceil$ levels (figure 1) and $n-1$ by binary summators.

Hold time of this summator $T_\Sigma = \lceil \log_2(n-1) \rceil \tau_\Sigma$, where τ_Σ time of operation of one binary summator modulo m_i.

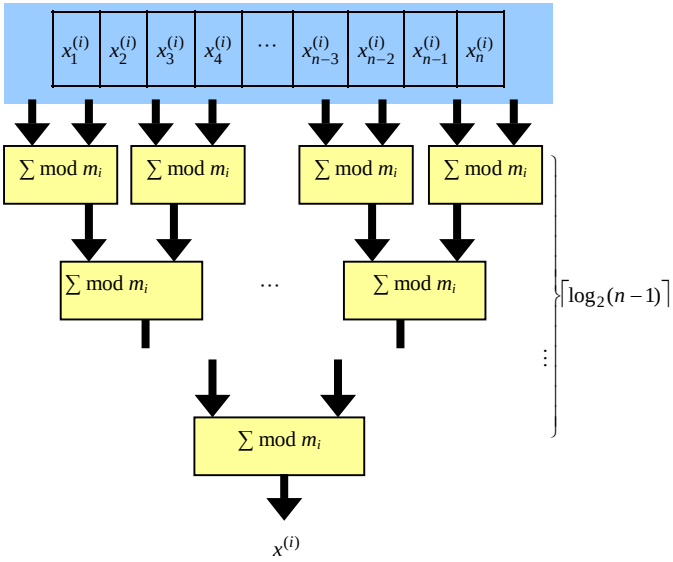


Figure 1. The "horizontal" *N*-ary summator modulo m_i of a parallel type

2.1.2. The "horizontal" *N*-ary summator modulo m_i of a sequential type. The traditional *N*-ary summator of a sequential type consists of the parallel-sequential register of shift and accumulating binary summator modulo m_i (figure 2). Amount of sync signals necessary for execution for the operation

$$y^{(i)} = \left| \sum_{j=1}^{n} x_j^{(i)} \right|_{m_i}$$

corresponds to total of addends and equally n.

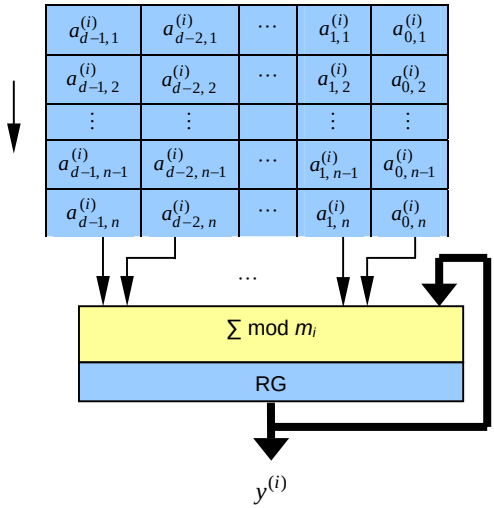


Figure 2. "Horizontal" *N*-ary summator modulo m_i of a sequential type

2.2. Algorithms and devices of *N*-ary addition modulo m_i, grounded on a "vertical" method

2.2.1. The algorithm of "vertical" *N*-ary addition modulo m_i. Let's consider algorithm *N*-ary addition module m_i, which works not with numbers $x_j^{(i)}$ $(j = 1, 2, \ldots, n)$, but with digits $a^{(i)}_{r,1}, a^{(i)}_{r,2}, \ldots, a^{(i)}_{r,n}$ $(r = d-1, \ldots, 1, 0)$.

Algorithm 2.1.

Step 1. Are realized d of count operations of units modulo m_i:

$$\xi^{(i)}_{d-1} = \left| \sum_{j=1}^{n} a^{(i)}_{d-1,j} \right|_{m_i}, \quad \ldots, \quad \xi^{(i)}_{1} = \left| \sum_{j=1}^{n} a^{(i)}_{1,j} \right|_{m_i}, \quad \xi^{(i)}_{0} = \left| \sum_{j=1}^{n} a^{(i)}_{0,j} \right|_{m_i}.$$

Step. 2. Realizing modular products of values $\xi_r^{(i)}$ $(r = 0, 1, \ldots, d-1)$ on 2^r $(r = 0, 1, \ldots, d-1)$:

$$\hat{\xi}_{d-1}^{(i)} = \left|\xi_{d-1}^{(i)} 2^{d-1}\right|_{m_i}, \ \ldots, \ \hat{\xi}_1^{(i)} = \left|\xi_1^{(i)} 2\right|_{m_i}, \ \hat{\xi}_0^{(i)} = \xi_0^{(i)}.$$

Step 3. The outcome of *N*-ary addition is:

$$y^{(i)} = \left|\sum_{j=0}^{d-1} \hat{\xi}_r^{(i)}\right|_{m_i}.$$

2.2.2. The "vertical" *N*-ary summator modulo m_i of a parallel type. The structure of algorithm 2.1 can be presented as follows:

The device for parallel implementation of algorithm 2.1 (figure 3)

$$\begin{array}{llll}
x_1^{(i)} = (a_{d-1,1}^{(i)}, & a_{d-2,1}^{(i)}, & \cdots, a_{1,1}^{(i)}, & a_{0,1}^{(i)}) \\
x_2^{(i)} = (a_{d-1,2}^{(i)}, & a_{d-2,2}^{(i)}, & \cdots, a_{1,2}^{(i)}, & a_{0,2}^{(i)}) \\
\vdots \quad \vdots & \vdots & \vdots & \vdots \\
x_{n-1}^{(i)} = (a_{d-1,n-1}^{(i)}, & a_{d-2,n-1}^{(i)}, & \cdots, a_{1,n-1}^{(i)}, & a_{0,n-1}^{(i)}) \\
x_n^{(i)} = (a_{d-1,n}^{(i)}, & a_{d-2,n}^{(i)}, & \cdots, a_{1,n}^{(i)}, & a_{0,n}^{(i)})
\end{array}$$

$$\begin{array}{llll}
\xi_{d-1}^{(i)} & \xi_{d-2}^{(i)} & \cdots \ \xi_1^{(i)} & \xi_0^{(i)} \\
\downarrow & \downarrow & \downarrow & \downarrow \\
\hat{\xi}_{d-1}^{(i)} & \hat{\xi}_{d-2}^{(i)} & \cdots \ \hat{\xi}_1^{(i)} & \hat{\xi}_0^{(i)}
\end{array}$$

$$\left|\hat{\xi}_{d-1}^{(i)} + \hat{\xi}_{d-2}^{(i)}\right|_{m_i} \qquad \left|\hat{\xi}_1^{(i)} + \hat{\xi}_0^{(i)}\right|_{m_i}$$

$$y^{(i)} = \left|\sum_{r=0}^{d-1} \hat{\xi}_r^{(i)}\right|_{m_i}$$

contains the register of storage of binary bit digits, d of parallel counters of units (devices of convolution) ST $\bmod m_i$ and $d-1$ of binary summators $\Sigma \bmod m_i$. If on outputs of parallel units counters the unitary code, the additional encoders CD-1 … CD-*d* of the unitary code in the code of a binary number system are installed. For the usually used module $2 \le m_i < 128$ a tree of binary summators — 1 … 3 levels. The

main device of the considered summator — counter of units ST $\bmod m_i$. The principles of construction of such devices are well known [16—18], however, probably best outcomes were obtained in [19, 20].

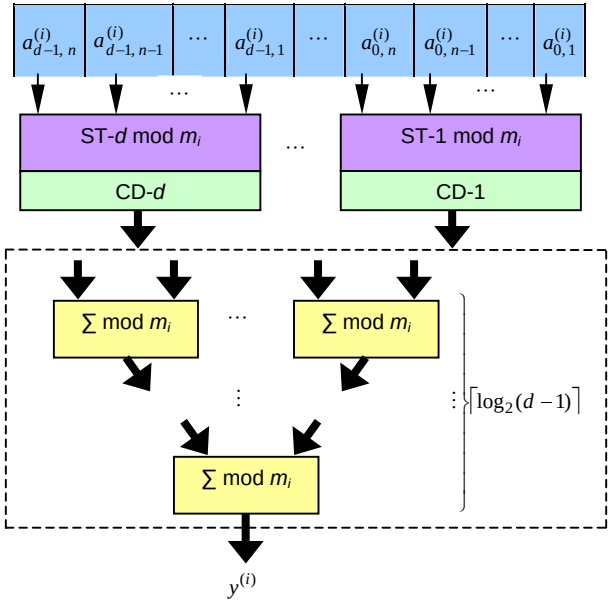
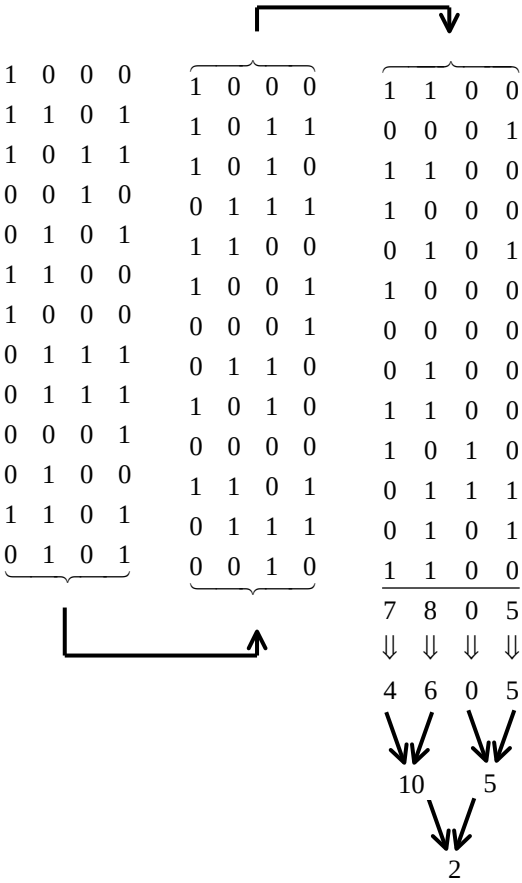


Figure 3. The "vertical" summator modulo m_i

Numerical example of addition of 39 operands modulo 13:



2.2.3. The "vertical" *N*-ary summator modulo m_i of a sequential type. We use numbers $\xi_{d-1}^{(i)},\ldots,\xi_1^{(i)},\xi_0^{(i)}$, from algorithm 2.1 for construction of the formula (principle of Horner):

$$y^{(i)}=\left|\xi_{d-1}^{(i)}2^{d-1}+\xi_{d-2}^{(i)}2^{d-2}+\ldots+\xi_1^{(i)}2+\xi_0^{(i)}\right|_{m_i}=$$

$$=\left|\left|\left(\ldots\left|\left(\left|\left(\left|\xi_{d-1}^{(i)}2\right|_{m_i}+\xi_{d-2}^{(i)}\right)2\right|_{m_i}+\xi_{d-3}^{(i)}\right)2\right|_{m_i}+\ldots+\xi_1^{(i)}\right)2\right|_{m_i}+\xi_0^{(i)}\right|_{m_i}.$$

.

The summator realizing this formula contains the parallel-sequential register of shift the counter of units ST mod m_i in the unitary code (encoder CD is not used) multiplying tube on 2 modulo m_i and binary adderaccumulator modulo m_i (figure 4).

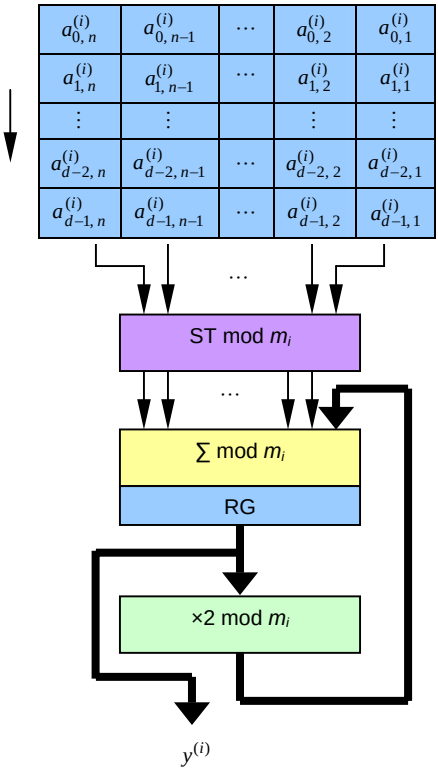


Figure 4. The "vertical" *N*-ary summator modulo m_i of a sequential type

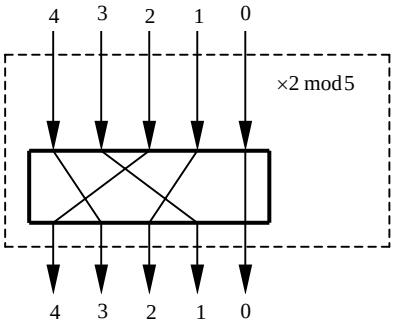


Figure 5. Structure of the multiplying tube on 2 modulo m_i

The principle of operation of *N*-ary summator is completely defined by the last obtained formula. The last clock tick of operation of the summator is used for addition $\xi_0^{(i)}$.

The unitary code is convenient for execution of the operation of multiplying on a constant in a finite ring. Thus the operation of multiplying is defined by the scheme of swaps of bits. The example of construction of the multiplying tube modulo 5 operation using this principle represented on a figure 5.

3. "Vertical" *N*-ary the converter of the code of a binary number system in the code modulo m_i

3.1. Algorithm of "vertical" *N*-ary conversion of a binary number system in the code modulo m_i

Let's consider the operator of the following sort:

$$y^{(i)} = \left| \sum_{j=1}^{s} X_j \right|_{m_i},$$

where for numbers $X_1, X_2, \ldots, X_s$ is satisfied condition

$$0 \le \sum_{j=1}^{s} X_j < M = m_1 m_2 \ldots m_k$$

and

$$
\begin{aligned}
X_1 &= b_{v-1,1}2^{v-1} + b_{v-2,1}2^{v-2} + \ldots + b_{1,1}2 + b_{0,1},\\
X_2 &= b_{v-1,2}2^{v-1} + b_{v-2,2}2^{v-2} + \ldots + b_{1,2}2 + b_{0,2},\\
&\vdots\\
X_{s-1} &= b_{v-1,s-1}2^{v-1} + b_{v-2,s-1}2^{v-2} + \ldots + b_{1,s-1}2 + b_{0,s-1},\\
X_s &= b_{v-1,s}2^{v-1} + b_{v-2,s}2^{v-2} + \ldots + b_{1,s}2 + b_{0,s},
\end{aligned}
$$

$b_{r,j} \in \{0,1\}$, $r = 0, 1, \ldots, v-1$; $j = 1, 2, \ldots, s$.

Algorithm 3.1.

Step 1. Are realized v of count operations of units modulo m_i:

$$
\beta_{v-1}^{(i)} = \left|\sum_{j=1}^{s} b_{v-1,j}\right|_{m_i}, \quad \ldots, \quad \beta_1^{(i)} = \left|\sum_{j=1}^{s} b_1^{(i)}\right|_{m_i}, \quad \beta_0^{(i)} = \left|\sum_{j=1}^{s} b_0^{(i)}\right|_{m_i}.
$$

Step. 2. Realizing modular products of values $\beta_r^{(i)}$ $(r = 0, 1, \ldots, v-1)$ on $\left|2^r\right|_{m_i}$ $(r = 0, 1, \ldots, v-1)$:

$$
\begin{aligned}
\hat{\beta}_{v-1}^{(i)} &= \left|\beta_{v-1}^{(i)}\left|2^{v-1}\right|_{m_i}\right|_{m_i},\\
&\vdots\\
\hat{\beta}_1^{(i)} &= \left|\beta_1^{(i)}|2|_{m_i}\right|_{m_i},\\
\hat{\beta}_0^{(i)} &= \beta_0^{(i)}.
\end{aligned}
$$

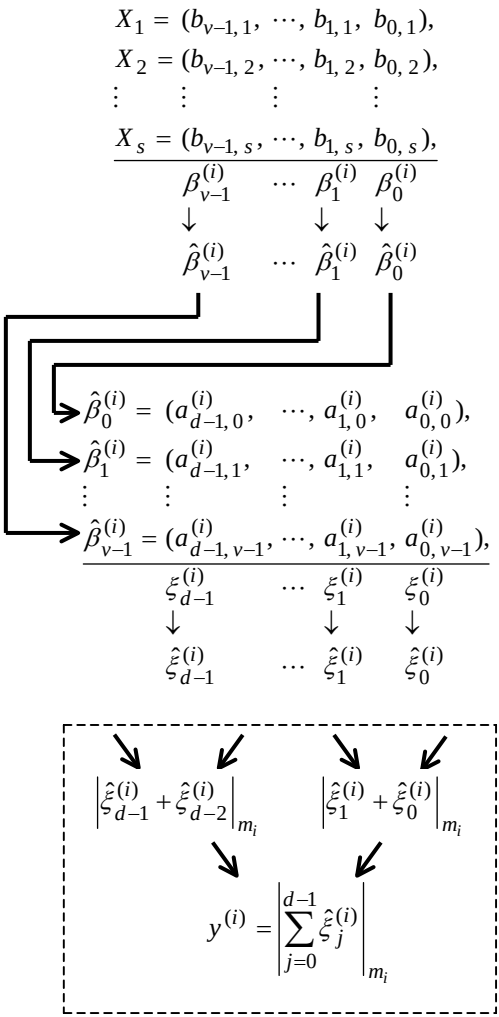
Step 3. Consider, that

$$
\hat{\beta}_0^{(i)} = x_1^{(i)}, \hat{\beta}_1^{(i)} = x_2^{(i)}, \ldots, \hat{\beta}_{v-2}^{(i)} = x_{n-1}^{(i)}, \hat{\beta}_{v-1}^{(i)} = x_n^{(i)}
$$

and fulfil algorithm 2.1.

3.2. "Vertical" *N*-ary the converter of the code of a binary number system in the code modulo m_i

The structure of algorithm 3.1 can be presented as follows:



The block diagram of a parallel summing converter on a figure 6 is represented. The algorithm of operation of a converter of a sequential type is defined by the formula:

$$x^{(i)} = \left|\beta_{v-1}^{(i)} 2^{v-1} + \beta_{v-2}^{(i)} 2^{v-2} + \ldots + \beta_1^{(i)} 2 + \beta_0^{(i)}\right|_{m_i} =$$

$$= \left|\left|\left(\ldots\left|\left(\left|\left(\left|\beta_{v-1}^{(i)} 2\right|_{m_i} + \beta_{v-2}^{(i)}\right)2\right|_{m_i} + \beta_{v-3}^{(i)}\right)2\right|_{m_i} + \ldots + \beta_1^{(i)}\right)2\right|_{m_i} + \beta_0^{(i)}\right|_{m_i}.$$

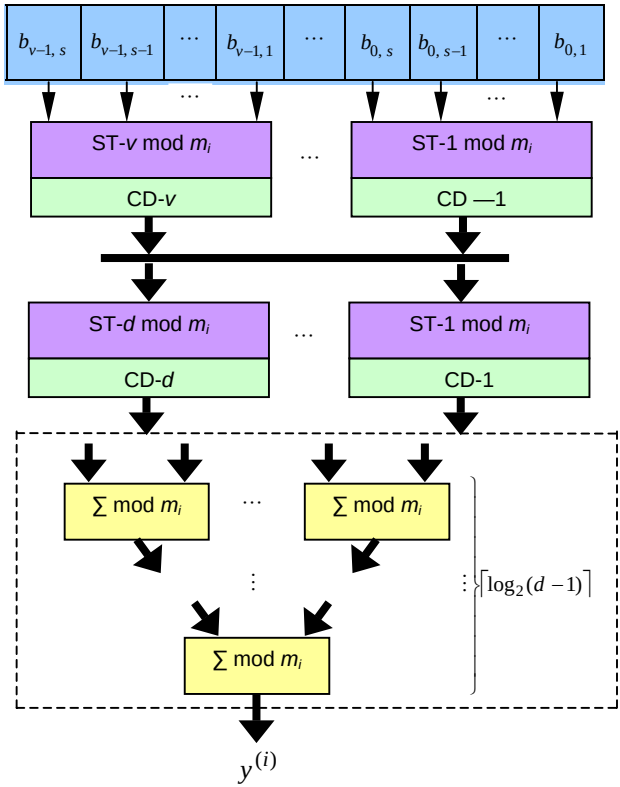


Figure 6. "Vertical" *N*-ary the converter of the code of a binary number system in the code modulo m_i

The schematic diagram of the *N*-ary converter of a sequential type will coincide with the scheme of the "vertical" summator of a sequential type (figure 4). But the sizes of an input of the serialparallel register of shift and amount of inputs of the counter of units ST-modm_i will be

distinguished. An amount of clock ticks of dating pulses necessary for operation of a converter equal *v*.

4. Principle of *N*-ary multiplying modulo m_i

For implementation of *N*-ary multiplying modulo m_i it is possible to use property of a discrete log:

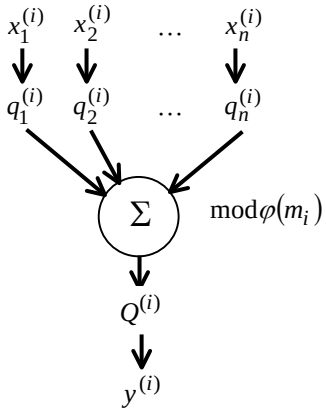
$$\left|\mathrm{Log}_g \prod_{j=1}^{n} x_j^{(i)}\right|_{\varphi(m)} = \left|\sum_{j=1}^{n} \mathrm{Log}_g x_j^{(i)}\right|_{\varphi(m)},$$

where $\mathrm{Log}_g A$ —discrete logarithm from A modulo m_i and basis g (g — primitive radical); $\varphi(m_i)$ — function of Euler from value m_i;

$$\varphi(m_i) = p_1^{\alpha_1 - 1} p_2^{\alpha_2 - 1} \quad \ldots \quad p_z^{\alpha_z - 1}(p_1 - 1)(p_2 - 1)\ldots(p_z - 1),$$

where $p_1, p_2, \ldots, p_z$ —simple factors.

Then the structure of algorithm of *N*-ary multiplying modulo m_i can be presented as follows:



here $q_j^{(i)} = \mathrm{Log}_g x_j^{(i)}$, where $j = 1, 2, \ldots, n$; $Q^{(i)} = \left|\sum_{j=1}^{n} \mathrm{Log}_g x_j^{(i)}\right|_{\varphi(m)}$;

$y^{(i)}$ outcome of calculation of an inverse discrete logarithm.

For implementation of *N*-ary multiplying modulo m_i the device, introduced on a figure 7 can be used. The device contains ROM-1.1 … ROM-1.*n*, intended for storage of values of a discrete logarithm; the *N*-ary summator modulo $\varphi(m_i)$ or $P_{m_i}(m_i)$; ROM-2 intended for storage of outcomes of inverse discrete logarithm.

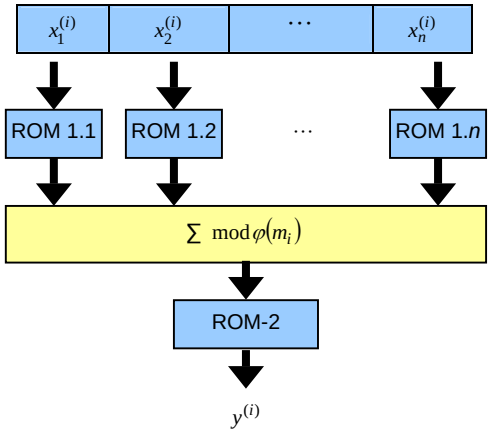


Figure 7. The *N*-ary multiplying tube modulo m_i

5. Conclusion

Thus outcomes obtained in this paper allowed to expand a circle of algorithmic and technical solutions for implementation of *N*-ary operations modulo m_i. The traditional devices use a so-called a "horizontal" method. The new algorithmic and technical solutions realize a "vertical" method. The advantage of a "vertical" method in comparison with a "horizontal" method at essential increase of an amount of operands —*N* is reached. The characteristics of binary arithmetic devices considerably depend on principles of construction of parallel counters of units modulo. The level of complexity of parallel counters is the linear function from number of entry arguments. Significant successes in the theory of synthesis of parallel counters recently are reached. The depth of the scheme was considerably reduced and the speed is raised.

The reached outcomes can become elements of the theory of *N*-ary finite ring computations.

References

1. F.J. Taylor, "Single Modulus ALU for Signal Processing", *IEEE Trans. Acoust., Speech, Signal processing*, vol. ASSP-33, N 5, October 1985, pp. 1302-1315.
2. M.A. Soderstrand, W.K. Jenkins, G.A. Jullien and F.T. Tailor, "*Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*", IEEE Press, New York, 1986.
3. H.J. Nussbaumer, "*Fast Fourier Transform and Convolution Algorithms*", Springer-Verlag, 1982.
4. J.M. Pollard, "The Fast Fourier Transform in a Finite Fild", *Math. Comp*., vol. 25, N 114, 1971, pp.365-374.
5. D.E.R. Denning, "*Cryptography and Data Security*", Adisson-Wesly, Reading MA., 1983.
6. Oleg A. Finko, "Methods of Problem-Oriented Representation and Data Processing in Resources of the Hardware Support of Intellectual Systems", Proceedings of the IEEE Computer Society international Conference on Artificial Intelligence Systems 2002 (IEEE AIS'02). Gelendzhik, Russia, September 5-10, 2002, pp. 453-454.
7. Oleg A. Finko, "Sverkhparallel'nie Logicheskie Vichislenia Metodami Moduliarnoi Arifmetiki (Superparallel Logical Computations by Methods of Modular Arithmetics)", Trudi Mezhdunarodnikh Konfereztii "Iskusstvennie Intellektual'nie Sistemi" (IEEE AIS'02) i "Intellektual'nie SAPR" (CAD-2002), *Izdatel'stvo Fiziko-Matematicheskoi Literaturi*, Moskow, 2002. pp. 448-455.
8. Kai Hwang, "*Computer Arithmetic: Principles, Architecture and Design*'', John Wiley, 1979.
9. D. Zhang, G.A. Jullien, W.C. Miller, "VLSI Implementations of Neural-Like Networks for Finite Ring Computations", Proceedings of the 32nd Mid-West Symposium on Circuits and Systems, Champaign, III, August 14-16, 1989, vol. 1, New York (N. Y.), 1990, pp 485-488.
10. N.S. Szabo and R.I. Tanaka, "*Residue Arithmetic and its Applications to Computer Technology*", McGraw-Hill, New York, 1967.
11. Donald E. Knuth, "*Iskusstvo programmirovaniia, T. 2: Poluchislennye algoritmy*", Moscow: Vil'iams, 2000.
12. A.G. Akritas, "*Elements of Computer Algebra*", John Wiley & Sons, 1989.
13. Oleg A. Finko, "Number Restoration in the System of Residual Classes with a Minimum Number of Radices", *Engineering Simulation*, Overseas Publishers Association, Vol. 16, 1999, pp. 329-334.
14. Author's Certificate 1557682, MKI h 03 M 7/18, "Preobrazovatel' Pozitsionnogo Koda v Kod Sistemy Ostatochnykh Khlassov (Converter of a

Positional code to the Code of the Residue Number Systems)" / O.A. Finko, V.A. Krasnobaeyv, and N.I. Shvetsov. *Bull*. No. 14, Published April 15, 1990. (In Russian).
15. Oleg A. Finko, "Check and Reconfiguration of Analog-to-Digital Devices Operating in the System of Residual Classes", *Engineering Simulation*, Overseas Publishers Association, Vol. 18, 2001, pp. 631-543.
16. Author's Certificate 1383365, MKI G 06 F 11/10, "Ustroistvo dlia Sviortki po Moduliu (The Device for Convolution Modulo)", / O.A. Finko, N.I. Cherviakov, N.I. Shvetsov, A.V. Palzhev, *Bull*. No. 11, Published March 23, 1988. (In Russian).
17. E. Earl, J.R. Swartzlander, "Parallel couters", *IEEE Trans. Comput*. vol. C-22, N 6, 1973, pp. 1021-1024.
18. S. Dormido, M.A. Canto, "An Upper Bound for the Syntesis of Generalized Parallel Couters", *IEEE Trans. Comput*. vol. C-34, N 8, 1982, pp. 802-805.
19. O.A. Muzichenko, "Ispol'zovanie Simmetrii Peremennikh Dlia Umen'shenia Slozhnosti Logicheskikh Skhem", *Avtomatika i Telemekhanika*, N 10, 2000, pp. 151-163.
20. O.A. Muzichenko, "Sintez Logicheskikh Skhem Modul'nogo Kontrolia v Unitarnikh Pozitsionnikh Dvoichnikh Kodakh", *Avtomatika i Telemekhanika*, N 3, 2001, pp. 159-173.

УДК. 681.3 Р.Г.Бияшев, Е.В.Горковенко, С.Е.Нысанбаева



Алгоритмы шифрования сообщений и формирования электронной цифровой подписи с заданной криптостойкостью

(Институт проблем информатики и управления МОН РК, г. Алма-Ата)

Предложены алгоритмы шифрования сообщений и формирования электронной цифровой подписи в непозиционной полиномиальной системе счисления. Определена криптостойкость алгоритмов, зависящая от длины ключа, выбранных полиномиальных оснований и их распределения. Приведена блок-схема, объединяющая обе процедуры

The algorithms of messages ciphering and formation of electronic digital signature in non-positional polynomial notation are proposed. The cryptostability of algorithms depending on the key length, choosed polynomial bases and their distribution is determined. The block diagram combining both procedures is presented.

При хранении, передаче и обмене электронной информацией в информационных сетях и системах возникают проблемы обеспечения ее конфиденциальности (защиты от атак), установления аутентификации (подлинности) автора и ее

целостности (отсутствия изменений в полученном электронном сообщении). Конфиденциальность может быть обеспечена применением криптографических методов (шифрования). Задачу установления целостности сообщения и подлинности его автора позволяет эффективно решать электронная цифровая подпись (ЭЦП) – относительно короткая дополнительная информация, передаваемая вместе с подписанным текстом.

Существуют различные алгоритмы шифрования сообщений и формирования (создания) электронной цифровой подписи. Государственный стандарт Республики Казахстан (СТ РК 1073-2002) [1] на средства криптографической защиты информации (СКЗИ) распространяется на те средства, которые предназначены для:

- защиты передаваемых или хранимых конфиденциальных данных;
- контроля целостности передаваемых, используемых или хранимых данных или программного обеспечения;
- аутентификации, в том числе отказа от авторства или приписывания авторства одним субъектом другому;
- генерации, формирования, распределения или управления ключами.

В зависимости от степени защиты информации для СКЗИ стандартом устанавливаются 4 уровня безопасности. Каждый уровень определяется конкретным материальным ущербом в зависимости от нарушения защищенности информации путем разглашения, навязывания или несанкционированного изменения конфиденциальных данных. Указаны также общие требования к СКЗИ, в том числе и для каждого уровня безопасности, часть из которых приведена в таблице 1.

Длина ключа является, безусловно, одним из показателей криптостойкости алгоритмов шифрования сообщений и создания ЭЦП, но не самым лучшим. Правильнее, на наш взгляд, в качестве критерия использовать не длину ключа, а криптостойкость алгоритма.

Таблица 1.

Уровень безопасности	Ущерб от разглашения, навязывания или несанкционированного изменения информации	Вычислительная сложность алгоритма ВКЗ	Длина ключа алгоритмов, бит		Длина хэш-кода бит	Длина ЭЦП, бит
			Симмет-ричные	Асимметричные		
1	$\leq 10^2$ МРК	$\geq 2^{48}$	≤ 56	≤ 384	≤ 112	≤ 112
2	$\leq 10^4$ МРК	$\geq 2^{96}$	≤ 112	≤ 1536	≤ 160	≤ 160
3	$\leq 10^6$ МРК	$\geq 2^{128}$	≤ 168	≤ 3072	≤ 256	≤ 256
4	$\leq 10^8$ МРК	$\geq 2^{192}$	≤ 256	≤ 8192	≤ 320	≤ 512

МРК - минимальный расчетный показатель,
ВКЗ – вскрытие криптографической защиты.

Известные методы шифрования, схемы формирования ЭЦП и стандарты разработаны для позиционных систем счисления. Существенно повысить криптостойкость алгоритмов шифрования, а также сократить длину хэш-значений и электронной цифровой подписи позволяют нетрадиционные методы криптографии на основе непозиционной полиномиальной системы счисления.

Предлагаются алгоритмы шифрования текста сообщений и формирования электронной цифровой подписи в непозиционной полиномиальной системе, в которой криптостойкость зависит не только от длины ключа, но и от выбранной системы полиномиальных оснований, а также их распределения (порядка следования).

Поскольку шифрование является составной частью формирования ЭЦП, то оба алгоритма целесообразно представить одной блок-схемой. Для этого рассмотрим этапы процессов шифрования сообщения заданной длины и формирования ЭЦП. Предполагаем, что электронное сообщение и ЭЦП имеют соответственно длины N и N_1 бит, причем N_1 намного меньше N, а база данных (БД) неприводимых многочленов содержит все неприводимые полиномы с двоичными коэффициентами степени не выше N.

Процедура шифрования сообщения заданной длины N состоит из двух этапов:

1. выбор системы полиномиальных оснований и порядка их

следования;

2. генерация гаммы с использованием ГПСЧ.

Эти два этапа описывают выбор одной (или одного варианта) системы оснований. Суть их состоит в следующем.

1. Пусть $p_1(x), p_2(x),..., p_S(x)$, $1 \le S \le N$, неприводимые многочлены с двоичными коэффициентами, используемые в качестве основного (рабочего) диапазона. Тогда сообщение длиной *N* можно интерпретировать как последовательность остатков $\alpha_1(x), \alpha_2(x),..., \alpha_S(x)$ от деления некоторого многочлена *F(x)* на рабочие основания $p_1(x), p_2(x),..., p_S(x)$ соответственно.

2. Ключевая последовательность длиной *N* бит также интерпретируется как последовательность остатков $\beta_1(x), \beta_2(x),..., \beta_S(x)$, но от деления некоторого другого многочлена *G(x)* по тем же рабочим основаниям системы. Тогда в качестве криптограммы $\omega_1(x), \omega_2(x),..., \omega_S(x)$ может рассматриваться некоторая функция *H(F(x),G(x))* , операции которой, в соответствии с операциями непозиционной системы счисления, выполняются параллельно по модулям полиномов, выбранных в качестве оснований системы.

Для процесса шифрования информации полным ключом, кроме многочлена *G(x)*, является и конкретный набор оснований, выбранных из всего множества неприводимых многочленов степени не выше *N*.

Общее число всех возможных и отличающихся друг от друга вариантов выбора систем оснований определяет криптостойкость алгоритма шифрования.

Пусть n_1 - число неприводимых многочленов с двоичными коэффициентами степени m_1. Полные системы вычетов по модулям этих многочленов содержат все многочлены с двоичными коэффициентами степени не выше $m_1 - 1$, для записи которых используется m_1 бит [2]. Пусть соответственно n_2 - число неприводимых многочленов с двоичными коэффициентами

степени m_2, n_3 - число неприводимых многочленов с двоичными коэффициентами степени m_3 и т.д., n_S - число неприводимых многочленов степени m_S. При $S=N$ (степень оснований равна N) для записи полных систем вычетов по модулям этих оснований необходимо N бит.

Тогда процедура выбора системы рабочих оснований сводится к нахождению коэффициентов в уравнении

$$k_1 p^{m_1}(x) + k_2 p^{m_2}(x) + ... + k_S p^{m_S}(x) = N, \qquad (1)$$

(где $0 \le k_i \le n_i$, $p^{m_j}(x)$ - многочлен степени m_j, $1 \le m_j \le S$, $k = k_1 + k_2 + ... + k_S$), определяющем количество k неприводимых многочленов из БД различных степеней, которые можно выбрать в качестве оснований системы, запись вычетов по которым покрывает длину заданного сообщения N.

С ростом порядка неприводимых многочленов с двоичными коэффициентами их количество стремительно растет (таблица 2), в связи с чем очевиден широкий выбор решений уравнения (1).

Таблица 2.

Степень неприводимых многочленов	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	…
Количество неприводимых многочленов	1	1	2	3	6	9	18	30	56	120	240	488	972	1938	3876	7749	…
Количество бит, покрываемых неприводим. многочленами	1	2	6	12	30	54	126	240	504	1200	2640	5856	12636	27132	58140	123984	…

Для определения криптостойкости зашифрованного сообщения нужно найти число способов отбора оснований. Число различных комбинаций выбора оснований для какой-либо одной степени определяется k_i-сочетаниями из всех n_i неприводимых многочленов степени m_i. В непозиционных системах счисления существенен и порядок расположения оснований, поэтому число

систем из k выбранных оснований будет равно

$$Z_1 = (k_1 + k_2 + ... + k_s)!\, C_{n_1}^{k_1}\, C_{n_2}^{k_2} \dots C_{n_s}^{k_s}, \qquad (2)$$

Тогда все варианты шифрования (выбор систем оснований и гаммы, распределение оснований) определится соотношением

$$2^N \sum_{k_1,k_2,...k_s} (k_1 + k_2 + ... + k_s)!\, C_{n_1}^{k_1}\, C_{n_2}^{k_2} \dots C_{n_S}^{k_S}, \quad (3)$$

в котором суммирование распространено на всевозможные комбинации целых положительных чисел $k_1, k_2, ..., k_S$, удовлетворяющих равенству (1), т.е. на все выборы систем оснований из числа неприводимых полиномов с двоичными коэффициентами степени $\le N$.

Обратная величина выражения (3) определяет криптостойкость шифрования сообщения длины N:

$$p_{kr} = \frac{1}{2^N \sum\limits_{k_1,k_2,...,k_s} (k_1 + k_2 + ... + k_S)!\, C_{n_1}^{k_1} C_{n_2}^{k_2} ... C_{n_S}^{k_S}}, \qquad (4)$$

Для длины сообщения в 256 байт можно, например, выбрать 80 многочленов 16-й степени, 60 многочленов 12 степени и 6 многочленов 8-й степени, т.е. всего 146 многочленов. В этом случае криптостойкость определяется выражением

$$p_{kr} = \frac{1}{2^{2048} 146!\, C_{7749}^{80} C_{488}^{60} C_{30}^{6}} \approx \frac{1}{10^{1146}},$$

что значительно меньше любой разумной величины, которая может быть задана в реальных условиях для системы шифрования.

Процедура формирования ЭЦП включает в себя три этапа:

1. восстановление функции $F(x)$: выбор системы полиномиальных оснований для сообщения длины N;
2. хэширование (сжатие) сообщения длины N до длины N_1 путем вычисления вычетов $F(x)$ по избыточным основаниям;
3. шифрование хэш-значения: выбор системы полиномиальных

оснований и их размещения, генерация гаммы с использованием ГПСЧ.

Рассмотрим подробнее содержание перечисленных этапов.

1. Этот этап полностью совпадает по содержанию, а поэтому и по обозначениям, с первым этапом процедуры шифрования. Определение (восстановление) многочлена *F(x)* производится по формуле

$$F(x) = \sum_{i=1}^{S} \alpha_i(x) B_i(x), \quad \text{где}$$

$$B_i(x) = \frac{\prod_{i=1}^{S} p_i(x)}{p_i(x)} M_i(x) \equiv 1 (\mathrm{mod}\, p_i(x)),$$

значения многочленов M_i (*x*) выбираются для выполнения сравнения [3].

2. Хэширование сообщения производится расширением на избыточные основания $p_{S+1}(x), p_{S+2}(x), ..., p_{S+U}(x)$, $1 \le U \le N_1$, выбранные произвольно из всех неприводимых многочленов степени, не превышающей N_1. Эта система оснований формируется независимо от выбора оснований $p_1(x), p_2(x), ..., p_S(x)$, но среди *U* избыточных оснований могут быть и совпадающие с некоторыми из рабочих. Вычеты $\alpha_{S+1}(x), \alpha_{S+2}(x), ..., \alpha_{S+U}(x)$ от деления восстановленного многочлена *F(x)* на дополнительные основания $p_{S+1}(x), p_{S+2}(x), ..., p_{S+U}(x)$ определяют длину хэш-значения N_1. Как видно, этот пункт повторяет первый этап шифрования.

3. Завершающим этапом создания ЭЦП является шифрование хэш-значения. Описание шифрования приводится в других обозначениях, так как и на этом этапе формирование системы полиномиальных оснований происходит независимо от выбора рабочих оснований.

3.1. Выбирается система оснований $r_1(x), r_2(x), ..., r_W(x)$, $1 \le W \le N_1$, из числа неприводимых многочленов с двоичными коэффициентами степени не выше N_1. В состав оснований

$r_1(x), r_2(x), ..., r_W(x)$ могут попасть некоторые многочлены как из рабочих оснований $p_1(x), p_2(x), ..., p_S(x)$, так и из избыточных $p_{S+1}(x), p_{S+2}(x), ..., p_{S+U}(x)$. Хэш-значение длины N_1 интерпретируется как последовательность остатков $\gamma_1(x), \gamma_2(x), ..., \gamma_W(x)$ от деления некоторого многочлена $F_1(x)$ на выбранные основания $r_1(x), r_2(x), ..., r_W(x)$ соответственно.

3.2. Ключевая последовательность генерируется с длиной N_1 и интерпретируется как последовательность остатков $\eta_1(x), \eta_2(x), ..., \eta_W(x)$ от деления некоторого полинома $G_1(x)$ на те же основания $r_1(x), r_2(x), ..., r_W(x)$. Тогда полученная в результате шифрования криптограмма $\lambda_1(x), \lambda_2(x), ..., \lambda_W(x)$ может быть представлена как некоторая функция $H_1(F_1(x), G_1(x))$.

С учетом перечисленных этапов все варианты формирования ЭЦП будут описываться выражением

$$2^{N_1} \sum_{k_1,k_2,...k_s} ((\,(k_1 + k_2 + ... + k_s)!\; C_{n_1}^{k_1}\, C_{n_2}^{k_2} \ldots C_{n_S}^{k_S} \times$$

$$\times \sum_{t_1,t_2,...t_U} (t_1 + t_2 + ... + t_U)!\; C_{d_1}^{t_1}\, C_{d_2}^{t_2} \ldots C_{d_U}^{t_U}\,) \times$$

$$\times \sum_{v,v_2,...,v_W} (v_1 + v_2 + ... + v_W)!\; C_{l_1}^{v_1}\, C_{l_2}^{v_2} \ldots C_{l_W}^{v_W}\,), \qquad (5)$$

В формуле (5) суммирование:

– $\sum_{t_1,t_2,...t_U}$ распространено на всевозможные комбинации целых положительных чисел $t_1, t_2, ..., t_U$, удовлетворяющих равенству (аналогу формулы (1))

$$t_1 p^{a_1}(x) + t_2 p^{a_2}(x) + ... + t_U p^{a_U}(x) = N_1,$$

где $a_1, a_2, ..., a_U$ и $d_1, d_2, ..., d_U$ - соответственно степени и число

неприводимых многочленов, используемых при выборе избыточных оснований, $0 \le t_i \le d_i$, $p^{a_j}(x)$ - многочлен степени a_j, $1 \le a_j \le U$, $t = t_1 + t_2 + ... + t_U$ - число избыточных оснований системы, запись вычетов по которым покрывает хэш-значение длины N_1;

– $\sum_{v, v_2, ..., v_W}$ производится по всевозможным комбинациям целых положительных чисел $v_1, v_2, ..., v_W$, определяемых из равенства

$$v_1 r^{b_1}(x) + v_2 r^{b_2}(x) + ... + v_W r^{b_W}(x) = N_1,$$

где $b_1, b_2, ..., b_W$ и $l_1, l_2, ..., l_W$ - степени и число неприводимых многочленов соответственно, используемых при выборе оснований $r_1(x), r_2(x), ..., r_W(x)$, $0 \le v_i \le l_i$, $r^{b_j}(x)$ - многочлен степени b_j, $1 \le b_j \le W$, $v = v_1 + v_2 + ... + v_W$ - система оснований системы, запись вычетов по которым покрывает шифруемое хэш-значение длины N_1.

Криптостойкость формирования ЭЦП определяется обратной величиной (5)

$$p_{sig} = \frac{1}{2^{N_1} \sum_{k_1, k_2, ..., k_s} ((Z_1 \sum_{t_1, t_2, ..., t_U} Z_2) \sum_{v_1, v_2, ..., v_W} Z_3)}, \qquad (6)$$

где $Z_2 = (t_1 + t_2 + ... + t_U)! \, C_{d_1}^{t_1} \, C_{d_2}^{t_2} \ldots C_{d_U}^{t_U}$, $Z_3 = (v_1 + v_2 + ... + v_W)! \, C_{l_1}^{v_1} \, C_{l_2}^{v_2} \ldots C_{l_W}^{v_W}$.

Выражение (6) показывает возможность формирования ЭЦП существенно меньшей длины, чем указано в СТ РК, при сохранении, а при необходимости и увеличения ее надежности.

На рис. 1 приводится блок-схема реализации описанных процедур шифрования сообщений и формирования ЭЦП с заданной криптостойкостью $P_{зад}$. Выбор системы оснований реализуют блоки 3-7. Процедура шифрования сообщения задается переменной Kr и блоками 8, 12-15, 20, 21, а формирования ЭЦП – переменной

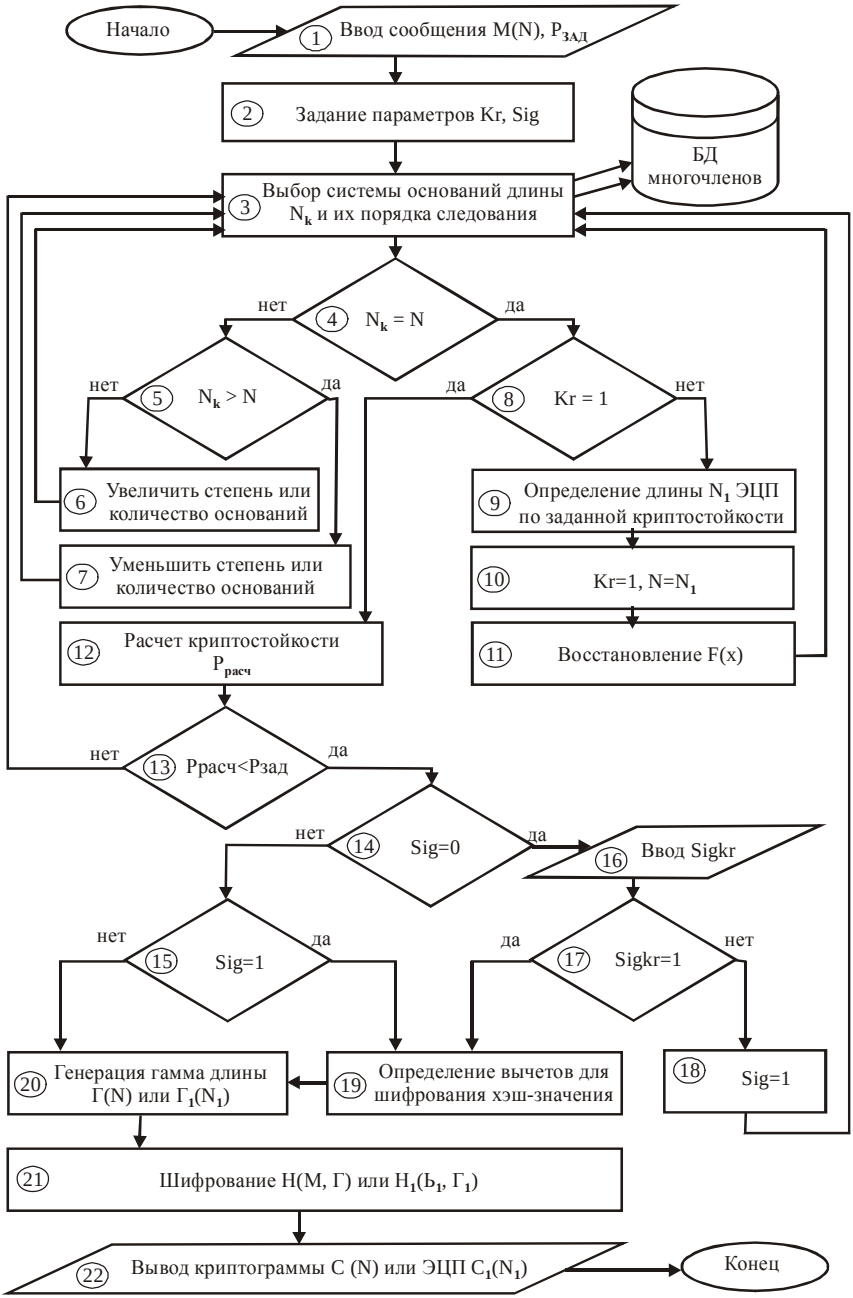


Рис. 1. Блок-схема шифрования сообщений и формирования ЭЦП

Sig и блоками 8-21. При создании ЭЦП в процедуре шифрования хэш-значений учтена также возможность применения как использованных при вычислении хэш-значений избыточных оснований, так и выбираемых через блок 3 других оснований (переменная Sigkr, блоки 15-19).

Литература

1. СТ РК 1073-2002 «Средства криптографической защиты информации» / Общие технические требования. - Астана: Госстандарт РК, 2002.
2. Гр. К. Моисил. Алгебраическая теория дискретных автоматических устройств. - М: Издательство иностранной литературы, 1963.
3. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. – М.: Советское радио, 1968.

УДК. 681.3

Р.Г.Бияшев, Р.К.Нысанбаев, Р.В. Егай



Применение модулярного шифрования в комплексе тестирования абитуриентов

(Институт проблем информатики и управления МОН РК,

г. Алма-Ата)

Осуществлено шифрование файлов с правильными ответами для их записи на магнитные носители и расшифровки при их считывании с обнаружением ошибок. Доступ к процедуре дешифрования производится в случае совпадения трех паролей. Алгоритмы шифрования и дешифрования построены на базе полиномиального варианта кодов Лагранжа.

Ciphering of the files with the right answers for their recoding on magnetic mediums and deciphering in their reading with the errors detection is realized. The access to the deciphering procedure is made in the case of coincidence of tree passwords. The ciphering and deciphering algorithms are constructed by the polynomial variant of Lagrangian codes.

Представляются программные модули, осуществляющие шифрование файлов с правильными ответами для их записи на магнитные диски и расшифровки при их считывании с обнаружением ошибок. При этом доступ к процедуре дешифрования разрешается лишь в случае совпадения трех паролей, вводимых на этапе шифрования

файлов.

Программные модули созданы для практической реализации криптографического метода и алгоритмов шифрования и дешифрования, основанных на использовании непозиционной мультипликативной композиции векторов и их представления в виде интерполяционных полиномов Лагранжа, а также метода помехоустойчивого кодирования на основе (n,k)-кода Лагранжа, примененного для обнаружения и исправления одиночных ошибок в криптограммах.

Отдельно был создан модуль, реализующий формирование электронной цифровой подписи (Уранаев Н.Т., Бияшев Р.Г., Малько К.А.) для обнаружения несанкционированных изменений в листах ответов абитуриентов после их сканирования и записи в БД.

Модули были встроены в программный комплекс тестирования абитуриентов Республики Казахстан и уже в течение ряда лет используются в нем без каких-либо изменений.

Криптографический метод шифрования и дешифрования информации

Вводится следующая интерполяция векторов линейного пространства E_n. Пусть $\omega_1, \omega_2, ..., \omega_n$ - n различных элементов поля F_q $(n \le q)$, упорядоченных некоторым образом.

Под вектором $\vec{a} = (\alpha_1, \alpha_2, ..., \alpha_n)$ понимается многочлен $a(x)$, который в точках $\omega_1, \omega_2, ..., \omega_n$ принимает значения $\alpha_1, \alpha_2, ..., \alpha_n$ соответственно. Тогда под композицией $\vec{a} * \vec{b}$ векторов $\vec{a} = (\alpha_1, \alpha_2, ..., \alpha_n)$ и $\vec{b} = (\beta_1, \beta_2, ..., \beta_n)$ понимается многочлен, принимающий в точках $\omega_1, \omega_2, ..., \omega_n$ значения $\alpha_1 * \beta_1, \alpha_2 * \beta_2, ..., \alpha_n * \beta_n$.

Суть криптографического метода заключается в следующем. Пусть $p = 2$, $p(x)$ - неприводимый многочлен степени m, порождающий поле Галуа $GF(2^m)$.

Предположим, что шифруется некоторое исходное сообщение M длины L двоичных разрядов. Сообщение M интерпретируется как многочлен

$$M(x) = (\alpha_1(x), \alpha_2(x), ..., \alpha_n(x)), \qquad (1)$$

где $\alpha_i(x), i = \overline{1,n}$ - значения, которые он принимает в точках $\omega_i \in GF(2^m), i = \overline{1,n}$. В (1) $\alpha_1(x), \alpha_2(x), ..., \alpha_n(x)$ выбираются так, что первым l_1 битам слова M ставятся в соответствие двоичные коэффициенты многочлена $\alpha_1(x)$, следующим l_2 битам – двоичные коэффициенты $\alpha_2(x)$, и так далее, наконец, последним l_n двоичным разрядам ставятся в соответствие двоичные коэффициенты полинома $\alpha_n(x)$.

В процессе шифрования исходного сообщения длина секретного ключа K задается равной длине сообщения. Тогда ключ, также как и исходное сообщение M интерпретируется как многочлен

$$K(x) = (\beta_1(x), \beta_2(x), ..., \beta_n(x)), \quad (2)$$

где $\beta_i(x), i = \overline{1,n}$ - значения, которые принимает многочлен $K(x)$ в точках $\omega_i \in GF(2^m), i = \overline{1,n}$.

Умножая (1) и (2) в выбранном поле, получаем некоторый многочлен

$$F(x) = M(x)K(x), \qquad (3)$$

значения которого в узлах ω_i вычисляются посимвольно

$$\gamma_i(x) \equiv \alpha_i(x)\beta_i(x) \text{ (modd 2, } p(x)), \quad (4)$$

тогда можно записать

$$F(x) = (\gamma_1(x), \gamma_2(x), ..., \gamma_n(x)). \qquad (5)$$

Здесь $\alpha_i(x), \beta_i(x), \gamma_i(x)$ - элементы поля $GF(2^m)$. Из этого

следует, что количество двоичных разрядов, соответствующих двоичным коэффициентам представления $\alpha_i(x), \beta_i(x), \gamma_i(x)$ для всех $i = \overline{1,n}$ имеет одинаковую длину, равную m, т.е. $l_i = m,\ i = \overline{1,n}$.

Двоичная запись длины L полученного многочлена $F(x)$ - зашифрованное сообщение (криптограмма).

В данном случае подлежат проверке $2^L \cdot n$ ключей, где L - длина ключа, а n - количество неприводимых многочленов с двоичными коэффициентами степени m.

Для дешифрования необходимо найти такой многочлен $K^{-1}(x)$, удовлетворяющий следующему сравнению

$$K^{-1}(x)F(x) \equiv M(x) \text{ (modd 2, } p(x)\text{)}, \qquad (6)$$

где

$$K^{-1}(x) = (\beta_1^{-1}(x), \beta_2^{-1}(x), ..., \beta_n^{-1}(x)). \qquad (7)$$

Для многочленов $\beta_i^{-1}(x), i = \overline{1,n}$ справедливы сравнения

$$K^{-1}(x) \equiv \beta_i^{-1}(x) \text{ (modd 2, } p(x)\text{)},$$

или

$$\beta_i^{-1}(x)\beta_i(x) \equiv 1 \text{ (modd 2, } p(x)\text{)}. \qquad (8)$$

Таким образом, определение элементов $\beta_i^{-1}(x), i = \overline{1,n}$ производится по выражению (8). Затем по формуле (6) восстанавливается значение многочлена $M(x)$, который и является исходным сообщением.

Алгоритм обнаружения и исправления одиночных ошибок в криптограммах на основе (n,k) - кода Лагранжа

Рассматривается код Лагранжа для исправления одиночных оши-

бок в криптограммах с использованием двух контрольных символов. Под ошибкой понимается любое искажение информационного или контрольного символа.

Пусть $\gamma_1(x), \gamma_2(x), ..., \gamma_n(x)$ - информационные символы, тогда контрольные символы вычисляются в соответствии с выражениями

$$\gamma_{n+1}(x) = \sum_{i=1}^{n} \gamma_i(x) L^{(i)}(x_{n+1}),$$

$$\gamma_{n+2}(x) = \sum_{i=1}^{n} \gamma_i(x) L^{(i)}(x_{n+2}), \quad (9)$$

и зашифрованное сообщение

$$\gamma_1(x), \gamma_2(x), ..., \gamma_n(x), \gamma_{n+1}(x), \gamma_{n+2}(x) \quad (10)$$

хранится, пересылается или передается на обработку. Здесь $\forall \gamma_i, L^{(i)}(x_{n+1}),\ L^{(i)}(x_{n+2}), 1 \le i \le n,$ являются элементами заданного конечного поля $GF(2^m)$. Если в кодовом слове произошла ошибка в i - том символе, $\overline{\gamma_i} = \gamma_i + \Delta\gamma_i$, то после повторного вычисления контрольных символов $\gamma'_{n+1}, \gamma'_{n+2}$ можно найти

$$\nabla_1 = \gamma'_{n+1}(x) \oplus \gamma_{n+1}(x) = \Delta\gamma_i(x) L^{(i)}(x_{n+1}),$$

$$\nabla_2 = \gamma'_{n+2}(x) \oplus \gamma_{n+2}(x) = \Delta\gamma_i(x) L^{(i)}(x_{n+2}). \quad (11)$$

Невязки ∇_1, ∇_2 однозначно определяют величину и местоположение ошибки. Очевидно, что если $\nabla_1 = \nabla_2 = 0$, ошибки нет; если $\nabla_1 \neq 0, \nabla_2 = 0$ или $\nabla_1 = 0, \nabla_2 \neq 0,$ ошибка соответственно в первом или втором контрольном символе; если

$\nabla_1 \neq 0, \nabla_2 \neq 0$, то ошибка произошла в одном из информационных символов.

Если выбрать $x_{n+1} = 0, \quad x_{n+2} = 1$, то

$$L^{(i)}(x_{n+1}) = 1, \qquad L^{(i)}(x_{n+2}) = i, \quad i = \overline{1,n}.$$

Выражение (9) для контрольных символов принимает вид

$$\gamma_{n+1}(x) = \sum_{i=1}^{n} \gamma_i(x),$$

$$\gamma_{n+2}(x) = \sum_{i=1}^{n} i\gamma_i(x), \tag{12}$$

а выражения для невязок (11) соответственно:

$$\nabla_1 = \gamma'_{n+1}(x) \oplus \gamma_{n+1}(x) = \Delta\gamma_i(x),$$

$$\nabla_2 = \gamma'_{n+2}(x) \oplus \gamma_{n+2}(x) = i\Delta\gamma_i(x). \tag{13}$$

В этом случае первое равенство (13) определяет величину ошибки, а номер ошибочного символа определяется умножением второго равенства (13) на мультипликативную инверсию величины ошибки, т.е.

$$\nabla_2 \nabla_1^{-1} = i\Delta\gamma_i(x)(\Delta\gamma_i(x))^{-1} = i.$$

Практическая реализация криптографического метода

Для описанного криптографического метода и алгоритма обнаружения ошибок в криптограммах на основе (n,k) - кода Лагранжа с целью шифрования и дешифрования информации, хранящейся на магнитных дисках, создано программное обеспечение на языке Delphi 4.0, работающее под операционной системой Windows

95/98. Программы шифруют и расшифровывают данные по блокам длиной $n = 254$ байта. Соответственно длина ключа также равна $n = 254$ байта. В качестве модуля был выбран неприводимый многочлен восьмой степени $p(x) = x^8 + x^4 + x^3 + x + 1$.

Программное обеспечение удовлетворяет следующим требованиям: 1) доступ к процедуре дешифрования осуществляется лишь в случае совпадения трех паролей; 2) если в зашифрованном файле с кодами правильных ответов появились любого рода ошибки, то при считывании файла с магнитного диска на экране монитора выдается соответствующее сообщение, и в этом случае дешифрование искаженного шифрованного файла не выполняется.

Описание алгоритма шифрования

Суть алгоритма шифрования заключается в следующем.

1) Начало выполнения программы.

2) Ввод имени исходного файла, содержащего исходное сообщение M.

3) Задание имени шифрованного (выходного) файла, в который будет записываться зашифрованное сообщение.

