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, $\mathbf{K}$, 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]:

$$X \xrightarrow{\text{СОК}} \left(|X|_{m_1}, |X|_{m_2}, \mathbf{K}, |X|_{m_L} \right) \qquad (1)$$

В СОК могут быть также представлены и отрицательные числа из диапазона $\{-(M-1)/2, \mathbf{K}, -1, 0, 1, \mathbf{K}, (M-1)/2\}$ для нечётного M и $\{-M/2, \mathbf{K}, -1, 0, 1, \mathbf{K}, (M/2)-1\}$ для чётного M. При этом, если $X < 0$, то:

$$X < 0 \xrightarrow{\text{СОК}} \left(|M - |X||_{m_1}, |M - |X||_{m_2}, \mathbf{K}, |M - |X||_{m_L} \right) \qquad (2)$$

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:

$$X * Y \xrightarrow{\text{СОК}} \left(\underbrace{\left| |X|_{m_1} * |Y|_{m_1} \right|_{m_1}}_{\text{канал } m_1}, \underbrace{\left| |X|_{m_2} * |Y|_{m_2} \right|_{m_2}}_{\text{канал } m_2}, \mathbf{K}, \underbrace{\left| |X|_{m_L} * |Y|_{m_L} \right|_{m_L}}_{\text{канал } m_L} \right), \qquad (3)$$

$$\forall X, Y \in Z(M), \quad * \in \{\oplus, \otimes\}.$$

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* $\{m_1, m_2, \mathbf{K}, m_L\}$ *и число* $X \in Z(M)$, *СОК-представление которого* $[x_1, x_2, \mathbf{K}, x_L]$ *определяется выражением:*

$$x_i = |X|_{m_i} \quad i = 1, 2, \mathbf{K}, L \tag{4}$$

Тогда:

$$Z(M) \cong Z(m_1) \oplus Z(m_2) \oplus \mathbf{K} \oplus Z(m_L) \tag{5}$$

Иными словами, кольцо целых чисел по модулю M, $Z(M)$, *изоморфно прямой сумме колец целых чисел по модулям* $m_1, m_2, \mathbf{K}, m_L$ *и существует единственное* $X \in Z(M)$, $M = \prod_{i=1}^{L} m_i$, *восстанавливаемое по остаткам из выражения (4) по формуле*:

$$X = \left| \sum_{i=1}^{L} \mathbf{m}_i \left| x_i \left| \mathbf{m}_i \right|_{m_i}^{-1} \right|_{m_i} \right|_M \tag{6}$$

где

$$\mathbf{m}_i = \frac{M}{m_i} \tag{7}$$

и $\left| \mathbf{m}_i \right|_{m_i}^{-1}$ *– обратный по умножению в кольце* $Z(m_i)$ *элемент для* $\mathbf{m}_i$*:*

$$\left| \mathbf{m}_i \left| \mathbf{m}_i \right|_{m_i}^{-1} \right|_{m_i} = 1 \tag{8}$$

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 $\{m_1, m_2, \mathbf{K}\, m_L\}$, после чего выполняются независимые вычисления для каждого модуля m_i. На выходе происходит обратное преобразование из СОК в ПСС.

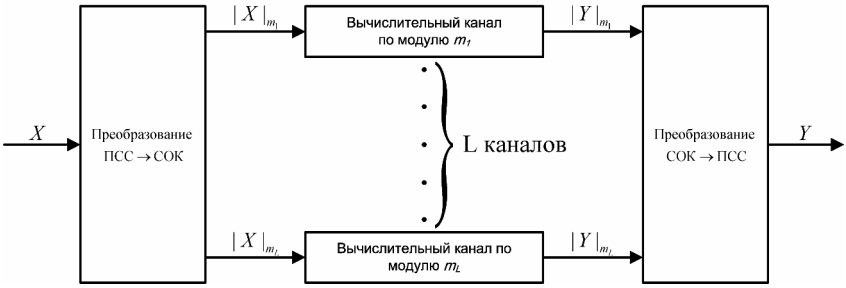


Рис. 1. Общая структура устройств цифровой обработки сигналов в СОК.

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].

