

**ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА**

P11-87-234

Н. М. Никитюк

**БЫСТРЫЕ И ЭКОНОМИЧНЫЕ АЛГОРИТМЫ
ДЛЯ СПЕЦИАЛИЗИРОВАННЫХ ПРОЦЕССОРОВ
В СПЕКТРОМЕТРАХ ФИЗИКИ
ВЫСОКИХ ЭНЕРГИЙ.
ЗНАКОВЫЕ ЛОГАРИФМЫ
И СИСТЕМА ОСТАТОЧНЫХ КЛАССОВ**

Направлено в журнал
"Приборы и техника эксперимента"

1987

1. ВВЕДЕНИЕ

Специализированные процессоры (СП) характеризуются малой длиной слова (10-16 бит), упрощенной архитектурой и простотой управления [1,2]. Область применения таких процессоров в настоящее время весьма широка. Важным параметром СП, применяемых в спектрометрах физики высоких энергий, является высокое быстродействие, которое в значительной степени зависит от того, насколько эффективно реализуются такие трудоемкие операции, как умножение и деление. Так, процессор, описанный в работе [3], вычисляет следующее неравенство:

$$|AX_1 - BX_2 + CX_3 - D| > Z, \quad (1)$$

где A, B, C, D и Z - константы, $X_1 + X_3$ - переменные. Множество задач сводится к вычислению выражений типа $\sum A_i X_i$, где часто повторяется операция умножения. Операция деления необходима при создании СП для определения методом деления заряда, координаты частицы, регистрируемой с помощью позиционно-чувствительного детектора [4].

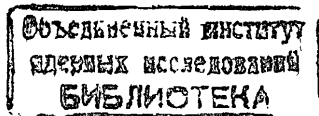
С целью упрощения схем умножения (деления) и увеличения быстродействия в ряде разработок операции умножения (деления) заменены на операции сложения (вычитания) логарифмов исходных чисел [5,6]. Однако для перевода логарифмов чисел в антилогарифмы необходимы дополнительные схемы и дополнительные затраты времени в тех случаях, когда требуются повторные вычисления. И, наконец, логарифмы отрицательных чисел, как известно, отсутствуют.

Целью данной работы является исследование таких малоизвестных алгоритмов для выполнения операций умножения и деления, которые можно было бы эффективно решать методами табличной арифметики с помощью постоянных программируемых ЗУ (ППЗУ). Ниже описаны два таких алгоритма - знаковые логарифмы и система остаточных классов (СОК). Эти системы счисления были использованы нами при создании СП для определения компланарности треков заряженных частиц. Особенностью знаковых логарифмов является возможность выполнения операций сложения и вычитания знаковых логарифмов чисел, причем исходные числа могут быть и отрицательными.

Система остаточных классов также имеет ряд полезных качеств, которые могут быть успешно использованы при построении СП.

2. ЗНАКОВЫЕ ЛОГАРИФМЫ

В системе счисления, основанной на знаковых логарифмах, некоторое действительное число A представляется знаком S_A и логарифмом L_A [7-9]:



если $|A| \geq 1/\mathcal{Z}$, то $L_A = \log_2 (\mathcal{Z}|A|)$,

если $|A| < 1/\mathcal{Z}$, то $L_A = 0$. (2)

Причем, если $A < 0$, то $S_A = -1$, если $A > 0$, то $S_A = 0$, и $S_A = 0$, или 1 , если $A=0$. В выражениях (2) \mathcal{Z} представляет собой константу масштабирования. Она используется для того, чтобы можно было получать логарифмы от чисел с отрицательным знаком. Число A и его знаковый логарифм связаны соотношениями

$$A = (1 - 2S_A)(1/\mathcal{Z}) 2^{L_A} \quad (3)$$

и

$$S_A = (1 - 2S_A)L_A, \quad (4)$$

где символом \sum_A обозначается знаковый логарифм. Для выполнения операций над знаковыми логарифмами с заданной точностью необходимо найти представление этих данных, аналогичное двоичной взвешенной системе. Дискретное представление K_A знакового логарифма \sum_A имеет вид:

$$K_A = [0,5 + 2^{b-1} L_A / 2^{b-1}] = [0,5 + 2^{b-1} \log_2 |\mathcal{Z}A|] / 2^{b-1}, \quad (5)$$

где $[X]$ — означает целое число, которое не превосходит X , и b — константа квантования. Другими словами, величина K_A представляет собой n — разрядное двоичное число:

$$K_A = (K_n K_{n-1} \dots K_b K_{b-1}) = \sum_{i=1}^n K_i 2^{i-b}.$$

Соотношение между \sum и A для $-16 \leq A \leq 16$ приведено на рис. 1 для $\mathcal{Z} = 16$. Поскольку логарифмическая функция является монотонной, то, как и следовало ожидать, в знаковой логарифмической системе можно сравнивать величины исходных чисел. Видно также, что, используя масштабный коэффициент \mathcal{Z} , можно вводить логарифмы отрицательных чисел. Следует также отметить, что около нулевой точки имеется область неопределенности шириной $2/\mathcal{Z}$.