4) Ввод трех паролей. В случае неправильного их задания, ввод паролей повторяется.

5) Ввод исходного сообщения M из исходного файла по блокам длиной n байт каждый.

6) Задание модуля $p(x)$ любым неприводимым многочленом восьмой степени. Шифрование производится поблочно. Каждому байту исходного сообщения ставятся в соответствие полиномы $\alpha_i(x)$, $i = \overline{1,n}$, которые являются остатками по модулю $p(x)$. Коэффициентами остатков $\alpha_i(x)$ будут биты соответственных байтов исходного сообщения.

7) Ввод ключа K, в качестве которого выбирается произвольная последовательность длины n байт. Тогда по аналогии с пунктом 6 каждому байту ключа K ставятся в соответствия

многочлены $\beta_i(x)$, $i=\overline{1,n}$.

8) Умножение каждого из многочленов $\alpha_i(x)$ на соответственный многочлен $\beta_i(x)$, $i=\overline{1,n}$. В результате получаются некоторые многочлен $A_i(x)=\alpha_i(x)\beta_i(x)$, $i=\overline{1,n}$, двоичные коэффициенты которых представляют собой пятнадцатиразрядное двоичное слово.

9) Вычисление по формуле (4) значений многочленов $\gamma_i(x)$, $i=\overline{1,n}$. Двоичные коэффициенты полученных остатков $\gamma_i(x)$ представляют собой зашифрованные байты.

10) Обнаружение ошибок в криптограмме: для этого используются два избыточных узла $x_{n+1}=0$ и $x_{n+2}=1$. Тогда контрольные символы $\gamma_{n+1}(x)$ и $\gamma_{n+2}(x)$ рассчитываются по формулам (12).

11) Запись всех вычисленных $\gamma_i(x),\ i=\overline{1,n+2}$ в выходной файл.

12) Проверка завершения считывания исходного сообщения по блокам: когда ввод блоков закончен, то выполнение программы завершается. Иначе повторение выполнения программы, начиная с пункта 8.

13) Конец выполнения программы.

На рис. 1 изображена блок-схема изложенного алгоритма шифрования.

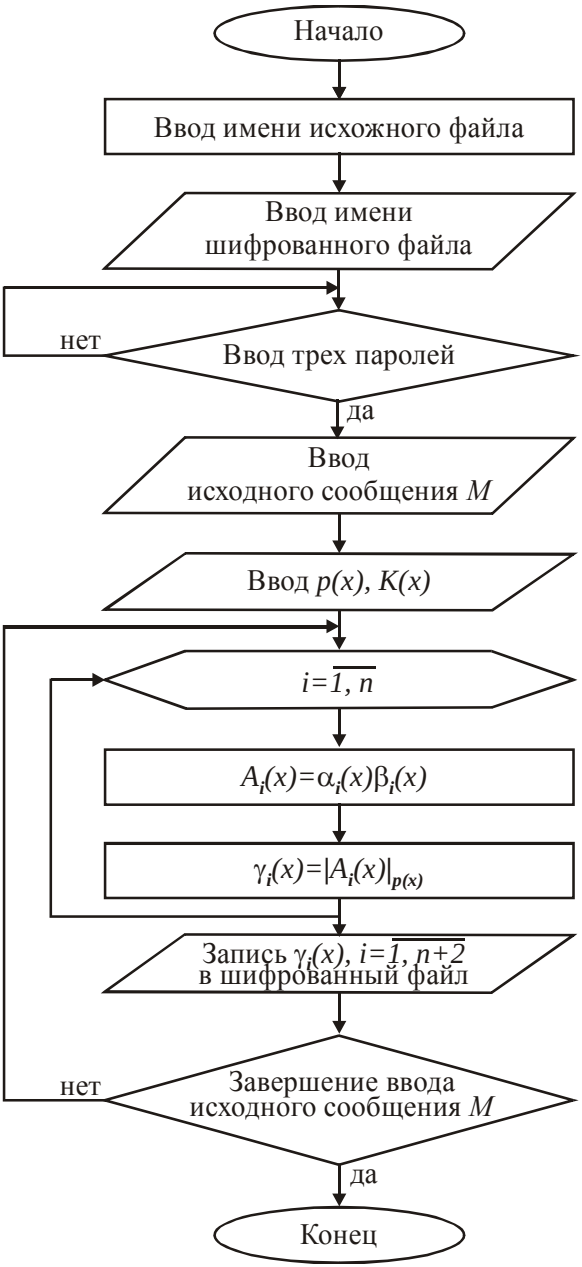


Рис. 1. Блок-схема алгоритма шифрования

Описание алгоритма дешифрования

Алгоритм дешифрования.

1) Начало выполнения программы.

2) Задание имени шифрованного файла, содержащего зашифрованное сообщение F.

3) Ввод имени дешифрованного файла, куда будет записываться расшифрованное сообщение.

4) Ввод трех паролей, которые указывались при шифровании исходного сообщения. Если один или несколько паролей заданы неверно, то необходимо повторить их ввод. Процедура ввода паролей может повторяться только три раза.

5) Если все три пароля введены верно, то задается шифрованное сообщение F по блокам длиной n байт каждый. Дешифрование производится поблочно.

6) Ввод модуля $p(x)$ и ключа $K(x)$, которые использовались в процессе шифрования. Каждому байту криптограммы ставятся в соответствие полиномы $\gamma_i(x)$, $i=\overline{1,n}$, которые являются остатками по модулю $p(x)$. Коэффициентами остатков $\gamma_i(x)$ будут биты соответственных байтов зашифрованного сообщения.

7) Обнаружение ошибок считывания зашифрованного сообщения: для этого вычисляются контрольные символы $\gamma'_{n+1}(x)$ и $\gamma'_{n+2}(x)$ по формулам (12).

8) Если $\gamma_{n+1}(x)=\gamma'_{n+1}(x)$ и $\gamma_{n+2}(x)=\gamma'_{n+2}(x)$, то ошибки в криптограмме нет, а если $\gamma_{n+1}(x)\neq\gamma'_{n+1}(x)$ и(или) $\gamma_{n+2}(x)\neq\gamma'_{n+2}(x)$, то в зашифрованном сообщении произошла ошибка.

9) Если в криптограмме есть ошибки, то выполнение программы завершается. Если же ошибок нет, то для каждого

$\beta_i(x)$ ключа K, $i=\overline{1,n}$ вычисляются инверсные многочлены $\beta_i^{-1}(x)$ по формуле (8).

10) Каждый из многочленов $\beta_i^{-1}(x)$ умножается на соответственный многочлен $\gamma_i(x)$. В результате получаются некоторые многочлены $A_i^{-1}(x),\ i=\overline{1,n}$.

11) По формуле (6) производится восстановление многочленов $\alpha_i(x),\ i=\overline{1,n}$.

12) Двоичные коэффициенты остатков $\alpha_i(x)$ являются байтами расшифрованного, т.е. исходного сообщения.

13) Запись в дешифрованный файл всех вычисленных $\alpha_i(x),\ i=\overline{1,n}$.

14) Проверка завершения считывания шифрованного файла. Если ввод завершен, то работа программы завершается, иначе управление передается на пункт 5.

15) Конец выполнения программы.

Блок-схема алгоритма дешифрования приведена на рис. 2

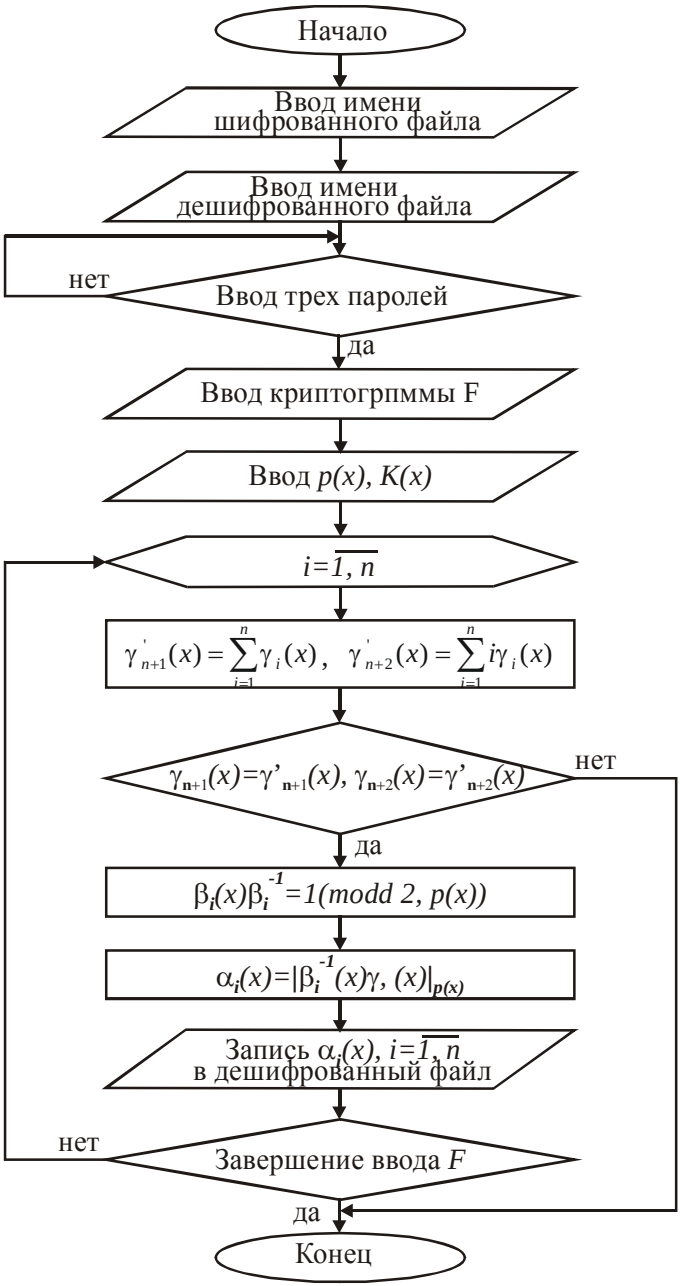


Рис. 2. Блок-схема алгоритма дешифрования

УДК 658.512.011.56

Д.Б. Малашевич

Недвоичные системы в вычислительной технике

(Московский государственный институт электронной техники)

Произведён краткий обзор наиболее перспективных альтернативных не двоичных систем счисления, рассмотрены известные вычислительные системы на их основе. Рассмотрены пути реализации троичной систем.

Первые компьютеры (тогда электронные вычислительные машины – ЭВМ) в шестидесятых-семидесятых годах прошлого века создавались на основе дискретных элементов, сначала электронных ламп, затем полупроводниковых диодов и транзисторов. Дело было принципиально новое и разработчики сами определяли подходы к построению ЭВМ и пути их реализации.

Сначала ЭВМ строились на основе различных реализаций привычной для человека десятичной системы. Но приемлемого варианта десятичного элемента для их построения найдено так и не было. Гораздо экономичнее и проще в аппаратном исполнении оказалась двоичная система счисления, она постепенно и победила. Таким образом, проблемы построения электронной элементной базы изначально определили путь развития вычислительной техники. Это влияние оказывалось решающим на всех этапах ее развития, и не всегда положительным.

Пока элементная база имела дискретный характер (электронные лампы, диоды, транзисторы и т.п.), разработчики ЭВМ имели возможность практической реализации своих идей в построении ЭВМ, и они пользовались этой возможностью. В те времена каждая ЭВМ была уникальна, никакой унификации между ними, тем более общепринятой стандартизации, еще не было. Первые ЭВМ вообще делались в одном экземпляре, затем их начали выпускать промышленно, с ничтожным в нынешнем понимании тиражом. Так первая в СССР серийная ЭВМ «Стрела» выпускалась три года (1953 - 1956 гг.), всего было выпущено 7 экземпляров. А первая «массовая» ЭВМ «Урал-1» выпускалась 5 лет (1956 – 1961 гг.), выпущено 183 машины.

Логические элементы для каждой ЭВМ разработчик проектировал сам. Интеллект разработчика ЭВМ еще ничто и никто не ограничивало, в результате появились и были реализованы самые разнообразные архитектурные, структурные и схемотехнические решения в построении ЭВМ. Не все главные конструкторы ЭВМ поддались легкости реализации двоичной системы счисления. Она несла с собой и серьезные неприятности, например проблемы округления результатов, представление отрицательных чисел, длинные цепочки переносов, затягивающие процесс вычислений, и т.п. Поэтому ученые искали иные пути построения высокопроизводительных ЭВМ, в том числе и в недвоичных системах счисления. Вот некоторые примеры, получившие реальную реализацию, в хронологической последовательности их зарождения.

Недвоичные ЭВМ

Троичные ЭВМ

ЭВМ «Сетунь» [1]**.** В 1959-60 гг. в вычислительном центра МГУ им. М.В. Ломоносова была разработана ЭВМ «Сетунь» (главный конструктор Н.П. Брусенцов). ЭВМ была рассчитана на использование в вузах, научно-исследовательских учреждениях и конструкторских бюро для решения научно-технических задач средней сложности. В 1961 – 65 гг. было выпущено около 50 комплектов ЭВМ «Сетунь».

Сетунь была машиной последовательного действия с блоком быстрого умножения. Ее главные особенности:

- троичная симметричная (с положительными и отрицательны-

ми значениями цифр) система представления чисел и команд,

- трехзначная логика,
- страничная двухуровневая организация памяти,
- пороговая реализация трехзначной логики на электромагнитных элементах с двухпроводной передачей трехзначных сигналов.

При длине слова 9 тритов, что эквивалентно 14,26 бит (1 трит соответствует 1,58 бита [2]) и наборе всего лишь из 24 команд, Сетунь была весьма эффективна при реализации широкого спектра алгоритмов, в том числе с плавающей запятой. Длина операндов - 9 тритов и 18 тритов, троичный порядок числа с плавающей запятой - 5 тритов. Сетунь отличалась от современников естественностью троичной логики, легкостью понимания, освоения и применения, сочетанием простоты и высокой эффективности программирования.

ЭВМ «Сетунь-70». В 1970 г. так же в ВЦ МГУ была разработана ЭВМ «Сетунь-70» с троичной симметричной системой представления данных и программ (главный конструктор Н.П. Брусенцов). Область применения: решение научно-технических задач средней сложности.

Особенности ЭВМ: троичная симметричная система представления данных и программ, трехзначная логика в пороговой реализации на пороговых электромагнитных элементах с однопроводной передачей сигналов, страничная двухуровневая организация памяти, двухстековая архитектура, послоговое кодирование программ, управление ходом программы в духе структурированного процедурного программирования. Набор операций ЭВМ включает 81 операцию: 27 основных (тестирование и преобразование данных, управление ходом программы), 27 служебных (управление магнитным барабаном, внешними устройствами, системой прерываний), 27 макроопераций, микропрограммируемых пользователями. Идентификаторами операций и адресов служат трайты (шестерки тритов).

ЭВМ «Сетунь-70» не была освоена в серийном производстве: не нашлось завода. Опытный образец машины 17 лет проработал на факультете ВМиК МГУ в составе диалоговой системы структурированного программирования «ДССП» и автоматизированной системы обучения "Наставник" до замещения его серийным микрокомпьютером "Электроника НЦ 80-20" (ДВК-2) в 1987 г.

На этом история создания троичных ЭВМ в стране была прервана.

Модулярные ЭВМ

ЭВМ «Т-340А» и «К-340-А» [3]**.** В 1963-64 гг. в НИИ-37 (ныне НИИДАР) по инициативе его директора Ф.В. Лукина был разработан экспериментальный образец первой в стране модулярной ЭВМ Т-340А (главный конструктор Д.И. Юдицкий, научный руководитель И.Я. Акушский). ЭВМ предназначалась для расчетов радиолокационных данных в радиолокаторе дальнего наблюдения системы ПРО. ЭВМ проработала ряд лет в составе полигонного варианта РЛС до его демонтажа.

По результатам испытаний и опытной эксплуатации Т340-А в 1963-66 гг. были произведены определенные доработки и ЭВМ с обозначением «К-340А» была освоена в серийном производстве. В течение 1966 – 73 гг. было выпущено около 50 комплектов ЭВМ «К-340А», ставшей базовой ЭВМ для всех РЛС, разрабатываемых в те годы в НИИ-37. По производительности (2,4 млн. оп/с) К-340А превосходила всех своих современников. ЭВМ и сейчас работают в действующих РЛС.

ЭВМ «Алмаз» и «5Э53». В 1963 г. директор НИИ-37, инициатор разработки ЭВМ Т-340А и К-340А был назначен генеральным директором создаваемого в будущем Зеленограде Центра микроэлектроники. Туда же он пригласил коллектив Д.И. Юдицкого, который по заказу Генерального конструктора ПРО Г.В. Кисунько приступил к разработке высокопроизводительной ЭВМ для второй очереди ПРО Московского промышленного района. Сначала (1967 – 68 гг.) это был эскизный проект ЭВМ «Алмаз» с изготовлением экспериментального образца, а после победы на конкурсе эскизных проектов – разработка технического проекта ЭВМ «5Э53» (1969-71 гг.) с изготовлением опытного образца и серийным производством на Загорском электромеханическом заводе. Главным конструктором обоих ЭВМ был Д.И. Юдицкий.

Производительность Алмаза на задачах ПРО в общепринятом тогда понимании была порядка 30 млн. оп/с, 5Э53 – около 40 млн. оп/с. Это тоже был наивысший в мире в те времена результат. Завод почти завершил подготовку серийного производства и начал изготовление ее устройств, но в 1972 г., в связи со свертыванием работ над второй очередью системы ПРО, изготовление5Э53 было ис-

ключено из плана завода. Другого применения и изготовителя для наиболее производительной в мире ЭВМ не нашлось, ЭВМ была погублена.

Самокорректирующий код

ЭВМ «Корень». В 1964 – 65 гг. в НПО «Агат» была разработана специализированная ЭВМ «Корень» (главный конструктор – А. К. Заволокин, зам. главного конструктора – Е. К. Юферова). ЭВМ предназначалась для корабельной системы ПВО и применялась на авианесущих крейсерах и кораблях противолодочной обороны.

В ЭВМ «Корень» для повышения надежности работы был использован самокорректирующий код, исправляющий одиночные ошибки и обнаруживающий двойные. Арифметическое устройство для реализации арифметических операций было построено на табличном принципе, т. е. все результаты арифметических операций хранились в долговременном запоминающем устройстве и выбирались по входным данным. В машине была обеспечена возможность замены неисправного блока в процессе работы.

СЦВМ “Корень” представляла собой одноадресную машину с фиксированной запятой, быстродействием 10 тысяч операций в секунду, объемом оперативной памяти — 1568 слов, объемом долговременного ЗУ — 8232 слова. Число команд — 24. Машина потребляла 12 кВт электроэнергии. Характеристики, как видим, не очень высокие для того времени, но достаточные для решения поставленных задач. Главным требованием к ЭВМ была сверхвысокая надежность, и она была обеспечена.

Информации о дальнейшем применении самокорректирующего кода в других ЭВМ не имеется.

Влияние микроэлектроники

Таким образом, в истории вычислительной техники нашей страны зафиксировано три случая создания недвоичных ЭВМ (возможно были и другие, но информация о них автору не известна), причем большинство из низ входило в ряд лучших ЭВМ своего времени. Но по разным причинам, далеким от науки, техники и экономики, работы по этим направлениям были директивными методами остановлены. И все эти ЭВМ были построены на основе дискретных элементов, когда разработчик ЭВМ еще имел возможность реали-

зовать любые свои идеи.

Ситуация в технологии создания ЭВМ резко изменилась с появлением интегральных схем высокой сложности и особенно микропроцессоров. С одной стороны, такое развитие микроэлектроники коренным образом повлияло на физические характеристики ЭВМ, переведя их из категории штучной, в категорию массовой продукции. Довольно быстро и успешно пошел процесс унификации и стандартизации присоединительных характеристик функциональных устройств ЭВМ, превративший их в стандартные модули. Процесс создания ЭВМ выродился в простой подбор таких модулей и комплексирование из них требуемого компьютера «без паяльника и осциллографа», появился даже термин, удачно характеризующий этот процесс – «отверточная технология». Все это конечно весьма прогрессивно и положительно, но здесь имеется и другая, негативная сторона. Правила построения ЭВМ стали диктовать поставщики элементной базы, а они могут экономически эффективно работать только при условии массового производства их продукции. В результате острой конкуренции число поставщиков интегральных схем для компьютеров резко сократилось и в настоящее время их общее количество во всем мире ограничено несколькими фирмами. А фирма «Intel» является фактическим глобальным монополистом по определению архитектуры микропроцессоров для персональных компьютеров. В результате количество коллективов, занимающихся практической разработкой архитектурных, структурных и схемотехнических принципов построения ЭВМ, сократилось в мире до нескольких, причем их творческая мысль связана фирменными правилами, традициями, многолетним технологическим заделом. Многочисленные коллективы – генераторы новых архитектурных, структурных и схемотехнических решений в ЭВМ, практически исчезли. Отдельные энтузиасты и их группы из промышленных предприятий, где они могли «в железе» реализовать свои идеи, переместились в учебные заведения, где их деятельность ограничена сугубо теоретическими изысканиями. В какой-то мере это позволяет сохранять творческий потенциал, продолжать разработки перспективных направлений в развитии вычислительной техники и ждать того часа, когда в них появится необходимость. А она назревает.

В настоящее время в специальных изданиях и Интернете отмечается заметное повышение научной и инженерной активности в облас-

ти исследований нетрадиционных путей построения вычислительных средств и систем счисления для применения в вычислительной технике. В том числе и в нашей стране. Это можно объяснить несколькими причинами [3]:

- Во-первых, широкое проникновение вычислительной техники во все сферы жизнедеятельности человека резко повысило актуальность решения таких, ранее редких, а теперь массовых задач, как обработка сигналов, изображений, распознавания образов, криптография, обработка многоразрядной информации и т.п. Все они требуют огромных вычислительных ресурсов, часто превышающих возможности.
- Во-вторых, традиционная микроэлектроника подходит к пределу своих технологических возможностей, размеры ее элементов измеряются нанометрами, числом атомов. А идущие ей на смену наноэлектроника, молекулярная электроника, микромеханика, биоэлектроника и т.п. находятся в "эмбриональном" состоянии, еще далеки от промышленного применения и их перспективы оцениваются по-разному.
- В-третьих – остро встает проблема безопасности. Об этом еще далеко не достаточно говорят, но для России это проблема национальная. Применение зарубежной электроники в стратегически важных системах таит в себе огромную скрытую потенциальную угрозу. Современный уровень микроэлектроники, когда в кристалле одной интегральной схемы содержатся миллионы транзисторов, функционально законченные устройства и системы, обеспечивает и возможности введения диверсионных "закладок". Компьютер с такой "закладкой" может многие годы прекрасно работать, а "закладка" будет спать. Но в нужный кому-то момент, по сигналу извне (Internet, радиосигнал и т.п.) она просыпается и творит с системой все, что захочет ее хозяин. Обнаружить такие "закладки" практически невозможно. Эта задача по силам только мощнейшим в мире микроэлектронным фирмам, стоимость такой операции соизмерима со стоимостью создания исследуемой микросхемы, при обилии номенклатуры таких микросхем задача становится непосильной для экономики любой страны. В настоящее время ни кто не может дать гарантии, что в компьютерах Генштаба, Банка России, Правительства, Федерального собрания и других стратегически важных органов не "спят" диверсионные "закладки", и что они не проснуться в самый неподходящий для страны момент. Выход только один – в

создании отечественных изделий микроэлектроники и стратегически важных систем на их основе. Только здесь процесс можно полностью контролировать и исключить появление "закладок". Но поскольку технологически мы отстаем от зарубежной микроэлектроники, необходимо привлекать другие средства повышения эффективности систем.

В этих условиях интерес к поиску системных методов повышения эффективности и надежности вычислительных средств пробуждается вновь. Ряд серьезных фирм начал, пока теоретические, задельные работы в этой области. Работы ведутся в разных направлениях, но наиболее перспективными представляются модулярные и троичные ЭВМ, уже на практике доказавшие свои ценные особенности. Причем в настоящее время, по предложению главного конструктора троичных ЭВМ "Сетунь" и "Сетунь-70" Н.П. Брусенцова, прорабатывается вопрос об объединении модулярности и троичности. Остановимся кратко на наиболее ценных их свойствах.

Модулярная арифметика

Главным преимуществом модулярной арифметики является естественное распараллеливание вычислений на уровне системы счисления. Представление операндов в виде набора остатков по малым основаниям позволяет избежать длинных межразрядных переносов переполнения заёма при выполнении модульных арифметических операций. Особенно сильно это преимущество модулярной арифметики проявляется при работе с числами очень большой разрядности, и растёт вместе с разрядностью операндов. Но при выполнении немодульных операций преимущества модулярной арифметики теряются и возникают новые проблемы, которых лишены обычные позиционные системы счисления. Поэтому считается, что модулярная арифметика наилучшим образом применима для обработки больших целых положительных величин. Тем не менее, существуют и развиваются методы выполнения и немодульных операций, что позволяет строить полноценные модулярные системы. При решении задач с высокой долей немодульных операций, целесообразно наличие в системе и модулярного сопроцессора.

Ещё одной особенностью модулярной арифметики является возможность введения дополнительных избыточных оснований, с помощью которых можно выполнять контроль и коррекцию

ошибок в процессе выполнения операций. Это одно из важнейших преимуществ СОК (арифметичность) перед всеми позиционными системами: ни одна из них не позволяет находить и, тем более, исправлять ошибки в процессе выполнения арифметических операций. Наоборот, в арифметическом устройстве они, раз возникнув, бесконтрольно размножаются. В ЭВМ, работающих в традиционных позиционных системах счисления, контроль и исправление ошибок (контроль на четность, избыточное кодирование, мажорирование и т.п.) обеспечиваются только в системах хранения и передачи информации. Арифметико-логические устройства – один из основных источников сбоев и ошибок в позиционных ЭВМ, остаются бесконтрольными.

Малая разрядность оснований обеспечивает возможность реализации табличного выполнения модульных операций, если их результат не выходит за пределы диапазона представления чисел в процессоре. Причем в качестве операции в табличной арифметике может выступать не только элементарная модульная операция (сложение, умножение и т.п.), но и сложные функции, при вычислении которых не используются немодульные операции, например функции, которые могут быть представлены в виде полинома. Это свойство модулярной арифметики ставит ее вне конкуренции по производительности на определенных классах задач перед любыми позиционными системами. И это же свойство определяет специальный характер модулярных процессоров, отводит им роль высокопроизводительных арифметических сопроцессоров для решения определенных задач.

Троичная система

Троичная система счисления и трехзначная диалектическая логика Брусенцова [4] с симметричным кодом (-1,0,+1) обладает рядом ценных свойств наиболее интересными из которых являются [5]:

- Простота и гарантированная точность округления результатов вычислений. Значение каждого разряда равно 1/3 следующего более старшего, т.е. всегда менее половины его значения. Следовательно округление сводится к простому отбрасыванию лишних младших разрядов.

- Естественность представления знака числа. Нет необходимости во введении дополнительного кода и хлопотах с ним в процес-

се обработки информации, как это имеет место в двоичной системе.

- Более высокая информационная емкость троичного кода, по сравнению с двоичным, уменьшает количество межразрядных переносов при соответствующих операциях.

- Для двоичных ЭВМ все данные одинаково ценны, они не могут игнорировать несущественную информацию или сделать выбор из равных условий. Как известно двоичный Буриданов осел умер от голода между двумя стогами сена. А троичный Брусенцовов осел прекрасно себя там чувствует.

- Троичная логика, где есть «да», «нет» и «может быть» более естественна и понятна для человека в отличие от двоичной, исключающей вероятностный ход событий. Да и нейрон человеческого мозга, оказывается, тоже троичный (http://www.sci-tech-today.com/story.xhtml?story%20id=30799).

Уже этих кратких иллюстраций полезных свойств модулярной арифметики и троичной системы Брусенцова достаточно, чтобы разработчики ЭВМ обратили на них серьезное внимание. Недавно основатель троичной вычислительной техники Н.П. Брусенцов выступил с идеей их объединения, а крупнейший специалист в модулярной арифметике академик В.М. Амербаев заинтересовался этой идеей. Есть все основания ожидать высоких результатов от этого объединения.

О различных аспектах модулярной арифметики на конференции сказано достаточно много, поэтому остановимся на троичной системе Брусенцова. Большинство авторов, пишущих на эту тему, считает, что главной проблемой на пути построения троичной ЭВМ является отсутствие троичных электронных элементов. Даже Кнут в своей монографии [2] отметил «*До сих пор уравновешенная* (по Брусенцову – симметричная, прим. авт.) *троичная система все еще не нашла серьезного применения, но возможно, что ее симметричность и простая арифметика окажутся в один прекрасный день весьма существенными (когда «флип-флоп» заменится на «флип-флэп-флоп»)*». С технической точки зрения они в какой-то мере правы, это действительно проблема. Но на этом пути есть еще и препятствия, далекие от научных или технических проблем. Об

этом свидетельствует судьба ЭВМ «Сетунь-70», элементная база для которой коллективом Н.П. Брусенцова была создана и опытный образец ЭВМ многие годы прекрасно работал. Именно эти препятствия, а так же подавляющее господство двоичной системы, и мешали созданию троичных элементов. Но вернемся к элементной базе.

Автором рассмотрены пути создания элементной базы для реализации симметричной троичной системы [6] (троичная арифметика и трехзначная диалектическая логика Брусенцова) на основе промышленной технологии ОАО «Ангстрем». Работа еще далеко не завершена и проводится в тесном контакте с Николаем Петровичем Брусенцовым.

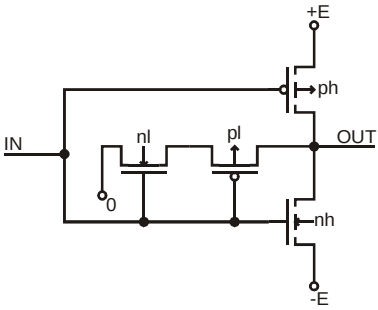
Рассмотрено два варианта построения троичных элементов:

- Однопроводный с электрическими сигналами трех уровней (+E, 0, -E).
- Двухпроводный с электрическими сигналами двух уровней (E, 0) и кодированием троичных значений, например 01 – «+1», 00 – «0» и 10 – «-1». Четвертое состояние «11» либо блокируется, либо используется в целях обеспечения безопасности или достоверности информации – этой проблемой заинтересовался академик Амербаев В.М., математик.

Оба варианта принципиально реализуемы, на основе технологии Ангстрема (и, по-видимому, большинства других полупроводниковых фирм), однако их готовность для практического применения, требуемые для этого затраты времени и средств существенно различаются.

Однопроводные троичные элементы

На данный момент проработана схемотехника нескольких троичных элементов на SPICE-моделях транзисторов технологии АТ-12 ОАО «Ангстрем», пример такого элемента, реализующего функцию троичного инвертора, приведен на рис. 1. Исследования показали принципиальную возможность построения таких элементов с точки зрения схемотехники и полупроводниковых технологий, но и вскрыли ряд неразрешимых пока проблем.



IN	OUT
-1	+1
0	0
+1	-1

Рис. 1. Схема однопроводного троичного инвертора: nl, pl – КМОП транзисторы с нормальным порогом, nh, ph – КМОП транзисторы с высоким порогом, и его таблица истинности.

Реализация однопроводного троичного элемента сопряжена с серьезными проблемами технологий производства и проектирования. Такой элемент имеет две особенности, отражающиеся на технологии его производства:

- Выходные транзисторы в некоторых кодовых комбинациях оказываются под двойным напряжением питания.
- Некоторые транзисторы необходимо гальванически изолировать от подложки.

Обе эти проблемы решены в Ангстреме, но нет такого производственного технологического маршрута по изготовлению ИС промышленного назначения, в котором они бы присутствовали оба. В технологии «кремний на изоляторе» эти обе проблемы решаются, но это очень дорогая технология для изготовления радиационностойких ИС, она не для экспериментов. Создание же нового технологического маршрута для производства ИС общепромышленного назначения технически не проблематично, но требует не только больших затрат времени и средств, но и гарантии рентабельной загрузки в последующем. Ни средств, ни гарантий в настоящее время никто дать не может.

Но главная проблема не в этом. Существующие системы автоматизации проектирования ИС (САПР), особенно их системы модели-

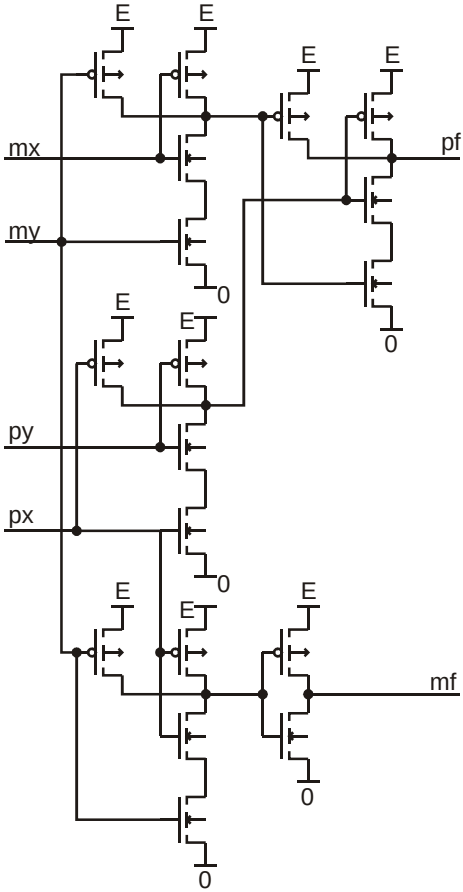
рования и синтеза, не умеют работать в троичной логике. И не просматриваются пути их адаптации к ней. Т.е. для проектирования однопроводных троичных элементов и троичных устройств требуется создание новых САПР. Это задача не только весьма трудоемкая и дорогостоящая, но и требующая переобучения разработчиков САПР на «троичное» мышление.

Таким образом, реализация однопроводного варианта троичной системы хотя потенциально технологически и возможна, но в настоящее время нереализуема по экономическим и организационным причинам. Эти препятствия могут быть преодолены только в том случае, если должностные лица и специалисты увидят преимущества троичной системы, достаточные для принятия решения о соответствующих вложениях и организации разработок технологий, САПР и изделий в троичной системе. А для этого нужно построить троичную ЭВМ иным способом.

Двухпроводные троичные элементы

Таким способом может быть двухпроводная реализация троичных элементов. Она обеспечивает возможность построения троичной ЭВМ на основе существующих промышленных полупроводниковых технологий и САПР без каких-либо доработок. На основе двухпроводного варианта возможно создание элементов, устройств и систем, полностью реализующих троичную арифметику и троичную диалектическую логику Брусенцова. По-существу по этому варианту была построена и первая троичная ЭВМ «Сетунь».

Такой вывод в какой-то мере подтверждается проведенными исследованиями. В частности проведена схемотехническая разработка элемента, реализующего сугубо троичную функцию «отношения следования», результаты которой приведены на рис. 2. Разработка проведена в SPICE-моделях транзисторов базового матричного кристалла (БМК) ОАО «Ангстрем» 1592ХМ1 (100 тыс. вентилей). Предполагается дополнить библиотеку стандартных элементов этого БМК троичными элементами, с тем, чтобы проводить разработку ЭВМ на его основе.



X	Y	m_x	p_x	m_y	p_y	m_f	p_f
-1	-1	1	0	1	0	0	1
-1	0	1	0	0	0	0	0
-1	+1	1	0	0	1	0	0
0	-1	0	0	1	0	0	0
0	0	0	0	0	0	0	0
0	+1	0	0	0	1	0	0
+1	-1	0	1	1	0	1	0
+1	0	0	1	0	0	0	0
+1	+1	0	1	0	1	0	1

Рис. 2. Схема двухпроводного троичного элемента «отношения следования» и его таблица истинности.

Таким образом, имеются все необходимые предпосылки для создания элементной базы троичной ЭВМ, работа над проектом которой в настоящее время начата Н.П. Брусенцовым в МГУ им. М.В. Ло-

моносова с участием ОАО «Ангстрем» и Санкт-петербургского государственного политехнического университета.

Литература

1. **Брусенцов Н. П.** и др. Общая характеристика малой цифровой машины «Сетунь-70». В кн.: Вычислительная техника и вопросы кибернетики, вып.10. Л., 1973, – С. 3-21.
2. **Кнут Д.** Искусство программирования для ЭВМ - Получисленные алгоритмы// М.: Мир, 1977. – С. 724
3. **Малашевич Б. М.** Неизвестные модулярные суперЭВМ// PC WEEK/RE, M., 2005. № 9. С. 44-45. № 10. – С. 52-54.
4. **Брусенцов Н. П.** Математическая теория силлогистики// Вычислительная техника и вопросы кибернетики. ЛГУ, 1971. – С. 154-176.
5. **Малашевич Д. Б.** Особенности применения троичной арифметики в вычислительных системах// Микроэлектроника и информатика – 2004. 11-я всероссийская межвузовская научно-техническая конференция студентов и аспирантов: Тезисы докладов. – М.: МИЭТ, 2004. – С. 88
6. **Малашевич Д. Б.** Анализ способов реализации троичной логики на КМОП схемах// Микроэлектроника и информатика – 2005. 12-я всероссийская межвузовская научно-техническая конференция студентов и аспирантов: Тезисы докладов. – М.: МИЭТ, 2005. – С. 105

УДК: 510.6 ; 681.31

Н.П. Брусенцов

Неадекватность двоичной информатики

(Московский государственный университет им. М.В. Ломоносова)

Непредставимое в двоичной информатике содержательное (смысловое) следование исчерпывающе охарактеризовано троичной четырехтритной шкалой.

О неполноценности двоичной информатики и лежащей в основе ее двухзначной логики убедительно свидетельствует запутанность ключевой логической проблемы – отношения содержательного (необходимого) следования. Известно, что так называемой “материальной импликации”, отождествляемой в двухзначной логике с отношением следования, присущи парадоксы: ”из ложного следует все, что угодно”, “истинное следует из чего угодно”. Многочисленные попытки устранить эти парадоксы конструированием исчислений “строгой”, “сильной”, “релевантной” и других импликаций цели не достигли и не могли достичь, потому что содержательное, непарадоксальное следование трехзначно, несовместимо с законом исключенного третьего.

Отношение, взаимосвязывающее термины x, y, отображается подмножеством декартова произведения

$$\{x, x'\}\times\{y, y'\} \equiv \{xy, xy', x'y, x'y'\}.$$

Материальная импликация представлена четким подмножеством $\{xy, x'y, x'y'\}$ и соответственно характеристической функцией:

$$(x \to y) \equiv xy \vee x'y \vee x'y' \equiv x' \vee y.$$

Эта функция принимает значение “1”, утверждая, что отношение соблюдено, при x=0, независимо от y, и при y=1, независимо от x, в чем и состоят парадоксы, означающие в сущности отсутствие взаимосвязи терминов. Таким образом, соблюденность материальной импликации есть не необходимость, а лишь не невозможность, т.е. *возможность* следования.

Нетрудно понять, что несовершенство материальной импликации обусловлено наличием в представляющем ее подмножестве члена $x'y$, однако, исключив его, имеем $\{xy, x'y'\}$ и характеристическую функцию $x \leftrightarrow y \equiv xy \vee x'y'$ отношения эквивалентности, а необходимое следование, четким подмножеством непредставимо. Требуется нечеткая (трехзначная) принадлежность элемента множеству и соответственно обобщение двухзначной булевой алгебры четких классов в трехзначную алгебру нечетких классов, допускающую помимо включенных и исключенных подклассов также *привходящие* (σ-классы), не включенные и не исключенные.

В трехзначной логике нечетких множеств и нечетких классов отношение содержательного естественноязыкового следования $x \Rightarrow y$ представлено декартовым нечетким подмножеством $\{xy, \sigma x'y, x'y'\}$ и характеристической функцией $xy \vee \sigma x'y \vee x'y'$, принимающей значение “0”, если следование невозможно, значение “1”, если выполняется с необходимостью, и значение “σ”, если возможно, но не необходимо.

В троичном компьютере с +, 0, – значениями тритов удобнее сопоставлять невозможности “–”, необходимости “+”, а собственно возможности “0”, условившись умалчивать возможные, но не необходимые (собственно возможные, нулевые) члены. При этом следование будет представлено подмножеством $\{xy, -xy', x'y'\}$ и характеристической функцией $xy \vee -xy' \vee x'y'$, кодируемыми четырехтритным кодом (четырехтритной ДК-шкалой) +–0+.

Экстенсиональное истолкование силлогистики, не позволяющее усмотреть в ее общеутвердительной посылке “Все x суть y” отношение следования с необходимостью y из x, обусловлено приняти-

ем в качестве “первых сущностей” единичных вещей [“Категории”, 2a11], а не тех несоставных особенностей, совокупностями которых эти вещи определены. Установив, что “...для уразумения через определение первее общее, а для чувственного восприятия - единичное” [“Метафизика”, 1018b32], Аристотель признал первичным единичное, оправдывая это тем, что все прочее находится в единичных вещах, и если бы они не существовали, “не могло бы существовать и ничего другого” [“Категории”, 2b1].

Верно, что сущности несоставных в рамках данного рассмотрения особенностей воспринимаются и познаются путем сопоставления вещей, которым эти особенности присущи, с вещами, которым они антиприсущи (необходимо не присущи). Но вместе с тем сущности единичных вещей представлены конъюнктивными совокупностями (множествами) существенных для цели рассмотрения несоставных особенностей. Так что и конструктивно, и по сути дела первичны все-таки не единичные вещи, а их несоставные особенности. Полагая первичными сущностями единичные вещи, приходится конструировать несоставные особенности их как дизъюнктивные совокупности (классы) индивидных конъюнкций-множеств терминов, которыми эти вещи представлены, в чем и состоит общепринятая экстенсиональная (объемная) трактовка силлогистики.

Принципиальное достоинство интенсионального истолкования заключается в том, что оно допускает диалектическую, адекватную живой реальности трактовку. Так, если экстенсиональный двухтерминный универсум представляет собой сосуществование мыслимых в нем четко определенных, фиксированных единичных вещей:

$$\mathrm{V}xy\mathrm{V}xy'\mathrm{V}x'y\mathrm{V}x'y'$$

то интенсиональная версия того же универсума оказывается выражением диалектического принципа сосуществования несоставных противоположностей:

$$\mathrm{V}x\mathrm{V}x'\mathrm{V}y\mathrm{V}y'$$

Примечательно, что строгая импликация Льюиса $\mathrm{V}'xy'$, означающая в экстенсиональном универсуме парадоксальное отношение материальной импликации

$$xy \vee x'y \vee x'y' \equiv x' \vee y$$

в интенсиональном становится полноценным содержательным следованием:

$$\mathrm{V}'xy'\mathrm{V}x\mathrm{V}x'\mathrm{V}y\mathrm{V}y' \equiv \mathrm{V}x\mathrm{V}'xy'\mathrm{V}y'$$

Соблюденность сосуществования противоположностей $\mathrm{V}x\mathrm{V}x'\mathrm{V}y\mathrm{V}y'$ означает, что термины x, y непременно должны быть переменными, не могут быть константами. Ведь парадоксы материальной импликации и возникают при $x \equiv 0$ и при $y \equiv 1$. Вместе с тем сущность несоставного термина может быть установлена лишь сопоставлением вещи, которой этот термин присущ, с вещью, которой он антиприсущ, так что реальное существование x - это сосуществование x и x', т.е. $\mathrm{V}x\mathrm{V}x'$ - собственно возможность x.

В базируемой на сосуществовании противоположностей трехзначной логике отсутствуют парадоксы и нелепости «классической» логики. Она безупречно соответствует здравому смыслу, неомертвленной реальности. В частности, силлогистика, обретя упущенную ненароком диалектичность, сводится к восьми видам двухтерминных отношений, компактно кодируемых четырехтритными шкалами:

$$\mathrm{A}xy \equiv \mathrm{A}y'x' \equiv \ \mathrm{A}xy \cap \mathrm{A}y'x' \ \equiv +\text{–}0+$$
$$\mathrm{A}yx \equiv \mathrm{A}x'y' \equiv \mathrm{A}yx \cap \mathrm{A}x'y' \equiv \ +0\text{–}+$$
$$\mathrm{I}xy \equiv \mathrm{I}x'y' \equiv \ \mathrm{A}xy \cup \mathrm{A}x'y' \equiv \ +00+$$
$$\mathrm{E}xy \equiv \mathrm{E}yx \equiv \mathrm{A}xy' \cap \mathrm{A}yx' \equiv -{+}{+}0$$
$$\mathrm{E}x'y' \ \equiv \mathrm{E}y'x' \equiv \mathrm{A}x'y \cap \mathrm{A}y'x \equiv 0{+}{+}\text{–}$$
$$\mathrm{O}xy \equiv \mathrm{O}x'y' \equiv \ \mathrm{E}xy \ \cup \mathrm{E}x'y' \ \equiv 0{+}{+}0$$
$$x \Leftrightarrow y \ \equiv \mathrm{A}xy \cap \mathrm{A}yx \equiv + \text{––}+$$
$$x \Leftrightarrow y' \equiv \mathrm{E}xy \cap \mathrm{E}x'y' \equiv \ -{+}{+}\text{–}$$

При этом все правильные модусы силлогизмов доказуемы посредством стандартной процедуры манипулирования кодирующими посылки шкалами [1]. Например, модус Barbara:

$$\mathrm{A}xy \cap \mathrm{A}yz \equiv (+\text{-}0+)_{xy} \cap (+\text{-}0+)_{yz} \equiv$$

$$\equiv (+{+}\text{--}00{+}{+} \cap +\text{-}0+{+}\text{-}0+)_{xyz} \equiv (+\text{---}000+)_{xyz} \Rightarrow (+\text{-}0+)_{xz} \equiv \mathrm{A}xz$$

Литература

1. Брусенцов Н.П. Реанимация аристотелевой силлогистики // Реставрация логики. – М.: Фонд «Новое тысячелетие», 2005. С.140-145.

УДК: 510.6 ; 681.31 Н.П. Брусенцов

Заметки о троичной цифровой технике

(Московский государственный университет)

Цель этих заметок - охарактеризовать с практической точки зрения особенности и возможности троичной цифровой техники.

Часть 1

http://www.computer-museum.ru/histussr/12-1.htm

Интерес к троичной технике возник уже на начальном этапе развития вычислительной техники в связи с замечательными арифметическими свойствами симметричного кода чисел [1], открытие которых происходило затем снова и снова [2, 3]. Впрочем, имеется мнение [4], будто интерес этот обусловлен ошибочным представлением об исключительной экономности троичного кода. Как бы то ни было, но интерес к троичной технике существует, и сама она, хотя и не быстро, но развивается. Укажем, например, что на многочисленных симпозиумах по многозначной логике ряд докладов прикладного характера было посвящено троичной технике.

В условиях интегральной технологии и микроэлектроники привлекательность троичной техники увеличивается: сложность трехзначных вентилей теперь не так страшна, а сокращение количества соединений и уменьшение рассеиваемой мощности особенно ценны. Новые преимущества троичного кода выявлены благодаря раз-

витию цифровой связи [6] – области, в которой троичная техника стала использоваться с появлением кабельного телеграфа и успешно применяется в новейших системах.

Однако не так важны частные выгоды и преимущества, как гармоничность и эффективность троичной техники в целом, неприсущность ей недостатков и неудобств, свойственных двоичной технике.

Что такое троичная цифровая техника?

Современная цифровая техника основана на двузначных сигналах и двухстабильных элементах памяти – это двоичная цифровая техника. Объекты, принимающие более чем два значения, реализуются в ней как совокупности двузначных элементов (битов). Например, десятичные цифры представляются четверками битов, символы алфавита, включаюшего буквы, цифры и некоторые другие знаки – восьмерками битов (байтами).

Соответственно все операции над недвузначными объектами реализуются как последовательности операций двузначной логики, производимых над битами, совокупностями которых представлены эти объекты. Например, арифметические операции над двоичными числами.

Важными преимуществами двоичной техники, обусловившими ее быстрое развитие и широкое распространение, являются: простота физической реализации битов и операций двузначной логики, некритичность допусков на параметры двузначных сигналов и двухстабильных устройств.

Троичная цифровая техника базируется на трехзначных сигналах и трехстабильных элементах памяти (тритах).

Объекты, принимающие более чем три значения, реализуются в ней как совокупности тритов. Операции над этими объектами осуществляются как последовательности операций трехзначной логики. Аналогом байта служит шестерка тритов - трайт. Двузначные объекты и операции над ними содержатся в троичной технике как вырождения тритов и операций трехзначной логики.

Практическая целесообразность троичной техники не очевидна. Ясно, что троичная техника равноценна двоичной технике в том

смысле, что все, осуществимое в одной из них, с тем или иным приближением осуществимо и в другой.

Ясно также, что трехзначные вентили и элементы памяти должны быть сложнее и дороже, чем двузначные, а трехзначная логика заведомо сложнее двузначной. Но с другой стороны, трехзначные элементы памяти мощнее (трит - это приблизительно 1,585 бита) и операционные возможности трехзначных вентилей богаче. Следовательно, обработка данных в условиях троичной техники осуществляется, при одном и том же физическом быстродействии элементов, быстрее, а структура троичного устройства, как правило, оказывается проще, чем структура функционально равноценного двоичного устройства. Другими словами, троичная техника характеризуется по сравнению с двоичной усложнением элементов, благодаря которому возможно упрощение создаваемых из них структур и увеличение скорости обработки данных. Замечательно, что троичная техника является единственной недвоичной техникой, не связанной с необходимостью ужесточения действующими в двоичной технике допусков на параметры сигналов и характеристики элементов.

Увеличение значности с двух до трех без ужесточения допусков достигается за счет недоиспользуемой двоичной техникой возможности различать сигнал как по амплитуде, так и по полярности. При этом троичный сигнал x можно рассматривать как суперпозицию его положительной x+ и отрицательной x- двоичных составляющих.

Основанная на таком представлении интерпретация троичной техники, как оперирующей не только с троичными сигналами, но и с их положительными и отрицательными двоичными компонентами, которые можно отделять от троичного сигнала, обрабатывать по отдельности и снова соединять в троичный сигнал, позволяет естественно и просто осуществить неформальное построение троичных цифровых устройств [7]. Физически такой подход выражается в том, что вентили, обладающие двоичным выходом положительной полярности, используются совместно с вентилями, обладающими двоичным выходом отрицательной полярности. На входах этих вентилей допустимы сигналы как положительной, так и отрицательной полярности, т. е. применяется трехзначная логика. Выходы положительной и отрицательной полярности можно объединять, благодаря чему в 1,5 - 2 раза увеличивается интенсивность исполь-

зования соединительных проводов и соответственно сокращается количество соединений между вентилями.

Практичность трехзначной логики

Одним из барьеров, сдерживающих развитие и распространение троичной техники, является неверное представление о необычности и трудной постижимости трехзначной логики. Современная формальная логика (как традиционная, так и математическая) основана на принципе двузначности. В числе ее фундаментальных законов имеется закон исключенного третьего: "Третьего не дано", истолковываемый обычно в том смысле, что правильная логика ничего, кроме "Да" и "Нет", допустить не может. Трехзначная логика при этом ассоциируется с интуиционизмом, модальностями, микромиром и другими таинственными вещами, но только не с обыденной действительностью, которая по сложившемуся на протяжении веков убеждению будто бы устроена и функционирует по двоичным правилам. Окутанная подобным научным туманом и характеризуемая, например, тем, что число двухместных функций, равное при двузначных переменных 16, в случае трехзначных переменных составляет 19683 (!), трехзначная логика естественно действует устрашающе. На самом деле трехзначная логика не только вполне корректна и адекватна действительности, но является даже более удобной и привычной для людей формой мышления, чем двузначная логика. Покажем это на примерах.

В качестве первого примера рассмотрим рычажные весы (рис. 1), представляющие собой характерное троичное устройство, трем состояниям которого соответствуют три возможных отношения: A>B, A=B, A<B.

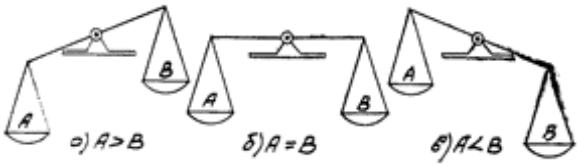


Рис. 1. Троичные весы

Для сравнения рассмотрим также двоичные весы, которые могут принимать только два состояния, соответствующие, например, отношениям A>B, AB (рис. 2). Ясно, что двоичные весы существенно менее удобны, чем троичные. Только в случае A>B результат

взвешивания на них определяется сразу, а в остальных двух случаях необходимо производить повторное взвешивание, поменяв местами А и В. Практическая работа с такими весами имеет смысл разве как средство переубеждения приверженцев двузначной логики, а с точки зрения других применений их можно рассматривать лишь как испорченные троичные весы. Двоичные цифровые устройства по сравнению с троичными устройствами в отношении логической эффективности занимают примерно такое же положение.

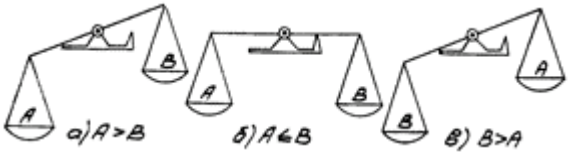


Рис. 2. Двоичные весы

Другой пример - ветвление по знаку величины X (рис.3) - не обладает физической наглядностью примера с весами, но явно демонстрирует принципиальное отличие трехзначной логики от двузначной.

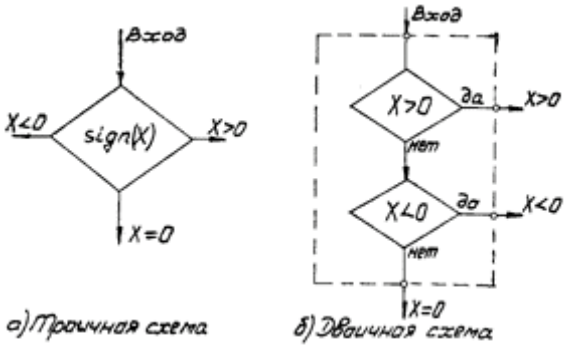


Рис. 3. Ветвление по знаку

Это отличие состоит не в том, что трехзначная логика, как нередко полагают, будто бы позволяет выразить нечто, невыразимое в двузначной логике, а в том, что в точности одно и то же в трехзначной логике может быть выражено более компактно и может быть выполнено за меньшее число шагов. В рассматриваемом примере троичное ветвление по знаку величины X описано заданием единственной трехзначной операции sign(X) и выполняется за один шаг, в то время как такое же ветвление, осуществляемое средствами

двузначной логики, связано с необходимостью двух операций и выполняется, вообще говоря, за два шага.

Нетрудно построить аналогичную схему ветвления в зависимости от отношений, которыми могут быть связаны две величины: X>У, X=У, X<Y. В жизни трехзначные отношения, укладывающиеся в данную схему, встречаются очень часто. Например:

увеличить - не изменять - уменьшить,
вперед - стой - назад,
выигрывает А - ничья - выигрывает В,
избыток - норма - недостаток,
дружественный - нейтральный - враждебный,
рано - своевременно - поздно,
влево - прямо – вправо
и т. п.

Многие вопросы предполагают тройственный ответ. На это указал еще Аристотель [8]: "Будет ли завтра в полдень морской бой?" - "Да" - "Нет" - "Может быть". Логики утверждают, что этот пример свидетельствует о неприменимости закона исключенного третьего к высказываниям о будущем. Но спросите у вашего соседа, был ли вчера дождь в Батуми. Если только сосед не прилетел из Батуми сегодня утром или не переговорил с кем-то, находящимся в Батуми, по телефону, то ответ его будет ни "Да", ни "Нет", а "Не знаю" или "Может быть". Но ведь высказывается он не о будущем, а о прошедшем!

Ясно, что тройственность ответа обусловлена не тем, что вопрос касается будущего, а тем, что ответчик не располагает информацией, необходимой для того, чтобы дать утвердительный или отрицательный ответ. Если в этой, весьма типичной в жизни ситуации ваш сосед окажется двоичным соседом, т. е. таким, который может отвечать только "Да" или "Нет", то во избежание недоразумений Вы должны спрашивать о дожде в Батуми в два приема. Сначала следует спросить, знает ли он о том, был или не был вчера дождь: в Батуми. И только в случае утвердительного ответа на этот вопрос можно спросить, был ли дождь в Батуми. В обычных разговорах с людьми, конечно, не приходится прибегать к подобным ухищрениям, так как всякий нормальный человек владеет трехзначной формой ответа. Однако в двоичных системах эти ухищрения неизбежны, потому что третье в двузначной логике "не дано".

Например, понятие "бит, значение которого не всегда определено", фактически сводится к тому, что должен быть другой вспомогательный бит, содержащий информацию о том, определено в данный момент значение основного бита или не определено. При этом, поскольку обращение к основному биту имеет смысл только в том случае, когда значение его определено, то всякий раз надо сперва обратиться к вспомогательному биту и лишь затем обращаться к основному, если вспомогательный оказался в состоянии "значение определено".

Приведенные примеры показывают, что трехзначная логика не есть нечто противоестественное или необыкновенное. Она не только доступна для людей, но позволяет рассуждать более просто и более быстро по сравнению с рассуждениями в условиях двузначной логики. На практике люди пользуются, по-видимому, преимущественно трехзначной логикой. Во всяком случае система умозаключений, составляющая логическую основу естественных языков, - силлогистика - построена на принципе трехзначности [9].

Эффективность троичной арифметики

Подобно тому как добавление третьего значения позволяет в значительной мере преодолеть неудобства двузначной логики, введение третьей цифры в систему машинного представления чисел оказывается достаточным для того, чтобы можно было, устранить практически все дефекты двоичной (и десятичной) арифметики. Дело не в том, что число три ближе других целых чисел к основанию натурального логарифма e (хотя усматривать в этом намек на исключительность троичного, конечно, можно), а в том, что три цифры - это минимум, уже позволяющий непосредственно обозначить положительное, отрицательное и нуль. Двоичная система неполноценна в том смысле, что в ней недостает одного из трех этих элементов. Ради простоты ограничимся рассмотрением целых чисел.

Назовем естественным представлением числа такое представление в позиционной системе счисления с положительным целым основанием p, при котором запись числа в виде ряда цифр a1a2...an конечной длины n однозначно определяет значение этого числа по формуле

В двоичной системе (p=2) возможно естественное представление либо только неотрицательных (цифры 0 и 1), либо только неположительных (цифры 0 и -1) чисел, либо

положительных и отрицательных, но неоднозначное и без нуля (цифры 1 и -1). Практически всегда используют цифру 0 и 1, т. е. естественное представление имеет место только для неотрицательных чисел. Поэтому двоичная арифметика проста и эффективна, пока операции производятся над числами без знака, например, в адресном пространстве двоичной памяти. Заметим, что при этом несложно реализовать выполнение операций с операндами разной длины, поскольку добавление ведущих нулей не изменяет числового значения операнда и, следовательно, может служить средством автоматического выравнивания длин. Если же длина слова фиксирована, то при естественном представлении чисел наиболее просто обнаруживается переполнение сумматора: при переполнении и только при переполнении цифра переноса из старшего разряда сумматора отлична от нуля.

Представление отрицательных чисел в рассматриваемой системе с цифрами 0 и 1 можно получить, производя вычитание большего числа из меньшего. Например, вычитая 1 из 0, имеем где точки означают бесконечное повторение ведущей цифры, обусловленное тем, что имеет место неограниченно распространяющийся влево заем. Таким образом, запись ...111 представляет число -1. Аналогично получим: ...110 для обозначения числа -2, ...101 для -3 и т. д. Замечательно, что при таком представлении отрицательных чисел арифметика, разработанная для двоичных чисел без знака, автоматически распространяется на числа со знаком в том смысле, что алгоритмы выполнения операций над неотрицательным, и числами в естественном представлении остаются в силе при добавлении представленных данным способом отрицательных чисел. Однако, введя точки как символ бесконечного повторения цифры, мы использовали как раз то третье, которого нет в двоичной системе. Пользуясь только цифрами 0 и 1, мы должны явно записывать отрицательные числа словами бесконечной длины. Ведь если просто оборвать распространяющийся влево заем, то отрицательное число нельзя будет отличить от некоторого положительного числа в естественном представлении. Например, отбросив точки в записи ...101 (число -3), получим 101 - положительное число 5. Впрочем, при фиксированной длине слова можно условиться, что в случае, когда

заем отсутствует, первый (левый) бит слова должен содержать цифру 0. Полное представление числа при этом будет получаться просто распространением влево за пределы слова той цифры, которая содержится в его первом бите.