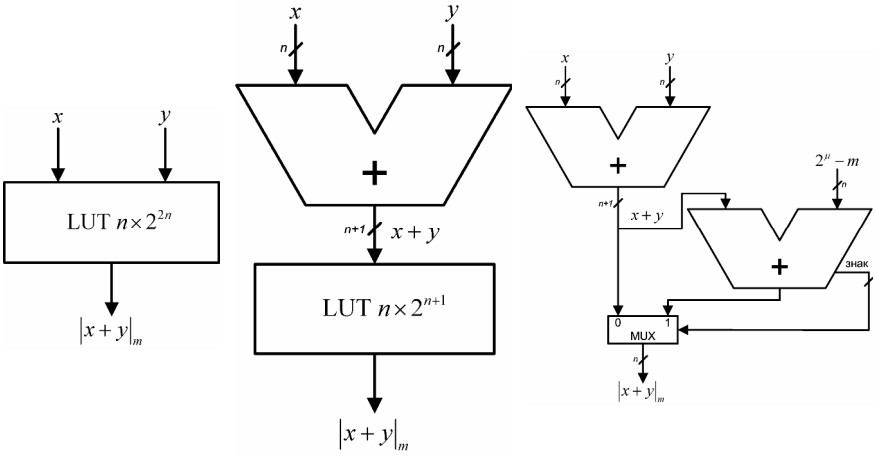
Поскольку в СОК используются модулярные операции, для высокой эффективности необходимо использовать специально спроектированные для СОК сумматоры и умножители.

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 $|x+y|_m$ 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, m – 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:

$$\begin{aligned} |xy|_m &= \left|\Phi(s^+) - \Phi(s^-)\right|_m \\ \Phi(s) &= \left|s^2\right|_m \\ s^+ = \frac{x+y}{2} \quad s^- &= \frac{x-y}{2}, \end{aligned} \tag{11}$$

и произведение $|xy|_m$ можно вычислять по формуле:

$$|xy|_m = \left|\left|\frac{1}{4}\right|_m \left|(x+y)^2 - (x-y)^2\right|_m\right|_m . \tag{12}$$

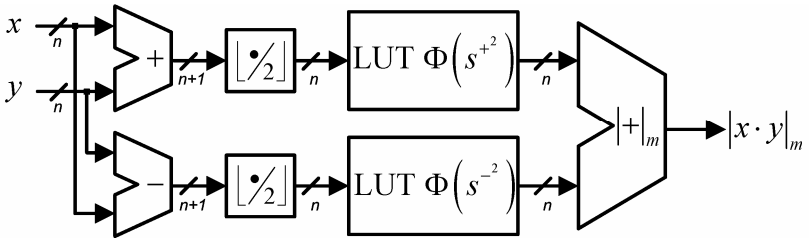


Рис. 3. Схема модулярного умножителя по модулю m на основе закона квадратов.

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 $Q=\{1,2,\mathbf{K},m-1\}$, и аддитивной по модулю (m_j-1) группой $I=\{0,1,\mathbf{K},m-2\}$. Этот изоморфизм может быть установлен следующим образом:

$$\forall q_n \in Q \quad \exists i_n \in I \quad : \quad q_n = g^{i_n} \tag{13}$$

и умножение над полем $GF(m)$ может производиться по формуле:

$$\left| q_j q_k \right|_m = g^{\left| i_j + i_k \right|_{m-1}} \tag{14}$$

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.

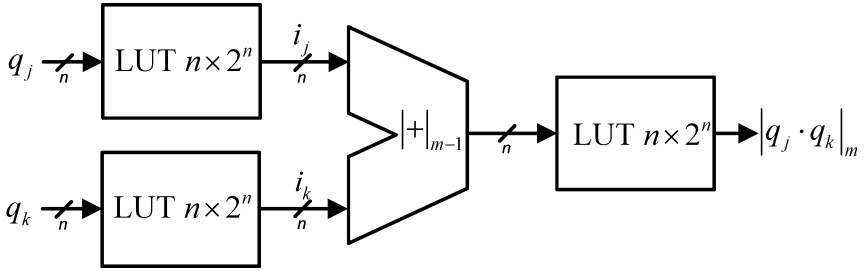


Рис. 4. Схема умножителя, основанного на исчислении степеней (умножитель Галуа).

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 $Q=\{1,2,\mathbf{K},31\}$ с элементом аддитивной группы $I=\{0,1,\mathbf{K},30\}$. 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, $\left|14\cdot 28\right|_{31}=20$.

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.

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

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.