Когда число отрицательно, т. е. когда заем имеет место, этой цифрой будет 1, а в случае неотрицательного числа ею будет 0. Обычно говорят, что первая цифра слова является знаком числа, хотя на самом деле она лишь указывает, отрицательно число или нет. Такое предоставление чисел со знаком (называемое дополнительным кодом) не обладает уже преимуществами естественного представления в отношении операций с разной длиной операндов и индикации переполнения, однако является самым употребительным в современных цифровых машинах, потому что позволяет более эффективно, чем другие двоичные коды, реализовать арифметику. Чтобы показать, насколько троичная арифметика эффективней арифметики, основанной на двоичном дополнительном коде, сопоставим программу, реализующую сложение двух чисел в дополнительном коде на миникомпьютере PDP-8 [10], с программой, осуществляющей равносильную операцию на троичном миникомпьютере, подобном PDP-8 в отношении архитектуры. Машина PDP-8 обладает 12-битовым аккумулятором Ac (0:11), между крайними битами которого включен однобитовый регистр связи Lk(1:1). В процессе сложения L14 воспринимает цифру переноса Cr из старшего разряда Ac(0) аккумулятора: Lk:=Lk+Cr. Имеется возможность тестировать Ac и Lk, а также "очищать" их - присваивать значение 0. В кольце LkAc можно производить сдвиг влево: LkAc:==AcLk и вправо: AcLk: =LkAc.

Сложение чисел в дополнительном коде осуществляется программой, которая использует в качестве слагаемых содержимое ячеек A и B главной памяти, а результат сложения помещает в ячейку SUM. На первом этапе сопоставляются знаки слагаемых и устанавливается одно из трех:

- знаки противоположны - OPPSGN,
- оба слагаемые отрицательны - BTHNEG,
- оба слагаемые положительны - BTHPOS.

В каждом из этих случаев сложение выполняется отдельной подпрограммой, причем при переполнении аккумулятора производится переход на POSERR, если слагаемые положительны, и на NEGERR, если отрицательны.

Ниже приведен текст программы на языке ассемблера с поясняющими комментариями.

START,	CLA CLL TAD A AND MASK TAD B SZL JMP BTHNEG RAL SZL CLA JMP OPPSGN JMP BTHPOS	/Ac: =0; Lk: =0; /Ac: =Ac+A; /Ac (1, 11): = 0; /Ac: =Ac+B; /if Lk=0 /then go to BTHNEG; /LkAc: =AcLk; /Ac: =0; if Lk=0 /then go to OPPSGN /else go to BTHPOS;
OPPSGN,	TAD A TAD B DCA SUM HLT	/Ac: =Ac+A; /Ac: =Ac+B; /SUM: =Ac; Ac: =0; /STOP;
BTHNEG,	CLA CLL TAD A TAD B SMA JMP NEGERR DCA SUM HLT	/Ac: =0; Lk: =0; /Ac: = Ac+A; /Ac: =Ac+B; /if not(Ac<0) /then go to NEGERR; /SUM: =Ac; Ac: =0; /STOP;
BTHPOS,	TAD A TAD B SPA JMP POSERR DCA SUM HLT	/Ac: =Ac+A; /Ac: =Ac+B; /if not(Ac>=0) /then go to POSERR; /SUM: =Ac; Ac: =0; /STOP
MASK, SUM, A, B, POSERR,	4000 0 nnnn nnnn	
NEGERR,	... ...	

Теперь покажем, как данная процедура может быть реализована в условиях троичной техники. Представим себе троичный вариант процессора PDP-8, т. е. работающую в троичном коде машину с такой же конфигурацией регистров Ac и Lk (заметим, что для обеспечения точности, соответствующей 12 битам, достаточно иметь в аккумуляторе 8 тритов) и аналогичным набором команд. В системе с тремя цифрами можно не получить никаких преимуществ перед двоичной системой, если принять неотрицательные (0, 1, 2) или неположительные (0, -1, -2) значения цифр. Мы используем симметричный набор цифр (-1, 0, 1), обеспечивающий однозначное естественное представление всех чисел: положительных, отрицательных и нуля. В этой системе арифметика чисел со знаком так же проста, как арифметика неотрицательных чисел в двоичной системе с цифрами 0, 1. В частности, чтобы обнаружить переполнение аккумулятора, не требуется анализировать знаки слагаемых: признаком переполнения служит ненулевая цифра переноса из старшего разряда в Lk. Если же необходимо не только обнаружить переполнение, но и произвести переход на одну из подпрограмм в зависимости от знака слагаемых, как в рассмотренной выше двоичной программе, то достаточно проанализировать цифру переноса, поскольку она обладает тем же знаком, что и слагаемые.

Анализ цифры в троичной машине естественно выполнять командой трехзначного перескока (пропуска). Такую команду применительно к анализу, Lk мы в духе мнемоники языка ассемблера для PDP-8 обозначим SLS - Skip on Link's Sign. Она предписывает следующую модификацию значения программного счетчика PC:

if Lk=0 then PC:=PC+2 else if Lk=l then PC:=PC+1;

Программа для троичного миникомпьютера, функционально равноценная приведенной выше двоичной программе для PDP-8, состоит из 8 команд:

START,	CLA CLL	/Ac: =0; Lk: =0;
	TAD A	/Ac: =Ac+A;
	TAD B	/Ac: =Ac+B;
	SLS	/if Lk= - 1
	JMP NEGERR	/then go to NEGERR; if Lk= 1
	JMP POSERR	/then go to POSERR;
	DCA SUM	/SUM: =Ac; Ac: =0;
	HLT	/STOP;

В случае если ветвление по знаку слагаемых не требуется, а необходимо только сигнализировать о переполнении, троичная программа сложения двух чисел с учетом их знаков допускает дальнейшее сокращение:

```
START,  CLA CLL     /Ac: =0; Lk: =0;
        TAD A       /Ac: =Ac+A;
        TAD B       /Ac: =Ac+B;
        SZLS        /if Lk=0
        JMP OVRFLW  /then go to OVRFLW;
        DCA SUM     /SUM: =Ac; Ac: =0;
        HLT         /STOP;
```

Эта программа, как и следовало ожидать, в точности совпадает с программой, реализующей на PDP-8 сложение чисел без знаков.

Рассмотренные примеры убедительно демонстрируют высокую эффективность троичной арифметики. Троичная арифметика для чисел со знаком так же проста и эффективна, как двоичная арифметика для чисел без знака.

Литература

1. Shannonc. E.A symmetrical notation for numbers. - "The American Mathematical Monthly", 1950, 57, N 2, p, 90 - 93,

2. Reid J.B. Letter to the editor. - "Comm. ACM", 1960, 3, N 3, p. A12 - A13.

3. Howden P.F. Weigh-counting technique is faster then binary.- "Electronics", 1974, 48, N 24, p. 121 - 122.

4. Байцер Б. Архитектура вычислительных комплексов, т. 1. М., "Мир", 1974.

5. Proceedings of the Sixth International Symposium on Multiple-Valued Logic, May 25 - 28 1976. IEEE Press, 1976.

6. CroisierA. Introduction to pseudoternary transmission codes.- "IBM Journal of Research and Development", 1970, 14, N 4, p. 354 - 367.

7. Брусенцов Н.П. Электромагнитные цифровые устройства с однопроводной передачей трехзначных сигналов. - В кн.: Магнитные элементы автоматики и вычислительной техники. XIV Всесоюзное совещание (Москва, сентябрь 1972 г.). М., "Наука", 1972, с. 242 - 244.

8. Аристотель. 06 истолковании. СПб., 1891.

9. Брусенцов Н.П. Диаграммы Льюиса Кэррола и аристотелева силлогистика. - В кн.: Вычислительная техника и вопросы кибернетики, вып. 13. Изд-во МГУ, 1976, с. 164-182.

10. Introduction to programming. PDP-8 handbook series. Digital Equipment Corporation, 1972.

ЧАСТЬ 2

http://www.computer-museum.ru/histussr/12-2.htm

В первой части были обсуждены логические и инженерные аспекты реализации троичных цифровых устройств. При этом было показано, что естественным и весьма эффективным путем физического воплощения трехзначной логики является использование пороговых элементов с положительными и отрицательными весами входов, а также с сигналами положительной и отрицательной полярности. Данная техника была разработана при создании троичных машин "Сетунь" и "Сетунь 70" [1] и отлично показала себя на всех этапах конструирования, производства и эксплуатации аппаратуры.

Пороговое осуществление операций с положительными и отрицательными сигналами легко осваивается людьми, поэтому как понимание логики построенных из пороговых элементов троичных устройств, так и разработка устройств, реализующих заданные функции трехзначных переменных, не связаны с особыми затруднениями. Что же касается надежности функционирования, то, вопреки беспочвенному утверждению [2, с.28], будто "плохая надежность аппаратуры и неадекватное программное оснащение" воспрепятствовали продолжительному использованию "Сетуни", следует сказать, что первая троичная машина была значительно надежнее большинства ее двоичных сверстниц и имела добротное и не такое уж бедное для минимашины того времени программное оснащение [3].

Экспериментальный образец машины "Сетунь", построенный в 1957 - 1958 гг., находился в эксплуатации 15 лет, причем из 4 тыс. использованных в нем пороговых элементов типа быстродействующих магнитных усилителей с питанием импульсами тока были заменены вследствие отказов только 3 элемента (все 3 на первом году эксплуатации): 2 из-за пробоя диодов типа Д1, по-видимому, обладавших дефектами изготовления, и 1 из-за нарушения изоляции между обмотками импульсного трансформатора. Машина ус-

тойчиво работала при значительной нестабильности напряжения питающей электросети и в достаточно широком диапазоне температур окружающей среды (от +15 до +30° С). Серийные экземпляры машин "Сетунь" успешно эксплуатировались в различных климатических зонах как с холодным, так и с жарким, а также резко континентальным климатом (например, в Ашхабаде, Душанбе, Махачкале, Иркутске, Якутске, Одессе), причем без какого-либо сервисного обслуживания и практически без запасных частей. Едва ли это может свидетельствовать о плохой надежности аппаратуры.

Нельзя согласиться и с имеющимся в [2] заключением, будто "Сетунь" была арифметической машиной, совершенно неподходящей для какой-либо оценки троичных логических операций, и что не было предпринято развернутых попыток использования ее для критического сравнения двоичных и троичных машин в отношении арифметических операций.

Разработка "Сетуни", в отличие от упоминаемого в [2] моделирования на двоичной микропрограммной машине троичного компьютера TERNAC, не предназначалась специально для исследования возможностей троичной техники и сравнения ее с двоичной техникой. Целью разработки было создание недорогой и простой в использовании малой цифровой машины широкого назначения (нечто в роде того, что десять лет спустя стали называть миникомпьютером), и эта цель была успешно достигнута. То же, что требуемая машина получилась троичной, было, пожалуй, делом случая: ввиду ненадежности имевшихся в то время транзисторов были созданы магнитные логические элементы, выполненные на нелинейных трансформаторах импульсов тока с диодами (быстродействующие магнитные усилители с питанием импульсами тока), причем оказалось, что эти элементы не только весьма удобны для построения троичных цифровых устройств, но и что троичные устройства получаются существенно более экономными в отношении количества оборудования и потребляемой мощности, более быстрыми и структурно более простыми, чем двоичные устройства, реализованные на тех же

элементах [4]. Однако, хотя исследование троичной цифровой техники и не было самоцелью разработки "Сетуни"; разработка эта внесла в понимание проблем и возможностей троичной техники больше, чем теоретическое изучение экономности троичного кодирования и методов синтеза функций трехзначной логики.

Принципиальные результаты разработки "Сетуни" настолько естественны и немудрены, что, с точки зрения формальных теоретиков многозначной логики, их попросту не видно.

Похоже, что именно этим объясняется приведенное выше заключение о слабости логических возможностей "Сетуни" и неиспользовании ее для сравнения троичной арифметики с двоичной . Результаты состояли в том, что было экспериментально доказано, что троичная машина, по меньшей мере в условиях электромагнитной техники, оказывается существенно экономнее, быстрее, проще и математически совершенней функционально эквивалентной двоичной машины, выполненной на элементах того же типа.

Кроме того, было показано, что троичные устройства могут быть эффективно и просто реализованы на основе способа выполнения логических операций, названного впоследствии пороговой логикой, причем именно в трехзначном варианте с положительными и отрицательными весами логических входов данный способ становится практически приемлемым, благодаря значительному ослаблению требований к точности и стабильности параметров физических элементов и сигналов. Не менее важным было и то, что трехзначная огика с ее 33 одноместных и 39 двуместных операций, трактуемая некоторыми философами как логика, таинственного микромира, предстала перед инженером как давно известная ему логика положительного, отрицательного и равного нулю тока (или заряда), а перед программистом - как логика элементарных чисел: 0, 1, - 1 или логика значений, принимаемых алгебраическим знаком числа: +, -, 0. При этом выяснилось, что, хотя эта трехзначная логика сложнее двузначной, она вместе с тем удобнее для человека, легче осваивается и применяется.

Вывод о неразвитости логических возможностей "Сетуни" сделан из того, что в наборе ее команд имеется только одна логическая операция - операция умножения тритов (поразрядное умножение), вследствие чего якобы машина не подходила для оценки троичных логических операций.

Невозможно понять, почему для оценки логических операций (и что это за оценка) надо строить машину, и "Сетунь", как уже было сказано, была создана для иных целей, но указанной единственной операции для программирования логики было достаточно. Может быть, как раз это и является той оценкой троичных операций, кото-

рую имеют в виду критики. Операция умножения тритов была включена в набор команд "Сетуни" как наиболее часто встречающаяся и легко понимаемая. Это аналогично наличию единственной логической операции конъюнкции в наборе команд PDP-8.

Однако операция умножения тритов несравнимо богаче двузначной конъюнкции [4]. Использование ее в сочетании с константами и арифметической операцией сложения позволяет осуществить произвольную функцию трехзначной логики. Впрочем, последнее имеет, пожалуй, лишь теоретическое значение, так как для программиста основными средствами выражения логики являются команды условного перехода, которых у "Сетуни" три, а команда умножения тритов используется главным образом для выделения, очистки или изменения знака частей слова.

Следует отметить, что при создании "Сетуни" идеалом было не предоставление программисту как можно большего набора различных команд, а, напротив, стремление сделать этот набор как можно меньшим, но вместе с тем обеспечивающим возможность эффективно запрограммировать широкий круг различных применений. Это должно было способствовать удешевлению машины и, с другой стороны, облегчить освоение ее программистами. Указанное стремление увенчалось успехом в известной степени благодаря симметричному троичному коду - в наборе команд "Сетуни" всего 24 команды.

АРХИТЕКТУРА ЦИФРОВОЙ МАШИНЫ "СЕТУНЬ"

Подтвердим сказанное кратким описанием архитектуры машины "Сетунь" [5]. Ее можно охарактеризовать как одноаккумуляторную, с индекс-регистром и одноадресным форматом команд. Длина аккумулятора 18 тритов, длина ячеек главной памяти 9 тритов, что эквивалентно 14,3 бита. Команды занимают по одной ячейке памяти, а операнды могут занимать как по одной, так и по две ячейке, рассматриваемые как одно 18-тритное слово.

Девять тритов, кодирующих команду К(1:9), разделяются на код операции К(6:8) и адресную часть К(1:5)К(9). Первые четыре трита К(1:4) адресной части составляют адрес пары 9-тритных ячеек, обращение к которым производится трояко в зависимости от значения К(5): при К(5)=0 доступна одна из этих ячеек, при К(5) =1 - другая, при К(5)=-1-18-тритное слово, младшей частью которого

является ячейка, соответствующая К(5)=1, а старшей - ячейка, соответствующая К(5) =0. Трит К(9) управляет индексацией так, что исполнительный адрес ЕА определяется выражением ЕА(1:5)=К(1:5)+К(9)*F(1:5),где F(1:5) - индекс-регистр. Поскольку К(9) принимает значения 0, 1, - 1, то ЕА равно либо К(1:5), либо К(1:5)+F(1:5), либо К(1:5)-F(1:5).

Адресное пространство главной памяти простирается от -(35-1)/2 до (35-1)/2, т. е. от -121 до +121, но так как каждый третий адрес использован для указания пары ячеек, обладающих каждая отдельным адресом, то адресовать можно только 162 девятитритных ячейки и 81 пару ячеек.

Последовательная выборка команд производится с пропуском адресов, соответствующих парам ячеек, т. е. программный счетчик принимает следующие значения: -120, -119, -117, -116, ..., -3, -2, 0, 1, 3, 4, ..., 117, 118, 120, 121.

При обменах с внешней памятью, которая реализована на магнитном барабане и является основной памятью машины, а также при вводе и выводе главная память представляется разделенной на страницы, содержащие по 54 ячейки.

Номером страницы служит значение старшего трита принадлежащих этой странице адресов. Другими словами, память структурирована как массив m(-1:1, -80:81, 1:9), в котором средняя компонента индекса пропускает каждое третье значение. Обмен с магнитным барабаном производится целыми страницами, причем трит К(1) в команде обмена указывает номер участвующей в обмене страницы главной памяти, а триты К(2:5) - номер страницы магнитного барабана. Ввод/вывод в девятеричном коде осуществляется целыми страницами, а в алфавитно-цифровом коде - записями произвольной длины с окончанием по литере "стоп" и по исчерпании страницы памяти.

Тройка тритов К(6:8) командного слова, используемая в качестве кода операции, принимает 27 различных значений, чем ограничивается количественный состав набора команд машины. Однако в действительности набор состоял из 24 команд, так как 3 значения кода операции были зарезервированы и в серийных машинах не использовались.

Из предоставленных программисту 24 команд две команды управляют страничным обменом между главной памятью и магнитным барабаном, одна команда задает операции ввода/вывода, причем номера устройств и режимов работы указываются в поле К(2:5). Операции, соответствующие остальным командам, определены над регистрами процессора и ячейками главной памяти.

В процессоре имеется пять программно доступных регистров: аккумулятор S(1:18), регистр множителя К(1:18), индекс-регистр F(1:5), программный счетчик С(1:5), регистр знака результата w(1:1) .

Над аккумулятором и указываемой адресной частью ячейкой памяти определены следующие операции:

засылка в аккумулятор S:=m(EA);
сохранение результата m(EA):=S;
сложение S: = S+m (EA);
вычитание S:=S - m(EA);
умножение тритов S:=SДm(EA);
сдвиг S:=S*3m(EA, 1:5);
нормализация m(EA):=Sнорм; S:=0; S(1:5):=nсдв;

Значение доставляемого из главной памяти операнда m(EA) может быть в зависимости от К(5) либо длинным, 18-тритным, либо коротким, 9-тритным. В последнем случае оно соответствует старшей половине 18-тритного аккумулятора, т. е. дополняется до 18-тритного 9 нулями справа. Запятая, отделяющая дробную часть числа от целой, подразумевается между вторым слева и третьим тритами аккумулятора.

В операции сдвига операнд m(EA, 1:5) является целым числом, представленным пятью старшими тритами соответствующей ячейки памяти.

Операция нормализации заключается в преобразовании путем сдвига значения S в Sнорм, удовлетворяющее условию 0,5<|Sнорм|<1,5. При этом Sнорм сохраняется в главной памяти, а число произведенных сдвигов nсдв помещается в S(1:5). В случае S=0 принято Sнорм=0, nсдв=0.

Регистр множителя К используется совместно с S в командах с операцией умножения. Таких команд три:

1) R:=S; S:=R*m(EA);

2) S:=S+R*m(EA);
3) S=R*S+m(EA).

Кроме того, имеется команда засылки в R, т. е. R:=m(EA), и команда засылки в К с остановом процессора:

К:=m(EA);
stop.

При выполнении операций сложения, вычитания и умножения возможно переполнение аккумулятора. Переполнение вызывает останов процессора, причем на пульте машины наряду с индикацией содержимого S и R отображаются также значения двух тритов переполнения S(-1:0).

С индекс-регистром F(1:5) связаны четыре операции:

засылка F:=m(EA, 1:5);
сохранение результата m(EA):=0; m(EA, 1:5):=F;
сложение F:=F+m(EA, 1:5);
сложение с С F:=C+m(EA, 1:5)

Все перечисленные выше операции над регистрами S,R,F сопровождаются фиксацией в однотритном регистре w знака полученного результата. В зависимости от значения w производится выбор продолжения программы при выполнении

команд условного перехода, которых имеется три:

1) if w=0 then C:=EA;
2) if w=1 then C:=EA;
3) if w=-1 then C:=EA.

Кроме того, имеется команда безусловного перехода C:=EA и команда сохранения текущего значения программного счетчика m(EA)=C.

Ручное управление машиной осуществляется с пульта управления кнопками "стоп", "пуск", "начальный пуск", "команда ПУ". Последняя позволяет выполнить команду, код которой набран на ключах пульта. Имеются также ключи для задания останова по данному значению программного счетчика и по команде с данным значением исполнительного адреса, причем задается тип команды: ввода/вывода, обмена с магнитным барабаном, прочие.

На первый взгляд описанная архитектура не отличается от архитектуры двоичных одноаккумуляторных машин: те же регистры, те

же операции. Однако при внимательном рассмотрении выявляются следующие принципиальные и важные для практики особенности.

1. Числа всех типов представлены единым натуральным кодом. В двоичных машинах для представления чисел разных типов и даже разного назначения приходится применять различный код, например: числа с фиксированной запятой представляют обычно дополнительным или обратным кодом, мантиссы чисел с плавающей запятой - прямым кодом, порядки - кодом с избытком, адреса памяти - натуральным двоичным кодом.

2. Операции определены над операндами, длина которых варьируется и может быть неодинаковой у первого и второго операнда. При этом не требуется никаких усложнений операционного устройства, ни вспомогательных команд в роде двоичной команды "расширения знака".

3. При усечении слова, например в случае присваивания длинного значения короткому регистру, автоматически получается наилучшее при данной укороченной длине представление первоначального значения, и вместе с тем сохраняемая часть слова копируется неизменной. В двоичной машине соответствующие возможности с известными оговорками можно обеспечить, лишь предусмотрев два варианта операций - с округлением и без округления.

4. Единственная операция сдвига выполняет функции всех двоичных операций сдвига - логического, арифметического, с округлением и без округления, причем выполняет

безукоризненно, чего нельзя сказать, например, об операции двоичного арифметического сдвига [6].

5. Знак числа в соответствии с общепринятой математической трактовкой этого понятия представлен трехзначной функцией, и разбиение чисел по знаку производится на положительные, отрицательные и равные нулю в противоположность сбивающей с толку двоичной традиции двузначного знака и отнесения нуля к положительным числам.

6. Интервал значения мантиссы нормализованного числа - от 0,5 до 1,5 по абсолютной величине - характеризуется значительно лучшей устойчивостью по сравнению с используемыми в двоичных машинах интервалами 0,5 - 1,0 и 1,0 - 2,0.

Эти и некоторые другие особенности троичной архитектуры обусловили математическое совершенство, компактность и простоту реализации набора команд "Сетуни", а главное, естественность, легкость понимания и применения машины пользователями.

АРХИТЕКТУРА ЦИФРОВОЙ МАШИНЫ "СЕТУНЬ 70"

Экспериментальный образец малой цифровой машины "Сетунь 70" эксплуатируется в МГУ с 1970 г., причем с 1974 г. на его основе функционирует автоматизированная система обучения "Наставник", обеспечивающая поточное обучение студентов языку программирования Фортран, проведение коллоквиумов и тестов, разработку учебных материалов, дидактические и психофизические эксперименты. Необычность архитектуры машины стимулировала разработку оригинальных системных программ и новой версии структурированного программирования.

В техническом отношении "Сетунь 70" характеризуется рядом усовершенствований по сравнению с "Сетунью". Так, реализация однопроводной передачи трехзначных сигналов позволила почти в 2 раза уменьшить число электрических соединений, логические элементы стали проще, миниатюрней и при большей релейности потребляют в 2,5 раза меньше энергии, значительно улучшены параметры троичной памяти и магнитной записи троичного кода. Дальнейшее развитие получила пороговая техника осуществления операций трехзначной логики. Разработанная применительно к электромагнитным средствам эта техника переносима и на полупроводниковые элементы, например, типа И2Л. Все же с точки зрения практики в настоящее время больший интерес представляют архитектурные особенности "Сетуни 70".

Минимальная непосредственно адресуемая единица главной памяти "Сетуни 70" - 6-тритный трайт (~9,5 бита) - на редкость удобна. Трайт лишь немногим больше 8-битного байта, но уже достаточно велик, чтобы закодировать, например, алфавит, включающий русские и латинские заглавные и строчные буквы, цифры, математические и служебные знаки. В трайте целое число как 9-ричных, так и 27-ричных цифр. Два трайта - это 19 битов, три трайта - почти 29 битов и т. д.

Представление чисел симметричным троичным кодом позволило легко реализовать последовательное выполнение арифметических

операций с варьируемой длиной операндов от 1 до 3 трайтов и длиной результата до 6 трайтов.

Благодаря симметричности кода просто и естественно реализованы реверсивные счетчики и указатели стеков, играющих в архитектуре машины важную роль.

Существенной чертой "Сетуни 70" является стековая организация процессора. Введение арифметического стека, т. е. магазина для автоматического запоминания промежуточных результатов, было обусловлено выбором в качестве языка машины польской инверсной записи (ПОЛИЗ), которую предпочли как удобный выходной язык трансляторов и вследствие ее компактности [7]. Машинная программа представляется в ПОЛИЗ последовательностью слов (или слогов), в которой различаются операционные и адресные слова. Адресное слово означает засылку в стек значения, хранимого в главной памяти по соответствующему адресу.

Операционное слово указывает одну из определенных над стеком операций, требуя выполнения этой операции процессором.

Ради экономии дорогой в те годы памяти были приняты короткие 6-тритные слова (трайты). Операционный трайт в двух старших тритах содержит нули, а в остальных 4-х тритах - код одной из 81 операций. Операции разбиты на три класса: 27 основных, 27 служебных и 27 макроопераций, программируемых пользователем.

Соответственно этим классам имеется три режима работы машины: основной или пользователя, прерывания и макро.

В адресном трайте первый (старший) трит указывает длину операнда, второй трит - один из трех регистров "приписки", содержащих номера открытых для доступа страниц памяти, остальные четыре трита - адрес операнда (старшего трайта операнда) на выбранной по регистру "приписки" странице. Счетчик команд и указатель арифметического стека имеют собственные регистры "приписки". Ширина стека - три трайта. Помимо стека в операционном устройстве имеется регистр порядков, длиной в 1 трайт, и регистры множителя и младшей части результата, по 3 трайта каждый.

Для автоматического сохранения адреса возврата при обработке прерывания и при выполнении макрооперации имеется второй (системный) стек, число позиций которого в первоначальном вари-

анте машины было ограничено двумя, а с введением команд структурированного программирования увеличилось до 26.

В 1975 г. машина подверглась модернизации, выразившейся в небольшом по объему переделок, но принципиальном усовершенствовании архитектуры на основе идеи структурированного программирования Э. Дейкстры [8, 9]. Двухстековая организация процессора и ПОЛИЗ оказались исключительно благоприятными для реализации структурированного программирования на уровне языка машинных команд. При этом в условиях новой дисциплины программирования стали несущественными затруднения, возникавшие в связи с мелкостраничной структурой памяти.

Все, что потребовалось сделать - это ввести команды ветвления, цикла и вызова подпрограммы вместо практически не употреблявшихся команд приращения, убавления и установки нуля в регистре порядков. Новые команды, в отличие от обычных в ПОЛИЗ однословных команд, представлены словосочетаниями. Например, команда JSR A вызова подпрограммы A включает операционный трайт JSR и следующий за ним трайт A, интерпретируемый как указатель начала подпрограммы, находящейся в 9-страничной оперативной памяти. Тело подпрограммы заканчивается трайтом RMC, означающим возврат к продолжению программы, из которой произведен вызов подпрограммы.

Команда тройственного ветвления BRT A1 A2 A3 состоит из 4-х трайтов и задает вызов одной из трех подпрограмм в зависимости от знака текущего значения вершины арифметического стека: если минус, то Al, если нуль, то A2, если плюс, то A3. Команда цикла DOWA повторно вызывает подпрограмму A, пока текущее значение вершины не равно нулю.

С появлением данных команд старые команды безусловного и условных переходов вышли из употребления, хотя и сохранены в машине. Развитая на основе новых команд версия структурированного программирования стала истоком системы ДССП [10] для двоичных микрокомпьютеров.

Литература

1. Брусенцов Н.П. Пороговая реализация трехзначной логики электромагнитными средствами. - В кн.: Вычислительная техника и вопросы кибернетики. Вып. 9. М.: Изд-во МГУ, 1972, с. 3 - 35.

2. Epstein G., Frieder G., Rine D. C. The development of multiple - valued logic as related to computer science. - Computer, 1974, Sept., vol. 7, N 9, p. 20-32.

3. Аннотированный указатель программ для вычислительной машины "Сетунь". Составители: Н. П. Брусенцов, В. А. Морозов. - Фонд алгоритмов и программ Минвуза СССР. М.: Изд. ВЦ МГУ. Вып. 1; 1968. Вып. 2, 1971.

4. Брусенцов Н.П. Опыт разработки троичной вычислительной машины. - Вестн. Моск. ун-та. Сер. 1: математика, механика, 1965, # 2, с. 39-48.

5. Брусенцов Н.П., Маслов С.П., Розин В.П., Тишулина А.М. Малая цифровая вычислительная машина "Сетунь". М.: Изд-во МГУ, 1965.

6. Steele G.L. Arithmetic shifting considered harmful. - SIGPLAN Notices, Nov. 1977, vol. 12, N 11, p. 61-69.

7. Брусенцов Н.П., Жоголев Е.А. Структура и алгоритм функционирования малой вычислительной машины. - В кн.: Вычислительная техника и вопросы кибернетики. Вып. 8. Л.: Изд-во ЛГУ, 1971, с. 34-51.

8. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975.

9. Брусенцов Н.П., Рамиль Альварес Х. Структурированное программирование на малой цифровой машине. - В кн.: Вычислительная техника и вопросы кибернетики. Вып. 15. М.: Изд-во МГУ, 1978, с. 3-8.

10. Брусенцов Н.П., ЗлаткусГ.В., Руднев И.А. ДССП - Диалоговая система структурированного программирования. - В кн.: Программное оснащение микрокомпьютеров. М.: Изд-во МГУ, 1982, с. 11 - 40.

УДК 583.1

М. В. Харинов

Недвоичная логика запоминания информации в изображении

(*Санкт-Петербургский институт информатики и автоматизации РАН*)

Рассматриваются элементы формализма представления видеоинформации для компьютерной обработки. Описываются системы счисления, предназначенные для точного моделирования запоминания информации.

The elements of formalism for image information computational interpretation are presented. The nonbinary number systems intended for accurate simulation of information storing are described.

Введение

Полвека назад главными характеристиками первых ЭВМ являлись объем памяти, быстродействие и надежность, которые определяли принципиальную возможность требуемых вычислений и являлись достаточными аргументами в пользу разработки и создания вычислительных устройств с недвоичными системами счисления [1–2]. При современных темпах развития вычислительных средств, когда доступная память обеспечивает комфортное программирование, а рост быстродействия ЭВМ снижает смысл усилий по ускорению расчетов в ограниченное число раз, исторические результаты [1–2] сохраняют прежнее значение, например, для портативных вычислительных устройств, которые сталкиваются с теми же проблема-

ми, что и первые ЭВМ. Однако для полноценного использования достигнутых решений в современных условиях, вероятно, требуется развитие обоснования их уникальных преимуществ в дополнительных классах приложений.

Числовые представления на основе системы остаточных классов (СОК) и тернарные числовые представления имеют общие области приложения, к которым относится цифровая обработка сигналов (ЦОС), включающая задачи автоматического распознавания и задачи защиты информации. В задачах распознавания СОК привлекательна тем, что обеспечивает логическое распараллеливание вычислений на низком уровне арифметических действий, выполняемых с возможностью компенсации нарушения данных. Тернарная логика запоминания результатов вычислений в ЦОС позволяет без дополнительных соглашений считать точнее и запоминать больше данных, чем двоичная логика. Имеется в виду, что, огрубляя результаты вычислений в тернарной логике, нетрудно воспроизвести результаты вычислений в двоичной, но не наоборот. Целью статьи является пояснение без технических деталей принципов применения тернарной логики и систем счисления в ЦОС. Термин «тернарный» употребляется в смысле обобщения термина «троичный» на случай применения, наряду с троичной, также и «псевдотроичной» системы счисления, которая описывается в следующем разделе.

Псевдотроичная система счисления.

Ключевой проблемой ЦОС, в частности, обработки изображений является зависимость ее результатов от изменения освещённости, геометрии съёмки, смены окружения объектов, а также видоизменения самих объектов. Одним из способов решения является преобразование изображения к некоторому инвариантному представлению, при котором компенсируется влияние изменения тех или иных входных параметров [3].

Формальной основой представления изображения в инвариантном виде является псевдотроичная система счисления (рис. 1), в которой неотрицательные целые числа I_0 раскладываются по степеням 2, как в обычной двоичной системе, но записываются в виде последовательности цифр λ со значениями от 0 до 2, как в троичной системе:

$$I_0 = \lambda_0 + 2 \cdot \lambda_1 + 4 \cdot \lambda_2 + ... + 2^k \cdot \lambda_k + ...,$$

где коэффициенты разложения λ_k определяются рекуррентными соотношениями:

$$\lambda_k = I_k - 2I_{k+1}; \quad I_{k+1} = \begin{cases} [I_k / 2], I_k - \text{нечётное}, \\ 2[I_k / 4], I_k - \text{чётное}, \end{cases} \quad k = 0, 1, 2...,$$

а квадратные скобки обозначают «целую часть».

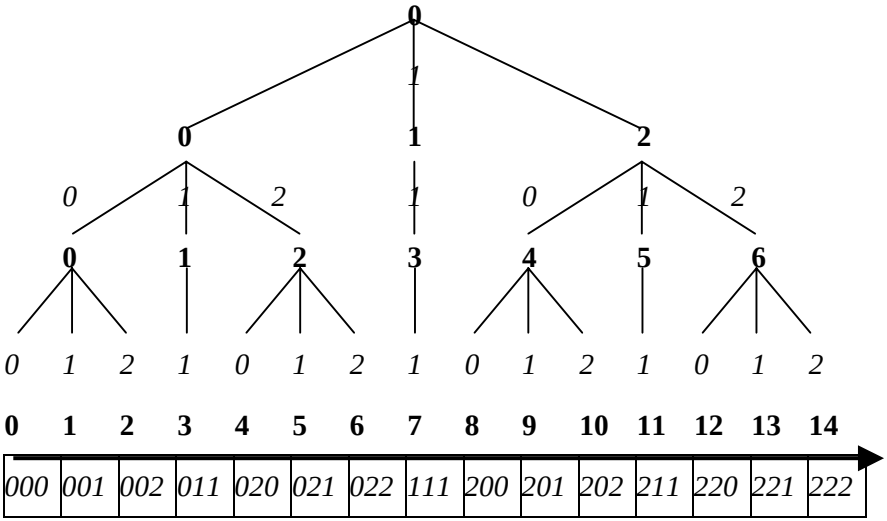


Рис. 1. Неотрицательные целые числа

На рис. 1 показана схема кодирования чисел в псевдотротроичной системе счисления. Курсивом выписаны цифровые обозначения в псевдотроичной системе, обычным шрифтом выписаны числа в десятичной системе. Коэффициенты разложения λ_k сопоставляются дугам некоторого дерева с узлами I_k, которое строится на начальном отрезке неотрицательных целых чисел. Код числа описывает путь по дугам дерева к числу, заданному на числовой оси.

В отличие от классических позиционных систем счисления [4], однозначность псевдотроичного представления чисел достигается тем, что допускаются не все сочетания цифр (чётные числа описываются чередованием 0 и 2, а нечётные - чередованием 0 и 2 с заключительной последовательностью из одних 1). Поэтому псевдо-

троичную систему можно называть также позиционной псевдосистемой счисления.

Таким образом, число 21, например, кодируется в виде 2021, обозначающем разложение этого числа по степеням 2. Обычную двоичную запись можно трактовать как представление числа в псевдотроичной системе, если первый неустановленный бит считать разделителем между записью числа посредством 1 в младших разрядах и записью посредством 0 и 2 в старших разрядах (рис. 2)

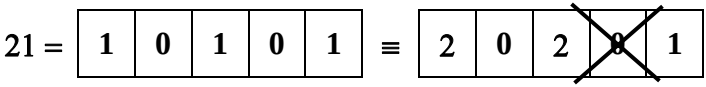


Рис. 2.

На рис.2 показана связь кодирования в двоичной системе счисления (слева) и псевдотроичной системе (справа). Биты выписаны в порядке убывания слева направо. При этом младшее число из одних 1 представляет собой число Мерсенна (степень 2 без 1) либо 0, а старшее число, кратное соответствующей степени 2 кодируется чередованием 0 и 2, заданной последовательностью старших битов. Указанная интерпретация позволяет выполнять поразрядные действия с трехзначными элементами чисел непосредственно в битовом представлении.

Если кодирование чисел в двоичной системе выполняется на основе традиционной двузначной (бинарной) логики, то кодирование и запоминание чисел в псевдотроичной системе выполняется в вырожденной троичной логике с импликацией «из 1 следует 1». Указанная логика и система счисления без неиспользуемых кодов троичной системы счисления и ошибок округления двоичной системы счисления поддерживает выбор между равноправными альтернативами в алгоритмах разделения конечных множеств.

В обработке изображений псевдотроичная система счисления используется для преобразования изображения в некоторое новое инвариантное представление, которое коммутирует с преобразованием изображения в негатив и не зависит от его стандартных преобразований — упаковки, растяжения, эквидистантной нормировки по яркости и др. [3]. В рамках указанных преобразований компен-

сируется изменчивость условий получения изображения и повышается надежность распознавания объектов.

Благодаря псевдотроичной системе инвариантное представление строится и запоминается как изоморфный по яркостному порядку образ изображения, который посредством арифметических преобразований яркостных значений определяет иерархическую последовательность гомоморфных образов и, в свою очередь, поддерживает вычисления в модели «виртуальной видеопамяти». Понятие виртуальной памяти вводится в классической позиционной троичной системе счисления и используется для описания обработки изображения в терминах считывания и записи информации.

Модель виртуальной видеопамяти.

В модели виртуальной видеопамяти информация изображения раскладывается на инвариантную и переменную компоненты. Обработка описывается как обратимое встраивание кодов произвольного сообщения за счет модификации переменной компоненты информации, которое не влияет на вычисление инвариантной компоненты. Встраивание сообщения описывается как его запоминание в «виртуальной» памяти, которая приписывается изображению формально, но используется для записи и считывания произвольных кодов сообщения подобно реальной компьютерной памяти. Полагается, что виртуальная память состоит из запоминающих элементов, которые вводятся посредством обобщения понятия битов.

Разряды виртуальной памяти определяются последовательностью вложенных диапазонов шкалы яркости, которые вычисляются в алгоритме [5] итеративного разделения шкалы яркости по гистограмме на приблизительно равновесные части. Итеративное разбиение яркостной шкалы продолжается до тех пор, пока каждый диапазон не выродится в диапазон, содержащий единственную яркость, которая сопоставляется последовательности стягивающихся к ней диапазонов и на каждой итерации принадлежит одному из них.

Полагается, что ячейки виртуальной памяти отвечают точкам изображения и состоят из последовательных запоминающих элементов. Значение очередного элемента i-й ячейки виртуальной памяти, в зависимости от номера итерации разбиения яркостной шкалы, определяется положением i-й яркости относительно центра оче-

редного диапазона яркости. При этом данному элементу ячейки виртуальной памяти приписывается положительное, отрицательное, либо нулевое значение знака разности яркости точки и центральной яркости рассматриваемого диапазона (рис. 3).

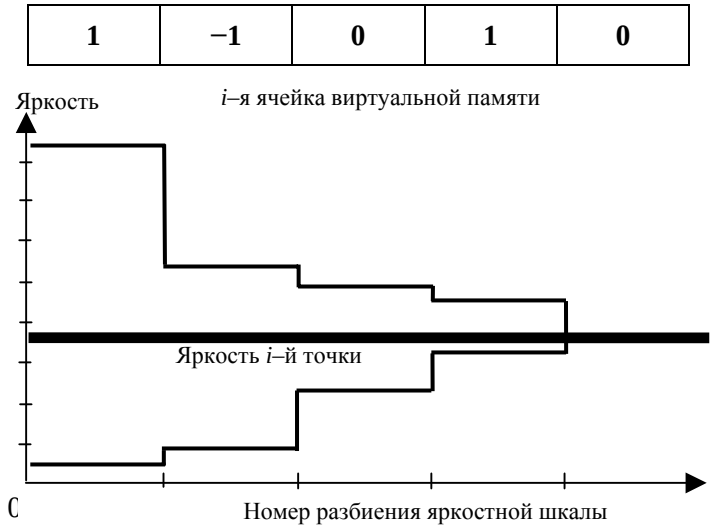


Рис. 3.

Тем самым определяется считывание троичных единиц информации, которые согласно Н.П. Брусенцову называются тритами [2]. Триты, вычисленные для данного разбиения шкалы яркости, составляют каналы виртуальной памяти и считаются упорядоченными по уменьшению вложенных диапазонов яркости. При этом самый старший трит каждой ячейки виртуальной памяти вычисляется по рабочему диапазону яркости, который содержит яркости всех точек изображения. Запись сообщения в триты виртуальной памяти связывается с отражением яркостного значения относительно центра соответствующего диапазона и выполняется последовательно от старших тритов — к младшим. Яркостное значение, оказавшееся в центре диапазона, очевидно, при отражении не меняется. Поэтому триты с нулевыми значениями при записи сообщения не подлежат модификации и считаются неактивными. К неактивным относят также триты, изменение которых влечет модификацию предшествующих тритов.

Основные принципы встраивания сообщения в виртуальную память включают следующие требования инвариантности:

- инвариантность изображения с сообщением относительно повторного встраивания сообщения (идемпотентность встраивания);
- инвариантность исходного изображения относительно обратного встраивания извлеченного из него сообщения;
- инвариантность сообщения относительно линейных и нелинейных изоморфных (по яркости) преобразований изображения с сообщением;
- инвариантность объема сообщения относительно преобразования исходного изображения в негатив.

Избыточность изображения выражается повторениями тритов по координатам и каналам виртуальной памяти. Искажение кодов сообщения в процессе передачи компенсируется их простым суммированием с последующим вычислением знака полученной суммы. Для учета случая равновесного распределения альтернативных значений битов сообщения, в качестве запоминающих элементов виртуальной памяти необходимо использовать именно триты, а не биты. Особенности единиц представления информации в виртуальной памяти, по сравнению с исходным битовым представлением в компьютерной памяти отражены в таблице.

Таблица.

Единицы представления и запоминания видеоинформации

Атрибуты	Единицы	
	Биты	Триты
Порядковый номер	0, 1, 2, ..., 7	0, 1, 2, ..., Ch
Состояние	0, 1	± 1, 0
Статус	RW	R, RW

где Ch — число каналов виртуальной памяти, R и RW — обозначения неактивных и активных значений яркости.

Заключение.

Практическим обоснованием модели виртуальной памяти служит реализация на ее основе метода многоканальной адаптивной стеганографии [6], в котором достигается повышение объема встраиваемых сообщений до 20–30% от объема исходного изображения за

счет их наложения по координатам. Теоретическое обоснование опирается на комбинаторный подход А. Н. Колмогорова к определению понятия количества информации [7]. При этом, однако, вопреки сложившимся стереотипам алгоритмы строятся не в бинарной, а в тернарной логике.

Противопоставление двоичной, троичной или иной системы счисления другим системам, вероятно, ограничивает возможности формализации понятия информации для компьютерного вычисления. Общей проблемой для позиционных и отличных от них систем счисления остается оптимизация переходов из одной системы в другую [4]. Возможно, сопоставление различных систем счисления со временем приведет к новым способам оптимизации представления чисел.

Литература

1. *Малашевич Б. М.* ЭЦВМ «5Э53», 2005. http://www.computer-museum.ru/histussr/5e53.htm
2. *Брусенцов Н.П.* Вычислительная машина «Сетунь» Московского государственного университета. — В кн.: Новые разработки в области вычислительной математики и вычислительной техники. — Киев, 1960. — С. 226–234. http://www.computer-museum.ru/histussr/setun2.htm
3. *Харинов М. В., Горохов В. Л.* Псевдотроичная система счисления и анализ изображений // Известия вузов. Радиоэлектроника / Вып. 2, — СПб, 2003. — С. 49–53.
4. *Кнут Д.* Искусство программирования для ЭВМ. Получисленные алгоритмы, — М.: Мир, 1977. — т.2, — С. 201–224.
5. *Прэтт У.* Цифровая обработка изображений. Том 1–2. — М.: Мир, 1982. — 1200 с.
6. M. *V. Kharinov.* Representation of Image Information for Machine Computation // Pattern Recognition and Image Analysis / Vol. 15. No 1, 2005. — pp. 212–214.
7. *Колмогоров А. Н.* Три подхода к определению понятия «Количество информации» // Проблемы передачи информации. — 1965. — Вып. 1, — том 1. — С. 3–8.

23 мая 2005 года ушел из жизни один из активных приверженцев Модулярной арифметики, доктор технических наук, профессор, проректор по информатизации и образовательным технологиям, заведующий кафедрой информационно-вычислительных систем Чувашского госуниверситета им. И.Н. Ульянова

Евгений Константинович Лебедев.

Наука и высшая школа понесли невосполнимую утрату. Е.К. Лебе-

дев стоял у истоков развития нового направления вычислительной техники – цифровой обработки сигналов, был организатором внедрения в регионе инновационных информационных технологий в образование, автором более 150 научных публикаций, монографий и учебных пособий по теории и методам ЦОС, информационным технологиям, в том числе и новейшей – дистанционному образованию. Им написан целый ряд учебников для студентов и школьников по информатике и микропроцессорным системам, спектральному анализу и поразрядной фильтрации.

Существенный вклад Евгений Константинович внес в развитие Модулярной арифметике, используя ее для обработки сигналов, в т.ч. в разработанном им изделии «Вычет». Результаты исследований и разработок Евгения Васильевича нашли отражение в его публикациях и монографиях.

Евгений Константинович родился 11 августа 1939 года в Звенигородском районе Московской области в семье военнослужащего, окончил школу с медалью и в 1957 году поступил на радиотехнический факультет Казанского авиационного института. В 1963 году он стал радиоинженером и успешно начал трудовую деятельность в ОКБ С.П. Королева.

В 1967 г. Е.К. Лебедев был приглашен на работу в Марийский политехнический институт им. М. Горького на открывшийся радиотехнический факультет. В 1968 г. он поступил в аспирантуру при кафедре 411 Московского авиационного института, где вскоре досрочно защитил диссертацию. В 1974 г. ему присвоено ученое звание доцента по кафедре «Радиотехника». Все последующие годы работы в вузе были отданы научной, педагогической и изобретательной деятельности. Блестящий оратор, талантливый педагог, крупный ученый и изобретатель – первое из многих своих авторских свидетельств СССР «Цифровое устройство селекции движущихся целей» он получил вскоре после окончания аспирантуры. Решением ВАК РФ от 22 декабря 1995 г. Евгению Константиновичу была присуждена ученая степень доктора технических наук, а 20 ноября 1996 г. – звание профессора по кафедре информационно-вычислительных систем. В этот период он возглавлял кафедру ИВС в Марийском государственном техническом университете им. М. Горького и параллельно занимал должность начальника аналитического отдела ОАО «Мартелком» Республики Марий Эл.

После встречи с ректором Чувашского государственного университета академиком Кураковым Л.П. на одной из научных конференций профессор Лебедев получил приглашение на работу в Чувашию.

21 октября 1996 года Е.К. Лебедев избирается на должность заведующего кафедрой информационных технологий и прикладной математики. С этого дня он отдает всего себя открытию в ЧГУ новой престижной специальности «Вычислительные машины, комплексы, системы и сети», ведет огромную организационную работу по оснащению кафедры современными средствами вычислительной техника и по обеспечению студентов учебными и учебно-методическими пособиями и материалами. Его лекторское мастерство, человеческие качества снискали заслуженное уважение студентов, аспирантов, коллег, будучи в течение ряда лет заместителем председателя профкома ЧГУ, он вел большую общественную заботу.

В 2000 году Евгений Константинович назначается на должность проректора по информатизации, осуществляет координацию всей деятельности университета в области новых информационных технологий и центра «Интернет», курирует работу Научной библиотеки, типографии, аспирантуры, РИО, редакции журнала «Вестник университета», вычислительного центра ЧГУ.

Красивый, порядочный, умный, талантливый человек, прекрасный семьянин оставил о себе светлую память. Все мы будем помнить его, выражая в эти горькие минуты прощания глубокое соболезнование жене, детям, родным и близким покойного.

Жизнь ученого и педагога оборвалась в результате тяжелой, коварной болезни, когда он был полон творческих и жизненных планов. Будучи одним из энтузиастов Модулярной арифметики, он планировал принять активное участие в Международной научно-технической конференции «50 лет Модулярной арифметике» и готовил материалы для нее, но, к величайшему сожалению, не успел завершить их.

Оргкомитет
Международной научно-технической конференции
«50 лет модулярной арифметике»

БИБЛИОГРАФИЯ

Труды по Модулярной арифметике в библиографии приведены в алфавитном порядке по фамилиям авторов, а авторские свидетельства и патенты на изобретения, в основном, – в порядке их номеров.

Важнейшие монографии

1. **Акушский И.Я., Юдицкий Д.И.** Машинная арифметика в остаточных классах. – М.: Советское радио, 1968, 440с.
2. **Амербаев В. М.** Теоретические основы машинной арифметики. — Алма-Ата: Наука, 1976. 324 с.
3. **Акушский И.Я., Амербаев В.М., Пак И.Т.** Основы машинной арифметики комплексных чисел. – Алма-Ата: Наука, 1970. – 248 с.
4. **Чернявский А.Ф., Данилевич В.В., Коляда А.А., Селянинов М.Ю.** Высокоскоростные методы и системы цифровой обработки информации. Мн.: Белгосуниверситет, 1996. 376 с.
5. **Евстигнеев В.Г.** Недвоичная машинная арифметика и специализированные процессоры. Под редакцией И.Я. Акушского. Москва, МИФИ СЕРВИС, 1992 г., 266 с. Тираж 5000 экз.
6. **Инютин С.А.** Арифметико-логические основы вычислительных систем. -Сургут: РИО, 2001. –120с.
7. **Инютин С.А.** Основы многоразрядной алгоритмики. - Сургут: РИО, 2002. -137с.
8. **Коляда А. А., Пак И. Т.** Модулярные структуры конвейерной обработки цифровой информации. — Мн.: Университетское, 1992. — 256 с.
9. **Лебедев Е. К.** Быстрые алгоритмы цифровой обработки сигналов: Монография. — Красноярск, 1989.
10. **Торгашев В.А.** Система остаточных классов и надёжность ЦВМ. – М.: Советское радио, 1973. – 120 с.
11. **Финько О.А.** Модулярная арифметика параллельных логических вычислений: Монография / Под. ред. В.Д. Малюгина; — М.: ИПУ РАН, 2003. — 224 с.
12. **Жихарев В.Я., Илюшко Я.В., Кравець Л.Г., Краснобаев В.А.** Методы и средства обработки информации в непозиционной системе счисления в остаточных классах. – Житомир: “Волынь”, 2005. – 220с.
13. **Краснобаев В. А. и др**. Методы повышения надежности специализированных ЭВМ систем и средств связи. Харьков: ХВВКМУ РВ, 1990. - 172 с.
14. **Краснобаев В. А., Приходько С. И., Снисаренко А.И.** Помехоустойчивое кодирование в АСУ. - Харьков: МО СССР, 1990. - 151 с.
15. **Синьков М.В., Губарени Н.М.** Непозиционные представления в многомерных числовых системах. – Киев: Наукова думка, 1979. – 137 с.

16. Торгашев В.А. Система остаточных классов и надёжность ЦВМ. – М.: Советское радио, 1973. – 120 с.
17. **Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А.** Модулярные параллельные вычислительные структуры нейропроцессорных систем / М.: Физматлит, 2003. – 288 с.
18. **Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н.** Нейрокомпыотеры в остаточных классах. Кн. 11 /— М.: Радиотехника, 2003. 272 с.
19. **Червяков Н. И., Сахнюк .А. и др.** Модулярные параллельные вычислительные структуры нейропроцессорных систем. — М.: Физматлиг, 2003. — 288 с.

Полная библиография

Agarwal D. P., "Modulo *2**n* + 1 arithmetic logic," *IEEJ. Electronic Circuits and Syst.*, vol. 2, no. 6, pp. 186-188, Nov. 1978.

Agarwal R. C. and Burrus C. S., "Number theoretic transforms to implement fast digital convolution," *Proc. IEEE.* vol. 63, no. 4, pp. 555-560, Apr. 1975.

Aiken H. H. and Semon W., "Advanced digital computer logic," Tech. Rep. WADC TR 59-472, Cambridge, MA, July 1959.

Aiken H. H., Theory of switching, Computation Lab., Harvard Univ., Cambridge, Mass., Rep. № BL-23, June 1959. Aiken H. H., Semon W., Advanced digital computer logic, WADC TR-59-472, July 1959.

Aiken H., Semon W., Advanced digital computer logic, Wright Air Dev. Ctr., Dayton, Ohio, Techn. Rept. № TR-59-472, July, 1959

Aiken H., Semon W., Advanced digital computer logic, Wright Air Dev. Ctr., Dayton, Ohio, Techn. Rept. № TR-59-472, July, 1959

Akers S. B. On a Theory of Boolean Functions // Society for Industrial and Applied Mathematics. - 1959. - V61. 7, № 4. - P. 487-498.

Alia G., Martinelli E. Optimal VLSI complexity design for high speed pipeline FFT using RNS // Comput. and Elec. Eng. 1998. Vol. 24, N3. P.167–182.

Arora R. K. and Sharma S., "Correction of multiple errors and detection of additive overflow in residue code," *Inform. Contr.*, vol. 39, no. 1, pp. 46-54, Oct. 1978.

Banerji D. K. and Brzozowski J. A., "On translation algorithms in residue number systems," *IEEE Trans. Comput.*, vol. C-21, no. 12, pp. 1281-1286, Dec. 1972.

Banerji D. K. and Brzozowski J. A., "Sign detection in residue number systems," *IEEE Trans. Comput.*. vol. C-18, pp. 310-320, Apr. 1969.

Banerji D. K., "A novel implementation method for addition and subtraction in residue number systems," *IEEE Trans. Comput.*, vol. C-23, pp. 106-109, Jan. 1974.

Banerji D. K., "On combinational logic for sign detection in residue number

systems," *3rd Symp. Computer Arithmetic,* pp. 113-116, Nov. 1975.
Banerji D. K., "On the use of residue arithmetic for computation," *IEEE Trans. Comput.*. vol. C-23, pp. 1315-1317, Dec. 1974.
Banerji D. K., Cheung To-Yat, and Ganesan V., "A high speed division method in residue arithmetic," in *Proc. 5th Symp. Computer Arithmetic,* pp. 158-164, May 1981.
Baraniecka A. Z. and Jullien G. A., "Hardware implementation of convolution using number theoretic transforms," in *Proc. 1979 IEEE Intern. Conf. Acous., Speech, and Signal Processing, ftp.* 490-493, Apr. 1979.
Baraniecka A. Z. and Jullien G. A., "On decoding techniques for residue number system realization of digital signal processing hardware," *IEEE Trans. Circuits Syst.*, vol. CAS-25, no. 11, pp. 935-936, Nov. 1978.
Baraniecka A. Z. and Jullien G. A., "Residue number system implementations of number theoretic transforms in complex residue rings," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-28, no. 3, pp. 285-291, June 1980.
Baraniecka A. Z., "Digital filtering using number theoretic techniques," Ph.D. dissertation, Dept. of Electrical Engineering, University of Windsor, Windsor, Ontario, Canada, 1980.
Baraniecki A. and Jullien G. A., "Quantization error and limit cycle analysis in residue number system coded recursive filters," presented at Proc. 1982 IEEE Intern. Conf. Acous., Speech, and Signal Processing, May 1982.
Barsi B. F. and Maestrini P., "Arithmetic codes in residue number systems with magnitude index," *IEEE Trans: Comput.,* vol. 27,no. 12,pp. 1185-1188, Dec. 1978.
Barsi F. and Maestrini P., "Application of residue arithmetic to recursive digital filters," in *Proc. 26th Midwest Symp. Circuits and Systems,* Puebia Mexico, pp. 394-397, Aug. 1983.
Barsi F. and Maestrini P., "Error codes in residue number systems with non-pairwise-prime moduli," *Information and Control.* vol. 46, no. 1, pp. 16-25, July, 1980.
Barsi F. and Maestrini P.. "Error correcting properties of redundant residue number systems," *IEEE Trans. Comput.*, vol. C-18, pp. 307-315, Mar. 1973.
Bartee T. C. and Schneider D. I., "Computation with finite fields," *Inform. Contr.*. vol. 6, pp. 79-98, 1963.
Baugh R. A. and Day E. C., "Electronic sign evaluator for residue number systems," Tech. Rep. TR-60-597-32, RCA, Camden, NJ, and Burlington, MA, Jan. 1961.
Bayoumi M. A. and Juliien G. A., "Analysis of microprocessor based digital filters," in *Proc. Intern. Symp. on Mini and Microcomputers and Their Applications,* San Francisco, CA, pp. 174-182, May 1983.
Bayoumi M. A. and Jullien G. A., "A high performance VLSI arithmetic unit using residue number systems for digital signal processing applications," in *Proc. ACM Computer Science Conf.*. Feb. 1983.

Bayoumi M. A., Juiiien G. A. and Miller W. C., "A VLSI model for residue number system architecture," *Integration^ VLSI J.*, 1984.
Bayoumi M. A., Juliien G. A. and Miller W. C., "A high speed VLSI adder," in *Proc. Intern. Electrical Electronics Conf.,* Toronto, ON, Canada, pp. 300-303, Sept. 1983.
Bayoumi M. A., Juliien G. A. and Miller W. C., "An area-time efficient NMOS adder," *Integration, VLSI J.*, vol. 1, no. 4, pp. 317-334, Apr. 1983.
Bayoumi M. A., Juliien G. A. and Miller W. C., "Models for VLSI implementation of residue number system arithmetic modules," in *Proc. 6th Symp. Computer Arithmetic,* Aar-hus, Denmark, June 1983.
Bayoumi M. A., Jullien B. A. and Miller W. C., "RNS modules for filters," *in, Proc. 26th Midwest Symp. On Circuits and Systems,* Louisville, KY, pp. 403-407, Aug. 1985.
Bayoumi M. A., Jullien G. A. and Miller W. C., "A hybrid VLSI architecture of FIR filters using the residue number systems," *Electron. Lett.*, Apr. 1985.
Bayoumi M. A., Jullien G. A. and Miller W. C., "A modular implementation of digital signal processing architectures using RNS," in *Proc. 15th Pittsburgh Conf. on Modelling and Simulation.* Apr. 1984.
Bayoumi M. A., Jullien G. A. and Miller W. C., "A systolic (VLSI) array using RNS for digital signal processing applications," in *Proc. 1984 ACM Computer Science Conf.*. pp. 115-120, Mar. 1984.
Bayoumi M. A., Jullien G. A. and Miller W. C., "A VLSI implementation of finite impulse response digital filters using residue numbersystems," *Canadian Electrical Eng. J.*. 1984.
Bayoumi M. A., Jullien G. A. and Miller W. C., "Highly parallel architectures for DSP algorithms using RNS," in *Proc. IEEE Intern. Symp. Circuits and Systems,* Kyoto, Japan, pp. 1395-1398,June 1985.
Bayoumi M. A., Jullien G. A. and Miller W. C., "Multi lookup table module for RNS systems implementations," *Electron. Lett.*, vol. 20, no. 2, pp. 94-95, Jan. 1984.
Bayoumi M. A., Jullien G. A. and Miller W. C., "The area-time complexity of a VLSI residue number system arithmetic unit," in *Proc. 8th Conf. Information Sciences and Systems.* Johns Hopkins University, Mar. 1983.
Bayoumi M. A., Jullien G. A. and Sid-Ahmed M. A., "A VLSI implementation of memory intensive residue number system architecture," in *Proc. 20th Afterton Conf.,* pp. 150-152, Oct. 1982.
Bayoumi M. A., Jutlien G. A. and Miller W. C., "An array implementation of digital filers," in *Proc. 1 984 IEEE Intern. Symp. Circuits Syst.,* Montreal, Canada, pp. 1268-1271, May 1984.
Beadles R. L., "A residue class arithmetic multiplication algorithm for prime number moduli," M.S.E.E. Dissertation, Dept. of Electrical Engineering, University of Pittsburgh, 1966.
Beckmann P. E., Musicus Bruce R. Fast fault-tolerant digital convolution using a polynomial residue number system // IEEE Trans. On Signal Processing. -

1993. - Vol. 41, № 7. - P. 2300-2313.
Bell M. J. and Jenkins W. K., "A unified mixed radix converter and error checker for a five-modulus residue number system," in *Proc. 1 984 IEEE Intern. Conf. ASSP.* San Diego, CA, pp. 4.1A4.1-4.1A4.4, Mar. 1984.
Bioul G., Davio M., and Quisquater J. J., "A computation scheme for an adder modulo 2 * **n* 4- 1," *Digital Processes* (Switzerland), vol. 1, no. 4, pp. 309-318, Winter 1978.
Blahut R. E., *East Algorithms for Digital Signal Processing.* Reading, MA: Addison-Wesley Publishing Co., 1985.
Blum T., Paar C. Montgomery modular exponentiation on reconfigurable hardware // 14-th IEEE Symp. on computer arithmetic (AIIITH14). April, 1999, Proceedings. - Los Alamitos, California: IEEE Computer Soc. Press, 1999. - P. 70-77.
Bool G. An Investigation of the Laws of Thought. London, Walton, 1984.
Brent R. P. and Kung H. T., "A regular layout for parallel adders," *IEEE Trans. Computers,* vol. C-31, no. 2, pp. 260-264, Mar. 1982.
Brule J. D., "Fast convolution with finite field fast transforms," *IEEE Trans. Acoustics, Speech, and Signal Processing.* vol. ASSP-23, p. 240, Apr. 1975.
Bupta H., *Selected Topics in Number Theory.* Kent, England: Abacus Press, 1980.
Carhoun D. O., Johnson B. L. and Redinbo G. R., "A synthesis algorithm for recursive finite field FIR digital filters," in *Proc. 1983 IEEE Intern. Symp. Circuits and Systems,* Newport Beach, CA, pp. 689-693, May 1983.
Cheney P. W., "A digital correlator based on the residue number system," *IRE Trans. Electron. Comput.,* vol. EC-11, pp. 63-70, Mar. 1961.
Cheney P. W., A digital correlator based on the residue number system, IRE Trans. on Electronic Computers, EC-10 (1961), March, 63—70.
Cheng V. S. and Huang C. H., "On the decoding of residue numbers," in *Proc. ISMM Intern. Symp. on Mim'and Micro Computers in Controls and Measurement,* San Francisco, May 1981.
Cho G. Y., Johnson L. G., Soderstrand M. A. New complex-arithmetic heterodyne filter. In *ISCAS (3)*, pages 593–596, 2004.
Chopper K. V. and Soderstrand M. A., "Implementation of very high speed recursive digital filters using commercially available devices," in *Proc. 1 985 Midwest Symp. Circuits and Systems.* Louisville, KY, pp. 667-670, Aug. 1985.
Clark G. A., Soderstrand M. A. and Johnson T. G., "Transform domain adaptive filtering using a recursive DFT," in *Proc. 1985 IEEE Intern. Symp. Circuits and Systems,* Kyoto, Japan, pp. 1113-1116, June 1985.
Cosentino R. J., "Fault-tolerant design of a systolic RNS FIR filter," Tech. Rep. MTR9613, MITRE Corp., Bedford, MA, Mar. 1985.
Cozzens J. H. and Finkelstein L. A., "Computing the discrete Fourier transform using residue number systems in a ring of algebraic integers," *IEEE Trans. Inform. Theory.* vol. IT-31, no. 5, pp. 580-588, Sept. 1985.

Debnath R. C. and Pucknell D. A., "On multiplicative overflow detection in residue number system," *Electron. Lett.*. vol. 14, no. 5, pp. 129-130, Mar. 1978.

Dickson L. E., *History of the Theory of Numbers, Vol. II.* Washington, DC: Carnegie Institution of Washington, 1920.

Dormido S., Canto M. A. An upper bound for the syntesis of generalized parallel counter // IEEE Trans. Cornput. - 1982. - Vol. C-34, X» 8. -P. 802-805.

Earl E., Swartzlander J. R. Parallel counters // IEEE Trans. Comput. -1973. - Vol. C-22, № 6. - P. 1021-1042.

Eichmann G., Keybl J., and Mammone R., "Number theoretic transforms," in *Proc. Society of Photo-Optic Instrumentation* (England), vol. 209, pp. 66-72, Oct. 1979.

Escott R. A. and Soderstrand M. A., "Applications of multiple-valued logic to residue number system computation," in *Proc. 25th Midwest Symp. Circuits and Systems,* Houghton, Ml, pp. 366-400, Aug. 1982.

Etzel M. and Jenkins W. K., "Redundant residue number systems for error detection and correction in digital filters," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-29, no. 5, Oct. 1980.

Etzel M. H. and Jenkins W. K., "Digital filters with fault tolerance," in *Proc. 1979 Joint Automatic Control Conf.,* Denver, CO,pp.187-192,June 1979.

Etzel M. H. and Jenkins W. K., "Error correction and overflow suppression properties of RRNS digital filters," in *Proc. 1980 IEEE Intern. Conf. Acous., Speech, and Signal Processing,* Denver, CO, pp. 1117-11 20, Apr. 1980.

Etzel M. H. and Jenkins W. K., "Hardware implementation of failure resistant residue number digital filters," in *Proc. IEEE Intern. Symp. Circuits and Systems,* Chicago, IL, pp. 88-91, Apr. 1981.

Etzel M. H. and Jenkins W. K., "The design of specialized residue classes for efficient recursive digital filter realization," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-30, no. 3, pp. 370-380, June 1982.

Etzel M. H., "Residue number system recursive digital filtering with error correction capabilities," Tech. Rep. T-93, Coordinated Science Laboratory, University of Illinois, Urbana, IL, Feb. 1980.

Etzel M. N., Jenkiw W. K. Redundant residue number system for error defection and correction in digital filters // IEEE Trans. On Acoust., and Speech and Signal Processing. - 1980. - FSSP-29, № 5. - P. 538-545.

Falkowski B. J. A Note on the Polynomial Form of Boolean Functions and Related Topics. // IEEE Trans. on Computers. - 1999. - Vol. 48, № 8. -P. 860-863.

Fields E. L. and Soderstrand M. A., "Performance characteristics of digital ladder networks," in *Proc. 1980 IEEE Intern. Symp. Circuits Syst.,* Houston, TX, pp. 1121-1124, Apr. 1980.

Fields E. L., "Applications of residue number system arithmetic to infinite impulse response digital filters," Ph.D dissertation, Dept. of Electrical and Computer Engineering University of California, Davis, CA, 1980.

Finko 0. A. Check and Reconfiguration of Analog-to-Digital Devices Operating in the System of Residual Classes // Engineering Simulation. — 2001. - Vol. 18. - P. 631-543.
Finko 0. A. Concordant Redundant Positional Notations // Engineering Simulation. - 1997. - Vol. 14. - P. 827-832.
Finko 0. A. Methods of problem-oriented representation and data processing in resources of the hardware support of intellectual systems // IEEE Conf. Artificial Intelligence Syst. (AIS'02). Gelendzhik, Russia, September 5-10, 2002. Proceedings. — Los Alamitos, California: IEEE Computer Soc. Press, 2002. - P. 453-454.
Finko 0. A. Number Restoration in the System of Residual Classes With a Minimum Number of Radices // Engineering Simulation. - 1999. -Vol. 16. - P. 329-334.
Finko O.A. Algorithms and Devices for N-ary Finite Ring Computations. /// Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 552-560.
Finko O.A. Introduction to New Parallel Computer Arithmetics Grounded on Factorisations of Operands'// Proc. International Congress "MATHEMATICS in XXI century. The role of the MMD of NSU in science, education, and business." 25-28 June 2003, Novosibirsk, Akademgorodok. http://www.sbras.ru/ws/MMF-21/.
Finko O.A. Methods of problem-oriented representation and data processing in resources of the hardware support of intellectual systems// IEEE Conf. Artificial Intelligence Syst. (AIS'02). Proceedings. — Los Alamitos, California: IEEE Computer Soc. Press, 2002. — P. 453–454. ISBN 5–94052–031–6.
Fouse S. D., Nudd G. R., Thorne-Booth G. M., Nygaard P. A. and Gichard F. D., "Residue-based image processor for very large scale integration (VLSI) implementation," *Proc. Intern. Society for Optics* (England), vol. 281, pp. 346-356, Apr. 1981.
Fouse S. D., Nudd G. R., Thorne-Booth G. M., Nygaard P. A. and Gichard F. D., "Image understanding of 1U algorithms," Tech. Rep. USCIPI 1010, Ch. 2, Hughes Research Laboratory, Malibu, CA, 1982.
Fraenkel A. S., "The use of index calculus and Mersenne primes for the design of a high speed digital multiplier," J. ACM, vol. 8, no. 1, pp. 87-96, Jan. 1961.
Fraser D. F. and Bryg N. J., "An adaptive digital signal processor based on the residue number system," in *Proc. AIAA 2nd Computers in Aerospace Conf.*, Oct. 1979.
Fraser D. F., "High speed adaptive digital filtering—Based on the residue number system," in *Proc. 26th Midwest Symp. Circuits and Systems.* Puebia Mexico, pp. 398-402, Aug. 1983.
Gajski D.D. Pat. 4187549 USA Modular modulo 3 module / Filed 17.11.77.
Games R. A., "Complex approximations using algebraic integers," *IEEE Trans.*

Inform. Theory, vol. IT-31, no. 5, pp. 565-579, Sept. 1985.
Gardner W. A. and Soderstrand M. A., "Design and implementation of multi-input adaptive signal extractors," Tech. Rep. SIPL 80-11, University of California, Signal and Image Processing Laboratory, Davis, CA, July 1980.
Gardner W. A. and Soderstrand M. A., "Design and implementation of multi-input adaptive signal extractors—-Vol. II," Tech. Rep. SIPL 82-12, University of California, Signal and Image Processing Laboratory, Davis, CA, Sept. 1982.
Garner H. L. Namber systems and arithmetic // Advances in Computers. - 1965. - Vol. 6. - P. 131-194.
Garner H. L. The residue number system // Ire transactions on electronic computers. - 1959. - Vol. 8, № 6. - P. 140-147.
Garner H. L., Error checking and the structure of binary addition, Ph. D. Diss., Univ. Michigan, Ann Arbor, Mich., 1958. 0 a r n e r H. L., The residue number system, IRE Trans. on Electronic Computers, EC-8 (1959), June, 140—147.
Gibson D. J. and Leon B. J., "An investigation into the use of residue number systems in digital filters," Tech. Rep. TREE 77-37, School of Electrical Engineering, Purdue University, West Lafayette, IN, Dec. 1977.
Goldereich 0., Ron D., Sudan M. Chinese remaindering with errors // IEEE Trans. On Information Theory. — 1999. — Preliminary version appered in proc. of 31st STOC 1999. - P. 225-234.
Good I. J. The relationship between two fast Fourier transforms // IEEE Trans. on Computers. — 1971. - № 3.
Goutzoulis A.P., Malarkey E.C., Davies D.K. et al.Optical processing with residue LED/LD lookup tables/ // Applied optic. 1988. Vol. 27, N9. P. 1674 – 1681.
Gregory R. T. and Matula D. W., "Base conversion in residue number systems," in *Proc. IEEE 3rd Symp. Computer Arithmetic.* Dallas, TX, pp. 117-125, Nov. 1975.
Gregory R. T., "Residue arithmetic with rational operands," in *Proc. 5th Symp. Computer Arithmetic,* pp. 144-145, May 1981.
Grosswald E., *Topics from the Theory of Numbers.* New York: McMillan, 1966.
Heidtmann K. D. Arithmetic Spectrum Applied to Fault Detection for Combinational Networks // IEEE Trails. Comput. 1991. Vol. 40, № 3. -P. 320 324.
Heindel L. E. and Horowitz E., "On decreasing the computing time for modular arithmetic," *Conf. Record 12th Symp. Switching and Automata Theory,* pp. 126-128, Oct. 1971.
Hellman M. E. The mathematics of public key cryptography. *Scientific American,* 241:146–157, February 1979.
Henderson D. S., Residue class error checking codes, Paper Presented at the 16th National ACM Conference, Los Angeles. September 5-8 1961
Herstein I. N., *Topics in Algebra.* Waltham, MA: Blaidsell Publishing Co.,

1964.
Hiasat A. A., Abdel-Aty-Zohdy S. H. Residue-to-binary arithmetic converter for the moduli set (2fe; 2^ 1; 2*[1]-[1] - 1) // IEEE Trails, on Circuits and Systems II: Analog and Digital Signal Processing. — 1998. Vol. 45, № 2. - P 204 209.
Hoffmann W., Muller H.E. Die quasi-logarithmischen Eigen-[sc]^^ Restklassensystems, Z. angew. Math. Mech, 40, Sonderheft OAMM-Tagung, Freiberg/Sa (1960), SS. T61 — T64
Howeil J. A. S., "On the reduction of a matrix to Frobenius form using residue arithmetic," Ph.D. dissertation, 1972.
Howell J. A. and Gregory R. T., "An algorithm for solving linear algebraic equations using residue arithmetic—Part I," *Bit* (Sweden), vol. 4, no. 3. pp. 200-224, 1969.
Howell J. A., "Exact solution of linear equations using in residue arithmetic," *Communications ACM,* vol. 14, no. 3, pp. 180-184, Mar. 1971.
Huang 0. H. and Taylor F. J., "A memory compression scheme for modular arithmetic," *IEEE Trans. Acous.^ Speech, and Signal Processing,* vol. ASSP-27, pp. 608-611, Dec. 1979.
Huang A., Tsunoda Y., Goodman J. W., and Ishihara S., "Optical computation using residue arithmetic," *Applied Optics,* vol. 18, no. 2, pp. 149-162, Jan. 1979.
Huang C. H. and Taylor F. J., "High speed DFTs using residue numbers," in *Proc. 1980 IEEE Intern. Conf. Acous., Speech, and Signal Processing.* Denver, CO, Apr. 1980.
Huang C. H., "A very fast residue to mixed-radix conversion algorithm for residue number applications," in *Proc. 16th Asilomar Conf. Circuits, Systems, and Computers,* Pacific Grove, CA, pp. 207-211, Nov. 1982.
Huang C. H., "High-speed two-dimensional filtering using residue arithmetic," in *Proc. 24th SPIE Intern. Tech. Symp.,* San Diego, CA, July 1980.
Huang C. H., "Large moduli residue number systems for very highspeed computing," in *Proc. Symp. Very High Speed Computing Technology,* Atlanta, GA, pp. IV.21-IV.31, Sept. 1980.
Huang C. H., Peterson D. G., Rauch H. E., Teague J. W. and Fraser D. F., "Implementation of a fast digital processor using residue number arithmetic," *IEEE Trans. Circuits Syst.,* vol. CAS-28, no. 1, pp. 32-38, Jan. 1981.
Huang H. and Taylor F. J., "A new technique for WFTA input/output reordering," *Intern. J. Computing and Inform. Sci.,* vol. 10, no. 1, pp. 27-37, Feb. 1981.
Ibrahim Khalid M., Saloum Saiat N. An amcient residue-to-binary converter design // At the same place. - 1988. — Cas.-35, № 9. -P. 1156-1158.
Inyutin S.A. Parallel Square Modular Computer Algebra // Lecture Notes in Computer Science: Parallel Processing and Applied Mathematics (PPAM). – German -Poland: Springer, 2004, -LNCS № 3019. –p. 993-997.
J.M., DaylandS.S., Gorgui-Naguib, Hinton O.R. Fault tolerant arithmetic unit

using duplication and residue codes // Integration, The VLSI Jurnal. - 1995. - Vol. 18, № 2-3. - P. 187-200.
Jam J., Bitner J., Pussell D. S., Abraham J. A. Probabilistic Verification of Boolean Functions // Formal Methods in System Design. Kluwer Academic Publishers. — 1992. — Vol. 1. - P. 61-115.
Jenkins W. K. and Etzel M. H., "Special properties of complement codes for redundant residue number systems," P/-OC. *IEEE,* vol. 69, no. 1, pp. 660-661 (correction vol. 69, no. 8, p. 1086, Aug. 1981), Jan. 1981. C.
Jenkins W. K. and Krogmeier J. J., "Error detection and correction in quadratic residue number systems," in *Proc. 26th Midwest Symp. Circuits and Systems,* Puebia Mexico, pp. 408-411, Aug. 1983.
Jenkins W. K. and Krogmeier J. V., "Complex digital filtering in quadratic modular number systems," in *Proc.1986 Intern. Symp. Circuits and Systems.* San Jose, CA, May 1986.
Jenkins W. K. and Lee C. F., "Complex residue number arithmetic for digital signal processing," in *Proc. 14th Asilomar Conf. Circuits Syst. and Computers,* Pacific Grove, CA, pp. 480-483, Nov. 1980.
Jenkins W. K. and Leon B. J., "A cost and performance evaluation of residue coded finite impulse response digital filters," in *Proc. 10th Asilomar Conf. Circuits Systems and Computers,* Pacific Grove, CA, Nov. 1976.
Jenkins W. K. and Leon B. J., "The use of residue coding in the design of hardware for nonrecursive digital filters," in *Proc. 8th Asilomar Conf. Circuits Systems and Computers,* Pacific Grove, CA, pp. 458-462, Dec. 1974.
Jenkins W. K. and Leon B. J., "The use of residue number systems in the design of finite impulse response digital filters," *IEEE Trans. Circuits Syst.,* vol. CAS-24, no. 4, pp. 191-201, Apr. 1977.
Jenkins W. K., "A highly efficient residue combinatorial architecture for digital filters," *Proc. IEEE,* vol. 66, no. 6, pp. 700-702, June 1978.
Jenkins W. K., "A new algorithm for scaling in residue number systems with applications to recursive digital filtering," in *Proc. 1977 IEEE Intern. Symp. Circuits Syst.*. pp. 56-59, Phoenix, AZ, Apr. 1977.
Jenkins W. K., "A new approach to the design of an error checker for failure resistant residue number digital filters," in *Proc. 1982 IEEE Intern. Symp. Circuits and Systems,* Rome, Italy, pp. 369-372, May 1982.
Jenkins W. K., "A standard computational element for the VLSI realization of digital processors using modular arithmetic," in *Proc. 16th Asilomar Conf. Circuits, Systems, and Computers,* Pacific Grove, CA, pp. 202-206, Nov. 1982.
Jenkins W. K., "A technique for high-precision digital filtering with multiple 8-bit microprocessors," in *Proc. 20th Midwest Symp. Circuits Syst.,* Lubbock, TX, pp. 58-62, Aug. 1977.
Jenkins W. K., "A technique for the efficient generation of projections for error correcting residue codes," *IEEE Trans. Circuits Syst.,* Feb. 1984.
Jenkins W. K., "Complex residue number arithmetic for high speed signal

processing," *Electron. Lett.*. vol. 16, no. 17,pp. 660-661, Aug.1980.
Jenkins W. K., "Composite number theoretic transforms for digital filtering," in *Proc. 9th Asilomar Conf. Circuits Systems and Computers,* pp. 458-462, Nov. 1975.
Jenkins W. K., "Hardware architectures for complex digital signal processors based on quadratic number codes," in *Proc. 1985 Intern. Symp. Circuit and Systems,* Kyoto, Japan, pp. 755-758,June 1985.
Jenkins W. K., "On the use of expended projection in residue number system error checking," in *Proc. 1 983 IEEE Intern. Symp. Circuits and Systems,* Newport Beach, CA pp. 698-700, May 1983.
Jenkins W. K., "Overflow detection in self-checking residue number digital processors," in *Proc. 1982 Intern. Conf. Circuits and Computers.* New York, NY, pp. 579-582, Sept. 1982.
Jenkins W. K., "Quadratic number codes for complex | signal processing," in *Proc. 1984 IEEE Intern. Symp.* y *Circuits and Systems,* Montreal, Canada, pp. 264-267, ^ May 1984.
Jenkins W. K., "Recent advances in residue number techniques for recursive digital filtering," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-27, no. 1, pp. 19-30, Feb. 1979.
Jenkins W. K., "Residue number system error checking using expanded projection," *Electron. Lett.,* vol. 18, no. 21,pp. 927-928, Oct. 1982.
Jenkins W. K., "Techniques for residue-to-analog conversion for high data rate digital filtering," in *Proc. 1978 IEEE Intern. Conf. Acous., Speech, and Signal Processing,* Tulsa, OK, pp. 804-807, Apr. 1978.
Jenkins W.K. The design of error checkers for self-checking residue number arithmetic // IEEE Trans. On Computers. — 1983. — C-32, № 4. — P. 388-396.
Jenkins W.K., Lao S. F. The design of an integrated RNS digital filter module based on serial-by-modulus arithmetic // IEEE Int. Cont. Comput. Design: VLSI Comput. and Process Rye Brook. NY. — 1987. -P. 634-637.
Johnson T. G., Soderstrand M. A. and Clark G. A., "Techniques for realization of high-speed recursive digital filters using residue number system arithmetic," in *Proc. 1986 IEEE Intern. Conf. ASSP,* Tokyo, pp. 2623-2626, Apr. 1986.
Juliien G. A. and Bayoumi A., "RNS modules for VLSI implementation of digital filters," in *Proc. 26th Midwest Symp. Circuits and Systems,* Puebia Mexico, pp. 403-407, Aug. 1983.
Jullien G. A. and Miller W. C., "A hardware realization of an NTT convolver using ROM arrays," in *Proc. 1980 IEEE Intern. Conf. Acous., Speech, and Signal Processing,* Denver, CO/ Apr. 1980.
Jullien G. A. and Miller W. C., "Applications of the residue number system to computer processing of digital signals," in *Proc. 4th IEEE Symp. Computer Arithmetic,* pp. 2ZO-225, Oct. 1978.
Jullien G. A. and Milter W. C., "A two-dimensional finite field processor for

image processing," in *Proc. 1981 IEEE Intern Conf. Acous., Speech, and Signal Processing,* Mar. 1981.
Jullien G. A., "Implementation of digitalsignal processing algorithms using parallel microcomputer arrays," in *Proc. 1979 Intern. Conf. Micro and Mini Computers.* Houston TX, Nov. 1979.
Jullien G. A., "Implementation of multiplication modulo a prime number with applications to number theoretic transforms," *IEEE Trans. Comput.,* vol. C-29, no. 10, pp. 899-905, Oct. 1980.
Jullien G. A., "Residue number scaling and other operations using ROM arrays," *IEEE Trans. Comput.,* vol. C-27, no. 4, pp. 325-336, Apr. 1978.
Jullien G. A., Jamali M. M. and Miller W. C., "Implementation of a spectrum analyzer using a memory intensive architecture," in *Proc. Canadian Communications and Energy Conf.,* Montreal, Canada, Oct. 1982.
Jullien G. A., Jamali M. M. and Miller W. C., "The viability of RNS based designs in digital signal processing hardware—A case study," in *Proc. 25th Midwest Symp. Circuits and Systems,* Houghton, Ml, pp. 166-170, Aug. 1982.
Jullien G. A., Miller W. C., Soltis J. J., Baraniecka A., and Tseng B. D., "Hardware realization of digital signal processing elements using the residue number system," in *Proc. 1 977 IEEE Intern. Conf. Acous., Speech, and Signal Processing,* April 1977.
Kameyama M. and Higuchi T., "A new scaling algorithm in symmetric residue number system based on multiple value logic," in *Proc. IEEE Intern. Symp. Circuits Syst.,* Tokyo, Japan, pp. 189-192,July 1979.
Karst E., Faktorenzerlegung Mersennescher Zahlen mittels programmgesteuerter Rechengerate, Numer. Math., 3, 1 (1961) 79-86
Katti S. A new residue arithmetic error correction scheme // IEEE Trans. On Computers. - 1996. - Vol. 45, № 1. - P. 13-19.
Kaushik S. and Arora R. K., "Sign detection in the symmetric residue number system," in *Proc. 5th Symp. Computer Arithmetic,* pp. 146-151, May 1981.
Kawamura S., KoAe M., Sano F. and Shimbo A. Cox-Rower Architecture for Fast Parallel Montgomery Multiplication // Proceedings EUROCRYPT 2000, LNCS 1807. - Springer Verlag, 2000. - P. 523-538.
Keir Y. A., Cheney P. W. and Tannenbaum M., "Division and overflow detection in residue number system," *IRE Trans. Electron. Comput.,* vol. EC-11, pp. 501-507, Aug. 1962.
Kim D. H. and Kirn J. K., "A theoretical consideration of complex processor using RNS," J. *Korea /nst. Electron. Engr.*. vol. 20, no. 6, pp. 69-74, Nov. 1983 (in Korean).
Kinoshita E., Kosako H., and Kojima Y., "General division in the symmetric residue number system," *IEEE Trans. Comput.*. vol. C-22, pp. 134-142, Feb. 1973.
Kinoshita E., Kosako H., and Kojima Y., "Logical implementation of arithmetic operations in the symmetric residue number system," Bulletin, University

of Osaka (Japan), vol. 22.no. 2,pp. 139-149,1973.
Kinoshita E., Kosako H., and Kojima YU., "A method of residue division," *Systems, Computers and Control,* vol. 1, no. 4, pp. 18-25, July 1970.
Kirn Moon Soo, Tomalechi Nolvhiro. Fault tolerant digital filters using pulse-train residue arithmetic circuits // Trans. hist. Electron. Inform. and Commun. Eng., 1987. - E 70, № 10. - P. 1009-1017.
Knowles J. B. and Olcayto E. M., "Coefficient accuracy and digital filter response," *IEEE Trans. Circuit Theory,* vol. CT-15,no.1,pp. 31-41, Mar. 1968.
Knuth D. E. *The art of computer programming,* volume 2. Addison-Wesley, 1969.
Kogge P. M., *The Architecture of Pipelined Computers.* New York: McGraw-Hill, 1981.
Kolyada, E. Otlivanchik, V. Revinsky, L. Vasilevitch // 5th Work-shop on DIP'94: Image Processing and Computer Optics. Samara, Russia, Aug. 22 – 26, 1994. Proc. SPIE. Vol. 2363. Washington, D.C., 1994. P. 147 – 151.
Koyano H., Saito T., and Hoshiko Y., "A digital convolver using a number theoretic transform," *Trans. Instrumentation, Electronics, and Communications Engineering* (Japan), vol. E-62, no. 6, pp. 446-447, June 1979.
Krishna K., Sun J. On theori and fast algoritms for error correction in residue number system product codes // IEEE Trans. On Computers. — 1993. - Vol. 42, July. - P. 840-852.
Krishnan R., Jullien G. A. and Miller W. C., "Complex , digital signal processing using quadratic residue number \ systems," *IEEE Trans. Acous., Speech^ and Signal Process- \ ing,* vol. ASSP-34, no. 1, pp. 166-186, Feb. 1986.
Krishnan R., Jullien G. A. and Miller W. C., "Computation of complex number theoretic transforms using quadratic residue number systems," in *Proc. 1986 IEEE Intern. Conf. ASSP,* Tokyo, pp. 233-236, Apr. 1986.
Kung H. T., Ruane L. M. and Yen D. W. L., *A Two Level Pipelined Systolic Array for Convolutions.* Computer Science Press, 1981.
Leon B. J., "Residue algebras and number theoretic transforms," in *Digital Signal Processing,* J. K. Aggarwal, Ed. Western Periodicals, 1979.
Liu Y. C., "Byte error correction in memory and arithmetic units," Ph.D. dissertation, 1970.
Mandelbamn D. M. Error correction in residue arithmetic // IEEE Trans. Comput. 1972. - Vol. 21, № 6. - P. 538-545.
Mandelbaum D. M. Further results on decoding arithmetic residue codes // IEEE Trans. On Information Theory. — 1978. - № 24. — P. 643-644.
Mandelbaum D. M. On a class of arithmetic codes and a decoding algorithm // IEEE Trans. On Information Theory. - 1976. - № 21. --P. 85-88.
Martic S. C. P. and Stanier B. J., "Microprocessor implementation of number theoretic transforms," *IEE' J. Electoric Circuits and Systems,* vol. 3, no. 1, pp. 21-26, Jan. 1979.
McClellan J. H. and Rader C. M., *Number Theory in Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1979.

McCoskey M., "Hybrid arithmetic," presented at Proc. 1985 Asilomar Conf. Circuits Systems and Computers, Pacific Grove, CA, Nov. 1985.
McCoskey M., "Prime or relatively prime radix data processing system," United States Patent 4,458.327, July 3/ 1984.
McLean M. A., A s p i n a 11 D., Decimal adder using a stored addition table, Proc. Inst. Electr. Engrs, 105B, 20 (1958), 129—135 144—146
Merkle R., Hellman M. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, IT-24:525–530, September 1978.
Merrill R. D., Jr., "Improving digital computer performance using residue number theory," *IEEE Trans. Electron. Corn-put.*, pp. 93-101, Apr. 1964.
Miller D. D., "Averaging modular number for phase measurement applications," in *Proc. 16th Asilomar Conf. Circuits, Systems, and Computers,* Pacific Grove, CA, pp. 465-468, Nov. 1982.
Miller D. D., Polky J. N. and King J. R., "A survey of Soviet developments in residue number theory applied to digital filtering," in *Proc. 26th Midwest Symp. Circuits and Systems,* Puebia Mexico, pp. 389-393, Aug. 1983.
Miller W. C. and Jullien G. A., "A ROM oriented 2-D digital filter structure," in *Proc. 12th Pittsburgh Conf. Modeling and Simulation,* Apr. 1981.
Mirsalehi Mir M., Shmir Joseph, Caulfield H. John. Residue arithmetic processing utiliring optical Fredkin gate arrays // Appl. Opt. — 1987. -Vol. 26, № 19. - P. 3940-3946.
Mizumachi M. and Kamiya Y., "On decoding algebraic codes constructed of the basis of the Chinese remainder theorem," *Trans. Instrumentation, Electronics and Communications Engineering* (Japan), vol. E-62, no. **12,** pp. 890-891, Dec. 1979.
Murakami H. and Reed I. S., "Recursive realization of finite impulse filters using finite field arithmetic," *IEEE Trans. Inform. Theory,* vol. IT-23, pp. 232-242, Mar. 1977.
Nagpal H. K., Juliien G. A. and Miller W. C., "High-speed convolver for image processing—A hardware approach using number theoretic concepts," presented at Proc. 1983 OHMCON Conf., Detroit, Ml, June 1983.
Nagpal H. K., Jullien G. A. and Miller W. C., "Architectures of a 2-D FFT/NTT processor for real-time processing of images," in *Proc. 25th Midwest Symp. Circuits and Systems,* Houghton, Ml, pp. 171-174, Aug. 1982.
Nagpal H. K., Jullien G. A. and Miller W. C., "Processor architectures for two-dimensional convolvers using a single multiplexed processor element," *IEEE Trans. Compute* vol. C-32, no. 11,pp. 989-1000, Nov.1983.
Nussbaumer H. J., "Relative evaluation of various number theoretic transforms for digital filtering applications," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-26, pp. 88-93, Feb. 1978.
Nussbaumer H., "Digital filtering using cotransforms in finite fields," *Electron. Lett.*, vol. 12, no. 5, pp. 113-114 Mar. 1976.
Oarner H. L., The residue number system, IRE Trans. on Electronic Computers, EC-8 (1959), June, 140—147.

O'Keefe K. H. and Wright J. L., "Remarks on base extension for modular arithmetic," *IEEE Trans. Comput.*, vol. C-22, pp. 134-142, Sept. 1973.
O'Keefe K. H., "A digital signal processor which uses the residue number system," in *Proc. 1 971 IEEE Intern. Mexico Conf. Systems, Networks, and Computers*, Oaxtepec, Mexico, Jan. 1 971.
O'Keefe K. H., "A note on fast base extension for residue number systems with three moduli," *IEEE Trans. Comput* voI.C-24.pp. 1132-1133, Nov. 1975.
Orion G.A., Peppard L.E., Tavares S.E. New fault tolerant techniques for residue number systems // IEEE Trans. On Computers. - 1992. — Vol. 41, № 11. - P. 1453 1464.
Ovcharenko L.A., Bolkunov A.A. Comparison of Computational Structures in Positional and Nonpositional Number System// Engineering Simulation. – 2001. – vol. 18. – P.423-427.
Papachristou C.A. Associative table look up processing for multioperand residue arithmetic // J. Assoc. Comput. 1987. Vol. 34, N 2. P. 376 – 396.
Paul D. F., Jenkins W. K. and Davidson E. S., "Residue arithmetic for real-time applications: High throughput and reliability using customized models," in *Proc. 1984 Intern. Conf. Circuits and Computers.* Rye, NY, pp. 689-694, Oct. 1 984.
Paulier P. Low-cost double-size modular exponentiation or how to stretch your cryptoprocessor. la H. Imai and Y. Zheng, editors, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC'99, LNCS-1560, - Springer Veriag, 1999. -P. 223-234.
Pjaenkel A. S., The use of index calculus and Mersenne primes for the design of a high-speed digital multiplier, У. ACM, 8 1 (1961) 87-96
Plessmann K. A parallel highly modular object-oriented computer architecture // 10 юбил. Международн. Симп. по пробл. модулярных инф.- выч. сист. и сетей. – Санкт-Петербург, Россия, 13–18 сент., 1993. – Пленар. докл. – М., 1996. – С.97–109.
Plessmann K., Wollert J. and others. A modular multi-PC system for real-time applications // 10 юбил. Международн. Симп. по пробл. модулярных инф.- выч. сист. и сетей. – Санкт-Петербург, Россия, 13–18 сент., 1993. – Пленар. докл. – М., 1996. – С. 110–119.
Polky J. N. and Miller D. D., "Architecture of the residue \ arithmetic adaptive filter processor," in *Proc. 16th Asilomar Conf. Circuits, Systems, and Computers*, Pacific Grove, CA, pp. 172-173, Nov. 1982.
Pollard J. M., "The fast Fourier transform in a finite field," *Mathematics of Computation*, vol. 25, Apr. 1971.
Posch K. C., Posch R. Residue number systems, a key to parallelism in public key cryptography // Proc. of Fourth IEEE Symp. on Parallel and Distributed Processing. — 1992. — P. 432^35.
Posch K. C., Posch R. RNS-Modulo Reduction in Residue Number Systems // IEEE Trans. on Parallel and Distributed Systems. - 1995. — Vol. 6, № 5. - P. 449--454.

Psaltis D. and Casasent D., "Optical residue arithmetic: A correlation approach," *Applied Optics*, vol. 18, no. 2, pp. 163-171,Jan. 1979.
Rader C. M., "Discrete convolution via Mersenne transforms," *IEEE Trans. Comput.*, vol. C-21, no. 12, pp. 1269-1273, Dec. 1972.
Ralston A., Ed., *Encyclopedia of Computer Science*. New York: Van Nostrand Reinhold, 1983.
Ramachandran V., "Single residue error correction in residue number systems," *IEEE Trans. Comput.*, vol. C-32, pp. 504-507, May 1983.
Ramnarayan A. S. and Taylor F. J., "On the structure of IIR filters using residue arithmetic," in *Proc. 1981 IEEE Intern. Conf. Acous., Speech, and Signal Processing*, pp. 251-254, Apr. 1981.
Ramnarayan A. S., "Practical realization of Mod p, p prime multiplier," *Electron. Lett.*, vol. 19, no. 15, pp. 466-467, June 1980.
Rao R. N. and Trehan A. K., "Binary logic for residue arithmetic using magnitude index," *IEEE Trans. Comput.*, vol. C-19, no. 8, pp. 752-757, Aug. 1970.
Rao T. R. N., "Arithmetic of finite fields," in *Proc. 5th Symp. Computer Arithmetic*, pp. 2-5, May 1981.
Reddy N. S. and Reddy V. U., "Convolution algorithms for small-word-length digital filtering applications," *IEE J. Electronic Circuits Syst.*, vol. 3, no. 6, pp. 253-256, Nov. 1979.
Reddy N. S. and Reddy V. U., "Implementation of Winograd's algorithm in modular arithmetic for digital convolutions," *Electron. Lett*, vol. 14, no. 7, pp. 228-229, Mar. 1978.
Redinbo G. R., "Fault tolerant digital filtering structures for wafer scale VLSI," in *Proc. 1 986 IEEE Intern. Conf. ASSP*, Tokyo, pp. 1189-1192, Apr. 1986.
Reed I. S. and Truong T. K., "Complex integer convolutions over a direct sum of Galois fields," *IEEE Trans. Inform. Theory*, vol. IT-21, no. 6, pp. 657-661, Nov. 1975.
Rivest R. L., Shamir A., Adleman L. *On Digital Signatures and Public Key Cryptosystems*, pages 120–125. Technical Report, MIT/LCS/TR-212, January 1979.
Rivest R. L., Shamir A., Adleman L. Some options in the design of a residue arithmetic. *Commununications of ACM*, 21(2):120–126, April 1978.
Sasaki A., "Addition and subtraction in the residue number system," *IEEE Trans. Comput.*. vol. EC-16, pp. 157-164, Apr. 1967.
Sasaki A., "The basis for implementation of additive operations in the residue number system," *IEEE Trans. Comput.*, vol. C-17, no. 11, pp. 1066-1073, Nov. 1968.
Schonhage A. and Strassen V., "Schnetle multiplikation grosser zahlen," *Computing* (Germany), vol. 7, pp. 281-292, 1971.
Shah A. R., Sid-Ahmed M. A., Jullien G. A., "A proposed hardware structure for two-dimensional recursive digital filters using the residue number system," *IEEE Trans. Circuits Syst.*, vol. CAS-32, no. 3, pp. 285-288, Mar. 1985.

Shamir A. *A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem*, volume IT-30, pages 699–704. IEEE Trans. Inform. Theory, September 1984.

Shamir A. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

Shiozaki A., Nishida F., and Tanaka H., "An error-correction method using the residue number system," *Electronics and Communications* (Japan), vol. 58, no. 9, pp. 11-19, Sept. 1975.

Shubs Y. V., "Feasibility of utilizing a system of residual classes in signal processing equipment," */zv. Vuz Ra-dioelektron* (USSR), vol. 23, no. 1, pp. 75-76, Sept. 1975.

Shu-Hung Leung, "Application of residue number system to complex digital filters," in *Proc. 15th Asilomar Conf. Circuits Syst. and Computers*, Pacific Grove, CA, pp. 70-74, Nov. 1981.

Silverman R.D. Parallel polynomial arithmetic over finite ring // J. Parallel. and Distribut. Comput. – 1990. – Vol. 10, N 3. – P. 265–270.

Skavantzos A., Taylor F.J. On the polynomial residue number system // IEEE Trans. Signal Process. – 1991. – Vol. 39, N 2. – P. 376–382.

Slotnick D. L., "Modular arithmetic computing techniques," Tech. Rep. ASD-TDR-63-280, Westinghouse Electric Corp., Air Arm Division, Baltimore, MD, Feb. 1 963.

Smith J. C. Taylor F. J. A fault tolerant GEQRNS processing element for linear systolic array DSP applications // IEEE Trans. On Computers. — 1995. - Vol. 44, № 9. - P. 1121-1130.

Smith W., "SWIFT," in *Proc. Symp. Very High Speed Computing Technology.* Atlanta, GA, pp. VII.13-VIII.21, Sept. 1980.

Soderstrand M. A. and Chang Bill, "Design of a high-performance FIR digital filter on a CMOS semi-custom VLSI chip," in *Proc. 1986 ISMM Intern. Conf. on Mini and Micro Computers* (MIMl'86), Beverly Hills, CA, Feb. 1986.

Soderstrand M. A. and Escott R. A., "A high-speed digital adaptive filter using the residue number system, in *Proc. 1 983 IEEE Intern. Symp. Circuits and Systems*, Newport Beach, CA, pp. 705-708, May 1983.

Soderstrand M. A. and Escott R. A., "VLSI implementation in multiple-valued logic of an FIR digital filter using residue number system arithmetic," *IEEE Trans. Circuits and Systems*, vol. CAS-31, no. 1, Jan. 1986.

Soderstrand M. A. and Fields E. L., "Digital filters with 10-100 MHz through-put using bipolar PROM's and residue number arithmetic," in *Proc. 1978 IEEE Intern. Symp. Circuits Syst.*, New York, NY, May 1978.

Soderstrand M. A. and Fields E. L., "Multipliers for residue number arithmetic digital fitters," *Electron. Lett.*, vol. 13, no. 6, pp. 164-166, Mar. 1977.

Soderstrand M. A. and Fields E. L., "Tenth order digital filter using residue number arithmetic," in *Proc. 20th Midwest Symp. Circuits Syst.*, Lubbock, TX, pp. 899-903, Aug. 1977.

Soderstrand M. A. and Johnson T. G., "A microprocessobased complex multiplier/accumulator using the quadratic complex residue number system," *Intern. J. Mini and Micro Computers.* Sept. 1985.

Soderstrand M. A. and Johnson T. G., "Hardware implementations of a moving discrete Fourier transform," in *Proc. ISMM 7984 Conf. on Mini and Micro Computers (MlMl[f]84).* Ceasar's Palace, Las Vegas, NV, pp. 151-153, Dec. 1984.

Soderstrand M. A. and Johnson T. G., "Use of generalized quadratic RNS arithmetic in adaptive filter," in *Proc. 1985 Asilomar Conf. Circuits Systems and Computers,* Pacific Grove, CA, Nov. 1985.

Soderstrand M. A. and Keliey J. K., "'Design of a low cost FIR filter with sampling rate in excess of 10 MHz," in *Proc. IEEE Intern. Symp. Circuits and Systems,* Chicago, IL, pp. 432-434, Apr. 1981.

Soderstrand M. A. and Poe G. D., "Application of quadratic-like complex residue number system arithmetic to ultrasonics," in *Proc. 1984 IEEE Intern. Conf. ASSP.* San Diego, CA/ Mar. 1984.

Soderstrand M. A. and Sinha B., "A pipelined recursive residue number system digital filter," *IEEE Trans. Circuits Syst.*. vol. CAS-31, no. 4, pp. 415-417, Apr. 1984.

Soderstrand M. A. and Vernia C., "A high-speed low-cost modulo multiplier with RNS arithmetic applications," *Proc. IEEE,* vol. 68, no. 4, pp. 529-532, Apr. 1980.

Soderstrand M. A. and Vernia C., "An improved RNS digital to analog converter," in *Proc. IEEE Intern. Symp. Circuits and Systems,* Chicago, IL, pp. 85-87, Apr. 1981.

Soderstrand M. A. and Vernia C., "General modulo Pi multiplier with RNS arithmetic applications," in *Proc. IEEE Intern. Symp. Circuits Syst.,* Houston, TX, pp. 375-378, Apr. 1980.

Soderstrand M. A. and Vernia C., "Microprocessor controlled development system for adaptive filtering using parallel processing and residue number arithmetic," in *Proc. ISMM Intern. Conf. Mini and Micro Computers.* Montreal, Canada, Sept. 1980.

Soderstrand M. A. and Vernia C., "Microprocessor controlled development system for adaptive filtering using parallel processing and residue number arithmetic," *intern. J. Mini and Micro Computers,* vol. 3, no. *3,* pp. 39-43, July 1981.

Soderstrand M. A. and Vernia C., Paulson D., and Vigil M. C., "Microprocessor controlled adaptive digital filter," in *Proc. IEEE Intern. Symp. Circuits Syst.,* Houston, TX, pp. 142-146, Apr. 1980.

Soderstrand M. A., "A high-speed low-cost recursive digital filter using residue number arithmetic," *Proc. IEEE,* vol. 65, no. 7, pp. 1065-1067, July 1977.

Soderstrand M. A., "A microprocessor-based complex multiplier accumulator using the quadratic complex residue number system," in *Proc. MIMI'83,* San Antonio, TX, pp. 126-128,Dec. 1983.

Soderstrand M. A., "A new hardware implementation of modular adders for residue number systems," in *Proc. 26th Midwest Symp. Circuits and Systems*, Puebia Mexico, pp. 412-415, Aug.1983.
Soderstrand M. A., "Adaptive recursive filters," in *Proc. 1979 Intern. Colloq. Circuits and Systems*, Taipai, Taiwan, July 1979.
Soderstrand M. A., "Applications of microprocessors in digital signal processing," in *Proc. 21st Midwest Symp. Circuits Syst.*, Ames, IA, pp. 153-157, Aug. 1978.
Soderstrand M. A., "Experimental results from a microprocessor based digital filter designed for brainwave monitoring," in *Proc. ISMM Intern. Symp. Mini and Micro Computers*, Cambridge, MA, pp. 95-99, July 1982.
Soderstrand M. A., "High speed data conversion using residue number arithmetic A/D and D/A converters," in *Proc. 22nd Midwest Symp. Circuits Syst.*, Philadelphia, PA, pp. 6-10, June 1979.
Soderstrand M. A., "High-speed digital filters using residue number arithmetic," in *Proc. 9th Asilomar Conf. Circuits, Systems, and Computers*, Pacific Grove, CA, pp. 416-420, Nov. 1975.
Soderstrand M. A., "New hardware for high speed adaptive digital filtering," in *Proc. 24th Midwest Symp. Circuits and Systems*, Albuqurque, NM, pp. 63-68, June 1981.
Soderstrand M. A., "Practical applications of the residue number system to digital filtering," Tech. Rep. SAND77-8718, Sandia Laboratories, Livermore, CA, 1977.
Soderstrand M. A., "Selected papers on multi-input adaptive signal extractors and RNS implementations—Vol. II," Tech. Rep. SIPL 82-13, University of California, Signal and Image Processing Laboratory, Davis, CA, Sept. 1982.
Soderstrand M. A., "Techniques for computer based digital signal processing," in *Proc. 1981 ISMM Intern. Symp. on Mini and Micro Computers in Controls and Measurement*, San Francisco, CA, pp. 87-89, May 1981.
Soderstrand M. A., Buteau J. J. and Kelley J. K., "Selected papers on multi-input adaptive signal extractors and residue number arithmetic digital filters," Tech. Rep. SIPL 80-12, University of California, Signal and Image Processing Laboratory, Davis, CA, July 1980.
Soderstrand M. A., Jenkins W. K., Jullien G. A. and Tailor F. J. Residue Number System Arithmetic: Modem Applications in Digital Signal Processing. - N. Y.: IEEE Press, 1986.
Soderstrand M. A., Johnson T. G. and Chopper K., "Use of quadratic and quadratic-like complex residue number systems in digital signal processing," in *Proc. 1985 IEEE Intern. Symp. Circuits and Systems*, Kyoto, Japan, pp. 1385-1388,June 1985.
Soderstrand M. A., Johnson T. G. and Clark G. A., "Hardware realizations of frequency-sampling adaptive filters," in *Proc. IEEE Intern. Symp. Circuits and Systems*, San Jose, CA, May 1986.
Soderstrand M. A., Johnson T. G., Chopper K. V. and Clark G. A., "Adaptive

filtering using a recursive complex number theoretic transform (CNTT) in a generalized quadratic residue number system (GQRNS)," in *Proc. 1 985 Midwest Symp. Circuits and Systems,* Louisville, KY, pp. 464-467, Aug. 1985.

Soderstrand M. A., Kelley J. K. and Buteau J. J., "A microprocessor-based RNS arithmetic totally adaptive digital filter for system identification," in *Proc. 14th Asilomar Conf. Circuits Systems and Computers,* Pacific Grove, CA, Nov. 1980.

Soderstrand M. A., Sinha B. and Chopper K., "Comparison of three new techniques for pipelining I1R digital filters," in *Proc. 1984 Asilomar Conf. on Circuits Systems and Computers,* Pacific Grove, CA, Nov. 1984.

Soderstrand M. A., Vernia C. and Chang J., "An improved residue number system digital to analog converter," *IEEE Trans. Circuits and Systems,* vol. CAS-28, no. 12, pp. 1164-1169, Dec. 1983.

Soderstrand M. A.. "A universal digital filter for use with 8-bit microprocessors," in *Proc. 20th Midwest Symp. Circuits Syst.,* Lubbock, TX, pp. 55-57, Aug. 1977.

Soderstrand M. A.and Sinha B., "New technique for highspeed pipelined IIP RNS digital filters," in *Proc. 1984 IEEE Intern. Symp. on Circuits and Systems,* Montreal, Canada, pp. 1455-1458, May 1984.

Stallings W. T. and BouIIion T. L., "Computation of pseudoinverse matrices using residue arithmetic," *SIAM Review,* vol. 14, no. 1, pp. 152-163, Jan. 1972.

Su C. C., Lo H. Y. An algorithm for scaling and single residue error correction iii the residue number systems // IEEE Trans. On Computers. - 1990. - Vol. 39, № 8. P. 1053-1064.

Svoboda A, Valach M.; Operatorove obvody, StroJe na Zpracovam Informacf, Sbornfk III, Nakl. CSAV, Praha, 1955, 247—295 »4.

Svoboda A. and Valach M., "Decimal arithmetic unit," *Stroje Na Zpracovani.* vol. 8, Nakl. CSAU, Praha, 1962.

Svoboda A. Valach and M., "Operational circuits," *Stroje Na Zpracovafni Informaci.* vol. 3, Nakl. CSAV, Praha, 1955. A. Svoboda, "Rational numerical system of residual classes," *Stroje Na Zpracovani Informaci,* vol. 5, pp. 9-37, Prague, 1957.

Svoboda A., "Computer progress in Czechoslovakia II— The numerical system of residual classes," in *Digital Information Processor,* W. Hoffman, Ed. New York: Wiley, 1962.

Svoboda A., "The numerical system of residual classes in mathematical machines," *Information Processing (Proc. of UNESCO Conf.* June 1959), pp. 419-422, 1960.

Svoboda A., Rational numerical system of residual classes, Stroje na Zpracovanf Informacf, Sbornik V, Praha, 1957.

Svoboda A., The numerical system of residual classes in mathematical machines, Information Processing (Proceedings of UNESCO Conference, June

1959), 1960, 419—422.
Svoboda A., The numerical system of residual classes in mathematical machines, Proc. Congreso Internacional de Automatica, Madrid, October 11—12, 1958.
Svoboda A., Use of the Korobov sequence in mathematical machines» Stroje na Zpracovam Informacf, Sbornik, Nakl. CSAV, Praha, 1954.
Swartzlander E. E. Fault tolerant arithmetic via time-shared TMR // Proc. of the SPIE, the international society for optical engineering. --1999. - Vol. 3807. - P. 84 92.
Szabo N. S. and Tanaka R. I., "Report on residue (modular) arithmetic survey," Tech. Rep., Lockheed Missiles and Space Co., Sunnyvale, CA, Dec. 1963.
Szabo N. S., Tanaka R. I. Residue arithmetic and its applications to computer technology. — N. Y.: McGraw-Hill, 1967.
Szabo N., Sign detection in nonredundant residue systems, IRE Trans. on Electronic Computers, EC-11, 4 (August, 1962), 494--500.
SzabAo N. S., "Sign detection in nonredundant residue systems," *IRE Trans. Elctron. Comput.*, vol. EC-11, pp. 494-500, Aug. 1962.
Tai A., Cindrich I., Fineup J. R. and Aleksoff C. C., "Optical residue arithmetic computer with programmable computation modules," *Applied Optics,* vol. 18. no. 6, pp. 2812-2823, Aug. 1979.
Tan C. I. and Mclnnis B. C., "Adaptive digital control implemented using residue number systems," in *Proc. 20th IEEE Conf. Decision and Control,* vol. 2, pp. 808-812, Dec. 1981.
Tan C. I. and Mclnnis B. C., "Adaptive digital control implemented using residue number systems," *IEEE Trans. Automatic Control,* vol. AC-27, no. 2, 1982.
Tanaka R. I. *Some options in the design of a residue arythmetic,* volume III. “Proc.Nat.El.Conf”, Chicago, 1963.
Tanaka R. I., "Modular arithmetic techniques," Tech. Rep. 2-38-62-1A, ASTDR, Lockheed Missiles and Space Co., Nov. 1962.
Taylor F. J. and Ramnarayanan A. S., "An efficient residue-to-decimal converter," *IEEE Trans. Circuits and Systems,* vol. CAS-28, no. 12, pp. 1164-1169, Dec. 1981.
Taylor F. J. and Huang C. H., "A comparison of DFT algorithms using a residue architecture," *Computer and Electrical Engineering* (England), vol. 8, no. 3, pp. 161-171, Sept. 1981.
Taylor F. J. and Huang C. H., "A floating-point residue s arithmetic unit," *J. Franklin Inst.,* vol. 311, pp. 33-53, Jan 1981.
Taylor F. J. and Huang C. H., "An autoscale residue multiplier," *IEEE Trans. Computer,* vol. C-31, no. 4, pp. 321-325, Apr. 1982.
Taylor F. J. and Wilmhoff R., "On hard errors in RNS architectures," *IEEE Trans. Comput.*. pp. 772-7"74, June 1983.
Taylor F. J., "A VLSI residue arithmetic multiplier," *IEEE Trans. Comput.*. vol. C-31, no. 6, pp. 540-546, June 1982.

Taylor F. J., "Large moduli multipliers for signat processing," *IEEE Trans. Circuits and Systems,* vol. CAS-28, no. *7,* July 1981.
Taylor F. J., "Large VLSI modulus multipliers," in *Proc. IEEE Intern. Symp. Circuits Syst.,* Houston, TX, pp. 379-383, Apr. 1980.
Taylor F. J., "Residue arithmetic: A tutorial with examples," *Computer,* vol. 17, no. 5, pp. 50-63, May 1984.
Taylor F. J., "The impact of residue arithmetic on digital signal processing," in *Proc. IASTED Intern. Symp.*. Paris, France, June 19-21, 1985.
Taylor F. J., *Digital Filter Design Handbook.* New York: Marcel-Dekker, 1983.
Taylor F. J., Papadourakis G., Skavantzos A. and Stouratis A., "A radix-4 FFT using complex RNS arithmetic," *IEEE Trans. Comput.*. vol. C-34, no. 6, pp. 573-576, June 1985.
Taylor F., "A new residue-to-decimal converter," presented at Proc. 26th Midwest Symp. Circuits and Systems, Puebia Mexico, Aug. 1983.
Thong T. and Liu B., "Limit cycles in the combinatorial implementation of digital filters," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-24, no. 3, pp. 248-256,June 1976.
Thornton M. A., Dreschler R., Miller D. M. Spectral Techniques in VLSI CAD. Kluwer Academic Publishers, 2002.
Tkachenko A. V., Finko 0. A. Transformation and Syntesis of Complex Structures of Excessive Calcules // Atuomation and Remote Control. — 1995. - Vol. 56, № 5. - P. 765-770.
Tomabechi N., "Residue arithmetic using ring counters and its error correcting circuit," *Trans. Instrumentation, Electronic, and Communication Engineering* (Japan), vol. E-64, no. 6, June 1981.
Tomabechi N., Kameyama M. and Higuchi T., "Efficient residue arithmetic circuit using multiple-valued ring counters and its application to digital signal processing," in *Proc. 12th Intern. Symp. Multiple-Valued Logic,* pp. 107-112,Feb. 1982.
Tomabechi N., Kameyama M. and Higuchi T., "Pulse rate arithmetic circuit based on residue number system and its application to digital filters," *Trans. Institute Electronic and Communications Engineers* (Japan), vol. E-65, no. 2, p. 137,Feb. 1982.
Tseng B. D., Jullien G. A., and Miller W. C., "An error analysis of an FFT implementation using the residue number system," in *Proc. 1 978 Intern. Conf. Acous., Speech, and Signal Processing,* Tulsa, OK, pp. 800-803, Apr. 1978.
Tseng B. D., Jullien G. A., and Miller W. C., "Implementation of FFT structures using the residue number system," *IEEE Trans. Comput.,* vol. C-28, no. 11, pp. 831-845, Nov. 1979.
Tseng B. D., Miller W. C. and Jullien G. A., "Analysis of quantization error in a ROM oriented FFT processor," in *Proc. 25th Midwest Symp. Circuits and Systems,* Houghton, Ml, pp. 6-8, Aug. 1982.
Uspensky J V., Heaslet M. A., Elementary number theory, N Y' McOraw-Hill,

1939; Chapt. VIII, par. 6: Indices, p. 237 ff.
Va1ach M., Prevod cfsel ze soiistavy zbytkovych trfd do polyadicke sou-stavy zmenou ^meritka periody, Stroje na Zpracov^m Informacf Sbornfk IV, Nakl. CSAV, Praha, 1956, 53—64.
Valach M. Vznik kodu a ciselne zbytkovych tzid Stroje na zpracovani informaci // Sbomik III. Nak. 1 CSAV. 1955.
Valach M., "Origin of the code and number system of remainder classes," *Stroje Na Zpracovani Informaci,* vol. 3, Nakl. CSAV, Prague, 1955.
Vanwormhoudt M. C., "Structural properties of complex residue rings applied to number theoretic Fourier transforms," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-26, pp. 99-104, Feb. 1978.
Vegh E. and Leibowitz L. M., "Fast convolution in finite rings," *IEEE Trans. Acous., Speech, and Signal Processing,* vol. ASSP-24, pp. 343-344, Aug. 1976.
Vyshynskyy V. A. and Petushchak V. D., "A method of multiplication in a residue class system," *Soviet Automat. Contr.*. vol. 5, no. 1, pp. 77-78, Jan. 1972.
Vyshynskyy V. A. and V. D. Petushchak, "Algorithms for determination of the reciprocal of a number in a residue class system," *Soviet Automatic Control,* vol. 6, no. 3, pp. 58-61, May 1973.
W. K. Jenkins, "The design or error checkers for self-checking residue number arithmetic," *IEEE Trans. Comput.,* vol. C-32, no. 4, pp. 388-396, Apr. 1983.
W. K. Jenkins, D. F. Paul, and E. S. Davidson, "A custom-designed integrated-circuit for the realization of residue number digital filters," in *Proc. 1985 Intern. Conf. ASSP*. Tampa, FL, pp. 220-223, Mar. 1985. 1985.
Waser S. and Flynn M. J., *Introduction to Arithmetic for Digital Systems.* New York: Holt, Rinehart and Winston, 1982, Ch. 2.
Watson R. W. and Hastings C. W., "Self-checked computation using residue arithmetic," *Proc. IEEE,* pp. 1920-1931, Dec. 1966.
Weinmann K. D. and Soderstrand M. A., "Influences of hardware implementation on a high-speed digital adaptive filter using the residue number system," in *Proc. 25th Amidwest Symp. Circuits and Systems,* Houghton, Ml pp 376-380, Aug.1982.
Weinmann K. D., Soderstrand M. A. and Shebani Soliman, "Evaluation of new hardware for high-speed digital adaptive filter using the residue number system," in *Proc. 16th Asilomar Conf. Circuits, Systems, and Computers,* Pacific Grove, CA, pp. 187-191, Nov. 1982.
Wheaton L. B. and Current C. W., "A quaternary threshold logic modulo-four multiplier circuit for residue number system nonrecursive digital filters," in *Proc. 11th Intern. Symp. Multiple-Valued Logic,* pp. 48-53, May 1981.
Wigley N. M., Jullien G. A. On modulus replication for residue arithmetic r-omputations of complex inner products. IEEE Trans. On Computers. – 1990. – Vol. 39, № 8. – P. 1065-1076.
Wigley N. M., Jullien G. A., Reaume D. Large Dynamic Range Computa-

tions over Small Finite Rings // IEEE Trans. On Computers. — 1994. – Vol. 43, № 1. – P. 78-86.
Wilmhoff R. and Taylor F. J., “On hard errors in RNS architecture,” in *Proc. IEEE Intern. Conf. Acoustics Speech and Signal Processing,* June 1983.
Winograd S., “On the time required to perform addition,” *J. ACM,* vol. 12, no. 2, pp. 277-285, 1965.
Winograd S., “On the time required to perform multiplication,” *J. ACM.* Vol. 14, no. 4, pp. 793-802, 1967.
Wolenty R. and Jenkins W. K., “An experimental hardware realization of a multiple microprocessor residue number digital filter,” in *Proc. 1980 IEEE Intern. Conf. Acous., Speech, and Signal Processing,* Denver, CO, pp. 1097-1100, Apr. 1980.
Wolenty R. G., “An experimental hardware realization of a multiple-microprocessor residue number digital filter,” Master of Science thesis, Dept. of Electrical Engineering, University of Illinois, Urbana-Champaign, IL, 1979.
Yanushkeoich S., Dziurzanski P., Shmerko V. Word-Level Models for Efficient Computation of Multiple-Valued Functions, Part 1: LAR // IEEE 32th Int’1 Symp. On Multiple-Valued Logic. Boston. USA. — 2002. –P. 202-208.
Yanushkevich S., Shmerko V., Dziurzanski P. Word Level Decision Diagrams Upon the Conditions of Linearity // IEEE Trans. Comput. Design of Integrated Syst. — 2001. – Vol. XX**,** month. — P. 1-40.
Yau S. S. and Chung J., “On the design of modulo arithmetic units based on cyclic groups,” *IEEE Trans. Comput.*. vol. C-25, no. 11, pp. 1057-1067, Nov. 1976.
Yau S. S. and Liu Y. C., “Error correction in redundant residue number systems,” *IEEE Trans. Comput.*. vol. C-22, pp. 5-11,Jan. 1973.
Yun’ushkevich S. Arithmetical Canonical Expansions of Boolean and MVL Functions as Generalized Reed-MuUer Series // Proc. IFIP WG 10.5 Workshop on Applications of the Reed-MuUer Expansions in Circuit Design. – Japan. – 1995. – P. 300-307.
Zhang D. Parallel VLSI neural system designs. // Springer, 1998. P. 257.
Zhang D., Jullien G. A., Miller W. C. A neural-like approach to finit ring computation // IEEE Trans. Circuits and Syst. — 1990. Vol. 37, № 8. –P. 1048-1052.
Zhang D., Jullien G. A., Miller W. C. Recursive reduction in finite ring computations // 23rd Asilomar Conf. Signals, Syst. And Comput., Pasific Grove, Calif., Oct. 30 Nov. 1, 1989: Conf. Rec., vol. 2, San Jose (Calif.). – 1989. – P. 854-857.
Zhang D., Jullien G. A., Miller W. C. VLSI implementation of neural-like networks for finite ring computations // Proc. 32nd Midwest Symp. Circuits and Syst., Champaign, III, Aug. 14-16, 1989. Vol 1, N. Y., 1990. – P. 485-488.
Zhang C. N., Shirazi B., Jim D., Y. Residue number conversion // Explor.

Technol.: Today and tomorrow. Fall Joint Comput. Conf., Dallas. Tex., 25-29 Oct., 1987. Washington: D. C., 1987. – P. 390-396.

——, "A BLSI implementation of an NTT multiplier using RNS for digital signal processing applications," in *Proc. 2nd European Signal Proc. Conf.,* Germany, pp. 837-841, Sept. 1983.

——, "A comparison of residue number multipliers and 2's-compliment multipliers implemented by stored multiplication tables," in *Proc. 1978 IEEE Intern. Symp. Circuits Syst.*. New York City, NY, pp. 297-301, May 1978.

——, "A modular implementation of digital signal processing architectures using RNS," in *Proc. 1984 IEEE Intern. Symp. Circuits and Syst.*. Montreal, Canada, pp. 1069-1072, May 1984.

——, "A modular implementation of digital signal processing architectures using RNS," in *Proc. 16th Ann. Pittsburgh Conf. on Modeling and Simulation,* Apr. 1985.

——, "A new technique using biased addition for generating projections in error-correcting residue codes," in *Proc. 1983 IEEE Intern. Symp. Circuits and Systems,* Newport Beach, CA, pp. 701-704, May 1983.

——, "A VLSI implementation of an FFT/NTT computational unit," in *Proc. IEEE Intern. Conf. ASSP.* pp. 192-195, May 1985.

——, "A VLSI implementation of RNS-based architectures using RNS," in *Proc. IEEE Intern. Symp. Circuits and Systems,* Kyoto, Japan, pp. 1527-1530, June 1985.

——, "An algorithm for solving linear algebraic equations using residue arithmetic—Part II," *Bit* (Sweden), vol. 4, no. 4, 1969.

——, "An efficient VLSI adder for DSP architectures based on RNS," in *Proc. IEEE Intern. Conf. ASSP*, pp. 1457-1460, May 1985.

——, "Architectures for microprocessor-based adaptive filters," in *Proc. 21st Midwest Symp. Circuits Syst.*. Ames, IA,pp. 148-152, Aug. 1978.

——, "Bit-parallel based filters using residue number system," in *Proc. 27th Midwest Symp. Circuits and Systems.* pp. 304-307, June 1984.

——, "Failure resistant digital filters based on residue number system product codes," in *Proc. 1982 IEEE Intern. Conf. Acous., Speech, and Signal Processing,* Paris, France, pp. 60-63, May 1982.

——, "Floating-point arithmetic algorithms in the symmetric residue number system," *IEEE Trans. Comput.,* vol. C-23.no.1,pp. 9-20,Jan. 1974.

——, "Modeling of residue number system arithmetic chips," in *Proc. 14th Pittsburgh Conf. Modeling and Simulation,* Apr. 1983.

——, "Redundant complex residue number systems," in *Proc. Very High Speed Computing Symp.,* Georgia Institute of Technology, Atlanta, GA, pp. IV.3-IV.20, Sept. 1980.

——, "Techniques for residue to analog conversion for residue encoded digital filters," *IEEE Trans. Circuits Syst.,* vol. CAS-25, no. 7, pp. 555-562, July 1978. _

——, "The area-time complexity of a VLSI digital filter using residue number

systems," *Computers and Electrical Eng.,* 1984.
——, "The implementation of the generalized Lagrange FIR filter structure defined over finite fields on rings," in *Proc. Int. Conf. ASSP,* Tokyo, pp. 2583-2586, Apr. 1986.
——, "The numerical system of residual classes in mathematical machines," in *Proc. Congreso Internacional De Automatica.* Madrid, Spain, Oct. 1958. H. L. Garner, "The residue number system," *IRE Trans. Electron. Comput.*. vol. EC-8, pp. 140-147, June 1959.
——, "Using ROM arrays to implement computer arithmetic," in *Proc. 1979 Intern. Conf. Micro and Mini Computers,* Houston, TX, Nov. 1979.
——, "VLSI implementation of RNS .modules and their proposed applications," in *Proc. 21st Ann. Allerton Conf* pp. 1003-1013, Oct. 1983.
А, с. 1332317, СССР, МКИ[4] О 06 Р 7/72. Устройство для нормализации чисел в модулярной системе счисления /А. А. Коляда, М. Ю. Селянинов //Открытия. Изобрет. 1987. № 31.
А. с. 1007098, СССР, МКИ[3] О 06 Р 5/02. Устройство для формирования позиционных признаков непозиционного кода/А. А. Ко-ляда//0ткрытия. ИзобреТ. 1983. № 11.
А. с. 1015382, СССР, МКИ[3] С 06 Р 7/72. Устройство для умножения чисел в непозиционной системе счисления /А. А. Коляда // Открытия. Изобрет. 1983. № 16.
А. с. 1042028, СССР, МКИ[3] С 06 Р 15/332. Арифметическое устройство для процессора быстрого преобразования Фурье/А.А. Коляда, Л. Н. Василевич, В. В. Ревинский, А. Ф. Чернявский//0ткрытия. Изобрет. 1983. № 34.
А. с. 1095172, СССР. Устройство для возведения чисел в квадрат по модулю /В.А. Краснобаев, Е.И. Бороденко, В.И. Стеценко. Опубл. В БИ. 1984. № 20.
А. с. 1095178, СССР, МКИ[3] С 06 Р 7/72. Устройство для умножения чисел по модулю Р / В. А. Краснобаев, Е. И. Бороденко, В. И. Стеценко, А. Ю. Семенов //Открытия. Изобрет. 1984. № 20.
А. с. 1104501, СССР, МКИ[4] О 06 Р 5/02. Устройство для определения ранга числа / В. Н. Ахременко, А. А. Коляда, В. К. Кравцов и др. // Открытия. Изобрет. 1984. № 27.
А. с. 1134941, СССР, МКИ[4] С 06 Р 11/08. Устройство для обнаружения и исправления ошибок в непозиционном коде/А. А. Коляда//0ткрытия. Изобрет. 1985. № 2.
А. с. 1136165, СССР, МКИ[4] О 06 Р 11/08. Устройство для исправления ошибок в непозиционном коде/А.А. Коляда//Там же. № 3.
А. с. 1149254 СССР, МКИ[4] С 06 Р 7/72. Устройство для умножения чисел в системе остаточных классов/А. А. Коляда// Открытия. Изобрет. 1985. № 13.
А. с. 1166317 СССР, МКИ» О 06 Р 11/08. Устройство для контроля информации в системе остаточных классов / В. А. Краснобаев // Откры-

тия. Изобрет. 1985. № 25.
А. с. 1168947 СССР.Устройство для резервирования / В. А. Краснобаев. Опубл. В БИ. 1985. № 27.
А. с. 1176326 СССР, МКИ[4] С 06 Р 7/72, Н 03 М 7/00. Арифметическое устройство в системе остаточных классов / В. М. Амербаев, В. Т. Бородин, В. Н. Колосов, П. И. Рец // Открытия. Изобрет. 1985. № 32.
А. с. 1179546 СССР, МКИ[4] Н 03 М 7/00. Преобразователь непозиционного кода в позиционный код / В. М. Амербаев, А. А. Коляда, В. К. Кравцов, А. Ф. Чернявский // Открытия. Изобрет. 1985. № 34.
А. с. 1180897 СССР, МКИ[4] С 06 Р 11/08. Устройство для коррекции ошибок в непозиционном аддитивном коде/И. Я. Акушский, И. Т. Пак, С. А. Инютин, В. И. Максимов//0ткрытия. Изобрет. -1985. № 35.
А. с. 1190381 СССР, МКИ[4] С 06 Р 7/72. Устройство для округления числа в модулярной системе счисления /В. Н. Ахременко, А. А. Коляда, М. Ю. Селяников, А. Ф. Чернявский // Там же. 1985. № 41.
А. с. 1216777 СССР, МКИ[4] О 06 Р 5/02. Устройство для формирования интегральных характеристик модулярного кода /В. М. Амербаев, А. А. Коляда, В. К. Кравцов, В. В. Ревинский // Там же. 1986. № 9.
А. с. 1236472 СССР, МКИ[4] О 06 Р 7/72. Устройство для умножения в системе остаточных классов /В. Г. Евстигнеев, А. Н. Ко-шарновский, А. В. Маркин, А. С. Новожилов //Открытия. Изобрет. 1986. № 21.
А. с. 1241240 СССР, МКИ[4] О 06 Р 7/72. Устройство для деления чисел в интервально-модулярном коде/А. А. Коляда// Открытия. Изобрет. 1986. № 24.
А. с. 1244665 СССР, МКИ[4] С 06 Р 7/72. Вычислительное устройство в модулярной системе счисления /А. А. Коляда, М. Ю. Селянинов // Открытия. Изобрет. 1986. № 26. 250
А. с. 1259487 СССР, МКИ 4 Я 03 М 1/28. Преобразователь перемещения в код системы остаточных классов / С. Н. Хлевной, О. А. Финь-ко // Открытия. Изобрет. — 1986. - № 35.
А. с. 1263096 СССР. Устройство для резервирования / В. А. Краснобаев. Опубл. В БИ. 1986. № 40.
А. с. 1266009 СССР, МКИ[4] Н 03 М 7/18. Устройство для формирования интегральных характеристик модулярного кода / А. А. Коляда // Открытия. Изобрет. 1986. № 39.
А. с. 1275439 СССР, МКИ[4] С 06 Р 7/72. Устройство для нормализации числа в интервально-модулярном коде/А. А. Коляда// Открытия. Изобрет. 1986. № 45.
А. с. 1280625 СССР, МКИ[4] С 06 Р 7/72. Устройство для умножения комплексных чисел в модулярной системе счисления / А. А. Коляда // Открытия. Изобрет. 1986. № 48.
А. с. 1282116 СССР, МКИ[4] О 06 Р 7/04. Устройство для сравнения чисел в системе остаточных классов / М. К- Буза // Открытия. Изобрет. 1987. № 1.

А. с. 1287152 СССР, МКИ[4] О 06 Р 7/72. Устройство для деления чисел в системе остаточных классов / А. А. Коляда // Открытия. Изобрет. 1987. № 4.
А. с. 1305678 СССР, МКИ[4] С 06 Р 7/72. Устройство для масштабирования числа в интервально-модулярном коде/А. А. Коляда// Открытия. Изобрет. 1987. № 15.
А. с. 1322268 СССР, МКИ[4] О 06 Р 7/544, 7/72. Устройство для вычисления функций в модулярной системе счисления / А. А. Коля-да//0ткрытия. Изобрет. 1987. № 25.
А. с. 1322278 СССР, МКИ[4] О 06 Р 7/72. Устройство для сложения чисел в модулярной системе счисления / А. А. Коляда, М. Ю. Селянинов // Открытия. Изобрет. 1987. № 25.
А. с. 1322483 СССР, МКИ[4] Н 03 М 7/18. Преобразователь двоичного кода в код системы остаточных классов / В. И. Иванченко, П. Л. Прокопьев, В. Н. Торопов // Открытия. Изобрет. 1987. № 25.
А. с. 1330623 СССР, МКИ[4] О 06 Р 7/72. Устройство для масштабирования чисел в системе остаточных классов /А. М. Попов // Открытия. Изобрет. 1987. № 30.
А. с. 1343553 СССР, МКИ 4 Я 03 М 7/18. Преобразователь кода системы остаточных классов в позиционный код / II. И. Червяков, О. Е. Коршунов, О. А. Финько // Открытия. Изобрет. — 1987. — № 37.
А. с. 1352483 СССР, МКИ[4] О 06 Р 7/72. Устройство для умножения чисел в модулярной системе счисления / А. А. Коляда, В. В. Ревинский, М. Ю. Селянинов, А. Ф. Чернявский // Открытия. Изобрет. 1987. № 42.
А. с. 1354190. СССР, МКИ[4] О 06 Р 7/72. Арифметическое устройство в остаточной системе счисления / А, А. Коляда // Открытия. Изобрет. 1987. № 43.
А. с. 1368989 СССР, МКИ 4 Я 03 М 1/28. Аналого-цифровой преобразователь в код системы остаточных классов /О. А. Финько и др. // Открытия. Изобрет. — 1988. — № 3.
А. с. 1372620 СССР, МКИ 4 Я 03 М 1/28. Аналого-цифровой преобразователь в системе остаточных классов / О. А. Финько и др. // Открытия. Изобрет. — 1988. № 5.
А. с. 1383365 СССР, МКИ 4 С 03 Р 11/10. Устройство для свертки по модулю / Н. И. Червяков, И. И. Швецов, О. А. Финько, А. В. Пальцев // Открытия. Изобрет. 1988. — № 11.
А. с. 1388850 СССР, МКИ» О 06 /•• 7/49. Устройство для сложения и вычитания чисел по модулю Р /О. Н. Фоменко, В. А. Краснобаев, С. Н. Иванов и др. // Открытия. Изобрет. 1988. № 14.
А. с. 1388996 СССР, МКИ 4 Я 03 М 7/18. Преобразователь кода из системы остаточных классов в позиционный код / Н. И. Червяков, О. Е. Коршунов, О. А. Финько // Открытия. Изобрет. — 1988. - № 14.
А. с. 1388997 СССР, МКИ[4] Н 03 М 7/18. Преобразователь кода системы остаточных классов в позиционный код / Е. А. Смич-кус, В. Л. Баранов

// Открытия. Изобрет. 1988. № 14.
А. с. 1396283 СССР, МКИ[4] Н 03 М 13/00. Устройство для обнаружения ошибок в двухступенчатом модулярном коде /А. Б. Акулинчев, С. Н. Хлевной//0ткрытия. Изобрет. 1988. № 18.
А. с. 1398103 СССР, МКИ[4] Н 03 М 7/18. Преобразователь позиционного кода числа в модулярный код /А. Б. Акулинчев, С. Н. Хлевнои// Открытия. Изобрет. 1988. № 19.
А. с. 1410281 СССР, МКИ[4] Н 03 М 7/00. Устройство для преобразования непозиционного кода в позиционный код / Л. Н. Васи-левич, А. А. Коляда, В. В. Ревинский, М. Ю. Селянинов // Открытия. Изобрет. 1988. № 26.
А. с. 1432517 СССР, МКИ[4] О 06 Р 7/72. Арифметическое устройство в модулярной системе счисления /А. А. Коляда, М. Ю. Селянинов, А. Ф. Чернявский //Там же. 1988. № 39.
А. с. 1444961 СССР, МКИ 4 Я 03 М 1/28. Преобразователь числа в модулярный код / О. А. Финько и др. // Открытия. Изобрет. – 1988. — №46.
А. с. 1460772 СССР, МКИ[4] Н 03 М 7/18. Преобразователь позиционного кода в модулярный код / Н. И. Швецов, В. А. Краснобаев, В. Н. Телегин // Там же. 1989. № 7.
А. с. 1464293 СССР, МКИ[4] Я 03 М 7/18. Устройство для формирования интегральных характеристик модулярного кода / А. А. Коляда, М. Ю. Селянинов // Там же. 1989. № 9.
А. с. 1557682 СССР, МКИ 4 Я 03 М 7/18. Преобразователь позиционного кода в код системы остаточных классов / О. А. Финько и др. // Открытия. Изобрет. — 1990. — № 14.
А. с. 796846 СССР, МКИ[3] О 06 Р 11/08. Устройство для обнаружения и исправления ошибок арифметических операций в системе остаточных классов/Г. Г. Смолко, И. Я. Акушский, В. М. Бурцев// Открытия, изобретения, промышленные образцы и товарные знаки. 1981. № 2.
А. с. 857992 СССР, МКИ[3] О 06 Р 7/72. Арифметическое устройство в системе остаточных классов / Н. И. Червяков // Открытия, изобретения, промышленные образцы и товарные знаки. 1981. № 31.
А. с. 881745 СССР, МКИ[3] С 06 Р 7/72. Арифметическое устройство в системе остаточных классов/В. С. Василенко, С. И. Гри-горьев// Открытия. Изобрет. 1981. № 42.
А. с. 930317 СССР, МКИ[3] О 06 Р 17/72. Устройство для сложения чисел в системе остаточных классов/А. А. Коляда, В. К. Кравцов, А. Ф. Чернявский//0ткрытия, изобретения, промышленные образцы и товарные знаки. 1982. № 19.
А. с. 959062 СССР, МКИ[3] О 06 Р 5/02. Преобразователь двоичного кода в код системы остаточных классов / А. А. Коляда // Открытия. Изобрет. 1982. № 34.
А. с. 968802 СССР, МКИ[3] О 06 Р 5/02. Устройство для формирования позиционных характеристик непозиционного кола/ А. А. Коляда// Откры-

тия. Изобрет. 1982. № 39.
А. с. 999050 СССР, МКИ[3] О 06 Р 7/72. Арифметическое устройство в системе остаточных классов / Н. Ф. Сидоренко, А. Д. Дубовых, А. В. Королев, В. А. Краснобаев // Открытия. Изобрет. 1983. № 7.
А.с. 1401452 СССР МКИ4 G 06 F 7/49 Сумматор по модули три / О.Н. Музыченко (СССР)- № 4144092/24-24; Заявлено 04.11.86; Опубл. 07.06.88, Бюл. № 21 // Открытия, изобретения.-1988.-№ 21.-С.206.
А.с. 1401461 СССР МКИ4 G 06 F 11/00 Устройство для контроля количества единиц двоичного кода по модули К / О.Н. Музыченко, В.Н. Рыжевнин. В.В. Шлыков. В.И. Новиков (СССР) - № 4155025/24-24; Заявлено 02.12.86; Опубл. 07.06.88, Бюл. № 21 // Открытия, изобретения.-1988.-№ 21.- С.209.
А.с. 1401464 СССР МКИ4 G 06 F 11/10 Устройство для контроля количества единиц двоичного кода по модули К / О.Н. Музыченко (СССР)- № 4130943/24-24; Заявлено 08.10.86; Опубл. 07.06. 88, Бюл. № 21 // Открытия, изобретения.-1988.-№ 2l.- С.210.
А.с. 1403060 СССР МКИ4 G 06 F 7/49 Сумматор унитарных кодов по модули К / О.Н. Музыченко (СССР) - № 4034025/24-24; Заяв енно 07.03.86; Опубл. 15.06.88, Бюл. № 22 / Открытия, изобретения.-1988.-№ 22.- С.179.
А.с. 1492479 СССР МКИ4 Н 03 М 7/20 Устройство для преобразования двоичного кода в код по модули К / О.Н. Музыченко (СССР)- № 4234304/24-24; Заявлено 22.04.87; Опубл. 07.07.89, Бюл. № 25 // Открытия, изобретения.-1989.-№ 25.- С.256.
А.с. 1492479 СССР МКИ4 Н 03 М 7/20 Устройство для преобразования двоичного кода в код по модули К / О.Н. Музыченко (СССР)- № 4234304/24-24; Заявлено 22.04.87; Опубл. 07.07.89, Бюл. № 25 // Открытия, изобретения.-1989.-№ 25.- С.256.
А.с. 1527714 СССР МКИ4 Н 03 М 7/20 Устройство для преобразования количества единиц двоичного кода в код по модулю К / О.Н. Музыченко (СССР) - № 4363832/24-24; Заявлено 08.12.87; Опубл. 07.12.89, Бюл. № 45 // Открытия, изобретения.-1989.-№ 45.-С.249.
А.с. 1566342 СССР МКИ5 G Об F 7/49 Сумматор по модули пять / О.Н. Музыченко (СССР)- № 4487196/24-24; Заявлено 28.09.88; Опубл. 23.05.90, Бюл. № 19 // Открытия, изобретения.-1990.-№ 19.-С.227.
А.с. 1569995 СССР МКИ5 Н 03 М 7/20 Устройство для подсчёта единиц двоичного кода /О.Н. Музыченко (СССР)- № 4340296/24-24; Заявлено 08.12.87; Опубл. 67.06.90, Бюл. № 21 // Открытия, изобретения.-1990.-№ 21.-С.268.
А.с. 1571771 СССР МКИ5 Н 03 М 7/18 Устройство для формирования вычета последовательного двоичного кода по модули / О.Н. Музыченко, В.Н. Рыжевнин, В.В. Шлыков (СССР) - № 4471443/24-24; Заявлено 09.08.88. Опубл. 15.06.90, Бюл. № 22 // Открытия. Изобретения.-1990.-№ 22.-С.266-267.

А.с. 1575186 СССР МКИ5 G 06 F 11/10 Устройство для формирования остатка по модули от числа / О.Н. Музыченко (СССР)- № 4485871/24-24;Заявлено 22.09.88; Опубл. 36.06.90, Бюл. № 24 // Открытия, изобретения.-1990.-№ 24.-С.213-214.

А.с. 1603374 СССР МКИ5 G 06 F 7/49 Сумматор по модули тридцать один / О.Н. Музыченко (СССР)- № 4636546/24-24; Заявлено 12.01.89. Опубл. 30.10.90, Бюл. № 40 // Открытия, изобретения.-1996.-№ 40.-С.211-212.

А.с. 1647560 СССР МКИ5 G 06 F 7/49 Устройство для умножения по модулю семь / О.Н. Музыченко (СССР) - № 4697148/24; Заяв енно 29.05.89; Опубл. О/.05.91, Бюл. № 17 // Открытия, изобретения.-1991.-№ 17.-С.181.

А.с. 1654812 СССР МКИ5 G 06 F 7/49 Сумматор по модулю три / О.Н. Музыченко (СССР) - № 4709244/24; Заявлено 23.06.89. Опубл. 07.06.91, Бюл. № 21 // Открытия, изобретения.-1991.-№ 21.-С.191.

А.с. 1674379 СССР МКИ5 Н 03 М 7/18 Устройство для формирования вычета по произвольному модулю от числа/ О.Н. Музыченко, В.Н. Рыжевнин. В.А. Зайцев (СССР) - № 4731578/24; Заявлено 22.08.89; Опубл. 30.08.91, Бюл. № 32// Открытия, изобретения .-1991.-№ 32.-С.256.

А.с, 1793546. Червяков Н.И., Микула Н.П., Васильев И.А., Квасов М.В., Лавриненко И.Н. Преобразователь двоичного кода в код системы остаточных классов. А.С, 1793546, БИ № 5, 1993.

А.с. 1001079. Червяков Н.И., Болтков А.П., Хлевной С.Н. Преобразователь двоичного кода в код системы остаточных классов. А.С. 1001079, БИ № 8, 1983.

А.с. 1001086 СССР. Устройство для умножения по модулю /В.А. Краснобаев, А.В. Королев. Опубл. В БИ. 1983. № 8.

А.с. 1012242 СССР. Устройство для вычитания по модулю /В.А. Краснобаев, И.Б. Давыдов. Опубл. В БИ. 1983. № 14.

А.с. 1013957 СССР. Устройство для обнаружения ошибок в системе остаточных классов /В.А. Краснобаев, И.Б. Давыдов. Опубл. В БИ. 1983. № 15.

А.с. 1030799 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, Е.И. Бороденко. Опубл. В БИ. 1983. № 27.

А.с. 1034036 СССР. Устройство для возведения чисел в квадрат по модулю /В.А. Краснобаев, Е.И. Бороденко. Опубл. В БИ. 1983. № 29.

А.с. 1037244СССР. Устройство для сравнения чисел в системе остаточных классов /В.А. Краснобаев, Е.И. Бороденко, А.И. Бецков и др. Опубл. В БИ. 1983. № 31.

А.с. 1038951 СССР. Устройство для моделирования сетевого графика / Е. И. Бороденко,В.А. Краснобаев, А.И. Бецков и др. Опубл. В БИ. 1983. № 32.

А.с. 1076899 (СССР) Балюк В.В., Выжиковски Р., Каневский Ю.С. Преобразователь n-разрядного двоичного кода в его представление по модулю М. // Б.И. 1984, № 8. С. 159.

А.с. 1078655 СССР. Устройство для исправления одиночных и обнаружения многократных ошибок /В.А. Краснобаев, Е.И. Бороденко, А.И. Бецков. Опубл. В БИ. 1984. № 9.
А.с. 1095178 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, Е.И. Бороденко, В.И. Стеценко и др. Опубл. В БИ. 1984. № 20.
А.с. 1096641 СССР. Устройство для возведения чисел в квадрат по модулю /В.А. Краснобаев, Е.И. Бороденко, А.Ю. Семенов и др. Опубл. В БИ. 1984. № 21.
А.с. 1100619, Евстигнеев В.Г., Белова Р.С., Сведе-Швец В.Н. Устройство для умножения в системе остаточных классов. Авторское свидетельство СССР. – № 1100619, Б.И. 1984 № 24.
А.с. 110311, Евстигнеев В.Г., Белова Р.С., Сведе-Швец В.Н. Преобразователь двоичного кода в код по модулю q. Авторское свидетельство СССР. – № 110311, 1982.
А.с. 1105896 СССР. Самойлов А.Л. Пирамидальная свертка по модулю три. // Б. И. 1984. № 28. С. 145.
А.с. 1107122 СССР. Арифметическое устройство в системе остаточных классов /Ю.В. Пшеничный, В.А. Краснобаев, Е.И.Бороденко и др. Опубл. В БИ. 1984. № 29.
А.с. 1111170, Евстигнеев В.Г. Сумматор в системе остаточных классов. Авторское свидетельство СССР. – № 1111170, Б.И. 1984 № 32, G06F11/10.
А.с. 1115607, Евстигнеев В.Г. Устройство для деления q-ичных чисел. Авторское свидетельство СССР. – № 1115607, 1982, ДСП.
А.с. 1116870, Евстигнеев В.Г., Белова Р.С., Сведе-Швец В.Н. Устройство для умножения в системе остаточных классов. Авторское свидетельство СССР. – № 1116870, 1982.
А.с. 1120325, Евстигнеев В.Г., Евстигнеева О.В. Арифметическое устройство по модулю. Авторское свидетельство СССР. – № 1120325, Б.И. 1987, № 39, G06F7/72.
А.с. 1121670 СССР. Устройство для сравнения чисел в системе остаточных классов /В.А. Краснобаев, Л.Г.Трусей. БИ. 1984. № 40.
А.с. 1126950 СССР. Устройство для умножения по модулю /В.А. Краснобаев, Л.Г. Трусей. Опубл. В БИ. 1984. № 44.
А.с. 1133669. Червяков Н.И., Хлевной С.Н., Швецов Н.И., Болтков А.П. Преобразователь кода системы остаточных классов в двоичный код. А.С. 1133669, БИ № 1, 1985.
А.с. 1134939. Червяков Н.И., Хлевной С.Н., Швецов Н.И., Цюпко В.А. Сумматор по модулю. А.С. 1134939, БИ № 2, 1985.
А.с. 1141400, Евстигнеев В.Г., Евстигнеева О.В., Куракин В.А. Устройство для деления в системе остаточных классов. Авторское свидетельство СССР. – № 1141400, Б.И. 1985, № 7, G06F7/49.
А.с. 1145338 СССР. Устройство для сравнения чисел в системе остаточ-

ных классо /В.А. Краснобаев. Опубл. В БИ. 1985. № 10.
А.с. 1148121. Червяков Н.И., Хлевной С.Н., Вершков Н.А. Преобразователь напряжения в код системы остаточных классов. А.С. 1148121, БИ №12, 1985.
А.с. 1151948. Червяков Н.И., Хлевной С.Н., Швецов Н.И., Болтков А.П. Преобразователь кода системы остаточных классов в позиционный код. А.С. 1151948, БИ № 15, 1985.
А.с. 1151970 СССР. Устройство для определения альтернативной совокупности чисел в СОК / В.А. Краснобаев и др. Опубл. В БИ. 1985. № 15.
А.с. 1160394 СССР. Устройство для сравнения чисел в системе остаточных классов /В.А. Краснобаев. Опубл. В БИ. 1985. № 21.
А.с. 1160397 СССР. Устройство для возведения чисел в степень по модулю /В.А. Краснобаев, А.Ю. Опубл. В БИ. 1985. № 21.
А.с. 1163321, Евстигнеев В.Г., Евстигнеева О.В. Устройство для сложения многоразрядных q-ичных чисел. Авторское свидетельство СССР. – № 1163321, Б.И. 1985, № 23, G06F7/49.
А.с. 1166098 СССР. Устройство для умножения в системе остаточных классов /В.А. Краснобаев. Опубл. В БИ. 1985. № 25.
А.с. 1166117 СССР. Устройство для контроля информации в СОК /В. А. Краснобаев. Опубл. В БИ. 1985. № 25.
А.С. 1166897 (СССР). Устройство для обнаружения ошибок в слабоарифметическом коде систем остаточных классов // Ин-т матем. И мех. АН КазССР. Авт. Изобрет. И.Т. Пак, С.А. Инютин.
А.с. 1168934 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, Е.И. Бороденко, Ю.В. Пшеничный. Опубл. В БИ. 1985. № 37.
А.с. 1175034. Червяков Н.И., Хлевной С.Н., Вершков Н.А., Швецов Н.И. Преобразователь кода системы остаточных классов в напряжение. А.С. 1175034, БИ № 31, 1985.
А.с. 11779547. Червяков Н.И., Хлевной С.Н., Швецов Н.И. Преобразователь непозиционного кода в двоичный код. А.С. 11779547, БИ № 34, 1985.
А.с. 1182511 СССР МКИ4 G 06 F 7/50 Сумматор унитарных кодов / О.Н.Музыченко (СССР) - № 3299426/24-24; Заявлено 15.06.82; Опубл. 30.09.85, Бюл. № 36 // Открытия, изобретения.-1985.-№ 36.-С.196.
А.с. 1185339 СССР. Преобразователь позиционного кода в вычеты по проивольному модулю / В.А. Краснобаев, А.И. Сахно, А.В. Королев. Опубл. В БИ. 1985. № 38.
А.с. 1187161 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, О.Н. Фоменко и др. Опубл. В БИ. 1985. № 39.
А.с. 1188731, Евстигнеев В.Г., Евстигнеева О.В. Устройство для сложения чисел в избыточной системе счисления. Авторское свидетельство СССР. – № 1188731, Б.И. 1985, № 40, G06F7/49.
А.с. 1195349 СССР. Преобразователь позиционного кода в вычет по произвольному модулю /В.А. Краснобаев, А.И. Сахно, А.В. Королев.

Опубл. В БИ. 1985. № 44.
А.с. 1200278. Червяков Н.И., Швецов Н.И., Сагдеев К.М. Арифметическое устройство. А.С. 1200278, БИ № 47, 1985.
А.с. 1200432 СССР МКИ4 Н 03 М 7/18 Преобразователь кода числа из системы остаточных классов в позиционный код/ В.П. Лукоянов, О.Н. Музыченко. Н.К. Кора, А.В. Ростовский (СССР)-№ 3409961/24-24; Заявлено 16.05.82; Опубл. 23.12.85, Бюл. № 47 // Открытия, изобретения.-1985.-№ 47.-С.264.
А.с. 1206961 СССР МКИ4 Н 03 М 7/18 Преобразователь кода числа из системы остаточных классов в позиционный код/ В.П. Лукоянов, А.Р. Мартьянов. О.Н. Музыченко (СССР)- № 3409962/24-24; Заявл.16.03.82; Опубл. 23.8l.86, Бюл. № 3 // Открытия, изобретения.-1986.-№ 3.-С.245.
А.с. 1206962 СССР. Устройство для коррекции ошибок информации, представленной в системе остаточных классов /В.И. Долгов, А.В. Брезгунов, В.А. Краснобаев. Опубл. В БИ. 1986. № 15.
А.с. 1224803 СССР. Устройство для сравнения чисел в системе остаточных классов /В.И. Долгов, В.А. Краснобаев, А.В. Брезгунов. Опубл. В БИ. 1986. № 18.
А.с. 1226670 СССР. Обратимый преобразоватеь позиционного кода в код системы остаточных классов /В.А. Краснобаев, Н.И. ШвецовА.В, А.И. Сахно и др. Опубл. В БИ. 1986. № 19.
А.с. 1230300 СССР. Устройство выбора канала с экстремальным среднем напряжением / В.И Долгов, А.В. Брезгунов, В.А. Краснобаев и др. Опубл. В БИ. 1986. № 20.
А.с. 1233154 СССР. Устройство для возведения чисел в квадрат по модулю /В.А. Краснобаев, О.Н. Фоменко и др. Опубл. В БИ. 1986. № 20
А.с. 1241482 СССР. Дешифратор /В.А. Краснобаев, В.С. Харченко, Г.Н. Тимонькин и др. Опубл. В БИ. 1986. № 24.
А.с. 1247868 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1986. № 28.
А.с. 1257643 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1986. № 34.
А.с. 1259255 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев. Опубл. В БИ. 1986. № 35.
А.с. 1275432, Евстигнеев В.Г., Евстигнеева О.В., Титов А.П. Устройство для умножения. Авторское свидетельство СССР. – № 1275432, Б.И. 1986, № 45, G06F7/52.
А.с. 1280390, Евстигнеев В.Г., Канаев А.Е., Кошарновский А.Н. Цифровой фильтр. Авторское свидетельство СССР. – № 1280390, Б.И. 1986, № 48, G06F15/353.
А.с. 1283948. Червяков Н.И., Хлевной С.Н., Сагдеев К.М. Устройство для определения позиционных характеристик непозиционного кода. А.С. 1283948, БИ

№ 2, 1987.
А.с. 1285468, Евстигнеев В.Г., Амирханов А.В, Кошарновский А.Н., Кудрявцев В.С., Станков В.С. Арифметическое устройство по модулю. Авторское свидетельство СССР. – № 1285468, Б.И. 1987, № 3.
А.с. 1290314, Евстигнеев В.Г., Евстигнеева О.В. Устройство для суммирования в избыточной системе счисления. Авторское свидетельство СССР. – № 1290314, Б.И. 1987, № 6, G06F7/72.
А.с. 1290350 СССР, МКИ[4] О 06 Р 15/332. Устройство .для БПФ/Е. К. Лебедев, В. Ю. Лапий//Б.И.. 1987. № 6.
А.с. 1290915, Евстигнеев В.Г. Устройство для деления n-разрядных q-ичных чисел. Авторское свидетельство СССР. – № 1290915, Б.И. 1985, ДСИ.
А.с. 1312572 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1987. № 19.
А.С. 1313315 (СССР). Преобразователь позиционного кода в модулярный кол // Ин-т матем. И мех. АН КазССР. Авт. Изобрет. И.Т. Пак, В.М. Амербаев, Р.Н. Турмухамбетов. 1987
А.с. 1322246, Евстигнеев В.Г., Добровольская И.А., Сафонов Е.Н. Таймер. Авторское свидетельство СССР. – № 1322246, Б.И. 1987, № 25, G06F1/04.
А.с. 1341722. Червяков Н.И., Болтков А.П., Хлевной С.Н. Преобразователь кода числа во временной интервал. А.С. 1341722, БИ № 36, 1987.
А.с. 1343553. Червяков Н.И., Коршунов О.Е., Финько О.А. Преобразователь кода системы остаточных классов в позиционный код. А.С. 1343553, БИ № 37, 1987.
А.с. 1357960 СССР МКИ4 G 06 F 11/10 Устройство для контроля количества единиц двоичного кода по модулю К/ О.Н. Музыченко (СССР)- № 3682843/24-24; Заявлено 30.12.83; Опубл. 07.12.87, Бюл. № 45 // Открытия, изобретения.-1987.-№ 45.-С.202.
А.с. 1360419 СССР. Резервированное устройство /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1987. № 46.
А.с. 1361557 СССР МКИ4 G 06 F 11/08 Устройство для контроля параллельного двоичного кода по модулю К / О.Н. Музыченко (СССР)- № 4032650/24-24; Заявлено 05.03.86; Опубл. 23.12.87, Бюл. № 47 // Открытия, изобретения.-1987.-№ 47.-С.202.
А.с. 1363214 СССР МКИ4 G 06 F 11/08 Устройство для формирования остатка по произвольному модулю от числа / О.Н. Музыченко (СССР)- № 4093760/24-24; Заявлено 23.07.86; Опубл. 30. 12.87, Бюл. № 48// Открытия, изобретения.-1987.-№ 48.-С.182.
А.с. 1363512 СССР. Устройство выбора каналов /В.И. Долгов, А.В. Брезгунов, В.А. Краснобаев и др. Опубл. В БИ. 1988. № 2.
А.с. 1368879 СССР. Устройство для сложения и вычитания чисел по модулям/В.И. Долгов, В.А. Краснобаев, А.В. Брезгунов и др. Опубл. В БИ. 1988. № 3.

А.с. 1368989 СССР. Аналогово-цифровой преобразователь в код системы остаточных классов /В.А. Краснобаев, О.Н. Фоменко, Н.И. Швецов и др. Опубл. В БИ. 1988. № 5.
А.с. 1372620 СССР. АЦП в системе остаточных классов /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1988. №
А.с. 1376245 СССР. Преобразователь позиционного кода в код системы остаточных классов /В.А. Краснобаев, О.Н. Фоменко, Н.И. Швецов и др. Опубл. В БИ. 1988. № 7.
А.с. 1381488 СССР. Сумматор по модулю три /В.И. Долгов, В.А. Краснобаев, А.В. Брезгунов и др. Опубл. В БИ. 1988. № 10.
А.с. 1383341 СССР. Устройство для сложения и вычитания чисел по модулям /В.А. Краснобаев, В.Д. Экста, В.И. Пироженко и др. Опубл. В БИ. 1988. № 11.
А.с. 1383349, Евстигнеев В.Г., Алексеев А.В., Бондаренко А.В., Куракин В.А. и др. Сумматор в знакоразрядной позиционно-остаточной системе счисления. Авторское свидетельство СССР. – № 1383349, Б.И. 1988, № 11, G06F7/72.
А.с. 1388550 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1988. № 14.
А.с. 1399743. Червяков Н.И., Писарев А.А. Устройство для обнаружения ошибок в системе остаточных классов. А.С. 1399743, БИ № 20, 1988.
А.с. 1403371. Червяков Н.И., Малофей О.П., Николаев Ю.И., Швецов Н.И., Болтков А.П. Преобразователь перемещения в код. А.С. 1403371, БИ № 22, 1988.
А.с. 1425656. Червяков Н.И., Камельчук М.Ю., Шайкин А.Е. Арифметическое устройство. А.С. 1425656, БИ № 35, 1988.
А.с. 1425666, Евстигнеев В.Г., Кошарновский А.Н. Арифметическое устройство по модулю. Авторское свидетельство СССР. – № 1425666, Б.И. 1988, № 35, G06F7/72.
А.с. 1425676 СССР МКИ4 G 06 F 11/08 Устройство для контроля параллельного двоичного кода по модули К / О.Н. Музыченко (СССР)- № 4101662/24-24; Заявлено 27.05.86; Опубл. 23.09.88, Бюл. № 35 // Открытия, изобретения.-1988.-№ 35.-С.210.
А.с. 1425845 СССР МКИ4 Н 03 М 7/12 Устройство для свертки двоичного кода в код по модули К/ О.Н. Музыченко (СССР) - № 4107828/24-24; Заявлено 12.09.88; Опубл. 23.06.86, Бюл. № 35 // Открытия, изобретения.-1988.-№ 35.- С.268.
А.с. 1427358 СССР. Устройство для сравнения чисел в системе остаточных классов / В.А. Краснобаев, И.Д. Горбенко, М.А. Гальцев и др. Опубл. В БИ. 1988. № 36.
А.с. 1427574 СССР МКИ4 Н 03 М 7/20 Устройство для подсчета числа единиц двоичного кода по модули К / О.Н. Музыченко, В.Н. Рыжевнин, В.В. Шлыков, В.П. Костромитин (СССР)- № 4155410/24-24; Заявлено 02.12.86; Опубл. 30.09.88, Бюл. № 36 // Открытия, изобретения.-1988.-

№ 36.- С.245.
А.с. 1429322 СССР МКИ4 Н 03 М 7/12 Преобразователь двоичного кода в код по модули К / О.Н. Музыченко (СССР)- № 4191976/24-24; Заявлено 04.02.87; Опубл. 07.10.88, Бюл. № 37 // Открытия, изобретения.-1988.- № 37.- С.245.
А.с. 1432503 СССР МКИ4 G Об F 7/49 Сумматор по модули три / О.Н. Музыченко (СССР) – М 4144102/24-24; Заявлено 04.11.86; Опубл. 23.10.88, Бюл. № 39 // Открытия, изобретения.-1988.-№ 39.- С.185.
А.с. 1432772 СССР. Преобразователь перемещения в код /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1988. № 40.
А.с. 1438006 СССР МКИ4 Н 03 М 7/20 Устройство для подсчёта числа единиц двоичного кода по модули К / О.Н. Музыченко, Б.Б. Трушкин. В.Н. Беляев (СССР) - № 4155411/24-24; Заявлено 02.12.86. Опубл. 15.11.88, Бюл. № 42 // Открытия, изобретения.-1986.-№ 42.- С.255-256.
А.с. 1441395 СССР.Сумматор-умножитель по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1988. № 44.
А.с. 1451690 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1989. № 2.
А.с. 1451691 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. В БИ. 1989. № 2.
А.с. 1460772 СССР. Преобразователь позиционного кода в модулярний код /В.А. Краснобаев, Н.И. Швецов, О.Н. Фоменко и др. Опубл. В БИ. 1989. № 7.
А.с. 1483450 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, А.И. Сахно, В.И. Глушко и др. Опубл. В БИ. 1989. № 20.
А.с. 1487035 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, В.Г. Евстигнеев, О.Н. Фоменко и др. Опубл. В БИ. 1989. № 22.
А.с. 1488968. Червяков Н.И., Акиншин М.А., Колесников И.А., Микулич В.И., Серавин Д.А. Устройство для преобразования чисел из кода системы остаточных классов в позиционный код с контролем ошибок. А.С. 1488968, БИ № 23, 1989.
А.с. 1509903 СССР. Устройство для свертки по произвольному модулю /В.А. Краснобаев, Г.М. Чигасов, В.Д. Экста и др. Опубл. В БИ. 1989. № 35
А.с. 1520667 СССР. Устройство для формирования остатка по произвольному модулю от числа /В.А. Краснобаев, Л.С. Сорока, Г.М. Чигасов и др. Опубл. В БИ. 1989. № 40.
А.с. 1532923 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. В БИ. 1989. № 48.

А.С. 1532924 (СССР). Устройство для формирования позиционного признака в модулярной арифметике // Ин-т матем. И мех. АН КазССР. Авт. Изобрет. И.Т. Пак, Р.Н. Турмухамбетов. 1990
А.С. 1541605 (СССР). Устройство для масштабирования чисел в формулярной арифметике // Ин-т матем. И мех. АН КазССР. Авт. Изобрет. И.Т. Пак, Р.Н. Турмухамбетов. 1990
А.с. 1546976 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский, В.П. Ирхин и др. Опубл. В БИ. 1990. № 8.
А.с. 1546977 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский, В.П. Ирхин и др. Опубл. В БИ. 1990. № 8.
А.с. 1552171 СССР. Устройство для сравнения в системе остаточных классов /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. В БИ. 1990. № 11.
А.с. 1557682 СССР. Преобразователь в системе остаточных классов / Н.И. Швецов, В.А. Краснобаев, О.Н. Фоменко и др. Опубл. В БИ. 1990. № 9.
А.с. 1571583 СССР. Арифметическое устройство по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1990. № 22.
А.с. 1580561 СССР. Устройство для формирования остатка по произвольному модулю от числа /В.А. Краснобаев, Л.С. Сорока, С.А. Чепига и др. Опубл. в БИ. 1990. № 30.
А.С. 1587639 СССР. Преобразователь модулярного кода в позиционный код// Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, Р.Н. Турмухамбетов. 1990
А.с. 1587641 СССР МКИ5 Н 03 М 7/20 Устройство для преобразования двоичного кода по МОДУЛЮ К / О.Н. Музыченко (СССР)- № 4339080/24-24; Заявлено 08.12.87; Опубл. 23.08.90; Бюл. № 31 // Открытия, изобретения.-1990.-№ 31.-С.262-263.
А.с. 1587642 СССР МКИ5 Н 03 М 7/20 Устройство для преобразования двоичного кода по модулю К / О.Н. Музыченко (СССР) - № 4363831/24-24; Заявлено 08.12.87; Опубл. 23.08.90, Бюл. № 31 // Открытия, изобретения.-1990.-№ 31.-С.258.
А.с. 1594541 СССР. Устройство для свертки чисел по модулю /В.А. Краснобаев, Л.С. Сорока, С.А. Чепига . Опубл. в БИ. 1990. №35.
А.с. 1599857 СССР. Устройство для сложения и умножения чисел по модулям /В.А. Краснобаев, В.И. Глушков, А.И. Сахно и др. Опубл. в БИ. 1990. № 38.
А.с. 1603371 СССР МКИ G 06 F 7/49 Сумматор по модули семь / О.Н. Музыченко (СССР)- № 4636229/24-24; Заявлено 12.01.89; Опубл. 30.10.90, Бюл. № 40 // Открытия, изобретения.-1990.-№ 40.-С.210.
А.с. 1603372 СССР МКИ5 £ Об F 7/49 Сумматор по модули семь / О.Н. Музыченко (СССР) - № 4636232/24-24; Заявлено Т2.01.89; Опубл.

30.10.90, Бюл. № 40 // Открытия, изобретения.-1990.-№ 40.-С.210-211.
А.с. 1603373 СССР МКИ5 G 06 F 7/49 Сумматор по модули тридцать один /О.Н. Мизыченко (СССР)- № 4636246/24-24; Заявлено 12.01.89; Опубл. 30.10.90, Бюл. № 40 // Открытия, изобретения.- 1990.-№ 40.-С.211.
А.с. 1603375 СССР МКИ5 G 06 F 7/49 Сумматор по модули пятнадцать / О.Н. Музыченко (СССР) - № 4636550/24-24; Заявлено 12. 01.89; Опубл. 30.10.90, Бюл. № 40 // Открытия, изобретения.-1990.-№ 40.-С.212-213.
А.с. 1603376 СССР МКИ5 G 06 F 7/49 Сумматор по модули шестьдесят три / О.Н. Музыченко (СССР) - № 4636551/24-24; Заявлено 12.01.89; Опубл. 30.10.90, Бюл. № 40 // Открытия, изобретения.-1990.-№ 40.-С.213-214.
А.с. 1605225 СССР МКИ5 G 06 F 7/49 Сумматор по модули пятнадцать /О.Н. Музыченко (СССР)- № 4636245/24-24; Заявлено 12. 01.89; Опубл. 07.11.90, Бюл. № 41 // Открытия, изобретения.- 1990.-№ 41.-С.201.
А.с. 1605226 СССР МКИ5 G 06 F 7/49 Сумматор по модули шестьдесят три / О.Н. Музыченко (СССР) - № 4636250/24-24; Заявлено 12.01.89; Опубл. 07.11.90, Бюл. № 41 // Открытия, изобретения.-1990.-№ 41.-С.202.
А.с. 1605227 СССР МКИ5 G 06 F 7/49 Сумматор по модулю шестьдесят три / О.Н. Музыченко (СССР) - Н 4636547/24-24; Заявлено 12.01.89; Опубл. 07.11.90, Бюл. № 41 // Открытия, изобретения.-1990.-№ 41.-С.203.
А.с. 1608645 СССР МКИ5 G 06 F 7/49 Сумматор по модулю семь / О.Н. Музыченко (СССР) - № 4636228/24-24; Заявлено 12.01.89; Опубл. 23.11.90, Бюл. № 43 // Открытия, изобретения.- 1990.-№ 43.-С.187.
А.с. 1608646 СССР МКИ5 G 06 F 7/49 Сумматор по модулю тридцать один // О.Н. Музыченко (СССР) - № 4636231/24-24; Заявлено 12.01.89; Опубл. 23.11.90, Бюл. № 43 // Открытия, изобретения.-1990.-№ 43.-С.187.
А.с. 1615714 СССР. Устройство для умножения по модулю /В.А. Краснобаев, Г.М. Чигасов, В.П. Ирхин и др. Опубл. в БИ. 1990. № 47.
А.с. 1617439 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, В. И. Глушков, В.П. Ирхин и др. Опубл. в БИ. 1990 № 48.
А.с. 1619403 СССР. Устройство для перевода числа, представленного в системе остаточных классов, в полиадическую систему счисления / В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1991. № 1.
А.с. 1624442 СССР МКИ5 G 06 F 7/49 Сумматор по модулю пятнадцать / О.Н. Музыченко (СССР)- № 4636230/24-24; Заявлено 12. 01.89; Опубл. 30.01.91, Бюл. № 4 // Открытия, изобретения.-1991.-№ 4.-С.143.
А.с. 1633399 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1991. № 9.

А.с. 1633400 СССР. Арифметическое устройство по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1991. № 9.
А.с. 1636844 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1991. № 11.
А.с. 1644131 СССР МКИ5 G 04 F 7/49 Устройство для умножения по модулю пять /О.Н. Музыченко (СССР) - № 4709466/24; Заявлено 23.06.89; Опубл. 23.04.91, Бюл. № 15 // Открытия, изобретения.-1991.-№ 15.-С.186-187.
А.с. 1647561 СССР МКИ5 Q 06 F 7/49 Устройство для умножения по модулю семь /О.Н. Музыченко (СССР) - № 4698020/24; Заявлено 25.09.89; Опубл. 07.05.91, Бюл. № 17 // Открытия, изобретения.-1991.-№ 17.-С.181-182.
А.с. 1647563 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, В.П. Ирхин, М.И. Цыба и др. Опубл. в БИ. 1991. № 15.
А.с. 1647909. Червяков Н.И., Васильев И.А., Микула Н.П. Преобразователь кодов из системы остаточных классов в двоичный позиционный код. А.С. 1647909, БИ № 17, 1991.
А.с. 1658142 СССР МКИ5 G 06 F 7/49 Сумматор по модулю пять / О.Н. Музыченко (СССР) - № 4666906/24; Заявлено 27.03.89; Опубл. 23.06.91, Бюл. № 23.// Открытия, изобретения.-1991.-№ 23.-С.169.
А.с. 1667055 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1991. № 28.
А.с. 1683011 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, В.И. Долгов, В.П. Ирхин и др. Опубл. в БИ. 1991. № 37.
А.с. 1683012 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский, В.П. Ирхин и др. Опубл. в БИ. 1991. № 37.
А.с. 1683014 СССР. Устройство для возведения чисел в степень по модулю три /В.А. Краснобаев, О.Н. Фоменко, В.П.Ирхин и др. Опубл. в БИ. 1991. № 30 (37?).
А.с. 1689949 СССР. Устройство для вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1991. № 41.
А.с. 1697079 СССР. Устройство для умножения по модулю / В.А. Краснобаев, В.П. Ирхин, В.И. Глушков и др. - Опубл. в БИ. 1991. № 45.
А.с. 1702366 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, В.И. Глушков, А.И. Сахно и др. Опубл. в БИ. 1991. № 48.
А.с. 1716511 СССР. Устройство для умножения чисел по модулю /В.А. Краснобаев, В.П. Ирхин, В.Д. Экста и др. Опубл. в БИ. 1992. № 8.
А.с. 1732470 СССР. Аналогово-цифровой преобразователь напряжения в код системы остаточных классов /В.А. Краснобаев, О.Н. Фоменко, В.А. Каревский и др. Опубл. в БИ. 1992. № 17.

А.с. 1734212. Червяков Н.И., Оленев А.А. Устройство для вычисления остатка по модулю 2+1. А.С. 1734212, БИ № 18, 1992.
А.с. 1751857. Червяков Н.И., Оленев А.А., Сагдеев К.М. Устройство для вычисления остатка по модулю от двоичного числа. А.С. 1751857, БИ № 28, 1992.
А.с. 1751858. Червяков Н.И., Оленев А.А. Устройство для вычисления остатка по модулю от двоичного числа. А.С. 1751858, БИ № 28, 1992.
А.с. 1755275 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, В.П. Ирхин, А.И. Сахно и др. Опубл. в БИ. 1992. № 30.
А.с. 1756881 СССР. Арифметическое устройство по модулю /В.А. Краснобаев, В.П. Ирхин, М.В. Юмашев и др. Опубл. в БИ. 1992. № 31.
А.с. 1775721 СССР. Арифметическое устройство по модулю /В.А. Краснобаев, В.П. Ирхин, М.В. Квасов и др. Опубл. в БИ. 1992. № 42.
А.с. 1807484 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, В.П. Ирхин, А.И. Сахно и др. Опубл. в БИ. 1993. № 13.
А.с. 1809437 СССР. Арифметическое устройство по модулю /В.А. Краснобаев, В.П. Ирхин, М.В. Квасов и др. Опубл. в БИ. 1993. № 14.
А.с. 1810889 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1993. № 15.
А.с. 1820379 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1993. № 21.
А.с. 1820380 СССР. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, О.Н. Фоменко, В.П. Ирхин и др. Опубл. в БИ. 1993. № 21.
А.с. 359652, Евстигнеев В.Г., Акушский И.Я. Устройство управления для цифровых вычислительных машин. Авторское свидетельство СССР. – № 359652, Б.И. 1972, № 35, G06F9/06.
А.с. 377767. Червяков Н.И. Устройство для преобразования чисел из десятичной системы счисления в систему остаточных классов. А.С. 377767, БИ № 18, 1979.
А.с. 402867, Евстигнеев В.Г., Акушский И.Я. Дешифратор кодов в СОК. Авторское свидетельство СССР. – № 402867, Б.И. 1973, № 42, G06F5/02.
А.с. 529669 (СССР). Устройство для формирования позиционного признака непозитивного кода // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.
А.с. 579611 (СССР). Устройства для формирования признака переполнения и знака при сложении и вычитании // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев
А.с. 579614 (СССР). Устройство деления // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев, А.О. Жаутыков.

А.с. 579617 (СССР). Устройство для умножения чисел // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.
А.с. 579618 (СССР). Устройство умножения // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев, Б.Е. Дуйсенов.
А.с. 584648 (СССР). Устройство умножения //' Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев, А.Н. Казангапов.
А.с. 601689 (СССР). Арифметическое устройство // Ин- т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев, А.О-Жаутыков.
А.с. 603987 СССР. Факторович М.Г., Полисский Ю.Д.. Устройство для определения максимального и минимального из "n" чисел, представленных в системе остаточных классов. А.с. 603987, М. Кл.^{2} G06 F 7/04, 1978.
А.с. 608155 СССР. Факторович М.Г., Полисский Ю.Д.. Устройство для сравнения чисел, выраженных в системе остаточных классов. А.с. 608155, М. Кл.^{2} G06 F 7/04, 1978.
А.с. 618739 Полисский Ю.Д. Факторович М.Г..Устройство для сравнения чисел в системе остаточных классов., М. Кл.^{2} G06 F 7/04, 1978.
А.с. 637809 СССР Полисский Ю.Д. Факторович М.Г.. Преобразователь кодов из СОК в полиадический код., М. Кл.^{2} G06 F 5/02, 1978.
А.с. 7638886, Евстигнеев В.Г., Амербаев В.М., Ицкович А.А. Дешифратор кодов СОК. Авторское свидетельство СССР. – № 7638886, Б.И. 1980, № 34, G06F5/02.
А.с. 885999 СССР. Устройство для умножения и сложения чисел по модулю / В.А. Краснобаев. Опубл. в БИ. 1981. № 44.
А.с. 894711, Евстигнеев В.Г., Амербаев В.М., Бияшев Р.Г., Черкасов Ю.Г. Устройство для обнаружения и исправления ошибок арифметических преобразований полиномиального кода. Авторское свидетельство СССР. – № 894711, Б.И. 1981, № 48, G06F11/08.
А.с. 896620 СССР. Устройство для умножения по модулю / В.А. Краснобаев. Опубл. в БИ. 1982. № 1.
А.с. 917541 СССР. Устройство переводов в системе остаточных классов // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, Р.Н. Турмухамбетов. 1982
А.с. 922731 СССР. Устройство для умножения в системе остаточных классов / В.А. Краснобаев. Опубл. в БИ. 1982. № 15.
А.с. 932499 (СССР). Устройство для исправления ошибок // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский. 1982
А.с. 947802 (СССР), Устройство умножения // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский и др. 1982
А.с. 951296 СССР. Устройство для умножения по модулю / А.В. Королев, В.А. Краснобаев, Б. И. Сергеев. Опубл. в БИ. 1982. № 30.

А.с. 959067 СССР. Устройство для вычитания по модулю / В.А. Краснобаев, А.В. Королев, Б.И. Сергеев. Опубл. в БИ. 1982. № 34.
А.с. 959068 СССР. Устройство для умножения по модулю / В.А. Краснобаев, А.В. Королев. Опубл. в БИ. 1982. № 34.
А.с. 964645 СССР. Устройство для обнаружения одиночных ошибок кода в системе остаточных классов / В.А. Краснобаев, А.И. Бецков, Е.И. Бороденко др. Опубл. в БИ. 1982. № 37.
А.с. 968808 СССР. Устройство для модульного умножения / А.В. Королев, В.А.Краснобаев. Опубл. в БИ. 1982. № 39.
А.с. 976440 СССР. Устройство для умножения чисел по модулю / В.А. Краснобаев, А.В. Королев. Опубл. в БИ. 1982. № 43.
А.с. 981990 СССР. Устройство для умножения по модулю /В.А. Краснобаев, А.В. Королев. Опубл. в БИ. 1982. № 46.
А.с. 981991 СССР. Устройство для умножения по модулю /В.А. Краснобаев, А.В. Королев. Опубл. в БИ. 1982. № 46.
А.с. 983701. Червяков Н.И., Болтков А.П., Хлевной С.Н. Преобразователь двоичного кода в код системы остаточных классов. А.С. 983701, БИ № 47, 1982.
А.с. 999050 СССР. Арифметическое устройство в системе остаточных классов / Н.Ф. Сидоренко, В.А. Краснобаев, А.В. Королев и др. Опубл. в БИ. 1983. № 7.
А.с. В-1501 СССР. Способ кодирования// Ин-т матем. и мех. АН КазСССР. Авт. изобрет. И.Т. Пак, И.Я. Акушский, Д.И. Юдицкий.
А.с. № 1005028, Червяков Н.И., Шамардинов В.А., Литвинов С.Н. Устройство для преобразования числа из системы остаточных классов в позиционный код. А.С. № 1005028, БИ № 10, 1983.
А.с. № 1008729. Червяков Н.И., Болтков А.П., Хлевной С.Н. Устройство для преобразования чисел из позиционной системы счисления в систему остаточных классов. А.С. № 1008729, БИ № 12, 1983.
А.с. № 1012237. Червяков Н.И., Шамардинов В.А. Устройство для преобразования чисел из позиционной системы счисления в систему остаточных классов. А.С. № 1012237, БИ № 14, 1983.
А.с. № 1552181. Червяков Н.И., Скиба Г.П., Журиков А.Г., Кечкин М.А. Устройство для определения знака числа, представленного в системе остаточных классов. А.С. № 1552181, БИ № 11, 1990.
А.с. № 376770 Червяков Н.И. Устройство для округления дробей, представленных в системе остаточных классов. А.С. № 376770, БИ № 17, 1973.
А.с. № 675420,. Червяков Н.И. Устройство для сравнения n-разрядных десятичных чисел. А.С. № 675420, Б.И. № 27, 1979.
А.с. № 705443. Червяков Н.И., Лисунов В.В. Устройство для перевода числа, представленного в системе остаточных классов, в полиадическую систему счисления. А.С. № 705443, БИ № 47, 1979.
А.с. № 813403. Червяков Н.И., Зайцев А.Н. Преобразователь кодов из системы остаточных классов в двоичный позиционный код. А.С. № 813403,

Б.И. № 10, 1981.
А.с. № 813408. Червяков Н.И., Иванов П.В. и др. Новые подходы к решения автономных и неавтономных π- задач // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с
А.с. № 855659. Червяков Н.И. Сумматор по модулю. А.С. № 855659, БИ № 30, 1981.
А.с. № 951305. Червяков Н.И. Устройство для округления числа в системе остаточных классов. А.С. № 951305, БИ № 30, 1982.
А.с. № 1142827. Червяков Н.И., Швецов Н.И, Хлевной С.Н. Устройство для определения позиционных характеристик непозиционного кода. А.С. № 1142827, БИ № 8, 1985.
А.с. № 1383365. Червяков Н.И., Финько О.А., Пальцев А.В. Устройство для свертки по модулю. А.С. № 1383365, БИ № 11, 1988.
А.с. № 1743002. Червяков Н.И., Оленев А.А. Устройство для обработки информации, представленной в системе остаточных классов. А.С. № 1743002, Б.И. № 23, 1992.
А.с. № 377771. Червяков Н.И. Сумматор в системе остаточных классов. А.С. № 377771, БИ № 18, 1973.
А.с. № 610102. Червяков Н.И. Устройство для перевода числа, представленного в системе остаточных классов в полиадическую систему счисления. А.С. № 610102, БИ № 21, 1978.
А.с. № 781812. Червяков Н.И., Колесницкий С.В. Устройство для выравнивания порядков чисел, представленных в системе остаточных классов. А.С. № 781812, БИ № 43, 1980.
А.с. № 798846. Червяков Н.И., Лисунов В.В. Устройство для обнаружения ошибок в информации, представленной в системе остаточных классов. А.С. № 798846, БИ № 3, 1981.
А.с. № 991407. Червяков Н.И., Болтков А.Н. Устройство для сопряжения АС. № 991407, БИ № 3, 1983.
А.с. №1173558. Червяков Н.И., Швецов Н.И, Хлевной С.Н., Вершков НА. Непозиционный цифро-аналоговый преобразователь. А.С. №1173558, БИ № 30, 1985.
А.с. №1732353 СССР. Устройство для вычисления ДПФ / Л.Н. Василевич, И.И. Гунько, А.А. Коляда // Откр. Изобр. – 1992. – №17.
А.с. №541164. Червяков Н.И., Колесницкий С.В. Устройство для сравнения чисел. А.С. №541164, БИ № 48, 1976.
А.С. И 66116 (СССР). Устройство для коррекции ошибок в непозиционном аддитивном коде // Ин-т матем. и мех. АН КазССР. Авт. изобрет. И.Т. Пак, С.А. Инютин.
А.с. 1084799. Черкасский Н.В., Митьков В.С., Аксарин Л.Л. Устройство для формирования остатка по модулю три. А.с. 1084799 СССР// Б. И. 1984. № 13. С. 177.
А.с. 1197090, Евстигнеев В.Г., Хлевной С.Н. Устройство для определения ранга числа. Авторское свидетельство СССР. – № 1197090, Б.И. 1985,

№ 45, G06F11/08.
А.с. 1259487 СССР, МКИ 4 Н 03 М 1/28. Преобразователь перемещения в код системы остаточных классов/ С.Н. Хлевной, О.А. Финько // Открытия. Изобрет. — 1986. — №35.
А.с. 1273925, Евстигнеев В.Г. S-ичный сумматор. Авторское свидетельство СССР. – № 1273925, Б.И. 1986, № 44, G06F7/72.
А.с. 1275440, Евстигнеев В.Г., Кошарновский А.Н., Евстигнеева О.В., Канаев А.Е. Устройство для умножения. Авторское свидетельство СССР. – № 1275440, Б.И. 1986, № 45, G06F7/72.
А.с. 1280624, Евстигнеев В.Г., Кошарновский А.Н., Маркин А.В Устройство для умножения чисел с плавающей запятой. Авторское свидетельство СССР. – № 1280624, Б.И. 1986, № 48, G06F7/72.
А.с. 1290315, Евстигнеев В.Г., Кошарновский А.Н., Новожилов А.С. Арифметическое устройство в системе остаточных классов. Авторское свидетельство СССР. – № 1290315, Б.И. 1987, № 6, G06F7/72.
А.с. 1305684. Черкасский Н.В. Устройство для формирования остатков по модулю. А.с. 1305684 СССР// Б. И. 1987. № 15. С. 250.
А.с. 1335998, Евстигнеев В.Г., Кошарновский А.Н., Свириденко В.А., Титов А.П. Устройство для умножения в системе остаточных классов. Авторское свидетельство СССР. – № 1335998, Б.И. 1987, № 33, G06F7/72.
А.с. 1339566. Черкасский Н.В., Титков В.Н. Устройство для формирования остатка по модулю $m=2^K+1$. А.с. 1339566 СССР// Б. И. 1987. № 35. С. 173-174.
А.с. 1343553 СССР, МКИ 4 Н 03 М 7/18. Преобразователь кода системы остаточных классов в позиционный код/ Н.И. Червяков, О.Е. Коршунов, О.А. Финько// Открытия. Изобрет. — 1987. — № 37.
А.с. 1361557, Музыченко О.Н. Устройство для контроля параллельного двоичного кода по модулю К. А.с. 1361557 СССР // Б. И. 1987. № 47. С. 202.
А.с. 1368989 СССР, МКИ 4 Н 03 М 1/28. Аналого-цифровой преобразователь в код системы остаточных классов/ О.А. Финько и др. // Открытия. Изобрет. — 1988. — № 3.
А.с. 1372620 СССР, МКИ 4 Н 03 М 1/28. Аналого-цифровой преобразователь в системе остаточных классов/ О.А. Финько и др. // Открытия. Изобрет. — 1988. — № 5.
А.с. 1381497, Евстигнеев В.Г., Марковский А.Д., Меликов Г.Г. и др. Устройство для извлечения квадратного корня. Авторское свидетельство СССР. – № 1381497, Б.И. 1988, № 10, G06F7/552.
А.с. 1383337, Евстигнеев В.Г., Маркин А.В., Кошарновский А.Н., Кузьмина Г.Ф. и др. Устройство для вычисления функции табличным методом. Авторское свидетельство СССР. – № 1383337, Б.И. 1988, № 11, G06F7/38.
А.с. 1383365 СССР, МКИ 4 G 03 F 11/10. Устройство для свертки по модулю/ Н.И. Червяков, Н.И. Швецов, О.А. Финько, А.В. Пальцев // От-

крытия. Изобрет. — 1988. — № 11.
А.с. 1387201. Черкасский Н.В. Устройство для формирования остатков по модулю. А.с. 1387201 СССР// Б. И. 1988. № 13. С. 259-260.
А.с. 1388996 СССР, МКИ 4 Н 03 М 7/18. Преобразователь кода из системы остаточных классов в позиционный код/ Н.И. Червяков, О.Е. Коршунов, О.А. Финько// Открытия. Изобрет. — 1988. — № 14.
А.с. 1401610. Черкасский Н.В. Устройство для формирования остатков по модулю. А.с. 1401610 СССР// Б. И. 1988. № 21. С. 256-257.
А.с. 1418702, Евстигнеев В.Г., Марковский А.Д., Пустовойтов О.И., Меликов Г.Г. и др. Устройство для изменения n-разрядного двоичного числа на единицу. Авторское свидетельство СССР. – № 1418702, Б.И. 1988, № 31.
А.с. 1425676, Музыченко О.Н. Устройство для контроля параллельного двоичного кода по модулю К. А.с. 1425676 СССР // Б. И. 1988. № 35. С. 210.
А.с. 1429322, Музыченко О.Н. Преобразователь двоичного кода в код по модулю К. А.с. 1429322 СССР // Б. И. 1988. № 37. С. 245.
А.с. 1444774, Черкасский Н.В. Устройство для формирования остатков по модулю. А.с. 1444774 СССР// Б. И. 1988. № 46. С. 220.
А.с. 1444961 СССР, МКИ 4 Н 03 М 1/28. Преобразователь числа в модулярный код/ О.А. Финько и др. // Открытия. Изобрет. — 1988. — № 46.
А.с. 1449986. Черкасский Н.В. Устройство для формирования остатков по модулю. А.с. 1449986 СССР// Б. И. 1989. № 1. С. 211.
А.с. 1462306, Евстигнеев В.Г., Кошарновский А.Н., Ермакова Т.Б. S-ичный сумматор. Авторское свидетельство СССР. – № 1462306, Б.И. 1989, № 8.
А.с. 1476614, Музыченко О.Н. Преобразователь двоичного кода. А.с. 1476614 СССР // Б. И. 1989. № 16. С. 250.
А.с. 1487035, Евстигнеев В.Г., Кошарновский А.Н., Краснобаев В.А. Устройство для суммирования по модулю. Авторское свидетельство СССР. – № 1487035, Б.И. 1989, № 22.
А.с. 1543550, Евстигнеев В.Г., Кошарновский А.Н. Преобразователь форматов. Авторское свидетельство СССР. – № 1543550, Б.И. 1990, № 6.
А.с. 1557682 СССР, МКИ 4 Н 03 М 7/18. Преобразователь позиционного кода в код системы остаточных классов/ О.А. Финько и др.// Открытия. Изобрет. — 1990. — № 14.
А.с. 1603371, Музыченко О.Н. Сумматор по модулю семь. А.с. 1603371 СССР // Б. И. 1990. № 40. С. 210.
А.с. 1603375, Музыченко О.Н. Сумматор по модулю пятнадцать. А.с. 1603375 СССР // Б. И. 1990. № 40. С. 212-215.
А.с. 1674121. Червяков Н.И., Куценко С.М., Фисаков А.Н. Устройство для определения знака числа, представленного в системе остаточных классов. А.С. 1674121, БИ № 32, 1991.
А.с. 1732472, Музыченко О.Н. Преобразователь двоичного кода в код по

модулю К. А.с. 1732472 СССР // Б. И. 1992. № 17. С. 227.
А.с. 1736006. Черкасский Н.В. Устройство для формирования остатков по модулю. А.с. 1736006 СССР// Б. И. 1992. № 19. С. 228-229.
А.с. 2235423, Оцоков Ш.А. Устройство для преобразования числа системы остаточных классов в позиционный код. Патент РФ № 2235423, Бюл. №24, 2004.
А.с. 605209, Факторович М.Г., Полисский Ю.Д..Устройство для перевода чисел из системы остаточных классов в полиадическую. Авт. свид. № 605209, М. Кл.^{2} G06 F 5/02, 1978.
А.с. 693367, Факторович М.Г., Полисский Ю.Д.. Устройство для сравнения чисел. Авт. свид. № 693367, М. Кл.^{2} G06 F 7/04, 1979.
А.с. 962942, Евстигнеев В.Г., Новожилов А.С., Белова Р.С., Сведе-Швец В.Н. Устройство для умножения в системе остаточных классов. Авторское свидетельство СССР. – № 962942, Б.И. 1982, № 36, G06F7/72.
А.с. 968800, Евстигнеев В.Г., Новожилов А.С., Сведе-Швец В.Н. Устройство для формирования позиционных признаков непозиционного кода. Авторское свидетельство СССР. – № 968800, Б.И. 1982, № 39, G06F5/02.
Айдарханов М.Б., Бияшев Р.Г., Горковенко Е.В., Сахариев Б.Б., Кулемзин А.А., Чебейко С.В., Шахмаев Р.А. Парирование угроз безопасности корпоративных информационных ресурсов. – Алматы, «Гылым», 2003 – 104 с.
Акаев А.А., Майоров С.А. Оптические методы обработки информации. – М.: Высш. шк., 1988. – 237 с.
Акушский И. Я-, Бурцев В. М., Пак И. Т. Алгоритмы деления с использованием ядерной характеристики // Теория кодирования и оптимизации сложных систем. Алма-Ата: Наука, 1977. С 26 — 33.
Акушский И. Я-, Бурцев В. М., Пак И. Т. Вычисление позиционной характеристики (ядро) непозиционного кода//Теория кодирования и оптимизации сложных систем. Алма-Ата: Наука, 1977. С. 17—25.
Акушский И. Я-, Пак И. Т. Вопросы помехоустойчивого кодирования в непозиционном коде//Вопр. кибернетики. 1977. Т. 28. С. 36 — 56.
Акушский И. Я-, Пак И. Т. Вопросы помехоустойчивого кодирования в непозиционном коде//Вопр. кибернетики. 1977. Т. 28. С. 36 — 56.
Акушский И. Я., Амербаев В. М., Пак И. Т. Основы машинной арифметики комплексных чисел. Алма-Ата: Наука, 1970. 248 с.
Акушский И. Я., Бурцев В. М., Пак И. Т. О новой позиционной характеристике непозиционного кода и ее применении//Теория кодирования и оптимизация сложных систем. Алма-Ата: Наука. 1977. С. 8— 16.
Акушский И. Я., Хацкевич В. Х. *Инверсные представления чисел в системе остаточных классов.* Сб. «Цифровая вычислительная техника и программирование, вып. 2», Москва, 1967.
Акушский И.Я., Амербаев В.М., Пак И.Т. Основы машинной арифметики

комплексных чисел. – Алма-Ата: Наука, 1970. – 248 с.
Акушский И.Я., Хацкевич В.Х. *О безранговых непозиционных представлениях чисел для одного класса оснований.* Сб. «Математическая и техническая кибернетика», Тб. Мецниереба, 1975.
Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. – М.: Советское радио, 1968, 440с.
Амербаев В. М. Теоретические основы машинной арифметики. — Алма-Ата: Наука, 1976. 324 с.
Амербаев В.М. Дьячков В.Н. Модулярная арифметика как криптографический примитив. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 187-193.
Амербаев В.М. Пак И.Т. Модулярной арифметике – 50 лет. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 5-22.
Амербаев В.М., Касимов Ю.Ф.. О сравнении чисел в непозиционных системах счисления. // Теория кодирования и оптимизации сложных систем. Алма-Ата: Наука, 1977. С 47 — 54.
Амербаев В.М., Пак И.Т. Параллельные вычисления в комплексной плоскости. – Алма-Ата: Наука, 1984. – 182 с.
Амербаев В.М., Стемпковский А.Л., Широ Г.Э. Модулярный быстродействующий согласованный фильтр. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 243-260.
Ахременко В. Н., Коляда А. А., Кравцов В. К. К вопросу о распределении поправки Амербаева для позиционных характеристик непозиционного кода // Оптимизационные задачи в автоматизированной системе плановых расчетов. Мн.: НИИ ЭМП при Госплане БССР, 1982. С. 128— 135.
Баженов А.А., Овчаренко Л.А., Сизов А.С. Прямой цифровой синтез гармонических колебаний в системе остаточных классов// Физика волновых процессов и радиотехнические системы. – 2003. – №2. – С.21-23.
Байоми М. А. Заказные матрицы СБИС для структур, основанных на системе счисления в остаточных классах // ТИИЭР. - 1987. — Т. 38. — № 12. - С. 134-139.
Балашов Е. П., Негода В. И., Пузанков Д. В. и др. Информационные системы. Табличная обработка информации/ Под. ред. Е. П. Балашова, В. Б. Смочова. - Л.: Энергоиздат, Л.О., 1985. — 184 с.
Баня Е.И.. Применение в вычислительных машинах системы счисления остаточных классов (СОК) с наименьшими по абсолютной величине остатками. Ж."Кибернетика", №3, 1967, с.36-38.
Бережной В.В. Нейросетевая структура для исправления двукратных

ошибок в модулярных нейрокомпьютерах. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 528-539.

Бияшев Р.Г., Горковенко Е.В Шифраторы информации с заданной криптостойкостью Труды международной научно-практической конференции «Состояние, проблемы и задачи информатизации в Казахстане», Издание КазНТУ им.Сатпаева, \ Алматы с.27-37.

Бияшев Р.Г., Горковенко Е.В. Информационная безопасность: состояние и проблемы. // Проффи, №9(1), 2003, с.2-5.

Бияшев Р.Г., Горковенко Е.В. Кодирование и шифрование информации. // Труды 2-й международной конференции. КазНУ им. Аль-Фараби. Алматы, 2004, с 130-134.

Бияшев Р.Г., Горковенко Е.В., Нысанбаева С.Е. Алгоритмы шифрования сообщений и формирования электронной цифровой подписи с заданной криптостойкостью. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 569-579.

Бияшев Р.Г., Нысанбаев Р.К., Егай Р.В. Применение модулярного шифрования в комплексе тестирования абитуриентов. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 580-591.

Блейхут Р. Теория и практика кодов, контролирующих ошибки. – М.: Мир, 1986. – 576 с.

Болкунов А.А., Овчаренко Л.А. Синтез структуры ЭВМ в системе остаточных классов// Электронное моделирование. – 2001. – № 4. – С.116-120.

Болкунов А.А., Овчаренко Л.А. Сравнение вычислительных структур в позиционной и непозиционной системах счисления// Электронное моделирование. – 2000. – № 3. – С.113 – 117.

Борисенко А.А., Губарев С.И., Булаенко С.И.. Оо одном способе преобразования чисел из позиционного кода в систему остаточных классов. Автоматизированные системы управления и приборы автоматики, вып.40, Харьков, 1976.

Борисенко А.А., Завизиступ Ю.Ю., Губарев С.И.. О взамисвязи кодов чисел, выраженных в системе остаточных классов, с их позиционным представлением. Автоматизированные системы управления и приборы автоматики, вып.35, Харьков, 1975.

Борисенко А.А., Завизиступ Ю.Ю., Штец Л.К.. Устройство для преобразования чисел, выраженных в системе остаточных классов, в их позиционное представление. Автоматизированные системы управления и приборы автоматики, вып.34, Харьков, 1975.

Бороденко Е. И., Краснобаев В. А. Контроль и повышение надежности функционирования дискретной техники. - Харьков: МО СССР, 1982. - 108 с.

Бороденко Е.И., Краснобаев В.А. Алгоритм коррекция ошибок в классе вычетов // НТС РТИ. Рязань. 1983. С. 52 –55.

Брусенцов Н.П. Заметки о троичной цифровой технике. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 611-634.

Брусенцов Н.П. Неадекватность двоичной информатики. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 607-610.

Буза М. К., Коляда А. Л. Некоторые исследования безранговых систем в остаточных классах//Там же. 1970. № 2. С. 14 — 19.

Василевич Л.Н., Коляда А.А. Структура арифметических устройств модулярных процессоров БПФ конвейерного типа // Электронное моделирование. 1989, Т. 2, № 6. С. 15-20.

Василевич Л.Н., Коляда А.А., Ревинский В.В. Высокоскоростная модулярная реализация адаптивных цифровых фильтров с конечной импульсной характеристикой // Весці АН Беларусі. Сер. фіз.-мат. навук. 1997. №1. С. 126 – 131.

Василевич Л.Н., Коляда А.А., Селянинов М.Ю., Чернявский А.Ф. Модулярные принципы построения процессоров для дискретного преобразования Фурье // Весці НАН Беларусі. Сер. фіз.- тэхн. навук. – 2001. – № 4. – С.108–117.

Вышинский В.А.. Некоторые алгоритмы преобразования чисел из позиционной системы счисления в систему остаточных классов. – В кн.: Семирар “Вопросы теории электронных цифровых математических машин.”, Вып.3, К. “Наукова думка”, 1967, с.3-16 (АН УССР, Науч. совет по кибернетике. Институт кибнрнетики).

Габашвили М. В., Хацкевич В. Х. К вопросу о матричной арифметике непозиционных систем счисления//Сообщ. АН Груз ССР. 1971. Т. 61, № 2. С. 405 — 408.

Габашвили Н. В., Хацкевич В. Х. О контроле по модулю машинных операций над псевдодвоичными словами. «Тр. Проблем. лабор. автоматики и вычислительной техники ГПИ», Тбилиси, 1971.

Габашвили. Н. В., Хацкевич В. Х. *К вопросу о мультипликативных характеристиках непозиционных систем*. «Тр. Проблем. лабор. автоматики и вычислительной техники ГПИ», Тбилиси, 1971.

Гаврилов Ю.В., Пучко А.М. Арифметические устройства быстродействующих ЭВМ.-М.: Сов. Радио, 1970.

Гамбереер Д. Высокоскоростные алгоритмы маршрутизации, основанные на целочисленной арифметике в остаточных классах// Передача ин-

формации. Экспресс-информация. 1988. № 21. С. 22—26.
Гегелия Г. Д., Хацкевич В. Х. *Об одном методе построения систем автоматического контроля*. «Тр. III научно-технической конференции по надежности», Ленинград, 1970.
Дадаев Ю. Г. Теория арифметических кодов. М.: Радио и связь. 1981. 272 с.
Дадаев Ю.Г. Арифметические коды, исправляющие ошибки. – М.: Советское радио, 1968. – 168 с.
Дадаев Ю.Г. Теория арифметических кодов. – М.: Радио и связь, 1981. – 272 с.
Дзегеленок И. И., Оцоков Ш. А. О распараллеливании безошибочных вычислений на ПМК-сети „Курс-2000" // Вычислительные сети. 2003. № 1.
Дзегелёнок И.И, Оцоков Ш.А. Подход к решению проблемы безошибочных вычислений с использованием ускоренного алгоритма отображения дробей Фарея. // Труды научной конференции, посвященной 75-летию со дня рождения академика В.А.Мельникова. РАН. М. 2004.
Дзегелёнок И.И. , Оцоков Ш.А. О распараллеливании безошибочных вычислений на ПМК-сети КУРС 2000 // Электронный журнал. Вычислительные сети. Теория и практика. http://network-journal.mpei.ac.ru.
Дзегелёнок И.И. , Оцоков Ш.А. Экспериментальное исследование модели безошибочных вычислений на ПМК-сети КУРС 2000 // Сб. трудов международной научной конференции «Информационные средства и технологии» М.:МЭИ (ТУ), 2003.- С.103-106.
Дзегеленок И.И., Оцоков Ш.А. Метод ускорения модулярной арифметики с самоисключением ошибок округления. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 355-361.
Долгов А.И. Диагностика устройств, функционирующих в системе остаточных классов. – М.: Радии и связь, 1982. – 64 с.
Долгов В.И.,Краснобаев В.А.,Кононова И.В. Метод и алгоритмы реализации арифметических операций в системе остаточных классов // Электрон.моделирование. 1989. № 5. С. 15 – 18.
Евдокимов А. А. Безопасность реализации криптографического нейросопроцессора на ПЛИС// Физико-математические науки: Материалы 49-й науч.-метод. конф. преподавателей и студентов «Университетская наука региону». – Ставрополь: Изд-во СГУ, 2004. – С. 184 – 185.
Евдокимов А. А. Согласованность геометрических моделей системы остаточных классов и нейронной сети CMAC// Труды участников международной школы-семинара по геометрии и анализу памяти Н. В. Ефимова. – Ростов-на-Дону: Изд-во ООО «ЦВВР», 2004. – С. 188 – 190.
Евдокимов А.А. Реализация модулярных нейронных вычислительных структур на базе ПЛИС. // Труды Юбилейной Международной научно-

технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 377-388.

Евдокимов А.А. Свидетельство об отраслевой регистрации разработки № 4086 от 03 декабря 2004 г. на разработку «Описания нейронной сети конечного кольца «VHDL»». Номер гос. регистрации 50200401423 от 09 декабря 2004 г. (Министерство образования РФ. Государственный координационный центр информационных технологий. Отраслевой фонд алгоритмов и программ)

Евдокимов А.А. Свидетельство об отраслевой регистрации разработки № 4088 от 03 декабря 2004 г. на разработку «Описание устройства для перевода чисел из системы остаточных классов в обобщенную позиционную систему счисления «VHDL»». Номер гос. регистрации 50200401425 от 09 декабря 2004 г. (Министерство образования РФ. Государственный координационный центр информационных технологий. Отраслевой фонд алгоритмов и программ).

Евдокимов А.А. Свидетельство об отраслевой регистрации разработки № 4087 от 03 декабря 2004 г. на разработку «Описание устройства для преобразования чисел из позиционного представления в систему остаточных классов «VHDL»». Номер гос. регистрации 50200401424 от 09 декабря 2004 г. (Министерство образования РФ. Государственный координационный центр информационных технологий. Отраслевой фонд алгоритмов и программ)

Евстигнеев В. Г. Компьютерные арифметики. Ретроспективный взгляд // Электроника, наука, технология, бизнес. 1998. — № 2. С. 19-22.

Евстигнеев В. Г. Недвоичная машинная арифметика и специализированные процессоры. - М.: МИФИ СЕРВИС, 1992. - 266 с.

Евстигнеев В. Г., Горская В. В., Филиппова Н. В. Некоторые вопросы масштабирования при решении задач в СОК в неизбыточном арифметическом диапазоне // Науч. тр. по проблемам микроэлектроники Моск. ин-та электронной техники. 1972. Вып. 9. С. 200 — 212.

Евстигнеев В.Г. Недвоичная машинная арифметика и специализированные процессоры. – М.: МИФИ СЕРВИС и АО «ИНСОФТ», 1992.

Евстигнеев В.Г. Недвоичные компьютерные арифметики. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, , тираж 150 экз., с 178-186.

Евстигнеев В.Г., Краснобаев В.А., Бороденко Е.И. и др. Функциональные преобразователи для системы остаточных классов // Радиотехника. 1984. Вып. 69. С. 57 -60.

Евстигнеев В.Г., Сведо –Швец А.В., Краснобаев В.А. Реализация арифметических операций и помехоустойчивое кодирование в позиционно-остаточной системе счисления // Радиотехника. 1983. Вып. 66. С. 13 - 16.

Евстигнеев В.Г. S-ичный сумматор. – Электронная техника, серия 10, выпуск 5(59), 1986.
Евстигнеев В.Г. Арифметика с плавающей запятой в позиционных системах с большим основанием. – Оборонная техника. Научно-технический сборник, выпуск 12, 1985.
Евстигнеев В.Г. Арифметические сопроцессоры. – Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 1(57), 1988.
Евстигнеев В.Г. Быстродействие и аппаратная сложность сумматоров и умножителей. – Оборонная техника. Научно-технический сборник, выпуск 12, 1985.
Евстигнеев В.Г. КОМПЬЮТЕРНЫЕ АРИФМЕТИКИ. Ретроспективный взгляд. –«ЭЛЕКТРОНИКА: Наука, Технология, Бизнес», 2/98.
Евстигнеев В.Г. Недвоичная машинная арифметика и специализированные процессоры. Под редакцией И.Я. Акушского. Москва, МИФИ СЕРВИС, 1992 г., 266 с. Тираж 5000 экз.
Евстигнеев В.Г. О возможности применения кода СОК для построения комбинаторских переключателей. – Теория кодирования и информационное моделирование, Наука, Алма-Ата, 1973.
Евстигнеев В.Г. Об архитектуре отказоустойчивой БЦВМ на основе позиционно-остаточной системы счисления. – Всесоюзная школа по проблемам математического обеспечения и архитектуры бортовых вычислительных систем, г. Ташкент, 1988.
Евстигнеев В.Г. Повышение производительности ЭВМ за счет отказа от двоичной системы счисления. – Второе Всесоюзное совещание по автоматизированному проектированию программного обеспечения систем управления движущимися объектами, г. Харьков, 1989.
Евстигнеев В.Г. Позиционно-остаточная система счисления для быстродействующих ЭВМ. – Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 1(41), 1984.
Евстигнеев В.Г. Преобразование чисел с плавающей запятой в формат с большим основанием и обратно. – Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 4(64), 1989.
Евстигнеев В.Г. Синтез S-ичных сумматоров. – Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 1(57), 1988.
Евстигнеев В.Г. Умножение S-ичных цифр в позиционно-остаточной системе счисления. – Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 2(42), 1984.
Евстигнеев В.Г. Умножение и деление позиционно-остаточных чисел, представленных в естественной форме. – Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 2(42), 1984.
Евстигнеев В.Г. Устройства для сложения позиционно-остаточных чисел, представленных в естественной форме. – Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 1(41), 1984.
Евстигнеев В.Г. Цифровые фильтры в недвоичных системах счисления. –

Вопросы оборонной техники. Научно-технический сборник, серия VIII, выпуск 3(47), 1985.
Евстигнеев В.Г., Акушский И.Я. Некоторые вопросы структуры специализированных ЦВМ. – Теория кодирования и информационное моделирование, Наука, Алма-Ата, 1973.
Евстигнеев В.Г., Бороденко Е.И., Краснобаев В.А. Коррекция ошибок в системе остаточных классов со взаимно непростыми основаниями. – Радиотехника, выпуск 67, Республиканский межвед. сборник г. Харьков, «Высшая школа», 1983.
Евстигнеев В.Г., Бороденко Е.И., Краснобаев В.А. Табличная реализация модульных операций в классе вычетов. – Республиканский межведомственный научно-технический сборник ХИРЭ, выпуск 64, 1983 г., «Высшая школа».
Евстигнеев В.Г., Бороденко Е.И., Краснобаев В.А. Табличная реализация операций модульного сложения в системе остаточных классов. – Радиотехника, выпуск 68. Республиканский межвед. сборник г. Харьков, «Высшая школа», 1984.
Евстигнеев В.Г., Бороденко Е.И., Сведе-Швец В.Н., Краснобаев В.А. Функциональные преобразователи для системы остаточных классов. – Радиотехника, выпуск 69, Республиканский межвед. сборник г. Харьков, «Высшая школа», 1984.
Евстигнеев В.Г., Евстигнеева О.В. Позиционно-остаточное представление чисел для повышения скорости выполнения арифметических операций в ЭВМ. – Всесоюзная конференция «Пути повышения эффективности создания ГАП в приборостроении и микроэлектронике», Москва, 1985, выпуск 1, с.27-28.
Евстигнеев В.Г., Королев А.В., Краснобаев В.А. Алгоритмы модульных операций в классе вычетов. – ХВВКУ. Научно-технический сборник, № 327, 1981.
Евстигнеев В.Г., Королев А.В., Краснобаев В.А. Доклад на спец. тему. – ПВКУ, научно-техническая конференция, 1981.
Евстигнеев В.Г., Королев А.В., Лысенко И.Э. Арифметические алгоритмы и система команд ЦВМ в СОК. – Тематический сборник научных трудов ХАИ, выпуск 4, 1982.
Евстигнеев В.Г., Кошарновский А.Н. Байтовый сумматор в системе остаточных классов. – Вопросы оборонной техники, Научно-технический сборник, серия VIII, выпуск 3(39), 1983.
Евстигнеев В.Г., Кошарновский А.Н., Новожилов А.С. Архитектура комплекта микропроцессорных БИС на основе недвоичной арифметики. – Отраслевая научно-техническая конференция МЭП «Схемотехнические и материаловедческие решения сверхскоростных схем с высокой степенью интеграции», 15-16 апреля, Москва, 1986.
Евстигнеев В.Г., Кошарновский А.Н., Новожилов А.С. Методы ускорения выполнения арифметических операций в современных ЦВМ. – Вопро-

сы оборонной техники, Научно-технический сборник, серия VIII, выпуск 2(38), 1983.
Евстигнеев В.Г., Кошарновский А.Н., Новожилов А.С. Системы счисления для ЭВМ. – Вопросы оборонной техники, Научно-технический сборник, серия VIII, выпуск 3(39), 1983.
Евстигнеев В.Г., Краснобаев В.А. Алгоритмы модульных операций сложения-вычитания в системе остаточных классов. – Академия им. Дзержинского. Сборник трудов. Москва, 1982.
Евстигнеев В.Г., Краснобаев В.А. Оптимизация выбора оснований системы остаточных классов. – ХВВКУ. Сборник трудов № 328, 1983.
Евстигнеев В.Г., Краснобаев В.А., Сведе-Швец В.Н. Арифметические алгоритмы для q-ичной системы счисления. – Тематический сборник научных трудов ХАИ, выпуск 4, 1982.
Евстигнеев В.Г., Краснобаев В.А., Сведе-Швец В.Н. Построение узлов и блоков ЦВМ, работающих в системе остаточных классов, с применением элементов оптоэлектронной техники. – ХВВКУ. Научно-технический сборник, № 327, 1981.
Евстигнеев В.Г., Кузьмина Г.Ф., Сафонов Е.Н. Байтовый процессорный элемент. – Вопросы оборонной техники, Научно-технический сборник, серия VIII, выпуск 3(39), 1983.
Евстигнеев В.Г., Розанов Л.А., Новожилов А.С. Обнаружение и исправление ошибок арифметических преобразований позиционно-остаточного кода. – Вопросы оборонной техники, Научно-технический сборник, серия VIII, выпуск 2(38), 1983.
Евстигнеев В.Г., Романов Л.Г. Комбинаторные матричные переключатели в системе остаточных классов. – Электронная техника, серия III, вып.35(39), 1972.
Евстигнеев В.Г., Романов Л.Г. Эквивалентные преобразования комбинаторских матричных переключателей в СОК. – Сборник ВИМИ «РИПОРТ» № 4, 1975.
Евстигнеев В.Г., Филиппова Н.В., Пьянзин А.Я. Симметрические трехточечные слабопозиционные системы. – Труды отдела экономических исследований БФСОАН СССР, 1971, 5 (9).
Жихарев В.Я., Илюшко Я.В., Кравець Л.Г., Краснобаев В.А. Методы и средства обработки информации в непозиционной системе счисления в остаточных классах. – Житомир: Изд-во “Волынь”, 2005. – 220с.
Жихарев В.Я., Илюшко Я.В., Краснобаев В.А. Влияние системы счисления на надежность ЭВМ. // Радіоелектронні і комп’ютерні системи. – 2004. - № 1(5).-С. 98 – 104.
Жихарев В.Я., Юнес Эль Хандасси, Краснобаев В.А. Методы и алгоритмы реализации арифметических операций в классе вычетов // Открытые информационные и компьютерные интегрированные технологии. – Х.: НАКУ (ХАИ). – 2003. – Вып. 20. – С. 84-101.
ЖихаревВ.Я., Юнес Эль Хандасси, Краснобаев В.А. Пути повышения

производительности и отказоустойчивости ЭВМ // Открытые информационные и компьютерные интегрированные технологии. – Х.: НАКУ (ХАИ). – 2003. – Вып. 19. – С. 269-282.
Журавлев Ю.П., Кателюк Л.А., Циклинский Н.И. Надежность и контроль ЭВМ. М.: Радио и связь, 1978.
Зольников В.К., Машевич П.Р. Логическая оптимизация блоков микропрограммного управления СБИС. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 423-433.
Зольников В.К., Машевич П.Р. Структурная декомпозиция блоков микропрограммного управления. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 407-422.
Инютин С. А. Алгоритм с линейной сложностью в помехозащитном модулярном кодировании // Сб. научных статей: системный анализ и обработка информации, вып. 2. –Сургут: РИО, 2004. -11с.
Инютин С. А. Параллельные вычисления в сверхбольших компьютерных диапазонах // I Междунар. конф. „Параллельные вычисления и задачи управления" (PACO-2001). Москва, 29—31 января 2001. Сборник трудов. — М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2001. - С. 76-87.
Инютин С. А. Проблема метрик в модулярном помехозащитном кодировании // Труды СурГПИ, вып. 2. –Сургут: РИО, 2004. -9с.
Инютин С. А. Проблема оценки вычислительной сложности алгоритмов //Проблемы информатизации нефтегазового комплекса. –Тюмень: ТГНГУ, 2004. -17с.
Инютин С.А. Алгоритмы синдромного и совмещенного декодирования арифметических модулярных кодов / PACO-2004. 2-ая международная конференция. Параллельные вычисления и задачи управления: сборник трудов. –М.: ИПУ РАН, 2004. –С. 1208 -1214.
Инютин С.А. Арифметико-логические основы вычислительных систем. - Сургут: РИО, 2001. –120с.
Инютин С.А. Вычислительные задачи большой алгоритмической сложности и модулярная арифметика // Вестник Тюменского госуниверситета. –Тюмень: -2002 ,- № 3. –с. 3-9.
Инютин С.А. Компьютерная модулярная алгебра квадратичного диапазона и область ее приложения // Вестник Тюменского госуниверситета. – Тюмень: -2001 , - № 2. –с. 141-148.
Инютин С.А. Модулярные вычисления в сверхбольших компьютерных диапазонах // Известия вузов. Электроника. -2001, -№ 6. –с. 34-39.
Инютин С.А. Основы многоразрядной алгоритмики. -Сургут: РИО, 2002. -137с.

Инютин С.А. Помехозащитные модулярные кодовые конструкции квадратичного диапазона // Вестник Тюменского госуниверситета. – Тюмень: -2003 ,- № 5. –с. 173-180.
Инютин С.А. Модулярные вычисления в сверхбольших компьютерных диапазонах // Известия вузов. Электроника. –2001. – № 6. – с. 34–39.
Инютин С.А. Модулярные вычисления для задач большой алгоритмической сложности. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 211-217.
Ирхин В.П. Алгоритмы определения позиционных характеристик кодов в системе остаточных классов. – Харьков: ХВКИУРВ. Тематический научно-технический сборник № 337, 1992, с. 33-36.
Ирхин В.П. Алгоритмы реализации операций модулярной арифметики. – Воронеж: Воронежская высшая школа МВД России. Прикладные вопросы защиты информации: Сборник статей / Под ред. С.В. Бухарина, 1996, с. 12-16.
Ирхин В.П. Алгоритмы табличной реализации остаточной арифметики. – Воронеж: ВИРЭ. Труды института: Методические основы развития способов и средств радиоэлектронной борьбы, 1995, с. 37-39.
Ирхин В.П. Проектирование непозиционных специализированных процессоров. – Воронеж: Издательство Воронежского государственного университета, 1999. – 136 с.
Ирхин В.П. Проектирование непозиционных специализированных процессоров. Воронеж: Изд. Воронеж. ун-та, 1999. 136 с.
Ирхин В.П. Табличная реализация операций модулярной арифметики. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 261-266.
Ирхин В.П. Улучшение основных характеристик операционных устройств спецпроцессоров. – Харьков: ХВКИУРВ. Тематический научно-технический сборник № 337, 1992, с. 31-33.
Ирхин В.П., Коровин В.М. Разработка тренажеров на базе непозиционных вычислительных устройств. Научно-методический сборник МО РФ № 49. - М., Воениздат, 2000, с. 94-98.
Ирхин В.П., Табуненко В.А. Алгоритм приведения двоичного числа по простому модулю: Харьков: НАНУ, ПАНИ, ХВУ. Системы информационного взаимодействия. Сб. научн. тр., 1996, с. 43-46.
Ирхин В.П., Табуненко В.А. Алгоритмы формирования исходного состояния содержимого кольцевых регистров сдвига. – Харьков, НАНУ, ПАНИ, ХВУ. Системы информационного взаимодействия, 1995, с. 23-26.
Ирхин В.П., Табуненко В.А. Использование остаточной арифметики при табличных методах обработки – Харьков, НАНУ, ПАНИ, ХВУ. Обра-

ботка информации. Сб. научн. трудов, 1996. – с 29-35.
Исмаилов Ш.А., Оцоков Ш.А. Разрядно-параллельный алгоритм и структура преобразования чисел из позиционной системы счисления в систему остаточных классов // Вестник. ДНЦ РАН. - 2001. - № 9.- С. 40-43.
Казангапов А.Н. и др. Волоконная оптика в измерительной и вычислительной технике / Алма-Ата: Наука, 1989, 245с.
Калашников В.С. Основные виды архитектур модулярных сумматоров для двух операндов // Микроэлектроника и информатика-2004. Одиннадцатая всероссийская межвузовская конференция студентов и аспирантов: Тезисы докладов, М.:МИЭТ, 2004.
Калашников В.С. Принципы построения двоичных и модулярных мультиоперандных сумматоров // Микроэлектроника и информатика-2005. Одиннадцатая всероссийская межвузовская конференция студентов и аспирантов: Тезисы докладов, М.:МИЭТ, 2005.
Калмыков И.А. Математическая модель нейронной сети для исправления ошибок непозиционного кода поля Галуа в частотной области/ «Нейрокомпьютеры: разработка и применение» 2004, №5-6, с.71-78;
Калмыков И.А. Разработка метода контроля и коррекции ошибок для непозиционного спецпроцессора с деградируемой структурой/Збірник наукових праць 2004, Київ, Національна Академія Наук Укріни, Выпуск № 25, с. 65-78.
Калмыков И.А., Бережной В.В. Многоступенчатая полиномиальная система классов вычетов в расширенных полях Галуа и ее нейросетевая реализация/Вестник Ставропольского Государственного Университета, 2004, Выпуск № 38 с. 16-24.
Калмыков И.А., Червяков Н.И., Щелкунова Ю.О., Бережной В.В. Архитектура отказоустойчивой нейронной сети для цифровой обработки сигналов/Нейрокомпьютеры: разработка, применение. №12, 2004. с.51-60.
Калмыков И.А., Червяков Н.И., Щелкунова Ю.О., Бережной В.В., Шилов А.А. Нейросетевая реализация в полиномиальной системе классов вычетов операций ЦОС повышенной разрядности/ Нейрокомпьютеры: разработка и применение, 2004, №5-6, с.94-101.
Калмыков И.А., Щелкунова Ю.О., Гахов В.Р., Горденко Д.В., Новиков В.И. Модель и структура нейронной сети для реализации ЦОС в расширенных полях Галуа/Збірник наукових прац. Выпуск 1. «Системи обработки інформації», 2003, С.29-41.
Калмыков И.А., Щелкунова Ю.О., Гахов В.Р., Шилов А.А. Математическая модель коррекции ошибок в полиномиальной системе класса вычетов на основе определения корней интервального полинома/Волновые процессы. №5, т.6, Самара, 2003 – С.30-34.
Касами Т., Токура Н., Ивадари Ё., Инагаки Я. Теория кодирования. М.: Мир, 1978.
Кейр И. А., Чини П. В., Таненбаум М. Деление и определение переполне-

ния в системах счисления остаточных классов//Кибернет. сб. М.: Мир, 1964. Вып. 8. С. 166— 178.
Клязник В. В., Ласкеев С. К- Применение системы остаточных классов при построении цифровых фильтров //Вычислительные средства в технике и системах связи. 1978. № 3. С. 69 — 73.
Кнут Д. Э. Искусство программирования, том 2. Получисленные алгоритмы, 3-е из. — М.: Издательский дом „Вильяме", 2000.
Коломейк.0 В. В., Петущак В. Д. Вопросы упрощения немодульных операций в специализированных ЭВМ, работающих в ССОК // Кибернетика. 1986. № 4. С. 104— 106.
Коляда А. А. Алгоритмы арифметики обобщенных СОК//Там же. 1980. № 1. С. 6— 12.
Коляда А. А. Интервально-модулярные коды с исправлением ошибок//Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1988. № 2. С. 33 — 36.
Коляда А. А. Метод знаковых чисел для формирования интегральных характеристик модулярного кода// Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1987. № 1. С. 3 — 5.
Коляда А. А. О нормированном ядре числа в системах остаточных классов и его вычислении//Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1983. № 3. С. 12— 16.
Коляда А. А. О структуре интегральных характеристик модулярного кода//Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1986. № 1. С. 46 — 49.
Коляда А. А. О ядре числа в системах остаточных классов// Кибернетика. 1982. № 2. С. 123— 125.
Коляда А. А. Обобщенные СОК//Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1972. № 3. С. 20 — 23.
Коляда А. А. Определение знака числа в обобщенных СОК// Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1976. № 1. С. 12 — 17.
Коляда А. А. Структура быстродействующих АУ в обобщенных СОК//Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1981. № 1. С. 19 — 25.
Коляда А. А. Умножение в обобщенных СОК// Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1981. № 3. С. 3 — 8.
Коляда А. А. Умножение чисел разных знаков в ядерно-модулярном коде//Вестн. Белор. ун-та. Сер. 1. Физ. Мат. Мех. 1983. № 2. С. 15— 18.
Коляда А. А., Кравцов В. К. О некоторых вопросах реализации арифметики обобщенных СОК// Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1981. № 2. С. 3—7.
Коляда А. А., Кравцов В. К. Об одном методе формирования позиционных характеристик непозиционного кода//Междунар. конф. «Мат. методы в исследовании операций» (София, Болгария, 24 — 29 окт., 1983): Тез. София, 1983. С. 39.
Коляда А. А., Пак И. Т. Модулярные структуры конвейерной обработки

цифровой информации. — Мн.: Университетское, 1992. — 256 с.
Коляда А. А., Пилиповец Ф. С. О нахождении оснований систем остаточных классов//Теория и применение мат. машин. Мн.: Изд-во БГУ им. В. И.Ленина, 1972. С. 16—28.
Коляда А. А., Пилиповец Ф. С. О распределении поправки Амербаева к неточному рангу числа в системах остаточных классов // Вести. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1982. № 2. С. 53—56.
Коляда А. А., Ревинский В. В. Сложение и вычитание в обобщенных СОК//Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1972. № 3. С. 75 — 76.
Коляда А. А., Селянинов М. Ю. Нормализация чисел в модулярной системе счисления// Вести. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1988. № 1. С. 50 — 53.
Коляда А. А., Селянинов М. Ю. О формировании интегральных характеристик кодов систем в остатках с симметричным диапазоном// Кибернетика. 1986. № 4. С. 20 — 24.
Коляда А. А., Селянинов М. Ю. Умножение дробей в модулярной системе счисления с использованием интервального индекса // Вести. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1986. № 3.
Коляда А. А., Селяншов М. Ю. О контроле модулярных вычислительных устройств конвейерного типа//Всесоюз. совещ. «Конвейерные вычислительные системы» (Киев, 18—19 сент., 1985): Тез. докл. и сообщ. Киев, 1985. С. 91 — 93.
Коляда А. А., Чернявский А. Ф. Быстродействие АУ в обобщенных СОК// Вестн. Белорус, ун-та. Сер. 1. Физ. Мат. Мех. 1981. № 2. С. 12— 16.
Коляда А.А., Аксенов А.М., Отливанчик Е.А. и др. Синтез компьютерной процедуры и архитектуры высокопроизводительного процесса ортогональных проекций дискретных сигналов на базе минимально избыточных модулярных систем счисления // ММРО-7: Математические методы распознавания образов. – Пущино, Россия, 25–30 сентября 1995 г. – Тез. Докл. – Москва. – 1995. – С. 105–106.
Коляда А.А., Коляда Н.А., Чернявский А.Ф. Мультипроцессорная технология модулярных вычислений. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 218-231.
Коляда А.А., Кравцов В.К., Чернявский А.Ф. Основы минимально избыточной интервально-модулярной арифметики с рекурсивной кодовой структурой// Информатика. – 2004. – №1. – С. 112–120.
Коляда А.А., Пак И.Т. Модулярные структуры конвейерной обработки цифровой информации. – Минск: Университетское, 1992. – 256 с.
Коляда А.А., Ревинский В.В., Селянинов М.Ю. и др. Теоретические основы модулярных вычислительных структур на конечных математических моделях // Современные вопросы оптики, радиационного мате-

риаловедения, информатики, радиофизики и электроники: Сборник научных трудов НИИ прикладных физических проблем им. А.Н.Севченко. – Мн.: Белгосуниверситет, 1996. – Ч. 2. – С. 4 – 9.

Коляда А.А., Ревинский В.В., Селянинов М.Ю., Чернявский А.Ф. Применение минимально избыточного модулярного кодирования для быстрого умножения комплексных чисел в системах цифровой обработки сигналов // Весці Акадэміі навук Беларусі. Сер. фіз.-тэхн. Навук. – 1996. – № 1.

Коляда А.А., Ревинский В.В., Селянинов М.Ю., Шабинская Е.В. Методы масштабирования минимально избыточной модулярной арифметики / // Весці НАН Беларусі. Сер. фіз.-мат. навук. – 1998, № 4. С. 132 – 137.

Коляда А.А., Ревинский В.В., Чернявский А.Ф. Минимально избыточные полиномиально-скалярные модулярные системы счисления // Весці НАН Беларусі. Сер. фіз.-мат. навук. – 1998. – №3. – С.103–107.

Коляда А.А., Селянинов М.Ю., Чернявский А.Ф. Минимально избыточные полиномиально-скалярные модулярные системы счисления// Актуальные проблемы социально-гуманитарных и естественных наук. Тез. научн. конф., посвящённой 70-ю Белгосуниверситета. – Т.1. – Минск, 1996. – С. 181–183.

Коляда А.А., Чернявский А.Ф. Модулярные вычислительные структуры: Вчера, Сегодня, Завтра. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 23-34.

Корнев М.Д. О структурных решениях в проекте ЭВМ 5Э53. /// Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 173-177.

Корнилов А.И., Исаева Т.Ю., Семенов М.Ю. Методы логического синтеза сумматоров с ускоренным переносом по модулю (2^n-1) на основе BDD-технологии // Известия ВУЗов. Электроника. – 2004. – Вып. 3. – С. 54-60.

Корнилов А.И., Семенов М.Ю, Калашников В.С. Методы аппаратной оптимизации сумматоров для двух операндов в системе остаточных классов // Известия ВУЗов. Электроника. – 2004. - № 1. – С. 75-82.

Корнилов А.И., Семенов М.Ю., Ласточкин О.В. Принципы построения модулярных индексных умножителей// Известия ВУЗов. Электроника. – 2004. - Вып. 2. – С. 48-55.

Корнилов А.И., Семенов М.Ю., Ласточкин О.В., Калашников В.С. Применение современных методов проектирования при реализации модулярных вычислительных процедур. /// Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 362-376.

Королев А.В.,Краснобаев В.А. Коррекция ошибок в классе вычетов // АСУ летательных аппаратов. Харьков:. ХАИ. Вып.4.1982. С. 145 –151.
Коряков И.В. Защищенная передача сигналов на основе модулярного преобразования. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 503-513.
Коряков И.В. Метод измерения частоты сигнала на основе системы остаточных классов. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 514-523.
Краснобаев В. А. и др. Методы повышения надежности специализированных ЭВМ систем и средств связи. Харьков: ХВВКМУ РВ, 1990. - 172 с.
Краснобаев В. А. Надежностная модель ЭВМ в системе останочных классноов // Электронное моделирование. — 1985. — Т. 7, № 4. С. 44 46.
Краснобаев В. А., Приходько С. И., Снисаренко А.И. Помехоустойчивое кодирование в АСУ. - Харьков: МО СССР, 1990. - 151 с.
Краснобаев В.А,.Евстигнеев В.Г., Сведо-Швец В.А.Арифметические алгоритмы для q-ичной системы счисления// АСУ летательных аппаратов. Харьков:. ХАИ. Вып.1982. С. 165 –168.
Краснобаев В.А. Алгоритмы модульных операций в классе вычетов. – Харьков. ХВКИВУ. Тематический НТС № 327, 1982. – С. 99 – 101.
Краснобаев В.А. Алгоритмы определения обратной мультипликативной величины числа в системе остаточных классов // Збірник наукових праць. Системи обробки інформації. - Харків: НАНУ ПАНМ, ХВУ, . 2002. – Вип. 4. (20). – С. 30 - 32.
Краснобаев В.А. Вариант выбора системы счисления ЭВМ // АСУ и приборы автоматики . 1989. Вып. 91. С. 120 -122.
Краснобаев В.А. Вариант математической модели надежности ЭВМ в системе остаточных классов // Кибернетика. 1987. № 1. С. 25 – 26, 38.
Краснобаев В.А. Влияние формы кодирования операндов на надежность систем обработки цифровой информации. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 343-354.
Краснобаев В.А. Илюшко Я.В. Методы обработки информации в системе остаточных классов // Радіоелектронні і комп'ютерні системи. – 2004. - № 2(6).-С. 101 – 109.
Краснобаев В.А. Искусственный интеллект и система остаточных классов // Проблемы бионики. 1987. Вып. 39. С. 53 -58.
Краснобаев В.А. Корректирующие l-коды в системе остаточных классов. Деп. ЦИВТИ. № 6393, № 2. 1981. Д 4767.

Краснобаев В.А. Коррекция однократных ошибок в системе остаточных классов с помощью l-кодов // Проектирование вычислительных систем и их устройств для летательных аппаратов. М:. МАИ. 1980. С. 70 –73
Краснобаев В.А. Математическая модель надежности ЭВМ в системе остаточных классов // Электрон.моделирование. 1990. № 5. С. 70 – 72.
Краснобаев В.А. Метод и алгоритмы коррекции ошибок в системах цифровой обработки информации // Радиотехника. 2002. Вып. 126. С. 231 – 237.
Краснобаев В.А. Метод коррекции ошибок в системе остаточных классов // АСУ и приборы автоматики . 1987. Вып. 82. С. 112 -115.
Краснобаев В.А. Метод коррекции ошибок в системе остаточных классов // АСУ и приборы автоматики . 1988. Вып. 88. С. 36 -38.
Краснобаев В.А. Метод парной нулевизации чисел, представленных в системе остаточных классов с предварительной выборкой цифр. Деп. ЦИВТИ. № 6213, № 9. 1980. Д 4591.
Краснобаев В.А. Метод реализации арифметических операций в системе остаточных классов. Деп. Электрон.моделирование. № 4. 1988. № 2002 – В88.
Краснобаев В.А. Методы арифметического сравнения чисел, представленных кодом системы остаточных классов // АСУ и приборы автоматики . 1987. Вып. 84. С. 74 -76.
Краснобаев В.А. Методы и алгоритмы возведения чисел в произвольную степень по модулю системы остаточных классов // АСУ и приборы автоматики . 1986. Вып. 80. С. 101 -103.
Краснобаев В.А. Методы повышения надежности специализированных ЭВМ систем и средств связи. Харьков: МО СССР, 1990. – 172с.
Краснобаев В.А. Методы реализации модульных операций в системах цифровой обработки информации // Радиотехника. 2001. Вып. 119. С. 130 – 134.
Краснобаев В.А. Методы сравнения операндов в системе остаточных классов // Радиотехника. 2003. Вып. 134. С. 223 – 228 .
Краснобаев В.А. Методы сравнения чисел, представленных кодом системы остаточных классов // Электрон.моделирование. 1988. № 2. С. 84 – 87.
Краснобаев В.А. Надежностный синтез вычислительных структур в системе остаточных классов // Кибернетика. 1990. № 5. С. 115 – 118.
Краснобаев В.А. О коррекции однократных ошибок в системе остаточных классов // Вопросы теории радиотехнических информационно-измерительных систем . М:. ВА им.Дзержинского .Вып. 20. 1983. С. 209 –211.
Краснобаев В.А. Оценка быстродействия коррекции ошибок в системе остаточных классов для систем передачи информации // Радиотехника. 1984. Вып. 71. С. 32 -35.
Краснобаев В.А. Построение узлов и блоков ЭВМ, работающих в системе

остаточных классов. – Харьков. ХВКИВУ. Тематический НТС № 327, 1982. – С. 94 – 96.
Краснобаев В.А. Применение нелинейных кодов для контроля информации в системе остаточных классов. Деп. ЦИВТИ. № 7195, № 8. 1982. Д 5480.
Краснобаев В.А. Принцип реализации арифметических операций в системе остаточных классов // АСУ и приборы автоматики . 1988. Вып. 86. С. 82 -85.
Краснобаев В.А. Способ помехоустойчивого кодирования в системе остаточных классов // Радиотехника. 1984. Вып. 70. С. 9 -17.
Краснобаев В.А. Способ арифметического помехоустойчивого кодирования в системе остаточных классов // Радиотехника. 1985. Вып. 73. С. 28 - 31.
Краснобаев В.А. Способ коррекции ошибок в системе остаточных классов с взаимно не простыми основаниями // Автоматика и приборостроение. 1983. Вып. 9(№ 204). С. 68 – 70.
Краснобаев В.А. Способ коррекции ошибок с помощью l-кодов. Деп. ЦИВТИ. № 7102, № 7. 1982. Д 5402.
Краснобаев В.А. Способ преобразования чисел из кода системы остаточных классов в десятичный код с контролем ошибок. Деп. ЦИВТИ. № 6212, № 9. 1980. Д 4589.
Краснобаев В.А., Илюшко Я.В. Построение систем искусственного интеллекта на основе использования непозиционного кодирования информации // Открытые информационные и компьютерные интегрированные технологии. – Х.: НАКУ (ХАИ). – 2004. – Вып. 24. – С. 286-298.
Краснобаев В.А., Илюшко Я.В., Замула А.А. Универсальные алгоритмы сжатия табличных цифровых данных результатов выполнения арифметических операций в системе остаточных классов // Радиотехника. Всеукр. Межвед. науч.-техн. сб. 2005. Вып. 140 . С.111 – 121
Краснобаев В.А., Ирхин В.П. Алгоритм реализации операции модульного умножения в системе остаточных классов// Электронное моделирование, 1993, №5, с. 20-26.
Краснобаев В.А., Ирхин В.П. Вариант оптимизации структуры ЭВМ, функционирующей в позиционно-остаточной системе счисления (Материалы международной НТК «Представление, обработка и передача информации».) Сочи-Харьков: АНУ, ПАНИ, 1992, с. 65-69.
Краснобаев В.А., Ирхин В.П. Вариант решения задачи оптимального резервирования в системе остаточных классов // Кибернетика. 1990. № 3. С. 123 – 125.
Краснобаев В.А., Ирхин В.П. Варианты определения обратной мультипликативной величины числа в системе остаточных классов. – Харьков: ХВКИУРВ. Тематический научно-технический сборник №337, 1992, с. 132-140.
Краснобаев В.А., Ирхин В.П. Пример решения обратной задачи опти-

мального резервирования в системе остаточных классов// Кибернетика. – 1990. – №3. – с. 123-125

Краснобаев В.А., Ирхин В.П., Квасов М. В. Вариант определения обратной мультипликативной величины числа в системе остаточных классов. Харьков. ХВКИВУ. Тематический НТС № 427, 1992. – С. 94 – 96.

Краснобаев В.А., Ирхин В.П., Кононова Н.В. Методы и алгоритмы реализации арифметических операций в системе остаточных классов//АСУ и приборы автоматики. – Харьков: Вища школа. Вып. 93. 1990.с. 46-53.

Краснобаев В.А., Ілюшко Я.В. Табличный метод обработки цифровой информации в классе вычетов // Моделювання та інформаційні технології. Київ: НАНУ. – 2004. – Вип. 26. – С.101 – 105.

Краснобаев В.А., Удников А. Выбор системы счисления при проектировании отказоустойчивых ЭВМ // Інформаційно-керуючі системи на залізничному транспорті. - 2001. - № 2. – С. 27 – 29.

Краснобаев В.А., Удников А. Н. Методы оценки достоверности функционирования цифровых систем обработки информации.// Збірник наукових праць. – Харків. ХВУ. 2002.-Вип.4 (42). - С. 313 – 315.

Краснобаев В.А., Удников А.Н. Метод обнаружения ошибок в цифровых системах обработки информации // Вісник ХДТУСГ Вип. 19. Том 2. 2003. С. 88-92.

Краснобаев В.А., Швецов Н.И. Надежностная модель вычислителя для обработки информации АСУ. – Харьков. ХВКИВУ. Тематический НТС № 330, 1986. – С. 125 – 126.

Краснобаев В.А.,Бороденко Е. И.,Евстигнеев В.Г. Оптимизация выбора оснований системы остаточных классов. – Харьков. ХВКИВУ. Тематический НТС № 328, 1983. – С. 35 – 37.

Краснобаев В.А.,Евстигнеев В. Г., Бороденко Е. И. Коррекция ошибок в системе остаточных классов со взаимно не простыми основаниями // Радиотехника. 1983. Вып. 67. С. 39 -43.

Краснобаев В.А.,Евстигнеев В. Г., Бороденко Е. И. Коррекция ошибок в системе остаточных классов со взаимно не простыми основаниями // Радиотехника. 1983. Вып. 67. С. 39 -43.

Краснобаев В.А.,Евстигнеев В. Г., Бороденко Е. И. Табличная реализация операции модульного сложения в системе остаточных классов // Радиотехника. 1984. Вып. 68. С. 57 -61.

Краснобаев В.А.,Евстигнеев В.Г., Бороденко Е.И. Табличная реализация модульных операций в классе вычетов // Радиотехника. 1983. Вып. 64. С. 34 -37.

Краснобаев В.А.,Ирхин В.П. Алгоритм реализации операции модульного умножения в системе остаточных классов // Электрон. моделирование. 1993. № 5. С. 20 – 26.

Краснобаев В.А..Надежностная модель ЭВМ в системе остаточных классов // Электрон.моделирование. 1985. № 4. С.44 – 46.

Краснобаев В.А.Основы создания вычислителей на основе остаточных

классов // Системи обробки інформації. – Харків: НАНУ, . - ПАНМ, ХВУ. – 2001. - Вип. 1 (11). – С. 3 – 7.
Краснобаев В.А.Техническая реализация метода коррекции ошибок в системе остаточных классов // АСУ и приборы автоматики . 1987. Вып. 81 С. 97 -101.
Краснобаєв В.А., Ілюшко Я.В. Метод корекції помилок у системах автоматизованої обробки даних // Вісник ХДТУСГ імені Петра Василенка. Вип. 27. Том 2. 2004. С. 182 - 185.
Краснобаєв В.А., Ілюшко Я.В. Метод та обчислювальна система обробки інформації, що представлена у системі залишкових класів // Збірник наукових праць. Системи обробки інформації. - Харків: НАНУ ПАНМ, ХВУ. 2004. – Вип. 7. (35). – С. 106 - 111.
Краснобаєв В.А., Кошман С.О. Застосування системи залишкових класів у машинній арифметиці // Вісник ХДТУСГ Вип. 19. Том 2. 2003. С. 134-136.
Краснобаєв В.А.Синтез та оптимізація обчислювальних структур у системі залишкових класів // Інформаційно-керуючи системи на залізничному транспорті. – 2000. - № 2.- С. 36 – 37.
Ласточкин О.В. Особенности реализации умножителей в устройствах построенных с применением принципов модулярной арифметики // Микроэлектроника и информатика-2004. Одиннадцатая всероссийская межвузовская научно-техническая конференция студентов и аспирантов: Тезисы докладов, М.:МИЭТ, 2004.-444с, стр.219.
Ласточкин О.В. Принципы построения IP-блоков двоичных и специализированных умножителей с применением языка Verilog HDL // Микроэлектроника и информатика-2005. Двенадцатая всероссийская межвузовская научно-техническая конференция студентов и аспирантов: Тезисы докладов, М.:МИЭТ, 2004.-444с, стр.104.
Лебедев Е. К. Быстрые алгоритмы цифровой обработки сигналов: Монография. — Красноярск, 1989.
Лебедев Е. К. Синтез нелинейных непозиционных устройств обработки марковских сигналов // Изв. вузов. Радиотехника. — 1987. — Т. 30, № 12. С. 69-72.
Лебедев Е. К. Цифровая фильтрация в системе остаточных классов // Изв. вузов. Радиотехника. — 1985. — Т. 28, № 8. - С. 58-62.
Лебедев, Е. К. Синтез нелинейных непозиционных устройств обработки марковских сигналов//Изв. вузов. Радиоэлектроника. 1987. Т. 30, № 12. С. 69 — 72.
Лопатин Д.С., Овчаренко Л.А., Ирхин В.П. Масштабирование чисел в модулярной системе счисления. Системы обработки информации. Сборник научных трудов. Вып. 6 (22). – Харьков: НАНУ, ПАНИ, ХВУ, 2002, с.209-215.
Лукин Ф.В. Доклады об ЭВМ «Алмаз» на конкурсной комиссии. // Труды Юбилейной Международной научно-технической конференции «50 лет

модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 161-172.
Малашевич Б. М. Разработка вычислительной техники в Зеленограде. Неизвестные супер-ЭВМ // Электроника, наука, технология, бизнес. — 2004. — № 2.
Малашевич Б.М., Малашевич Д.Б. Модулярная арифметика – взгляд изнутри. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 47-100.
Малашевич Б.М., Малашевич Д.Б. Отечественные модулярные и троичные ЭВМ. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 101-148.
Малашевич Д.Б. Недвоичные системы в вычислительной технике. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 592-606.
Малашевич Д.Б., Машевич П.Р. Элементная база для модулярных и троичных ЭВМ. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 389-406.
Мотоока Т., Хорикоси Х., Сакаути М. и др.Компьютеры на СБИС / – М.: Мир, 1988. – Кн. 2.
Музыченко О.Н. Методы Синтеза логических схем модульного контроля в унитарных непозиционных двоичных кодах. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 434-458.
Музыченко О.Н. Методы Синтеза логических схем модульного контроля в натуральных двоичных кодах. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 459-489.
Музыченко О.Н. Синтез логических схем модульного контроля в унитарных позиционных двоичных кодах. // Автоматика и телемеханика. 2001. № 3. С. 158 - 173.
Музыченко О.Н. Специализированные методы синтеза логических схем. кн.2. Методы синтеза логических схем модульного контроля, выполнения модульных операций и преобразования кодов. (Учебное пособие) С. Пб., БГТУ, 2004, 244 с.
Музыченко О.Н. Упрощение пороговых схем, синтезируемых методом промежуточного преобразования. // Автоматика и телемеханика. 1990.

№ 12. С. 164 - 170.
Музыченко О.Н., Рыжевнин В.Н. Быстродействующие последовательностные устройства модульного контроля параллельных двоичных кодов. Техника средств связи сер. ‘Техника проводной связи’, 1989, вып.5, с.72-78.
Музыченко О.Н., Рыжевнин В.Н. Проектирование быстродействующих последовательностных устройств модульного контроля двоичных кодов. Тезисы докладов 20-й отраслевой НТК 'Интег-ральные оптические сети связи', Л., 1989, т.2, с.77-78.
Музыченко О.Н., Рыжевнин В.Н. Проектирование устройств формирования вычета последовательного двоичного кода по произвольному модулю К Тезисы докладов20-й отраслевой НТК 'Интег-ральные оптические сети связи', Л., 1989, т.2, с.79-80.
Нейрокомпьютеры в остаточных классах. Кн.11 (Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н.): учеб. пособие для вузов. – М.: Радиотехника, 2003. – 272 с.
Нейроматематика. Кн. 6 / Червяков Н.И., Сахнюк П.А. и др.; Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2002. – 448 с.
Овчаренко Л.А. Вариант реализации основных операций в модулярном арифметическом устройстве// Телекоммуникации. – 2001. – №3. – С.8-11.
Овчаренко Л.А. Когерентный преобразователь модулярного кода// Телекоммуникации. – 2001. – №4. – С.40-44.
Овчаренко Л.А. Обоснование требований к точности установки сдвига фазы в управляемых фазовращателях когерентного модулярного сумматора// Телекоммуникации. – 2002. – №5. – С.2-4.
Овчаренко Л.А. Оценка достоверности выполнения операций в когерентном сумматоре модулярного арифметического устройства// Телекоммуникации. – 2001. – №12. – С.38-41.
Овчаренко Л.А. Реализация немодульных операций на когерентных модулярных сумматорах. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 329-342.
Овчаренко Л.А. Реализация цифрового трансверсального фильтра в системе остаточных классов// Известия вузов – Радиоэлектроника. – 2002. – №4. – С.50-57.
Овчаренко Л.А. Способ выполнения немодульных операций в модулярной системе счисления// Седьмая межвузовская научно-техническая конференция. Тезисы докладов. Труды института. – Воронеж: ВИРЭ. – 2001. – вып. 7. – С. 126.
Овчаренко Л.А. Устройство контроля выполнения арифметических операций в позиционном сумматоре// Телекоммуникации. – 2002. – №3. – С.26-30.

Овчаренко Л.А. Цифроаналоговый преобразователь кода системы остаточных классов контроллера управления динамическим объектом// Известия вузов – Радиоэлектроника. – 2002. – №11. – С.29-33.
Овчаренко Л.А. Цифровой фильтр в системе остаточных классов на когерентных модулярных сумматорах// Физика волновых процессов и радиотехнические системы. – 2004. – №2. – С.48-53.
Овчаренко Л.А., Баженов А.А. Спектральные характеристики прямого цифрового синтезатора частоты в системе остаточных классов// Известия вузов – Радиоэлектроника. – 2003. – №9. – С.53-60.
Овчаренко Л.А., Баженов А.А., Проскурина Н.Г. Формирование периодических сигналов произвольной формы цифровым синтезатором в модулярной системе счисления// IV Международная научно-техническая конференция «Кибернетика и технологии XXI века». Сборник докладов. – Воронеж: ВГУ. – 2003.– С.394-399.
Овчаренко Л.А., Баженов А.А., Проскурина Н.Г. Функциональный преобразователь кода модулярной системы счисления в напряжение// III Международная научно-техническая конференция «Кибернетика и технологии XXI века». Сборник докладов. – Воронеж: ВГУ. – 2002.– С.312-316.
Овчаренко Л.А., Баженов А.А., Сизов А.С. Анализ спектральных характеристик прямого цифрового синтезатора частоты в системе остаточных классов// Шестая международная конференция «Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации». Сборник материалов конференции. – Курск: КГТУ. – 2003. – С.182-183.
Овчаренко Л.А., Болкунов А.А. Многоуровневое модулярное арифметическое устройство// Седьмая межвузовская научно-техническая конференция. Тезисы докладов. Труды института. – Воронеж: ВИРЭ. – 2001. – вып. 7. – С. 122.
Овчаренко Л.А., Болкунов А.А. Оптимизация структуры ЭВМ в системе остаточных классов// Шестая межвузовская научно-техническая конференция. Тезисы докладов. Труды института. – Воронеж: ВИРЭ. – 2000. –вып. 6. –С. 219.
Овчаренко Л.А., Болкунов А.А. Принципы построения преобразователей «напряжение – код в системе остаточных классов»// Вестник Воронежского института МВД России. – 2000. – №2(7). – С.19-24.
Овчаренко Л.А., Болкунов А.А. Проектирование специализированных процессоров в модулярной системе счисления// Межвузовский сборник научных трудов. – Воронеж: ВИРЭ. – 2000. – С. 77 – 81.
Овчаренко Л.А., Болкунов А.А., Проскурина Н.Г. Сравнительный анализ схемотехнических решений вычислительных структур модулярной арифметики// IX Международная научно-техническая конференция «Радиолокация, навигация, связь». Сборник докладов. – Воронеж: ВГУ. – 2003.– С.298-303.

Овчаренко Л.А., Дидрих В.Е. Повышение быстродействия цифрового фильтра в модулярной системе счисления// Радиосистемы: Обработка сигналов и полей.(журнал в журнале Радиотехника – №4 – 2002)– 2002. – №5(59). – С. 49-55.

Овчаренко Л.А., Ирхин В.П., Болкунов А.А. Алгоритм реализации операций базового набора в системе остаточных классов// Шестая межвузовская научно-техническая конференция. Тезисы докладов. Труды института. – Воронеж: ВИРЭ. – 2000. –вып. 6. – С. 220.

Овчаренко Л.А., Корольков М.А. Расчет разностного уравнения цифрового фильтра в модулярной системе счисления// VIII Международная научно-техническая конференция «Радиолокация, навигация, связь». Сборник докладов. – Воронеж: ВГУ. – 2002. – том 3. – С.2130-2141.

Овчаренко Л.А., Лопатин Д.С. Вычисление функций в модулярной системе счисления// Международная научная конференция «Информационные технологии в естественных, технических и гуманитарных науках». Часть 2. – Таганрог: ТГРТУ. – 2002. – С.32–35.

Овчаренко Л.А., Лопатин Д.С. Деление числа в модулярном коде на основание системы счисления// Телекоммуникации. – 2002. – №6. – С.7-10.

Овчаренко Л.А., Лопатин Д.С. Масштабирование чисел в модулярной системе счисления// Сборник научных трудов. – Харьков: НАНУ, ПАНМ, ХВУ. – 2002. – вып. 6(22). – С. 209-215.

Овчаренко Л.А., Лопатин Д.С. Преобразование двоичного позиционного кода в код модулярной системы счисления// Телекоммуникации. – 2003. – №9. – С.8-11.

Овчаренко Л.А., Лопатин Д.С. Синтез минимально избыточных структур отказоустойчивых специализированных процессоров в модулярной системе счисления// Вестник Военного института радиоэлектроники. – Воронеж: ВИРЭ. – 2004. – №2.

Овчаренко Л.А., Лопатин Д.С. Синтез отказоустойчивых структур модулярных арифметических устройств// Вторая Международная научно-техническая конференция «Физика и технические приложения волновых процессов». Сборник материалов конференции. – Самара: Поволжская государственная академия телекоммуникаций и информатики. – 2003.

Овчаренко Л.А., Лопатин Д.С., Проскурина Н.Г. Оценка помехоустойчивости модулярного сумматора, реализующего n–местные арифметические операции// IV Международная научно-техническая конференция «Кибернетика и технологии XXI века». Сборник докладов. – Воронеж: ВГУ. – 2003.– С.388-393.

Овчаренко Л.А., Чекалин С.С. Обнаружение ошибок в модулярном арифметическом устройстве на основе применения контрольного основания// Сборник научных трудов. – Харьков: НАНУ, ПАНМ, ХВУ. – 2002. – вып. 5(21). – С. 55-61.

Овчаренко Л.А., Чекалин С.С., Лопатин Д.С. Алгоритм деления числа в

модулярном коде на основание системы счисления// Материалы второй региональной научно-методической конференции «Информатика как педагогическая задача». – Воронеж: ВГУ. – 2003.– С.111-113.

Овчаренко Л.А., Чекалин С.С., Лопатин Д.С. Реализация алгоритма контроля по модулю позиционных сумматоров// Материалы второй региональной научно-методической конференции «Информатика как педагогическая задача». – Воронеж: ВГУ. – 2003.– С.114-116.

Онищенко С.М. Применение гиперкомплексных чисел в теории инерциальной навигации. Автономные системы. – Киев: Наукова думка, 1983. – 208 с.

Орлов Л. А., Попов Ю. М. Оптоэлектронное АУ в СОК //Авто-метрия. 1972. № 6. С. 14.

Осепянц О.А., Исмаилов Ш-М. А. Методика генерации оптимального основания для представления чисел в системе остаточных классов. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 304-320.

Отказоустойчивость специализированных процессоров автоматизированных систем управления и средств связи. (Методы повышения надежности специализированных процессоров систем управления и связи). Учебное пособие. Н.И. Червяков, В.А. Краснобаев, В.П. Ирхин – Ставрополь: Ставропольское ВВИУС, 1991.

Оцоков Ш.А, Исмаилов Ш.А. Алгоритм вычисления элементарной функции в системе остаточных классов // Доклады научной конференции «Новые информационные технологии. Разработка и аспекты применения». – Таганрог, 2002. – С. 74-75.

Оцоков Ш.А. Алгоритм безошибочного суммирования чисел с фиксированной запятой. // Доклады научно-технической конференции «Новые информационные технологии» том 1, М.:МГАПИ, 2003. – С. 155-158.

Оцоков Ш.А. Об ускорении операции сложения чисел с плавающей точкой на основе модулярной арифметики. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 321-328.

Оцоков Ш.А., Исмаилов Ш.А. Применение методов безошибочных вычислений в гидрологии // Доклады международной конференции «Параллельные вычисления в газовых и жидких средах» /на англ. яз./. – М., 2003. – С.206-208.

Оцоков Ш.А., Исмаилов Ш.А. Разрядно-параллельный алгоритм и структура вычислительного устройства безошибочной обработки массивов числовых данных // Сб. трудов международной научной конференции «Информационные технологии в науке, образовании, телекоммуникации, бизнесе». - Украина, Крым, Ялта-Гурзуф, 2003.- С. 37-39.

Пат. 1143591 (Япония). Устройство анализа информации. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М- Бурцев. 117.Пат. 1143592 (Япония). Процессор вычислительной машины. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М.Бурцев.

Пат. 1152932 (Япония). Устройство деления. Авт. изобрет.И.Т. Пак, И.Я. Акушский, В.М. Бурцев.

Пат. 1235676 (Англия). A device for coding complex numbers in digital computers. Авт. изобрет. И.Т. Пак, И.Я.Акушский. Д.И. Юдицкий.

Пат. 1305783 (Франция). Processeur d'ordinateur Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.

Пат. 1505779 (Франция), Precede de codge des nombres dans les caleu iatricesnumeric ques et dispositif pour sa realisation. Авт. изобрет. И.Т. Пак, И.Я. Акушский, Д.И.Юдицкий.

Пат. 1507121 (Англия). Improvements in Relatying to Multiplying Device. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.

Пат. 1507121 (Англия). Processor of Computer. Авт.изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.

Пат. 1507776 (Англия). Data Analyzer. Авт. изобрет. И.Т.Пак, И.Я. Акушский, В.М. Бурцев.

Пат. 1549376 (ФРГ). Verfarben und Befindung zur verachelisselung complexer zahien in digitalen Rechcnanlagen. Авт. изобрет. И.Т. Пак, И.Я. Акушский, Д.И. Юдицкий.

Пат. 157560 РФ. Арифметическое устройство по модулю/ Болкунов А.А., Овчаренко Л.А., Ирхин В.П., Долгачев А.А. – 2000. – Бюл. № 28.

Пат. 2012041 РФ, Калмыков И.А. Устройство для вычисления сумм парных произведений/ Патент № 2012041// Открытия. Изобретения. - 1994. Бюл. № 8.

Пат. 2018936 РФ. Устройство для умножения чисел по модулю /В.А. Краснобаев, В.П. Ирхин, М.В. Квасов и др. Опубл. в БИ. 1994. № 16.

Пат. 2018950 РФ, Калмыков И.А., Бережной В.В., Оленев А.А. Систолический процессор ДПФ с коррекцией ошибки/ Патент № 2018950 //Открытия. Изобретения. - 1994. - Бюл. № 16.

Пат. 2023289 РФ. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, В.П. Ирхин, А.И. Возный и др. Опубл. в БИ. 1994. № 21.

Пат. 2023290 РФ. Устройство для умножения чисел по модулю /В.А. Краснобаев, В.П. Ирхин, Н.И. Можаев и др. Опубл. в БИ. 1994. № 21.

Пат. 2029437 РФ, Калмыков И.А., Оленев А.А. Систолический отказоустойчивый процессор дискретного преобразования Фурье/ Патент № 2029437// Открытия. Изобретения. – 1994. Бюл. №16.

Пат. 2109326 РФ. Устройство для сложения и вычитания чисел по модулю / В.П. Ирхин - Б.И., 1998, №11.

Пат. 2110087 РФ. Устройство для сложения чисел по модулю / В.П. Ирхин - Б.И., 1998, № 12.

Пат. 2110147 РФ. Устройство для формирования остатка по модулю / В.П. Ирхин - Б.И., 1998, № 12.
Пат. 2123202 РФ. Устройство для резервирования / В.П. Ирхин - Б.И., 1998, № 34.
Пат. 2131618 РФ. Устройство для сложения N чисел по модулю / В.П. Ирхин - Б.И., 1999, № 16.
Пат. 2133495 РФ. Устройство для вычитания по модулю / В.П. Ирхин - Б.И., 1999, № 20.
Пат. 2137181 РФ. Устройство для умножения чисел по модулю / В.П. Ирхин и др. - Б.И., 1999, № 25.
Пат. 2143723 РФ. Устройство для умножения чисел по модулю / В.П. Ирхин и др. - Б.И., 1999, № 36.
Пат. 2145112 РФ. Устройство для сложения и вычитания чисел по модулю / В.П. Ирхин и др. - Б.И., 2000, № 3.
Пат. 2149442 Устройство для умножения по модулю семь/ О.Н. Музыченко 2000 г.
Пат. 2156998 РФ. Устройство для сложения и вычитания чисел по модулю / В.П. Ирхин и др. - Б.И., 2000, № 27.
Пат. 2187886 РФ. Устройство для преобразования чисел из кода системы остаточных классов в полиадический код/ Овчаренко Л.А. – 2002. – Бюл. № 23.
Пат. 2188448 РФ. Устройство для сложения n чисел по модулю p / Овчаренко Л.А. – 2002. – Бюл. № 24.
Пат. 2192092 РФ. Устройство для преобразования n – разрядного двоичного позиционного кода в двоичный код остатка по модулю m/ Овчаренко Л.А., Турченяк В.И. – 2002. – Бюл. № 30.
Пат. 2209460 РФ. Устройство для формирования остатка по модулю от числа / В.П. Ирхин и др. - Б.И., 2003, № 21.
Пат. 2220441 РФ. Устройство для сложения n чисел по модулю p / Овчаренко Л.А., Лопатин Д.С., Чекалин С.С. – 2003. – Бюл. № 36.
Пат. 2256213 РФ. Червяков Н.И., Шапошников А.В. Нейронная сеть для коррекции ошибок в модулярных нейрокомпьютерах. Патент № 2256213, Бюл. 19, 2005
Пат. 2257615 РФ. Червяков Н.И., Малофей А.О., Рыбальченко, Щелкунова Ю.О. Нейронная сеть для вычисления позиционной характеристики непозиционного кода. Патент № 2257615, Бюл. 21, 2005
Пат. 2258257 РФ. Червяков Н.И., Сивоплясов Д.В. Нейронная сеть для преобразования полиадического кода в код системы остаточных классов. Патент № 2258257, Бюл. 22, 2005
Пат. 2305782 (Франция). Dispositiv D'analyse de Finformation. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М.Бурцев.
Пат. 2305784 (Франция). Dispositif de multiplication. Авт.изобрет. И.Т. Пак. И.Я. Акушский, В.М. Бурцев.

Пат. 235330 (Франция). Dispositif de multiplication. Авт.изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.
Пат. 3523068 (Англия). Division Device. Авт. изобрет. И.Т.Пак, И.Я. Акушский, В.М. Бурцев.
Пат. 4064400 (США). Devuce for Multi pliing Numbers represented in the System of Residial Classes. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев. 114.4121298 (США). Central Processing Unit for Num - bers Represented in the System of Residual Classes. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.
Пат. 42649 А (Украина). Полисский Ю.Д.. Устройство для сравнения чисел в системе остаточных классов. Пат. № 42649 А (Украина) А, 7 G06 F 7/04, 2001 (в оригинале на украинском языке: Ю.Д.Поліський. Пристрій для порівняння чисел у системі залишкових класів. Патент України № 42649 А, 7 G06 F 7/04, 2001)
Пат. 47630 А (Украина) Полисский Ю.Д.. Устройство для определения в системе остаточных классов числа, ближайшего к заданному. Патент Украины № 47630 А, 7 G06 F 7/04, 2002 (в оригинале на украинском языке: Ю.Д.Поліський. Пристрій для визначення у системі залишкових класів числа, найбдижчого до заданого. Патент України № 47630 А, 7 G06 F 7/04, 2002)
Пат. 488225 (Швейцария) Recheneinzichtung zum Rechenmit komplexen Zahlen. И.Т. Пак, И.Я. Акушский, Д.И. Юдицкий.
Пат. 7608741 (Франция). Improvement in Relatying to Multiplying Device. Авт. изобрет. И.Т. Пак, И.Я. Акушский, В.М. Бурцев.
Пат. 810966 (Италия) Metodo per cjdificare numori in calcolatori Jigitali e dispozitivo per Feaiizare tale medoto. Авт. изобрет. И-Т. Пак, И.Я. Акушский, Д.И. Юдицкий.
Пат. № 1797119 РФ. Червяков Н.И., Бережной В.В., Оленев А.А. Устройство для преобразования кода системы остаточных классов в позиционный код с исправлением ошибок. Патент РФ № 1797119, БИ № 7, 1993.
Пат. № 1820377 РФ. Червяков Н.И., Бережной В.В. Сумматор по модулю. Патент РФ № 1820377, БИ № 21, 1993.
Пат. № 2020756 РФ. Червяков Н.И., Ремизов С.Л. Устройство для определения позиционных характеристик непозиционного кода. Патент РФ № 2020756, БИ № 18, 1994.
Пат. № 2022972 РФ. Червяков Н.И., Бережной В.В., Оленев А.А. Устройство для контроля и исправления ошибок в избыточном модулярном коде. Патент РФ № 2022972, БИ № 20, 1994.
Пат. № 2022971 РФ. Червяков Н.И., Ремизов С.Л. Устройство для обнаружения ошибок в информации, представленной в системе остаточных классов., БИ № 20, 1994.
Пат. №2220501 РФ. Способ преобразования кода системы остаточных классов в напряжение/ Овчаренко Л.А., Чекалин С.С., Лопатин Д.С.–

2003. – Бюл. № 36.
Пат. №2231822 РФ. Устройство для деления числа в модулярном коде на основание системы счисления/ Овчаренко Л.А., Чекалин С.С., Лопатин Д.С.– 2004. – Бюл. № 9.
Пат. №2237274 РФ. Устройство для деления числа в модулярном коде на основание системы счисления/ Овчаренко Л.А., Лопатин Д.С., Чекалин С.С.– 2004. – Бюл. № 27.
Пат. №2237972 РФ. Синтезатор частоты/ Баженов А.А., Овчаренко Л.А., Сизов А.С. – 2004. – Бюл. № 6.
Пат. №2239281 РФ. Цифровой синтезатор гармонических колебаний/ Баженов А.А., Овчаренко Л.А., Нечаев Ю.Б., Николаев О.В. – 2004. – Бюл. № 30.
Пат. №2242085 РФ. Устройство для преобразования n – разрядного двоичного позиционного кода в двоичный код остатка по модулю m/ Баженов А.А., Болкунов А.А., Овчаренко Л.А., Овчаренко К.Л.– 2004. – Бюл. № 34.
Пат. №2246753 РФ. Устройство для масштабирования числа в модулярной системе счисления/ Овчаренко Л.А., Лопатин Д.С., Чекалин С.С.– 2005. – Бюл. № 5.
Пат/ 2018935 РФ. Устройство для сложения и вычитания чисел по модулю /В.А. Краснобаев, В.П. Ирхин, М.В. Квасов и др. Опубл. в БИ. 1994. № 16.
Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976.
Пол. решение ВНИИГПЭ РФ о выдаче патента на изобрет. по заявке № 97104346/09, МКИ б Н 03 М 7/18. Преобразователь кода системы остаточных классов, заданной модулями V < p, в позиционный код / О. А. Финько, С. Б. Елесин, В. В. Корниенко // Открытия. Изобрет. 1998. №5.
Пол. решение ВНИИГПЭ РФ о выдаче патента на изобрет. по заявке № 97104283/09, МКИ 6 Н 03 М 7/18. Преобразователь кода системы остаточных классов, заданной модулями $p_2 – p_1=1$, в позиционный код/ О.А. Финько, С.Б. Елесин, В.В. Корниенко// Открытия. Изобрет. — 1998. — № 7.
Полисский Ю.Д. Сравнение чисел в системе остаточных классов. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 267-283.
Путинцев Н.Д. Аппаратный контроль управляющих цифровых вычислительных машин. М.: Сов. Радио, 1966.
Сабо Н. Определение знака в неизбыточных системах счисления остаточных классов // Кибернет. сб. М.: Мир, 1964. Вып. 8. С. 149— 165.
Свобода А. Развитие вычислительной техники в Чехословакии. Система счисления остаточных классов // Кибернетический сборник. М.: Мир, 1964. Вып. 8. — С. 115-148 / Svoboda A. Computer progress in Czecho-

slovakia. II. The numerical system of residual classes (SRE) in Digital Informations Wandles, 1962.
Селлерс Ф. Методы обнаружения ошибок в работе ЭЦВМ. М.: Мир, 1972.
Селянинов М.Ю. Минимально избыточная модулярная архитектура адаптивного КИХ-фильтра // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. 2002. № 2. С. 79–88.
Селянинов М.Ю. Применение численно-аналитической модулярной вычислительной технологии для выполнения аддитивных и мультипликативных операций над сигналами в пространствах ортогональных проекций // Доклады НАН Беларуси. – 2002. – Т. 46, № 2. – С. 62–66.
Селянинов М.Ю. Теоретические основы модулярной кодификации алгебраических систем // Весці НАН Беларусі. Сер. фіз.-мат. навук. 2002. № 1. С. 114–119.
Семенов М.Ю., Калашников В.С., Ласточкин О.В Применение аппарата модулярной арифметики для построения фильтра с конечной импульсной характеристикой // Известия ВУЗов. Электроника. – 2005. - №3. – С. 46-50
Семенов М.Ю., Калашников В.С., Ласточкин О.В. Структура оптимизированных сумматоров, функционирующих в системе остаточных классов // Микроэлектроника и информатика-2003. Десятая всероссийская межвузовская конференция студентов и аспирантов: Тезисы докладов, М.:МИЭТ, 2003.
Синъков М. В., Губарени Н. М. Непозиционные представления многомерных числовых систем. — Киев: Наукова думка, 1977. — 149 с.
Синьков М.В., Губарени Н.М. Непозиционные представления в многомерных числовых системах. – Киев: Наукова думка, 1979. – 137 с.
Синьков М.В., Синькова Т.В., Федоренко А.В., Чапор А.А. Нетрадиционная система остаточных классов и ее основоположник И.Я.Акушский. Сайт http://www.icfcst.kiev.ua
Сложение чисел в ядерно-модулярном коде/В. Н. Ахременко, А. А. Коляда, В. К. Кравцов, В. В. Ревинский//Совершенствование методов планирования и повышение эффективности общественного производства: Тез. докл. 9-й межресп. конф. молодых ученых (Минск, НИИЭМП при Госплане БССР, 19—23 апр., 1982). Мн., 1982. С. 166— 167.
Смирнов А.А. Корреляционный анализ в системе остаточных классов. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 524-527.
Соборников Ю.П.. Об одном методе деления и определения мультипликативного переполнения для ЭВМ, работающих в СОК. “Журнал вычислительной математики и математической физики”, 5, №2, 1965.
Стемпковский А.Л., Корнилов А.И., Семенов М.Ю. Особенности реализации устройств цифровой обработки сигналов в интегральном исполнении с применением модулярной арифметики // Информационные тех-

нологии. – 2004. - Вып. 2. – С. 2-9.
Тейтельбаум В.Н.. Сравнение чисел в чешской системе счисления. ДАН СССР, 1958, т.121, №5.
Торгашев В.А. Система остаточных классов и надёжность ЦВМ. – М.: Советское радио, 1973. – 120 с.
Финъко О. А. Логические вычисления на основе теоретико-числовых преобразований // Вторая Междунар. конф. по проблемам управ. (МКПУ II). Москва, 16-20 июня 2003. Сборник трудов. - М-: Ин-т проблем управ, им. В. А. Трапезникова РАН, 2003. — Т. 2. — С. 159-166.
Финъко О. А. Параллельные логические вычисления, использующие избыточные представления чисел // Вторая Междунар. конф. „Идентификация систем и задачи управления" (SICPR.0'03). Москва, 29-31 января 2003. Сборник трудов (CD). М.: Ин-т проблем управ, им. В. А. Трапезникова РАН, 2003. С. 1716-1728.
Финько О. А. Проблемы арифметико-кодовой совместимости устройств обработки информации в изделиях ракетно-космической техники // Третья научно-техническая конф. „Перспективы использования новых технологий и научно-технических решений в изделиях ракетно-космической техники разработки ГКНПЦ им. М.В. Хруничева". Москва, 16-18 декабря 2003. Сборник трудов. М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2003. — С. 216-218.
Финько О. А. Арифметике-кодовая совместимость устройств обработки информации в распределенных АСУ при проблемно-ориентированном представлении данных // Вторая Междунар. конф. по проблемам управ. (МКПУ II). Москва, 17 19 июня 2003. Сборник трудов. — М.: Ин-т проблем управ, им. В. А. Трапезникова РАН, 2003. - С. 139.
Финько О. А. Вариант классификации арифметических форм представления логических функций // XIV Междунар. школа-семинар „Синтез и сложность управляющих систем". Н. Новгород, 27 октября — 1 ноября 2003. Сборник трудов / Под ред. академика РАН О. Б. Лупанова. — Н. Новгород: Изд-во Нижегородского педагогического ун-та, 2003. — С. 83-84.
Финько О. А. Восстановление числа в системе остаточных классов с минимальным количеством оснований // Электронное моделирование. - 1998. - Т. 20, № 3. - С. 56-61.
Финько О. А. и др. Методика защиты модулярных криптоироцессоров от аппаратных ошибок // Межвузовский сборник научных трудов. — Краснодар: Краснодарский военный ин-т, 2001. - С. 171-175.
Финько О. А. и др. Непозиционное представление данных в криптосистемах с открытым ключом // V научно-техническая конференция Ракетных Войск. Краснодар, 17—19 сентября 1997. Тезисы докладов. Часть I. - Краснодар: КВВКИУ РВ, 1997. С. 13.
Финько О. А. Контроль и реконфигурация аналого-цифровых устройств, функционирующих в системе остаточных классов // Электронное мо-

делирование. — 2000. — Т. 22, № 4. С. 92-103.
Финько О. А. Методика распараллеливания сложных операций в специализированных вычислителях АСУ при проблемно-ориентированном представлении данных: Дис. ... канд. техн. наук. — Краснодар, КВВКУ РВ, 1995. - 240 с. (Библиотека КВИ).
Финько О. А. Методы обработки больших массивов информации на основе арифметики в остаточных классах // Третья научно-техническая конф. „Перспективы использования новых технологий и научно-технических решений в изделиях ракетно-космической техники разработки ГКНПЦ им. М. В. Хруничева". Москва, 16—18 декабря 2003. Сборник трудов. — М.: Ин-т проблем управления им. В. А. Трапезникова РАН, 2003. - С. 211-216.
Финько О. А. Модулярные формы арифметических полиномов для реализации систем булевых функций // Междунар. конф. „Искусственные интеллектуальные системы" (IEEE AIS ' 03) и „Интеллектуальные САПР" (CAD-2003). Геленджик, 3—10 сентября 2003. Сборник трудов. - М.: Наука. Физматлит, 2002. — С. 548 560.
Финько О. А. Параллельная реализация оператора группового суммирования констант в позиционной арифметике // 5-й сессия Междунар. научно-технической школы-семинара „Передача, обработка и отображение информации". Сочи-Теберда, март-апрель 1998, Материалы. — Ставрополь: Ставропольский воен. авиац. ин-т, 1999.
Финько О. А. Параллельные логические вычисления методами модулярной арифметики // II Междунар. конф. „Параллельные вычисления и задачи управления" (PACO-2004). Москва, 4-6 октября 2004. Сборник трудов. -- М.: Ин-т проблем управ, им. В. А. Трапезникова РАН, 2004. (В печати).
Финько О. А. Полиномиальная арифметика функций многозначной логики по заданному модулю // Известия вузов. Приборостроение. — 2004. - № 3.
Финько О. А. Применение цифровой обработки сигналов для реализации интенсивных логических вычислений // 6-я Междунар. конф. ‚Дифровая обработка сигналов и ее применение" (DSPA-2004). Москва, 31 марта — 2 апреля 2004. Сборник трудов. — М.: Радиотехника, 2004. - Т. 1. - С. 265-268.
Финько О. А. Реализация систем булевых функций большой размерности методами модулярной арифметики // Автоматика и телемеханика. — 2004. — № 6. ~ С. 37-60. (Специальный выпуск; в печати).
Финько О. А. Ррупповой контроль ассиметричных криптосистем методами модулярной арифметики // XIV Междунар. школа-семинар „Синтез и сложность управляющих систем". Н. Новгород, 27 октября — 1 ноября 2003. Сборник трудов/ Под ред. академика РАН О. Б. Лупанова. — Н. Новгород: Изд-во Нижегородского педагогического ун-та, 2003. - С. 85-86.

Финько О. А. Сверхпараллельные логические вычисления методами модулярной арифметики // Междунар. конф. „Искусственные интеллектуальные системы" (IEEE AIS'02) и „Интеллектуальные САПР" (CAD-2002). Геленджик, 5-10 сентября 2002. Сборник трудов. — М.: Наука. Физматлит, 2002. — С. 448-155.

Финько О. А. Сверхпроводниковый аналого-цифровой преобразователь для устройств цифровой обработки сигналов, функционирующих в остаточных классах // 5-я Междунар. конф. ‚Дифровая обработка сигналов и ее применение" (DSPA-2003). Москва, 12-14 марта 2003. Сборник трудов. — М.: Радиотехника, 2003. — С. 575-579.

Финько О. А. Синтез арифметических форм булевых функций посредством теоретико-числовых преобразований // Перспективные информационные технологии и интеллектуальные системы. — Таганрог: Таганрогский гос. радиотехнический ун-т, 2003. — № 15. — С. 45- 53.

Финько О. А. Синтез параллельных электрооптических аналого-цифровых преобразователей для вычислителей, функционирующих в модулярной арифметике // Изв. ВУЗов. Приборостроение. - 1999. -Т. 42, № 3-4. - С. 30-32.

Финько О. А., Елесин С. Б. Принципы построения средств аппаратной поддержки криптосистем с открытым ключом // 4-й сессия Меж-дунар. научно-технической школы-семинара „Передача, обработка и отображение информации". Сочи-Теберда, март-апрель 1997, Материалы. — Ставрополь: Ставропольский воен. авиац. ин-т, 1998.

Финько О. А., Кузьменко А. С. Тестопригодное устройство цифровой обработки сигналов, функционирующее в остаточных классах с простым средством контроля // 5-я Междунар. конф. ‚Дифровая обработка сигналов и ее применение" (DSPA-2003). Москва, 12—14 марта 2003. Сборник трудов. — М.: Радиотехника, 2003. — С. 571-574.

Финько О. А., Кузьменко А.С. Простой способ контроля тестопри-годного устройства цифровой обработки сигналов, функционирующего в остаточных классах // Приборы и системы. Управление, контроль, диагностика. 2004. - № 1. - С. 37-39.

Финько О.А. Многоканальные модулярные системы, устойчивые к искажениям криптограмм. /// Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 545-551.

Финько О.А. Многоканальные системы, устойчивые к искажениям криптограмм // В коллект. моногр.: Методы криптозащиты в средствах радиосвязи с различной степенью защищенности / Под. ред. Е.М. Сухарева. Кн.4. — М.: Радиотехника, 2005.

Финько О.А. Параллельные логические вычисления – прикладная область модулярной арифметики. /// Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Рос-

сия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 187-210.

Финько О.А. Варианты Китайской теоремы об остатках, ориентированные на техническую реализацию// Междунар. конгресс «Математика в XXI в. Роль механико-математического факультета Новосибирского гос. ун-та в науке, образовании и бизнесе». Новосибирск. Академгородок, 25–28 июня 2003. Тез. докл. http://www.sbras.ru/ws/MMF-21/.

Финько О.А. Восстановление числа в системе остаточных классов с минимальным количеством оснований // Электронное моделирование. — 1998. — Т.20, №3. — С.56–61. ISSN 0204–3572. / Finko O.A. Number Restoration in the System of Residual Classes With a Minimum Number of Radices // Engineering Simulation. — 1999. — Vol. 16. — P. 329—334 (in USA).

Финько О.А. Групповой контроль ассиметричных криптосистем методами модулярной арифметики// XIV Междунар. школа-семинар «Синтез и сложность управляющих систем». Н. Новгород, 27 октября — 1 ноября 2003. Сборник трудов/ Под ред. академика РАН О.Б. Лупанова. — Н. Новгород: Изд-во Нижегородского педагогического ун-та, 2003. — С. 85–86.

Финько О.А. Контроль и реконфигурация аналого-цифровых устройств, функционирующих в системе остаточных классов// Электронное моделирование. — 2000. — Т.22, №4. — С. 92–103. ISSN 0204–3572. / Finko O.A. Check and Reconfiguration of Analog-to-Digital Devices Operating in the System of Residual Classes // Engineering Simulation. — 2001. — Vol. 18. — P. 631–543 (in USA).

Финько О.А. Логические вычисления на основе теоретико-числовых преобразований // Вторая Междунар. конф. по проблемам управ. (МКПУ II). Москва, 16–20 июня 2003. Избранные труды в двух томах. Том 2. — М.: ИПУ РАН, 2003. — Т. 2. — С. 159–166. ISBN 5–201–14967–7.

Финько О.А. Методы обработки больших массивов информации на основе арифметики в остаточных классах// Третья научно-техническая конф. «Перспективы использования новых технологий и научно-технических решений в изделиях ракетно-космической техники разработки ГКНПЦ им. М.В. Хруничева». Москва, 16–18 декабря 2003. Сборник трудов. — М.: ИПУ РАН, 2003. — С. 211—216.

Финько О.А. Модулярная арифметика параллельных логических вычислений: Монография / Под. ред. В.Д. Малюгина; — М.: ИПУ РАН, 2003. — 224 с.

Финько О.А. Модулярные формы арифметических полиномов для реализации систем булевых функций // Междунар. конф. «Искусственные интеллектуальные системы» (IEEE AIS'03) и «Интеллектуальные САПР» (CAD-2003). Геленджик, 3–10 сентября 2003. Сборник трудов. Том 2. — М.: Наука. Физматлит, 2002. — С. 548–560. ISBN 5–9221–

0447–0.
Финько О.А. Модулярные формы арифметической логики// Междунар. конф. «Теория и практика логического управления», посвященная 100-летию со дня рождения члена-корреспондента АН СССР М.А. Гаврилова. Москва, 10–11 ноября 2003. Сборник тр./ Под ред. А.А. Амбарцумяна. — М.: ИПУ РАН, 2003. — С. 110–114.
Финько О.А. Модулярные формы систем k-значных функций алгебры логики // Автоматика и телемеханика. — 2005. — №7. ISSN 0005–2310.
Финько О.А. Параллельные логические вычисления методами модулярной арифметики// II Междунар. конф. «Параллельные вычисления и задачи управления» (PACO-2004). Москва, 4–6 октября 2004. Сборник трудов (CD). — М.: ИПУ РАН, 2004. — С. 1120–1207. ISBN 5–201–14974–X.
Финько О.А. Поисковые методы гибкой параллельной достоверной реализации логических функций криптографических алгоритмов // В коллект. моногр.: Методы криптозащиты в средствах радиосвязи с различной степенью защищенности / Под. ред. Е.М. Сухарева. Кн.4. — М.: Радиотехника, 2005.
Финько О.А. Полиномиальная арифметика функций многозначной логики// Изв. вузов. Приборостроение. — 2004. — Т.47, №5. — С. 41–46. ISSN 0021–3454.
Финько О.А. Применение цифровой обработки сигналов для реализации интенсивных логических вычислений// Тр. Российского НТОРЭС им. А.С. Попова. Серия: Цифровая обработка сигналов и ее применение. Вып.: VI–1. Междунар. конф. DSPA-2004. — М.: Радиотехника, 2004. — Том1. — С. 265–268.
Финько О.А. Реализация систем булевых функций большой размерности методами модулярной арифметики // Автоматика и телемеханика. — 2004. №6. — С. 37–60. ISSN 0005–2310.
Финько О.А. Сверхпараллельные логические вычисления методами модулярной арифметики // Междунар. конф. «Искусственные интеллектуальные системы» (IEEE AIS'02) и «Интеллектуальные САПР» (CAD-2002). Геленджик, 5–10 сентября 2002. Сборник трудов. — М.: Наука. Физматлит, 2002. — С. 448–455. ISBN 5–94052–031–6.
Финько О.А. Сверхпроводниковый аналого-цифровой преобразователь для устройств цифровой обработки сигналов, функционирующих в остаточных классах // Тр. Российского НТОРЭС им. А.С. Попова. Серия: Цифровая обработка сигналов и ее применение. Вып.: V–2. Междунар. конф. DSPA-2003. — М.: Радиотехника, 2003. — С. 575–579. http://www.autex.spb.ru/dspa/.
Финько О.А. Синтез арифметических форм булевых функций посредством теоретико-числовых преобразований// Перспективные информационные технологии и интеллектуальные системы. — Таганрог: Таганрогский гос. радиотехнический ун-т, 2003. — №15. — С. 45–53.

Финько О.А. Синтез параллельных электрооптических аналого-цифровых преобразователей для вычислителей, функционирующих в модулярной арифметике // Изв. вузов. Приборостроение. — 1999. — Т.42, № 3–4. — С. 30–32. ISSN 0021–3454.

Финько О.А., Кузьменко А.С. Простой способ контроля тестопригодного устройства цифровой обработки сигналов, функционирующего в остаточных классах // Приборы и системы. Управление, контроль, диагностика. — 2004. — №1. — С. 37–39.

Финько О.А., Кузьменко А.С. Тестопригодное устройство цифровой обработки сигналов, функционирующее в остаточных классах с простым средством контроля // Тр. Российского НТОРЭС им. А.С. Попова. Серия: Цифровая обработка сигналов и ее применение. Вып.: V–2. Междунар. конф. DSPA-2003. — М.: Радиотехника, 2003. — С. 571–574. http://www.autex.spb.ru/dspa/.

ФинькоО.А. Варианты Китайской теоремы об остатках, ориентированные на техническую реализацию // Междунар. конгресс „Математика в XXI в. Роль механико-математического факультета Новосибирского гос. ун-та в науке, образовании и бизнесе". Новосибирск. Академгородок, 25—28 июня 2003. Тез. докл. http://www.sbras.ru/ws/MMF-21/.

Фурман И.А., Кошман С.А., Краснобаев В.А. Вариант синтеза процессора в системе остаточных классов // Радиоэлектроника и информатика. 2003. Вып. № 2(23). С. 94-96.

Фурман И.А., Краснобаев В.А. Новые возможности использования системы счисления в остаточных классах для построения высокоэффективных устройств обработки данных и управления // Вісник ХДТУСГ. 2000. Вип. 3.С. 27 – 31.

Фурман І.О., Краснобаєв В.А., Кошман С.О. Аналіз табличних алгоритмів реалізації модульних операцій в атоматизованих системах обробки цифрової інформації // Вісник ХДТУСГ імені Петра Василенка. Вип. 27. Том 2. 2004. С. 174 - 178.

Хацкевич В. Х., Чачанашвили А. Р. *Базисные представления для одного класса непозиционных систем*. Сб. материалов «Математическая и техническая кибернетика», «Мецниереба», Тбилиси, 1977.

Харинов М.В. Недвоичная логика запоминания информации в изображении. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 635-642.

Хацкевич В. Х. Вопросы построения избыточных непозиционных кодов // Вопр. вычислительной техники и управлени». 1978. Т. 18, № 2. С. 192 — 196.

Хацкевич В. Х. О расширении диапазона представления чисел для одного класса непозиционных кодов//Мат. и техн. кибернетика. Тбилиси: Мецниеребата, 1975. С. 45 — 63.

Хацкевич В. Х. *Адаптивный метод нахождения интегральных*

характеристик при непозиционных параллельных вычислениях. Сб. «Вопросы разработки и применения средств вычислительной техники». Материалы республиканской конференции, Тбилиси, 1982.

Хацкевич В. Х. *Базисные представления непозиционных систем в метрологических задачах*. Материалы пятой республиканской научно-технической конференции по метрологии, Тбилиси, 1978.

Хацкевич В. Х. *Быстрое умножение для высокопроизводительных систем*. Труды Международной конференции «Высокопроизводительные вычислительные системы в управлении и научных исследованиях», Москва, 1991.

Хацкевич В. Х. *Вопросы построения избыточных непозиционных кодов*. Сб. «Вопросы вычислительной техники и управления». Изд. «Мецниереба», Тбилиси, 1978.

Хацкевич В. Х. *Вопросы работы с приближенными числами в системе остаточных классов*. Сообщ. АН ГССР., Тбилиси, 1967.

Хацкевич В. Х. *Избыточность для декодирования на основе алгоритмов с модулярной арифметикой*. В сб. «Шестой международный симпозиум по теории информации». Часть II, Москва-Ташкент, 1984.

Хацкевич В. Х. *Избыточность для оптимизации специальных арифметических свойств кода*. Труды IX симпозиума по проблеме избыточности в информационных системах, Ленинград, 1986.

Хацкевич В. Х. *К вопросу о достоверности передачи информации между компонентами гибридных вычислительных комплексов*. Сб. «Гибридные вычислительные машины и комплексы». «Наукова думка», Киев, 1973.

Хацкевич В. Х. *Корректирующие возможности некоторых базисных представлений непозиционных систем*. Сб. «IV Всесоюзная школа-семинар по вычислительным сетям», Москва-Ташкент, 1979.

Хацкевич В. Х. *Корректирующие возможности сопряженных последовательностей парных произведений*. Сб. «X симпозиум по проблеме избыточности в информационных системах», Ленинград, 1989.

Хацкевич В. Х. *Минибазисные модели непозиционных систем*. Сб. «Математическая и техническая кибернетика», «Мецниереба», Тбилиси, 1977.

Хацкевич В. Х. *Модификации метода быстрого умножения для высокопроизводительных систем*. Сб. «Математическая и техническая кибернетика», «Мецниереба», Тбилиси, 1987.

Хацкевич В. Х. *Модификация криптографической системы для вычислительных сетей*. Сб. «V Всесоюзная школа-семинар по вычислительным сетям», Москва-Владивосток, 1980.

Хацкевич В. Х. *Модульная арифметика в системе остаточных классов*. Труды ТНИИСА, IX т., Тбилиси, 1969.

Хацкевич В. Х. *Некоторые методы декодирования в системе счисления в*

остаточных классах. Сб. «III Международный симпозиум по теории информации», Москва-Тбилиси, 1979.

Хацкевич В. Х. *О использовании структуры представления информации для защиты арифметической обработки в сетевых системах*. Труды XII Всесоюзной школы-семинара по вычислительным сетям, Москва-Одесса, 1987.

Хацкевич В. Х. *О неортогональных базисных системах непозиционных представлений чисел*. Сообщ. АН ГССР., Тбилиси, 1971.

Хацкевич В. Х. *О расширении диапазона представления чисел для одного класса непозиционных кодов*. Сб. «Математическая и техническая кибернетика», Мецниереба, Тбилиси, 1975.

Хацкевич В. Х. *О решении целочисленных линейных оптимизационных задач в остаточных классах*. Сб. «Математическая и техническая кибернетика». Изд. «Мецниереба», Тбилиси, 1981.

Хацкевич В. Х. *Об арифметических возможностях одного класса непозиционных кодов*. Сб. «Управл. выч. Машины». Энергия, М. – Л., 1967.

Хацкевич В. Х. *Об одном классе неортогональных базисных систем в СОК*. «Сборник научных трудов МО #75», Москва, 1967.

Хацкевич В. Х. *Переходные параметры ступенчатых непозиционных систем*. Сб. «Математическая и техническая кибернетика», т. 24, вып. 2, Тбилиси, 1984.

Хацкевич В. Х. *Сопряженные по рангу числовые представления*. Сб. «IX Всесоюзная конференция по теории кодирования и передачи информации», часть I, Одесса, 1988.

Хацкевич В. Х. *Специальные базисные представления непозиционных систем для метрологических задач*. Труды института вычислительной математики. «Математическая и техническая кибернетика», 1984.

Хацкевич В. Х., Даниленко П. Я. *К вопросу о повышении эффективности работы электронных цифровых систем путем применения методов непозиционного кодирования*. ВРЭ #9, Москва, 1969.

Хацкевич В. Х., Мгебришвили М. Н. *О некоторых кодовых произведениях в вычислительных сетях*. Сб. «IV Всесоюзная школа-семинар по вычислительным сетям», Москва-Ташкент, 1979.

Хацкевич В. Х., Ревазишвили Г. Г. Система Меркля-Хэллмэна с распараллеливанием вычислений. Сб. «VIII Всесоюзная школа-семинар по вычислительным сетям», Москва, 1983.

Хацкевич В. Х., Хескелл Л. Функциональная избыточность в модулярной арифметике и сопряженные задачи. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 35-46.

Хацкевич В. Х., Чачанашвили А. Р. *Вопросы построения однопараметрических двуступенчатых непозиционных кодов*. Сб.

«Математическая и техническая кибернетика», «Мецниереба» Тбилиси, 1979.

Хацкевич В. Х., Чачанашвили А. Р. *Избыточное представление чисел для матричных компонент вычислительных сетей*. Сб. «Вопросы кибернетики, кодирования и передачи информации в вычислительных сетях». АН СССР. Выпуск 42., Москва, 1978.

Хацкевич В. Х., Чачанашвили А. Р. *Устройство для определения знака числа в системе остаточных классов*. Авторское свидетельство на изобретение #1254480, Государственный реестр изобретений СССР, Бюллетень изобретений от 1.5.1986.

Хацкевич В. Х., Шакарян Р. А. *О применении одной модификации циклического AN-кода для построения модульной арифметики непозиционных систем*. Сб. «Вопросы вычислительной техники», Тбилиси, 1979.

Хацкевич В. Х., Шакарян Р. А. *О применении элементов теории вычетов для измерения линейных величин*. «Материалы второй II научной республиканской конференции по метрологии. т. I», Тбилиси, 1974.

Хацкевич В. Х., Шакарян Р. А. *Применение нормальных инверсных представлений в задачах цифрового регулирования*. Сб. «Материалы VI конференции по метрологии», Тбилиси, 1982.

Хацкевич В.Х., Чачанашвили А.Р.. Базисные представления чисел для одного класса непозиционных кодов//Мат. и техн. кибернетика. Тбилиси: Мецниеребата, 1977. С. 95 — 105.

Червяков Н. И. Отказоустойчивые непозиционные процессоры // Управляющие системы и машины. — 1988. — №3. – С. 3-7.

Червяков Н. И., Велигороша А.В., Калмыков И.А. и др. Цифровые фильтры в системе остаточных классов // Теоретическая радиотехника. — 1995. — Вып. 38, № 8. - С. 11-20.

Червяков Н. И., Евдокимов А. А. Динамическая система пролонгированной безопасности // Инфокоммуникационные технологии. – Самара: Изд-во ПГАТИ, 2004, № 4. – С. 31 – 35.

Червяков Н. И., Евдокимов А. А. Локализация ложных частей секрета в пороговой криптографической схеме и ее нейросетевая реализация// Материалы XVII НТК «Совершенствование цифровых интегральных сетей на основе применения волоконно-оптических систем передачи». – Ставрополь: Филиал РВИ РВ, 2004. – С. 68.

Червяков Н. И., Евдокимов А. А. Многоместные сумматоры формальных нейронов конечного кольца// Компьютерная техника и технология: Сб. трудов регион. науч.-техн. конф. Ставрополь: СевКавГТУ, 2003. – С. 76 – 80.

Червяков Н. И., Евдокимов А. А. Нейрокомпьютерные средства для решения задач пролонгированных криптосистем// Проблемы физико-математических наук: Материалы 48 научно-методической конференции преподавателей и студентов «Университетская наука – региону». –

Ставрополь: Изд-во СГУ, 2003. – С. 110 – 113
Червяков Н. И., Евдокимов А. А. Нейросетевой блок локализации ошибок криптографического нейропроцессора// Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2004, № 10. – С. 54 – 61.
Червяков Н. И., Евдокимов А. А. Нейросетевой генератор криптографических ключей пороговой схемы разделения секрета// Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2004, № 10. – С. 62 – 67.
Червяков Н. И., Евдокимов А. А. Пороговое разделение файла на базе китайской теоремы об остатках // Инфокоммуникационные технологии. – Самара: Изд-во ПГАТИ, 2004, № 1. – С. 38 – 43.
Червяков Н. И., Евдокимов А. А. Структурно-разрядный синтез нейронных сетей в $GF(p^n)$ в задаче обновления криптографических ключей// Проблемы физико-математических наук: Материалы 48 научно-методической конференции преподавателей и студентов «Университетская наука – региону». – Ставрополь: Изд-во СГУ, 2003. – С. 114 – 117.
Червяков Н. И., СахнюкП.А. и др. Модулярные параллельные вычислительные структуры нейропроцессорных систем. — М.: Физматлиг, 2003. — 288 с.
Червяков Н.И. Методы и принципы построения модулярных нейрокомпьютеров. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 232-242.
Червяков Н.И. Преобразователи цифровых позиционных и непозиционных кодов в системах управления и связи. – Ставрополь: СВВИУС, 1985. – 64 с.
Червяков Н.И. Преобразователи цифровых позиционных и непозиционных кодов в системах управления и связи / Учебное пособие для курсового и дипломного проектирования. Ставрополь, 1985.
Червяков Н.И. Применение системы остаточных классов в цифровых системах обработки и передачи информации. – Ставрополь: СВВИУС, 1984. – 84 с.
Червяков Н.И. Применение системы остаточных классов в цифровых системах обработки и передачи информации / Учебное пособие для курсового и дипломного проектирования. Ставрополь, 1984.
Червяков Н.И. Проблемные вопросы развития архитектуры нейрокомпьютера на основе свойств биологических нейронов // Материалы научной конференции с международным участием (21-28 февраля 2004). Хургада, Египет. Успехи современного естествознания, Москва “Академия естествознания”, №2, 2004.
Червяков Н.И., Бережной В.В., Оленев А.А., Калмыков И.А. Минимизация избыточности кода системы остаточных классов с одним контрольным основанием// Электронное моделирование. 1994. №1. Т.16.

С.56-61.
Червяков Н.И., Бережной В.В., Гончарова Е.Н., Калмыков И.А. Локализация и исправление арифметических ошибок в модулярных нейрокомпьютерах / Нейрокомпьютеры: разработка, применение. №7, 2003. С. 28-32.
Червяков Н.И., Горденко Д.В., Галкина В.А., Лавриненко С.В. Применение AN-кодов для коррекции ошибок в системе остаточных классов // Методы и алгоритмы прикладной математики в технике, медицине и экономике. Материалы IV Международной научно-практической конференции. Часть 1. Новочеркасск, 2004.
Червяков Н.И., Дьяченко И.В. Принципы построения модулярных сумматоров и умножителей. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 490-502.
Червяков Н.И., Евдокимов А. А. Локализация ложных частей секрета в пороговой криптографической схеме и ее нейросетевая реализация// Материалы XVII НТК «Совершенствование цифровых интегральных сетей на основе применения волоконно-оптических систем передачи». – Ставрополь: Филиал РВИ РВ, 2004. – С. 68.
Червяков Н.И., Евдокимов А. А. Нейрокомпьютерные средства для решения задач пролонгированных криптосистем// Проблемы физико-математических наук: Материалы 48 научно-методической конференции преподавателей и студентов «Университетская наука – региону». – Ставрополь: Изд-во СГУ, 2003. – С. 110 – 113
Червяков Н.И., Евдокимов А.А. Динамическая система пролонгированной безопасности // Инфокоммуникационные технологии. – Самара: Изд-во ПГАТИ, 2004, № 4. – С. 31–35.
Червяков Н.И., Евдокимов А.А. Нейросетевой блок локализации ошибок криптографического нейропроцессора// Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2004, № 10. – С. 54 – 61.
Червяков Н.И., Евдокимов А.А. Нейросетевой генератор криптографических ключей пороговой схемы разделения секрета// Нейрокомпьютеры: разработка, применение. – М.: Радиотехника, 2004, № 10. – С. 62 – 67.
Червяков Н.И., Евдокимов А.А. Пороговое разделение файла на базе китайской теоремы об остатках // Инфокоммуникационные технологии. – Самара: Изд-во ПГАТИ, 2004, № 1. – С. 38 – 43.
Червяков Н.И., Калмыков И.А., Велигоша А.В., Иванов П.Е. Цифровые фильтры в системе остаточных классов / Радиоэлектроника. Т.38. №8, 1995,С.11-20.
Червяков Н.И., Калмыков И.А., Галкина В.А., Щелкунова Ю.О., Шилов А.А.. Элементы применения компьютерной математики и нейроинформатики/ Под ред. Н.И. Червякова. – М.: ФИЗМАТЛИТ, 2003. – 216с.

Червяков Н.И., Калмыков И.А., Щелкунова Ю.О., Бережной В.В. Математическая модель нейронной сети для коррекции ошибок в непозиционном коде расширенного поля Галуа / Нейрокомпьютеры: разработка, применение. №8-9, 2003. С. 10-16.

Червяков Н.И., Калмыков И.А., Щелкунова Ю.О., Бережной В.В. Математическая модель нейронных сетей для исследования ортогональных преобразований в расширенных полях Галуа/Нейрокомпьютеры: разработка, применение. №6, 2003. с.61-68.

Червяков Н.И., Копыткова Л.Б., Непретимова Е.Н., Сахнюк П.А. Шапошников А.В. и др. Нейрокомпьютеры в системах обработки сигналов. Коллективная монография./ Под редакцией Гуляева Ю. Лушкина А.И. – М: Радиотехника, 2003. – 224 с.

Червяков Н.И., Краснобаев В. А. Методы повышения надежности спецпроцессоров АСУС. - Ставрополь : МО СССР, 1991. - 115 с.

Червяков Н.И., Краснобаев В. А. Функциональные блоки и узлы отказоустойчивых и высокопроизводительных систем - Ставрополь : МО СССР, 1989. - 96 с.

Червяков Н.И., Краснобаев В. А., Квасов М. В. И др. Основы цифровой техники систем управления и связи. - Ставрополь : МО СССР, 1989. - 86 с.

Червяков Н.И., Лаврищенко И.Н., Лаврищенко С.В., Мезенцева О.С. Методы и алгоритмы округления, масштабирования и деления чисел в модулярной арифметике. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 284-303.

Червяков Н.И., Ремизов С.Л. Локализация ошибки на основе метода расширенной проекции. // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 540-544.

Червяков Н.И., Сахнюк П. А., Шапошников А. В., Макоха А. Н. Нейрокомпьютеры в остаточных классах. Кн. 11: Учеб. пособие для вузов. – М.: Радиотехника, 2003. – 272 с.

Червяков Н.И., Сахнюк П.А. и др. Нейрокомпыотеры в остаточных классах. Кн. 11. — М.: Радиотехника, 2003. 272 с.

Червяков Н.И., Сахнюк П.А. Применение нейроматематики для реализации вычислений в конечных кольцах // Нейрокомпьютеры: разработка, применение, 1999. № 1. С. 75-84.

Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н. Нейрокомпьютеры в остаточных классах / Учебное пособие для вузов – М.: Радиотехника, 2003. – 272 с.

Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А. Модулярные параллельные вычислительные структуры нейропроцессорных систем // М.: Физматлит, 2003. – 288 с.

Червяков Н.И., Сивоплясов Д.В., Галкина В.А., Лавриненко С.В. Нейронная сеть для исправления ошибок данных, представленных в модулярном коде // Методы и алгоритмы прикладной математики в технике, медицине и экономике. Материалы IV Международной научно-практической конференции. Часть 1. Новочеркасск, 2004.

Червяков Н.И., Ткачук Р.В., Галкина В.А., Лавриненко С.В. Использование “блуждающих” ключей в распределенных вычислительных сетях // Методы и алгоритмы прикладной математики в технике, медицине и экономике. Материалы IV Международной научно-практической конференции. Часть 1. Новочеркасск, 2004.

Червяков Н.И., Шалин Б.С. Коррекция ошибок в модулярных нейрокомпьютерах // Нейрокомпьютеры: разработка, применение. №5-6, 2004.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Нейронный цифровой фильтр с модулярной обработкой данных // Нейрокомпьютеры: разработка, применение. – 2002. – № 11.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. Нейронный алгоритм расширения оснований модулярного кода // Нейрокомпьютеры: разработка, применение. – 2002. – № 11.

Червяков Н.И., Шапошников А.В., Сахнюк П.А., Калмыков И.А. Применение модулярных вычислений для нейрообработки // Труды VIII Всероссийской конференции “Нейрокомпьютеры и их применение” НКП – 2002. Под ред. проф. А. Галушкина. М., 2002. С. 1053-1056.

Червяков Н.И., Швецов Н.И, Хлевной С.Н. Устройство для определения ранга числа. А.С. № 1125619, БИ № 43,1984.

Червяков Н.И. Алгоритм разработки самопроверяемых схем встроенного контроля арифметических устройств, функционирующих в СОК // НТК. – М.: ВАД, 1992. – 1с

Червяков Н.И. Арифметическое устройство в системе остаточных классов. А.С. № 857992, БИ № 31, 1981.

Червяков Н.И. Геометрическая модель избыточного кода системы остаточных классов // Управляющие системы и машины. – 1988. – № 6. – С. 36-139.

Червяков Н.И. и др. Нахождение нормированной системы остаточных классов для заданного фильтра // Материалы VI НТК Вузовская наука – Северо-Кавказскому региону. Ч. 2. - Ставрополь, 2002. – С. 42

Червяков Н.И. Ирхин В.П. и др. Отказоустойчивость специализированных процессоров автоматизированных систем управления и связи. – Ставрополь: СВВИУС, 1991. – 115 с

Червяков Н.И. Корректирующие коды в системе остаточных классов: метод проекций // XV НТК, Проблемы совершенствования АСБУ и связи РВСН. – Ставрополь, 2002. – С. 58.

Червяков Н.И. Нейронная сеть для расширения кортежа числовой системы вычетов. № 2256226, Бюл. 19, 2005
Червяков Н.И. Нейросетевой метод управления трафиком при сопряжении ЛВС и АТМ-сети // Телекоммуникации. – 2002. – № 2.
Червяков Н.И. Непозиционные системы счисления в ЭВМ. – Ставрополь: СВВИУС, 1969. – 24 с.
Червяков Н.И. О надежности микропроцессорных систем высокой производительности с распределенной обработкой данных // III НТК. – МО СССР, 1987. – 2 с
Червяков Н.И. Об одном способе преобразования чисел из двоичной системе счисления в систему остаточных классов // НТСб № 1. – Ставрополь: СВВИУС, 1970. – С. 16-21.
Червяков Н.И. Организация арифметических расширителей в микропроцессорных системах, базирующихся на множественном представлении информации // Управляющие системы и машины. – 1987. – № 1. – С. 26-29.
Червяков Н.И. Повышение отказоустойчивости процессора в системе остаточных классов путем перераспределения обрабатываемых им данных // Помехоустойчивость и эффективность систем связи и управления. Вып. № 5. – Ставрополь: СВВИУС, 1987. – С. 59-63.
Червяков Н.И. Преобразователь двоичного кода в код системы остаточных классов. А.С. № 374596, БИ № 15, 1973.
Червяков Н.И. Преобразователь кода из системы остаточных классов в позиционный код. А.С. 1388996, БИ № 14, 1988.
Червяков Н.И. Преобразователь полиадического кода в код системы остаточных классов. А.С. № 809154, БИ № 8, 1981.
Червяков Н.И. Сравнительная оценка алгоритмов преобразования кодов остаточных классов в непозиционных процессорах // XVIII НТК. – Киев, 1991. – 1 с
Червяков Н.И. Ускоренный алгоритм определения позиционных характеристик и его нейросетевая реализация // Нейрокомпьютеры: разработка, применение. – 2001. – № 10. – С. 19-25
Червяков Н.И. Функциональные представления параметров арифметического устройства, функционирующего в системе остаточных классов // Помехоустойчивость и эффективность систем связи и управления. Вып. 6. – Ставрополь: СВВИУС, 1987. – С. 45-52
Червяков Н.И., Балан М.В. О нейронном алгоритме расширения оснований модулярного кода // Физико-математические науки в Ставропольском государственном университете. Материалы 50-й юбилейной научно-методической конференции преподавателей и студентов СГУ “Университетская наука – региону”, посвященная 60-летию Победы в ВОВ (5-25 апреля 2005 г.)
Червяков Н.И., Бережной В.В. К вопросу о выборе объема контролируемого оборудования устройств, функционирующих в СОК // Помехоустойчивость и эффективность систем связи и управления, вып. 9. – Ставрополь:

СВВИУС, 1991. – 4 с.
Червяков Н.И., Бережной В.В. Метод синтеза встроенного контроля для унитарного кода на основе ПЛМ // Помехоустойчивость и эффективность систем связи и управления, вып. 9. – Ставрополь: СВВИУС, 1991. – 6 с.
Червяков Н.И., Бережной В.В., Оленев А.А. Алгоритм коррекции ошибок в числах, представленных остаточным кодом системы остаточных классов // Межведомственный тематический научный сборник “Синтез алгоритмов сложных систем”, вып. 8. – Таганрог, 1992. – 7 с.
Червяков Н.И., Болтков А.П. Оптимизация структуры комбинационного преобразователя чисел ПСС в СОК по критерию минимальных аппаратурных затрат // Депонирована МО, № 630, вып. 2, 1987.
Червяков Н.И., Болтков А.П., Финько О.Е. Об одном алгоритме преобразования чисел из СОК в ПСС без выхода за пределы диапазона // Депонирована МО, № 627, вып. 2, 1987.
Червяков Н.И., Велигоша А.В. Алгоритм формирования остатка от числа // Тем. НТСб. № 12. - Ставрополь: СВВИУС, 1994. – С. 12-15.
Червяков Н.И., Велигоша А.В. и др. Отказоустойчивый цифровой фильтр обработки комплексных данных // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с.
Червяков Н.И., Велигоша А.В. и др. Показатели для оценки устойчивости функционирования непозиционного процессора с деградируемой структурой // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с.
Червяков Н.И., Велигоша А.В. и др. Пути повышения быстродействия цифровых фильтров на основе использования СОК // Сборник рефератов депонирования. Вып. 28. – МО, 1994. – 5 с.
Червяков Н.И., Велигоша А.В. Новый класс ЦФ на основе непозиционной арифметики // Тематический сборник. – Орел: ВИПС, 1995. – С. 17-27
Червяков Н.И., Велигоша А.В. Обнаружение ошибок в избыточной системе остаточных классов // Тем. НТСб. № 12. – Ставрополь: СВВИУС, 1994. – С. 16-18.
Червяков Н.И., Велигоша А.В. Принципы построения АУ ЦФ // Сборник тезисов 2-го международного Н.Т.С. Туапсе. – 1994. – С. 80.
Червяков Н.И., Велигоша А.В., Калмыков И.А. Анализ распределения задач в вычислительных системах, работающих в СОК, методами теории массового обслуживания // VII НТК. – Ставрополь, 1993. – 1 с
Червяков Н.И., Велигоша А.В., Калмыков И.А. Повышение отказоустойчивости СП, выполняющего параллельно-конвейерные вычисления, на остове использования метода реконфигурации // VII НТК. – Ставрополь, 1993. – 1 с.
Червяков Н.И., Велигоша А.В., Калмыков И.А. Таблица АУ звена цифрового фильтра // Сборник докладов Всероссийской научной конференции. – Воронеж: ВНИИС, 1996. – С. 1072-1082
Червяков Н.И., Велигоша А.В., Тынчеров К.Т. и др. Применение модулярного кодирования для синтеза высокоскоростных цифровых фильтров // Кибернетика и системный анализ. – 1998. – № 2.
Червяков Н.И., Велигоша А.В., Тынчеров Т.К. Однокристальный процессор СОК

для теоретико-числового преобразования // Материалы НТКА, 1992
Червяков Н.И., Галкина В.А., Стрекалов Ю.А., Лавриненко С.В.Архитектура адаптивной параллельно-конвейерной нейронной сети для коррекции ошибок в модулярных нейрокомпьютерных системах // М.: Нейрокомпьютеры: разработка, применение. – № 6 – 2003
Червяков Н.И., Галкина В.А., Стрекалов Ю.А., Лавриненко С.В.Нейронная сеть прямого распространения для обработки данных в конечных кольцах // Электромагнитная базопасность и защищенность инфокоммуникационных систем. Приложение к журналу "Инфокоммуникационные технологии". Выпуск 1. – Самара, ПГАТИ, 2005.
Червяков Н.И., Дъяченко И.В., Лавриненко И.Н., Лавриненко С.В., Кондрашов А.В. Эффективные методы обработки данных при множественном их представлении в модулярных нейрокомпьютерах. // Нейрокомпьютеры: разработка, применение. – 2005. № 7 – С. 51-63
Червяков Н.И., Евдокимов А.А. Пороговое разделение файла на базе китайской теоремы об остатках // Инфокоммуникационные технологии. – 2004. –№ 1.
Червяков Н.И., Зайцев А.Н. Преобразователь кодов из системы остаточных классов в двоичный позиционный код. А.С. № 813408, БИ № 10, 1981.
Червяков Н.И., Калмыков И.А. и др. Исследование и обоснование применения активной реконфигурации как способа повышения отказоустойчивости АУ СП СОК // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с.
Червяков Н.И., Калмыков И.А. К вопросу о выборе рациональных модулей СОК // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с
Червяков Н.И., Калмыков И.А. Отказоустойчивый спецпроцессор с изменяющейся структурой // VI НТК. – Ставрополь, 1992 – 1 с.
Червяков Н.И., Калмыков И.А. Разработка и исследование имитационной модели непозиционного процессора с деградируемой структурой // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с.
Червяков Н.И., Калмыков И.А. Спецпроцессор АСС с многоступенчатой структурой СОК // VI НТК. – Ставрополь, 1992 – 1 с
Червяков Н.И., Квасов М.В. Параллельный алгоритм прямого и обратного преобразования кодов в остатках // I Всесоюзная научно-техническая школа-семинар "Компьютерные методы …". – Харьков, 1991. – 3 с
Червяков Н.И., Квасов М.В., Микула Н.П. Организация микропроцессорных обменов в специализированных распределенных вычислительных системах // Помехоустойчивость и эффективность систем связи и управления, вып. 7. – Ставрополь: СВВИУС, 1989. – 7 с.
Червяков Н.И., Козленко А.В. Устройство для обнаружения ошибок в модулярном коде. А.С. № 1295528, БИ № 9,1987.
Червяков Н.И., Копыткова Л.Б. Масштабирование чисел, представленных в системе остаточных классов // Вестник СГУ. Выпуск 38, 2004
Червяков Н.И., Копыткова Л.Б. Способы представления комплексных чи-

сел в системе остаточных классов // Физико-математические науки в Ставропольском государственном университете. Материалы 50-й юбилейной научно-методической конференции преподавателей и студентов СГУ “Университетская наука – региону”, посвященная 60-летию Победы в ВОВ (5-25 апреля 2005 г.)

Червяков Н.И., Копыткова Л.Б. Номографические методы выполнения немодульных операций в СОК // Проблемы физико-математических наук. Материалы XLIII научно-методической конференции. – 1998. – С. 99-104

Червяков Н.И., Копыткова Л.Б., Непритимова Е.А. Нейронные цифровые фильтры с постепенной деградацией их структуры // Нейрокомпьютеры: разработка, применение. – 2001. – № 10.

Червяков Н.И., Копыткова Л.Б., Непритимова Е.В. и др. Применение вычетов для представления и обработки данных // Вестник СГУ. Вып. 18. Ф.-м. науки. – Ставрополь: изд-во СГУ, 1999. – С. 64-72.

Червяков Н.И., Копыткова Л.Б., Непретилова Е.В. и др. Организация микропроцессорных обменов в естественно-надежных специализированных вычислительных системах, базирующихся на множественном представлении информации // Вестник СГУ. – Ставрополь: СГУ, 1999. – № 20. – С. 61-71.

Червяков Н.И., Копыткова Л.Б., Непретилова Е.В. и др. Проблемы реализации естественно-надежных процессоров в распределительных вычислительных системах // Вестник СГУ. – Ставрополь: СГУ, 1999. – № 20. – С. 72-84

Червяков Н.И., Копытова Л.Б. Масштабирование чисел при обработке сигналов // XIV НТК. – Ставрополь, 2001. – С. 12.

Червяков Н.И., Копытова Л.Б. Сравнительный анализ методов перевода чисел из СОК в ПСС // Проблемы физико-математических наук. – Ставрополь, 2001. – С. 54-57

Червяков Н.И., Кравченко Г.В. Метод контроля целостности информации на основе модулярного кодирования // XIV НТК. – Ставрополь, 2001. – С. 16

Червяков Н.И., Кравченко С.В., Шапошников А.В. и др. Применение системы остаточных классов для безопасности хранения ключей // XIII научно-техническая конференция. – 2000. – С. 83

Червяков Н.И., Краснобаев В.А. Надежностный синтез цифровых систем управления и связи. – Ставрополь: СВВИУС, 1991. – 98 с.

Червяков Н.И., Краснобаев В.А. Функциональные блоки и узлы отказоустойчивых и высокопроизводительных систем. – Ставрополь: СВВИУС, 1989. – 95 с.

Червяков Н.И., Краснобаев В.А., Квасов М.В. и др. О динамическом диапазоне избыточной системы остаточных классов // Помехоустойчивость и эффективность систем связи и управления, вып. 9. – Ставрополь: СВВИУС, 1991. – 4 с.

Червяков Н.И., Лавриненко И.А., Ляшенко О.В. и др. Перспективы развития организации вычисления нераспараллеливаемых алгоритмов // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с

Червяков Н.И., Лавриненко И.Н., Копыткова Л.Б. и др. Обработка данных в со-

процессорах функционально распределенных вычислительных систем // Сборник научный трудов. Вып. 4. – 2000. – С. 91-96.
Червяков Н.И., Лавриненко И.Н., Мезенцева О.С, Сивоплясов Д.В. Метод расширения динамического диапазона модулярного нейрокомпьютера // Нейрокомпьютеры: разработка, применение. – 2005. № 7 – С. 64-69.
Червяков Н.И., Линец Г.И. и др. Развитие способов организации параллельной обработки информации // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с
Червяков Н.И., Малофей О.П., Шапошников А.В. и др. Нейронные сети в системах криптографической защиты информации // Нейрокомпьютеры: разработка, применение. – 2001. – № 10. – 14 с
Червяков Н.И., Мезенцева О.С. Вычисления элементарных функций от аргументов, представленных в СОК // XXX НТК СК СГТУ. – 2000. – С. 59.
Червяков Н.И., Мезенцева О.С. Применение арифметики СОК для повышения достоверности передачи и обработки данных // Сборник научных трудов. Выпуск 1. – 1998. – С. 90-95.
Червяков Н.И., Мезенцева О.С., Лавриненко И.Н., Лавриненко С.В., Сивоплясов Д.В., Подопригора Н.Б. Связность обобщенной позиционной системы счисления и системы остаточных классов и ее применение в модулярных нейрокомпьютерных технологиях // I Международная научно-техническая конференция “Инфотелекоммуникационные технологии в науке, производстве и образовании” (19 декабря 2004 г.). – Ставрополь: СевКавГТУ, 2004. – С. 164-181
Червяков Н.И., Микула Н.П. Арифметических расширитель в микропроцессорной системе цифровой обработки сигналов // IV НТК. – Ставрополь, 1990. – 1 с
Червяков Н.И., Микула Н.П. Об эффективных по быстродействию алгоритмах решения задач цифровой обработки сигналов // Всесоюзная НТК “Компьютерные методы исследования …”. – М.: “Радио и связь”, 1990. – 1 с
Червяков Н.И., Микула Н.П., Квасов М.В. и др. О динамическом диапазоне системы остаточных классов // IV НТК. – Ставрополь, 1990. – 1 с.
Червяков Н.И., Микула Н.П., Квасов М.В. Математическая постановка задачи оптимизации арифметического устройств, функционирующего в системе остаточных классов // Помехоустойчивость и эффективность систем связи и управления. Вып. 7. – Ставрополь: СВВИУС, 1989. – С. 33-38.
Червяков Н.И., Микула Н.П., Квасов М.В. Математическая постановка задачи оптимизации арифметического устройств, функционирующего в системе остаточных классов // Помехоустойчивость и эффективность систем связи и управления, вып. 7. – Ставрополь: СВВИУС, 1989. – С. 68-71.
Червяков Н.И., Микула Н.П., Квасов М.В. Организация процессорных обменов в специализированных распределенных вычислительных системах // Помехоустойчивость и эффективность систем связи и управления, вып. 7. – Ставрополь: СВВИУС, 1989. – С. 63-67.

Червяков Н.И., Микула Н.П., Оленев А.А. и др. Архитектура высокопроизводительной отказоустойчивой микропроцессорной системы для цифровой обработки сигналов в системе остаточных классов // Межрегиональная НТК "Цифровая обработка сигналов в системе связи и управления". – Львов, 1992. – 1 с

Червяков Н.И., Микула Н.П., Оленев А.А. и др. Повышение эффективности выполнения арифметических операций по модулю при реализации цифровых фильтров // Помехоустойчивость и эффективность систем связи и управления, вып. 8. – Ставрополь: СВВИУС, 1990. – 4 с.

Червяков Н.И., Оленев А.А. Алгоритм кодирования комплексных чисел в виде комплексных остатков // Межведомственный тематический научный сборник "Синтез алгоритмов сложных систем", вып. 8. – Таганрог, 1992. – 6 с

Червяков Н.И., Оленев А.А. Алгоритм нахождения и коррекции ошибки в кодах, представленных в системе счисления остаточных классов // IV НТК. – Ставрополь, 1990. – 1 с

Червяков Н.И., Оленев А.А. Анализ особенностей создания функциональных блоков специализированных процессоров // IV НТК. – Ставрополь, 1990. – 1 с

Червяков Н.И., Оленев А.А. Кодирование комплексных чисел в системе остаточных классов // Помехоустойчивость и эффективность систем связи и управления, вып. 9. – Ставрополь: СВВИУС, 1991. – С. 50-52.

Червяков Н.И., Оленев А.А. Метод коррекции чисел, представленных кодом системы остаточных классов// Депонирована МО, № 5221, вып. 17, 1991.

Червяков Н.И., Оленев А.А. Отказоустойчивость систолических процессоров // Помехоустойчивость и эффективность систем связи и управления, вып. 9. – Ставрополь: СВВИУС, 1991. – С. 54-58.

Червяков Н.И., Оленев А.А. Оценка аппаратурных затрат реализации дискретного преобразования Фурье параллельно-конвейерного типа // VII НТК. – Ставрополь, 1993. – 1 с.

Червяков Н.И., Оленев А.А. Параллельный алгоритм перевода кода остаточных классов в обобщенную полиадическую систему // НТК. – Харьков, 1991. – 1 с

Червяков Н.И., Оленев А.А. Перспективы совмещения систолической архитектуры с системой счисления в остаточных классах // IV НТК. – Ставрополь, 1990. – 1 с.

Червяков Н.И., Оленев А.А. Принципы построения специализированных процессоров цифровой обработки сигналов // VII НТК. – Ставрополь, 1993. – 1 с.

Червяков Н.И., Оленев А.А. Систолические процессоры, использующие арифметику в остаточных классах // XVIII НТК. – Киев, 1991. – 1с

Червяков Н.И., Оленев А.А. Устройство для вычисления квадратичных вычетов. Патент РФ № 2020757, БИ № 18, 1994.

Червяков Н.И., Оленев А.А., Бережной В.В. Алгоритм коррекции ошибок в кодах системы остаточных классов // V НТК. – Ставрополь, 1992. – 1 с

Червяков Н.И., Оленев А.А., Бережной В.В. Архитектура высокопроизводительной отказоустойчивой системы для цифровой обработки сигналов с использо-

ванием системы остаточных классов // Всесоюзная школа-семинар “Передача, обработка и отображение информации”. Теберда – Сочи, апрель-ноябрь, 1991. – 1 с..

Червяков Н.И., Оленев А.А., Бережной В.В. и др. Архитектура высокопроизводительной отказоустойчивой системы цифровой обработки сигналов // XVIII НТК. – Воронеж: “Техника средств связи”, 1992. – 1 с..

Червяков Н.И., Оленев А.А., Бережной В.В. Метод автоматизированного проектирования самопроверяемых схем встроенного контроля табличных арифметических устройств // XVIII НТК. – Воронеж: “Техника средств связи”, 1992. – 1 с.

Червяков Н.И., Оленев А.А., Бережной В.В. Отказоустойчивость в систолических процессорах, использующих арифметику в остаточных классах // НТК. – Харьков, 1991. – 1 с

Червяков Н.И., Оленев А.А., Бережной В.В. Оценка аппаратурных затрат на реализацию методов контроля цифровых непозиционных процессоров // VI НТК. – Ставрополь, 1992.– 1 с

Червяков Н.И., Оленев А.А., Бережной В.В. Перспективы развития АСУ на основе структур, использующих системы счисления в остаточных классах // XVIII НТК. – Киев, 1991. – 1с

Червяков Н.И., Оленев А.А., Калмыков И.А. и др. Алгоритм выбора рационального набора модулей системы остаточных классов // Сборник научных трудов № 10. – Ставрополь: СВВИУС, 1992. – С. 24-29

Червяков Н.И., Оленев А.А., Микула Н.П. и др. Оценка надежности микропроцессорных систем с распределенной обработкой данных в АСУ // Механизация и автоматизация управления – 1991. – № 2. – С. 35-38

Червяков Н.И., Оленев А.А., Сагдеев К.М. Алгоритм коррекции ошибок в корректирующих остаточных кодах // Помехоустойчивость и эффективность систем связи и управления, вып. 8. – Ставрополь: СВВИУС, 1990. – С. 50-51.

Червяков Н.И., Ремизов С.Л. Анализ речевых сигналов в инфокоммуникационных системах на основе модулярных преобразований // Инфокоммуникационные технологии. – 2005. – № 1.

Червяков Н.И., Ремизов С.Л. Локализация ошибки на основе метода расширенной проекции // VI НТК. – Ставрополь, 1992. – 1 с

Червяков Н.И., Рожнов А.В. Модель обработки информации нейроноподобным образованием на основе аппарата системы остаточных классов // VI Всероссийская конференция “Нейрокомпьютеры и их применение”. Сборник докладов. – 16-18 февраля, 2000. – С. 474-477

Червяков Н.И., Сахнюк П.А., Шапошников А.В., Макоха А.Н. Структура нового специализированного нейропроцессора // М.: Нейрокомпьютеры: разработка, применение. – № 6 – 2003

Червяков Н.И., Сахнюк П.А. Вычисления в полях Галуа для обработки сигналов в системах управления // Новые технологии управления движения техниче-

скими объектами. – Новочеркасск, 1999. – С. 20-23.
Червяков Н.И., Сахнюк П.А. Нейрокомпьютерные вычислительные средства с модулярной арифметикой для цифровой обработки сигналов // Нейрокомпьютеры и их применение: Труды VI Всероссийской конференции 16-18 февраля 2000 г. – М.: РЦН, 2000. – С. 121-123.
Червяков Н.И., Сахнюк П.А. Отказоустойчивая архитектура непозиционных нейрочипов для решения сложных задач в масштабе реального времени // Сборник научных трудов № 17. – Ставрополь: ФРВИРВ, 1999. – С. 141-142.
Червяков Н.И., Сахнюк П.А. Применение модулярной арифметики в нейросетевых структурах управления интеллектуального робота // Труды 1 международной конференции "Новые технологии управления техническими объектами". – НП НИИ Систем управления и приводов. СевКавГТУ, 1999. – С. 116-118.
Червяков Н.И., Сахнюк П.А. Применение нейроматематики для реализации вычислений в конечных кольцах по модулю чисел Ферма и Мерсена // Сб. докл. VI Всероссийской конференции с международным участием "Нейрокомпьютеры и их применение". – М.: Радио и связь, 2000. – С. 586-588.
Червяков Н.И., Сахнюк П.А. Применение нейроматематики для реализации вычислений в конечных кольцах по произвольному модулю // Сб. докл. VI Всероссийской конференции с международным участием "Нейрокомпьютеры и их применение". – М.: Радио и связь, 2000. – С. 598-601.
Червяков Н.И., Сахнюк П.А., Велигоша А.В. Нейросетевой алгоритм модулярных вычислений по модулю чисел Ферма для цифровой обработки сигналов // XIII научно-техническая конференция. – 2000. – С. 7-8
Червяков Н.И., Сахнюк П.А., Копыткова Л.Б. Применение нейронных сетей для прямого и обратного преобразования кодов системы остаточных классов // Вестник СГУ. – Старопль: СГУ, 1999. – № 18. – С. 57-63.
Червяков Н.И., Сахнюк П.А., Лавриненко И.Н. и др. Перевод чисел, представленных в системе остаточных классов с вектором одних модулей в вектор других модулей // XIII научно-техническая конференция. – 2000. – С. 5-6.
Червяков Н.И., Сахнюк П.А., Непритилова Е.В. и др. Отказоустойчивые непозиционные процессоры с использованием искусственных нейронных сетей // XIII научно-техническая конференция. – 2000. – С. 5.
Червяков Н.И., Сахнюк П.А., Шапошников А.В. и др. Минимизация количества итераций нейронной сети конечного кольца // Нейрокомпьютеры и их применение: Труды VII Всероссийской конференции 14-16 февраля 2001 г. – М.: РЦН, 2001. – С. 595-598.
Червяков Н.И., Сахнюк П.А., Шапошников А.В. и др. Нейрокомпьютеры в системах обработки сигналов. – М.: ИПРЖР, 2003. – 157 с.
Червяков Н.И., Сахнюк П.А., Шапошников А.В. и др. Обнаружение ошибок в цифровых фильтрах, функционирующих в дополнительных кодах избыточной системы остаточных классов // Проблемы физико-

математических наук. Материалы XLV научно-методической конференции преподавателей и студентов. – 2000. – С. 152-155

Червяков Н.И., Сахнюк П.А., Шапошников А.В. и др. Применение многоступенчатой системы остаточных классов для вычислений с большими числами // XIII научно-техническая конференция. – 2000. – С. 6-7

Червяков Н.И., Сахнюк П.А., Шапошников А.В. и др. Применение модулярной арифметики для повышения отказоустойчивости вычислительных систем на основе нейронных сетей // Ситеми обробки інформації. Збірник наукових праць. Вип. 2(6) – Харків: НАНУ, ПАНМ, ХВУ, 1999. – С. 78-83.

Червяков Н.И., Сахнюк П.А., Шапошников А.В. и др. Реконфигурация модулярного вычислительного устройства на основе нейронных сетей // Системы обработки информации. Сборник научных статей. Вып. 1(7). – Харьков: НАНУ, ПАНМ, ХВУ, 2000. – С. 72-76

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Иерархическая модульная нейронная сеть с деградирующей структурой, функционирующая в системе остаточных классов / Нейрокомпьютер: разработка, применение. – 2000. – № 2. – С. 63-71.

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Использование внутреннего параллелизма разрядной сетки нейрокомпьютера / Нейрокомпьютеры и их применение: Труды VII Всероссийской конференции 14-16 февраля 2001 г. – М.: РЦН, 2001. – С. 59-61

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Метод подсчета ортогональных базисов // Проблемы и перспективы совершенствования автоматизированного управления и контроля. – 2000. – С. 52-54

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Модель и структура нейронной сети для реализации арифметики системы остаточных классов / Нейрокомпьютеры: разработка, применение. – 2001. – № 10. – С. 5-12.

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Применение системы остаточных классов для повышения отказоустойчивости вычислительных систем на основе нейронных сетей // Проблемы и перспективы совершенствования автоматизированного управления и контроля. – 2000. – С. 49-51.

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Проблемы повышения отказоустойчивости нейрочипов сигнальной обработки // Проблемы физико-математических наук. Материалы XLV научно-методической конференции преподавателей и студентов. – 2000. – С. 156-160.

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Пути эффективного использования фрагментов нейронных сетей на ПЛИС Xilinx при аппаратной реализации цифровых фильтров с параллельной обработкой данных // Нейрокомпьютеры: разработка и применение. – 2001. – № 10. – С. 20-31.

Червяков Н.И., Сахнюк П.А., Шапошников А.В. Разработка методики построения отказоустойчивого непозиционного нейрокомпьютера на основе нейронных сетей // Проблемы и перспективы совершенствования автоматизирован-

ного управления и контроля. – 2000. – С. 46-49
Червяков Н.И., Сахнюк П.А., Шапошников А.В. Сокращение и расширение набора оснований системы остаточных классов // I-ая Всероссийская НТК "Математическое моделирование в научных исследованиях": Сборник тезисов докладов. – Ставрополь: СГУ, 2000. – С. 228-230.
Червяков Н.И., Сахнюк П.А., Шапошников А.В., Ряднов С.А. Модулярные параллельные вычислительные структуры нейропроцессорных систем /– М.: Физматлит, 2003. – 288 с.
Червяков Н.И., Сивоплясов Д.В. Нейронная сеть для преобразования модулярного кода в позиционный код с промежуточным переходом через обобщенную полиадическую систему счисления // Инфокоммуникационные технологии. – 2004. –№ 1. – С. 29-33
Червяков Н.И., Сивоплясов Д.В. Нейронная сеть конечного кольца прямого распространения с модульной операцией на финишной ступени // Вестник Северо-Кавказского государственного технического университета. Серия "Физико-химическая". – 2004. – № 1(8).– С. 168-174
Червяков Н.И., Сивоплясов Д.В., Горденко Д.В. Нейронная сеть для преобразования полиадического кода в код системы остаточных классов // Нейрокомпьютеры: разработка, применение. – 2003. – № 10-11. – С. 10-12.
Червяков Н.И., Сивоплясов Д.В., Сахнюк П.А. Реализация синаптических связей модулярных нейронных сетей на ПЛИС Xilinx // Инфокоммуникационные технологии. – 2005. – № 2. – С. 41-44
Червяков Н.И., Сивоплясов Д.В., Ткачук Р.В. Нейронная сеть для вычисления позиционной характеристики ранга числа, представленного в системе остаточных классов // Нейрокомпьютеры: разработка, применение. – 2003. – № 10-11. – С. 32-34
Червяков Н.И., Смирнов А.А. Применение чисел Мерсена и Ферма в качестве оснований системы остаточных классов в двоичном канале связи // Инфокоммуникационные технологии. – 2004. –№ 1.
Червяков Н.И., Тынчеров К.Т. Корректирующие коды СОК с использованием интервально-индексной характеристики // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с
Червяков Н.И., Тынчеров К.Т., Велигоша А.В. Высокоскоростная обработка сигналов с использованием непозиционной арифметики // Радиотехника. – 1997. – № 10. – С. 23-27.
Червяков Н.И., Тынчеров К.Т., Велигоша А.В. Корректирующие коды СОК с использованием интервальных номеров чисел // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с
Червяков Н.И., Тынчеров К.Т., Калмыков И.А. К вопросу об обнаружении и исправлении ошибок в СОК // Сб. тезисов VIII НТС. – Ставрополь, 1995. – 1 с.
Червяков Н.И., Финько О.А. Двоичный алгоритм аналого-цифрового преобразования в остаточном представлении // VI НТК. – Ставрополь, 1992. – 1 с
Червяков Н.И., Хатамова М.Х. Некоторые номографические методы обработки

информации в СОК // Проблемы физико-математических наук. Материалы XLIII научно-методической конференции. – 1998. – С. 75-82.
Червяков Н.И., Хлевной С.Н. Использование избыточных непозиционных кодов для исправления ошибок // II НТК.- МО СССР, 1984. – 27 с.
Червяков Н.И., Хлевной С.Н. Некоторые вопросы обнаружения ошибок в арифметических остаточных кодах с одним избыточным основанием // Депонирована ЦИВТИ, № 9030, 1985. – 15 с.
Червяков Н.И., Хлевной С.Н., Швецов Н.И. Некоторые вопросы построения специализированных процессоров для цифровой обработки сигналов // II НТК. – МО СССР, 1984. – 26 с.
Червяков Н.И., Хлевной С.Н., Швецов Н.И. Оценка эффективности кодов в остатках для повышения структурной скрытности информации // II НТК. – МО СССР, 1984. – 26 с.
Червяков Н.И., Хлевной С.Н., Швецов Н.И. Оценка эффективности многоступенчатых остаточных кодов для повышения структурной скрытости информации // XXXII НТК, т. 2. – МО СССР, 1984. – 58 с.
Червяков Н.И., Хлевной С.Н., Швецов Н.И. Применение многоступенчатых избыточных непозиционных кодов для повышения эффективности АСУ // XXXII НТК, т.2. – МО СССР, 1984. – 59 с.
Червяков Н.И., Шалин Б.С. Модифицированная сеть Хэмминга для коррекции ошибок в системе остаточных классов // Инфокоммуникационные технологии. – 2004. –№ 1.
Червяков Н.И., Шапошников А.В. Алгоритм несимметричной криптографии реального времени // Новые технологии управления движения техническими объектами. – Новочеркасск, 1999. – С. 5-8.
Червяков Н.И., Шапошников А.В. Алгоритм симметричной криптографии реального времени // Новые технологии управления движением технических объектов. – Новочеркасск, 1999. – С. 5-8
Червяков Н.И., Шапошников А.В. Нейросетевой метод защиты информации // Сборник научных трудов № 17. – Ставрополь: ФРВИРВ, 1999. – С. 143-145
Червяков Н.И., Шапошников А.В. Применение системы остаточных классов в алгоритме несимметричной криптографии // Пути развития теории и техники связи. – Новочеркасск, 2000. – С. 106-108.
Червяков Н.И., Шапошников А.В. Применение спектральных методов анализа при идентификации пользователей компьютерной сети по голосу // I Всероссийская конференция “Спектральные методы обработки информации в научных исследованиях”. – Москва, 2000. – С. 171-175.
Червяков Н.И., Шапошников А.В. Способ возведения числа в степень в алгоритме несимметричной криптографии // XIII научно-техническая конференция. – 2000. – С. 18-19.
Червяков Н.И., Шапошников А.В., Копыткова Л.Б. и др. Определение одновременного появления ошибки и переполнения динамического диапазона в цифровых фильтрах, функционирующих в системе остаточных классов // Про-

блемы физико-математических наук. Материалы XLV научно-методической конференции преподавателей и студентов. – 2000. – С. 142-152

Червяков Н.И., Шапошников А.В., Копыткова Л.Б. Сопряжение цифровых фильтров, функционирующих в системе остаточных классов, с традиционными двоичными вычислительными структурами // Проблемы физико-математических наук. Материалы XLV научно-методической конференции преподавателей и студентов. – 2000. – С. 136-142

Червяков Н.И., Шапошников А.В., Ремизов С.Л. Нейросетевой алгоритм определения позиционной характеристики чисел, представленных в модулярном коде // XV НТК, Проблемы совершенствования АСБУ и связи РВСН. – Ставрополь, 2002. – С. 60-61

Червяков Н.И., Шапошников А.В., Сапин А.П. Нейронная сеть преобразования чисел в код системы остаточных классов // Сборник научных трудов № 19. – Ставрополь: ФРВИ РВ, 2001. – С. 208-211.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Нейрокомпьютерные ВС с модулярной арифметикой для вычисления БПФ // XXX НТК СК СГТУ. – 2000. – С. 135.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Нейронная реализация преобразования данных по произвольному модулю // XXX НТК СК СГТУ. – 2000. – С. 136.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Приближение нейроподобных образований к архитектуре СОК // XXX НТК СК СГТУ. – 2000. – С. 138.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Приложение нейроматематики при вычислениях в конечных кольцах // XXX НТК СК СГТУ. – 2000. – С. 134.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Применение искусственных нейронных сетей в отказоустойчивых модулярных процессорах // XXX НТК СК СГТУ. – 2000. – С. 139.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Применение нейронных сетей Хопфилда для коррекции ошибок в модулярных нейрокомпьютерах // Нейрокомпьютеры: разработка, применение. – 2002. – № 11. – С. 10-16.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Применение непозиционных систем счисления для ускорения нейрообработки // XXX НТК СК СГТУ. – 2000. – С. 139.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Реализация вычислений по модулю чисел Ферма и Мерсена с помощью нейроматематики // XXX НТК СК СГТУ. – 2000. – С. 137.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. и др. Структура моделей соединения искусственных нейронных сетей и постоянных запоминающих устройств на основе сверхбольших интегральных схем // Нейрокомпьютеры: разработка, применение. – 2002. – № 11.

Червяков Н.И., Шапошников А.В., Сахнюк П.А. Использование системы остаточных классов при табличных методах обработки сигналов //

НТСБ № 18. – Ставрополь: ФРВИРВ, 2000. – С. 171-173
Червяков Н.И., Шапошников А.В., Сахнюк П.А. Модулярные нейронные сети прямого распространения // XV НТК, Проблемы совершенствования АСБУ и связи РВСН. – Ставрополь, 2002. – С. 62-63
Червяков Н.И., Шапошников А.В., Сахнюк П.А. Нейронная реализация криптографического сопроцессора // Сборник научных трудов № 18. – Ставрополь: ФРВИРВ, 2000. – С. 174-176
Червяков Н.И., Шапошников А.В., Сахнюк П.А. Оптимизация модулярных вычислений при нейрообработке // I-ая Всероссийская НТК "Математическое моделирование в научных исследованиях": Сборник тезисов докладов. – Ставрополь: СГУ, 2000. – С. 220-225
Червяков Н.И., Шапошников А.В., Сахнюк П.А. Применение нейронных сетей в задачах цифровой обработки сигналов // Нейрокомпьютеры: разработка, применение. – 2001. – № 10. – С. 31-39.
Червяков Н.И., Шапошников А.В., Сахнюк П.А. Реконфигурация модулярной иерархичной нейронной сети, функционирующей в системе остаточных классов // Сборник трудов СКГТУ. Серия физико-химическая. Вып. 4. – Ставрополь: СКГТУ, 2000. – С. 96-101.
Червяков Н.И., Шапошников А.В., Сахнюк П.А. Решение NP-полных задач с помощью нейронных сетей // XXX НТК СК СГТУ. – 2000. – С. 41.
Червяков Н.И., Шапошников А.В., Сахнюк ПА. Приближение системы остаточных классов и нейронных сетей для обеспечения отказоустойчивости вычислительных систем // Сборник трудов СКГТУ. Серия физико-химическая, вып. 4, 2000. – С. 102-109
Червяков Н.И., Шапошников А.В., Сильченко В.В. Нейросетевой алгоритм восстановления числа по вычетам // XV НТК, Проблемы совершенствования АСБУ и связи РВСН. – Ставрополь, 2002. – С. 59-60
Червяков Н.И., Шапошников Н.И. Применение спектральных методов анализа при идентификации пользователей компьютерной сети по голосу // 1-я всероссийская НТК «Спектральные методы анализа в научных исследованиях»: Сборник тезисов докладов. – Пущино: ИМПБ, 2000. – С. 171-175.
Червяков Н.И., Швецов Н.И, Бунто А.Н. Накапливающий сумматор А.С. № 1251074, БИ № 30, 1986.
Червяков Н.И., Швецов Н.И. Анализ путей построения сумматоров по произвольному модулю // Помехоустойчивость и эффективность систем связи и управления. – Ставрополь: СВВИУС, 1984. – С. 50-54.
Червяков Н.И., Швецов Н.И., Болтков А.П., Шахов А.В. Преобразователь позиционного кода в вычет по модулю. А.С. 1383506, БИ № 11, 1988.
Червяков Н.И., Швецов Н.И., Хлевной С.Н. Надежность и живучесть систем управления и связи, функционирующих в системе остаточных классов. – Ставрополь: СВВИУС, 1986. – 57 с.
Чернявский А.Ф., Аксенов А.М., Коляда А.А., Селянинов М.Ю. Теоретические основы мультипликативных процедур для минимально избы-

точных модулярных систем счисления // Доклады АН Беларуси. 1995. Т.39. №6. С. 5 – 10.

Чернявский А.Ф., Данилевич В.В., Коляда А.А., Селянинов М.Ю. Высокоскоростные методы и системы цифровой обработки информации. Мн.: Белгосуниверситет, 1996. 376 с.

Чернявский А.Ф., Евдокимов А.А., Коляда А.А., Ревинский В.В. Рекурсивные минимально избыточные интервально-модулярные системы счисления// Доклады НАН Беларуси. – 2004. –Т.48, №1. – С.10–14.

Чернявский А.Ф., Коляда А.А., Кравцов В.К. Теоретические основы минимально избыточных квадратичных модулярных систем счисления. // Доклады НАН Беларуси. 1998. Т.42. №1. С. 5 – 12.

Шалин Б.С., Евдокимов А.А. Свидетельство об отраслевой регистрации разработки №3842 от 03 сентября 2004 г. на разработку «Программа для порогового разделения файла». Номер гос. регистрации 50200401112 от 14 сентября 2004 г. (Министерство образования РФ. Государственный координационный центр информационных технологий. Отраслевой фонд алгоритмов и программ)

Шуба Ю. А. Оценка целесообразности применения системы остаточных классов в аппаратуре обработки сигналов // Радиотехника. — 1980. - Т. 25, № 1. - С. 75-76.

Юдицкий Д.И. Высокопроизводительная модулярная ЭВМ «Алмаз». // Труды Юбилейной Международной научно-технической конференции «50 лет модулярной арифметике», Россия, Москва, Зеленоград, 23-25 ноября 2005, издательство МИЭТ, тираж 150 экз., с 149-160.

Составители просят извинения
за неизбежные неполноту, неточности и ошибки.

М О Н О Г Р А Ф И Я [1]

УДК 519.7 ББК 22.18

Финько О.А.[2]

Ф59 Модулярная арифметика параллельных логических вычислений: Монография / Под. ред. В.Д. Малюгина; – М.: ИПУ РАН, 2003. – 224 с.

Монография[3] посвящена новому направлению математической логики — реализации параллельных логических вычислений посредством арифметико-логических форм. Впервые рассматривается отображение классической логики на модулярную арифметику, которое открывает уникальные возможности по достижению высоких уровней производительности и отказоустойчивости средств гибких логических вычислений.

Для специалистов в области защиты информации, прикладной математики, математической кибернетики, информатики и вычислительной техники, аспирантов и студентов соответствующих специальностей.

Рецензенты:

- лауреат Государственной премии СССР, академик Казахстанской НАН, доктор технических наук, профессор В.М. АМЕРБАЕВ;
- проректор Чувашского государственного ун-та по информатизации и образовательным технологиям, доктор технических наук, профессор Е.К. ЛЕБЕДЕВ.

---

[1] Книга выпущена на средства Министерства обороны РФ. Заинтересованные юридические и физические лица и организации могут получить книгу **БЕСПЛАТНО**. **Книга не может быть использована с целью получения прибыли.** Для получения книги необходимо выслать на имя руководителя организации соответствующее письмо с ходатайством и мотивацией предоставления книги. Адрес организации: 350035. г. Краснодар, ул. Красина, д. 4. Краснодарское высшее военное училище (военный институт). Начальнику военного училища.

[2] © Автор. Просьба замечания и отзывы на книгу направлять автору: E-mail: ofinko@yandex.ru.

3 С монографией можно ознакомиться в библиотеках ИПУ РАН (г. Москва) и Московского государственного института электронной техники (технический ун-т) (Зеленоград).

Оглавление

M O N O G R A F [4]

Finko O.A. Modular arithmetics of parallel logic evaluations. Under Maljugin's VD edition. Institute of problems of control of the Russian academy of sciences, Moscow, 2003.

The monography[5] is devoted to a new direction of a mathematical logic – to implementation of parallel logical computations by means of arithmetic-logic forms. For the first time map of classical logic to modular arithmetics which opens unique possibilities on reaching high levels of efficiency and fault tolerance of tools of flexible logical computations is considered.

The monography intends for usage by experts in the field of a guard of the information, an applied mathematics, mathematical cybernetics, computer science and computer facilities, post-graduate students and students of appropriate specialities

Reviewers:

- The winner of the State premium of the USSR, the academician of the Kazakhstan national academy of sciences, Dr.Sci.Tech., the professor
  V. Amerbaev;
- The pro-rector of the Chuvash state university on informatization and educational process engineerings, Dr.Sci.Tech., the professor E. Lebedev.

[4] The edition of the book financed the Ministry of Defence of the Russian Federation. The interested legal and physical persons and the organizations can receive the book **FREE-OF-CHARGE**. **The book cannot be used with the purpose of reception of the profit.** For reception of the book it is necessary to send addressed to the head of the organization the corresponding letter with the petition and motivation of granting of the book. The address of the organization: an index 350035. city of Krasnodar, street Khrasina, the house 4. Krasnodar high military school (military institute). To the chief of military school.

[5] It is possible to familiarize with the monography in libraries IPU of the Russian Academy of Science (Moscow) and the Moscow state institute of electronic technics (Technical university) (Zelenograd).

Contents

Информация об авторах научных трудов
Международной юбилейной конференции
"50 лет модулярной арифметике"

	Амербаев Вильжан Мавлютинович, академик НАН Республики Казахстан, доктор технических наук, профессор, лауреат Государственной премии СССР, главный научный сотрудник ГУП Научно-производственный центр «СПУРТ», т. (495) 531-00-00, E-mal: marketing@ancud.ru
	Бережной Виктор Васильевич, кандидат технических наук, доцент, доцент кафедры прикладной математики и информатики Ставропольского государственного университета, beregnoj@yandex.ru
	Бияшев Рустем Гакашевич, профессор, д.т.н., академик МАИ, заместитель директора Института проблем информатики и управления Министерства образования и науки Республики Казахстан, г. Алма-Ата, т: 91-18-28, E-mal: brg@ipic.kz

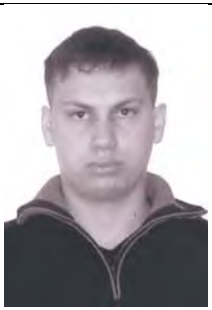
	Брусенцов Николай Петрович, кандидат технических наук, старший научный сотрудник, заслуженный работник высшей школы РФ, заслуженный научный сотрудник МГУ, заведующий лабораторией ЭВМ факультета вычислительной математики и кибернетики Московского государственного университета им. М.В. Ломоносова, г. Москва, т. (495) 939-38-56, E-mal: ramil@cs.msu.su .
	Горковенко Елена Владимировна, доцент, к.т.н., чл.корр. МАИ, ведущий научный сотрудник Института проблем информатики и управления Министерства образования и науки Республики Казахстан, г. Алма-Ата, т. 93-82-58, E-mal: gev@ok.kz
	Дзегелёнок Игорь Игоревич, профессор, доктор технических наук, Московский энергетический институт, г. Москва, (095) 3627558, E-mal: dsegel@mpei.ru
	Дьяченко Игорь Васильевич, ассистент Ставропольского государственного университета, г. Ставрополь, E-mal: rsx_magnus@inbox.ru

	Дьячьков Виктор Николаевич, кандидат технических наук, генеральный директор Государственного унитарного предприятия «Научно-производственный центр СПУРТ», г. Москва (Зеленоград), т. (495) 534-25-82, ф. 534-91-86, E-mail: office@spurt.compnet.ru
	Евдокимов Алексей Алексеевич, кандидат технических наук, ассистент Невинномысского технологического института (филиал) Северо-Кавказского государственного университета, г. Невинномысск, т. (86554) 7-17-78, E-mal: evd1980@mail.ru
	Евстигнеев Владимир Гаврилович, доцент, доктор технических наук, Московский научно-исследовательский телевизионный институт, г. Москва, т. (095) 460-04-88, E-mal: evstigneev_mniti@mail.ru, evstigneev_mniti@pochta.ru.
	Егай Роман Владиславович, начальник отдела ТОО «ИнформЭнерго», Республика Казахстан, г. Алма-Ата, т. 54-04-81

	Зольников Владимир Константинович, доктор технических наук, профессор, главный научный сотрудник ФГУП «НИИ Электронной техники», г. Воронеж, E-mail: wkz@rambler.ru
	Инютин Сергей Арнольдович, профессор, д.т.н., чл.–корр. МАНПО, проректор по научной работе Сургутского государственного педагогического института, г. Сургут, т. (3462) 318473, 319434, ф. 319438, E-mail: surgpi@surguttel.ru
	Ирхин Валерий Петрович, старший научный сотрудник, доктор технических наук, заслуженный изобретатель РФ, заведующий кафедрой Международного института компьютерных технологий, г. Воронеж, т. (0732) 21-79-06, E-mal: val_irkhin@mail.ru
	Исмаилов Ш-М. А., кафедра вычислительной техники Дагестанского государственного технического университета, E-mail: osepyants@rin.ru

	Калашников Вячеслав Сергеевич, инженер-исследователь Института проблем проектирования в микроэлектронике РАН, г. Москва (Зеленоград), т. (495) 787-28-02, E-mal: kvs@ici.ru
	Коляда Андрей Алексеевич, старший научный сотрудник, доктор физико-математических наук, лауреат Госпремии СССР, главный научный сотрудник НИИ Прикладных физических проблем им. А.Н. Севченко БГУ, г. Минск, Беларусь, т. (8017)271-96-09, E-mal: Razan@tut.by
	Коляда Назар Андреевич, младший научный сотрудник НИИ Прикладных физических проблем им. А.Н. Севченко БГУ, г. Минск, Беларусь, т. (8017)271-96-09, E-mal: Razan@tut.by
	Корнев Михаил Дмитриевич, кандидат технических наук, НИИ Дальней радиосвязи, г. Москва, т. (495) 962-63-36, E-mal: mkornev@niidar.ru

	Корнилов Александр Иванович, кандидат технических наук, заведующий сектором Института проблем проектирования в микроэлектронике РАН, г. Москва (Зеленоград), т. (495) 787-28-02, 531-88-90, E-mal: korn@ici.ru
	Коряков Игорь Витальевич, главный конструктор ООО НВФ «Криптон», г. Киев, E-mal: korjakov@crypton-ua.com
	Краснобаев Виктор Анатольевич, доктор технических наук, профессор, заслуженный изобретатель Украины, почетный радист СССР, профессор кафедры автоматизации и компьютерных технологий Харьковского национального технического университета сельского хозяйства им. Петра Василенка, г. Харьков, т. 8-1038-057-7123537, E-mal: krasnobaev_va@rambler.ru
	Лавриненко Ирина Николаевна, учитель физики средней школы № 19, г. Ставрополь, т. (8652) 77-56-26

	Лавриненко Сергей Викторович, студент Северо-Кавказского государственного технического университета, г. Ставрополь, т. (8652) 77-56-26
	Ласточкин Олег Викторович, инженер-исследователь Института проблем проектирования в микроэлектронике РАН, г. Москва (Зеленоград), т. (495) 787-28-02, E-mal: lastochkin@ici.ru
	Лукин Федор Викторович, профессор, доктор технических наук, четырежды лауреат Государственной премии СССР, первый директор зеленоградского Центра микроэлектроники
	Малашевич Борис Михайлович, гл. специалист ОАО «Ангстрем», г. Москва (Зеленоград), т. (495) 532-80-39, E-mal: mbm@angstrem.ru

	Малашевич Денис Борисович, аспирант, старший преподаватель кафедры ПКИМС Московского государственного института электронной техники, г. Москва (Зеленоград), т. 8(495)532-9861, E-mail: dicd@miee.ru
	Машевич Павел Романович, кандидат технических наук, заместитель генерального директора ОАО «Ангстрем», Москва (Зеленоград), т. (495) 531-03-06, mashevich@angstrem.ru
	Мезенцева Оксана Станиславовна, кандидат физико-математических наук, доцент, доцент кафедры информационных систем и технологий Северо-Кавказского государственного технического университета, г. Ставрополь т. 95-68-01
	Музыченко Олег Николаевич, профессор, доктор технических наук, Балтийский государственный технический университет «Военмех» им. Д.Ф. Устинова, г. Санкт-Петербург, E-mal: npfmeridian@peterlink.ru

	Нысанбаев Рустэм Камильевич, кандидат технических наук, технический директор IT RESEARCH CENTER, г. Астана, Республика Казахстан, т. 93-61-35.
	Нысанбаева Сауле Еркебулановна, старший научный сотрудник, кандидат физико-математических наук, старший научный сотрудник Института проблем информатики и управления Министерства образования и науки Республики Казахстан, г. Алма-Ата, т. 72-77-15, 92-83-58, 93-82-57, E-mal: nyssanbayeva@ipic.kz
	Овчаренко Леонид Александрович, старший научный сотрудник, доктор технических наук, начальник научно-исследовательского отдела Военного института радиоэлектроники, г. Воронеж, т. (0732) 36-89-75, E-mal: vire@vire.vrn.ru, leo@box.vsi.ru
	Осепянц О. А., кафедра вычислительной техники Дагестанского государственного технического университета, E-mail: osepyants@rin.ru

	Оцоков Шамиль Алиевич, кандидат технических наук, Московский энергетический институт, г. Москва, E-mal: OtsokovShA@mpei.ru, shamil25@rambler.ru
	Пак Иван Тимофеевич, доктор технических наук, профессор, зам. директора Института математики Национальной Академии Наук Республики Казахстан, г. Алма-Ата, Тел.: +7(3272) 91-37-15, Факс: +7(3272) 62-74-79, E-mail: park@math.kz
	Полисский Юрий Давидович, кандидат технических наук, старший научный сотрудник, заслуженный изобретатель Украины, 40 лет проработал в Научно-исследовательском институте автоматизации черной металлургии (НИИАчермет), г. Днепропетровск, т. (8-056) 744-33-65 E-mail: POLISSKY@MAIL.RU
	Ремизов Сергей Леонидович, преподаватель Сибирского государственного университета телекоммуникаций и информатики, г. Новороссийск, т. 8-913-894-65-88, E-mail: micnatserg@rambler.ru

	Семенов Михаил Юрьевич, кандидат технических наук, главный специалист Научно-исследовательского центра «Микростайл», г. Москва (Зеленоград), т. (495) 787-28 02, E-mal: sem@ici.ru
	Смирнов Александр Александрович, кандидат технических наук, Ставропольский государственный университет, г. Ставрополь, т. (8652)35-42-05, (8652)93-62-54, E-mal: shursun@mail.ru
	Стемпковский Александр Леонидович, академик РАН, доктор технических наук, профессор, директор Института проблем проектирования в микроэлектронике РАН, г. Москва (Зеленоград), т. 7 (495) 531 8890, E-mail: secretar@ippm.ru
	Финько Олег Анатольевич, доктор технических наук, доцент, Краснодарское высшее военное училище (военный институт), г. Краснодар, т. (861) 253-05-86, E-mile: ofinko@yandex.ru

	Харинов Михаил Вячеславович, кандидат технических наук, доцент, старший научный сотрудник Санкт-Петербургского института информатики и автоматизации РАН, г. Санкт–Петербург, т.(812) 328–4369, факс: (812) 328–4450, E-mal: khar@iias.spb.su.
	Хацкевич Вильям Харитонович, доктор технических наук, Vice President Computer Algorithm & Program Development Corp., New York, USA, т. (212) 228-72-71, Email: hackevic@hotmail.com
	Хэскелл Леви, President Computer Algorithm & Program Development Corp., New York, USA, т. (917) 355-28-94, Email: levi.haskell@hotmail.com
	Червяков Николай Иванович, профессор, доктор технических наук, профессор кафедры информатики и информационных технологий в системах управления Ставропольского военного института связи ракетных войск, г. Ставрополь, т. (88652)26-60-14, E-mal: k-fmf-primath@stavsu.ru

	Чернявский Александр Федорович, академик НАН Беларуси, доктор технических наук, профессор, лауреат Государственной премии СССР, Государственной премии БССР, премии Совета Министров СССР, заслуженный деятель науки и техники БССР, директор НИИ Прикладных физических проблем им. А.Н. Севченко БГУ, г. Минск, Беларусь, т. (8017)212-48-16, E-mal: Niipfp@bsu.by
	Широ Георгий Эдуардович, доктор технических наук, профессор, профессор кафедры вычислительной техники Московского государственного института электронной техники (технический университет), г. Москва (Зеленоград), т. 8(495)532-98-47, E-mail: shiro@olvs.miee.ru
	Юдицкий Давлет Исламович, профессор, доктор технических наук, главный конструктор модулярных ЭВМ Т-340А, К-340А, Алмаз и 5Э53

«50 лет модулярной арифметике»
Юбилейная Международная научно-техническая конференция
Сборник научных трудов
ISBN 5-7256-0409-8

Подписано к печати с оригинал-макета 14.07.06. Формат 60x84 1/16. Печать офсетная. Бумага офсетная. Гарнитура Times New Romans. Усл. печ. л. 43,50. Уч.-изд. л 37,75. Тираж 150 экз. Зак. 227.
Отпечатано в типографии ЗАО «Петит-А».
124460, Москва, Зеленоград, Южная промзона, пр-д 4806, стр. 2.

Содержание

«50 лет модулярной арифметике»
Юбилейная Международная научно-техническая конференция
Сборник научных трудов
ISBN 5-7256-0409-8

Подписано к печати с оригинал-макета 14.07.06. Формат 60x84 1/16.
Печать офсетная. Бумага офсетная. Гарнитура Times New Romans.
Усл. печ. л. 43,50. Уч.-изд. л 37,75. Тираж 150 экз. Зак. 227.
Отпечатано в типографии ЗАО «Петит-А».
124460, Москва, Зеленоград, Южная промзона, пр-д 4806, стр. 2.