Выполнение арифметических операций. Так же, как и в обычной системе счисления, используя знаковые логарифмы, можно эффективно выполнять такие трудоемкие операции, как умножение и деление. Однако, основное достоинство знаковых логарифмов состоит в том, что используя табличные методы решения, можно с достаточной для СП точностью выполнять операции сложения и вычитания логарифмов чисел.

Алгоритм сложения двух чисел имеет вид:

$$S = A + B \quad \text{или} \quad A = A(1 + B/A). \quad \text{Далее} \quad S = A \varphi(B/A),$$

где $\varphi(x) = 1+x$. Тогда логарифм суммы $L_S = L_A + L_\varphi$.

Если сложение выполняется в знаковых логарифмах, то $K_S = K_A + K_\varphi$.

Возможны два случая. Так, если $K_A \geq K_B$, то знак суммы $S_S = S_A$

и

$$K_S = K_A + \beta(K_B - K_A), \quad (6)$$

где

$$\beta(x) = [0,5 + 2^{b-1} \log_2 (1 + 2^x)] / 2^{b-1}. \quad (7)$$

Если же $K_B > K_A$, то $S_S = S_B$ и

$$K_S = K_B + \beta(K_A - K_B). \quad (8)$$

Операция вычитания $A-B$ выполняется следующим образом. Если $K_A > K_B$,

то разность K_P равна

$$K_P = K_A + \gamma(K_B - K_A), \quad (9)$$

где

$$\gamma(x) = [0,5 + 2^{b-1} \log_2 (1 - 2^x)] / 2^{b-1}. \quad (10)$$

Если же $K_B > K_A$, то знак разности S_P равен знаку S_B и

$$S_P = K_B + \gamma(K_A - K_B). \quad (11)$$

Если $K_A = K_B$, то S_P равен S_A или S_B и $S_P = 0$.

На рис. 2 приведена блок схема суммирующего/вычитающего устройства. Емкость ПЗУ, в котором хранятся табличные значения β и γ , составляет 2^n n — разрядных слов. Причем, b — разрядные знаковые логарифмы имеют динамический диапазон представления чисел, эквивалентный 16 двоичным числам.

Точность вычислений в знаковых логарифмах зависит от константы квантования b . Ошибка E вносится в процессе преобразования логарифма

L_A в его дискретное представление K_A :

$$K_A = L_A + E.$$

Величина E оценивается как $-2^{-b} \leq E \leq 2^{-b}$, а средне-квадратичное отклонение ошибки $\sigma_E^2 = 2^{-2b}/3$. Практически это значит,

что для повышения точности вычислений необходимо использовать ПЗУ

большой емкости. Используя выпускаемые промышленностью ПЗУ емкостью 512x8 бит, можно выполнять вычисления с точностью до целых чисел. Практическое применение знаковых логарифмов предполагает наличие специальных таблиц, в которых содержатся значения чисел X и соответствующие им величины $K_X, \beta(X)$ и $\gamma(X)$ при фиксированных значениях τ и ν . Такие таблицы можно вычислить с помощью ЭВМ. В табл. 1, 2 приведены фрагменты таблиц, описанных в работе [7]. Использование таких таблиц рассмотрим на конкретных примерах.

Пример 1. Сложить числа 2 и 3. Из таблицы 2 имеем $K_A=5,000$ и $K_B=5,625$. Решаем равенство (6): $K_S=5,625 + \beta(5,000-5,625)$. Из таблицы 1 имеем $\beta(-0,625) = 0,750$. Тогда $K_S = 6,375$. Из таблицы 2 имеем, что при $K_S = 6,375$ $X=5$.

Пример 2. Получить разность 3-4. Из таблицы 2: $K_A=5,625$ и $K_B=6,000$. Решаем равенство (9): $K_P=6,000 + \gamma(5,625-6,000)$. Используя таблицу 1, получаем: $\gamma(-0,375) = -2,125$. Тогда $K_P=3,875$. В таблице 2 наиболее близкой величиной числу 3,875 является цифра $K_X=4,000$, которой соответствует значений $X=1$. Этот пример иллюстрирует приближенный характер вычислений над числами, представленными в знаковых логарифмах. Операции умножения и деления в знаковых логарифмах выполняются так же, как с помощью обычных логарифмов.

Таблица 1

$\beta(x)$ и $\gamma(x)$ в зависимости от X			
$\tau=16 \quad \nu=4$			
X	$\beta(X)$	X	$\gamma(X)$
- 0,500 и - 0,625	0,750	- 0,125	- 3,625
- 0,750 до - 1,000	0,625	- 0,250	- 2,625
- 1,125 до - 1,375	0,500	- 0,375	- 2,125
- 1,500 до - 2,000	0,375	- 0,500	- 1,750

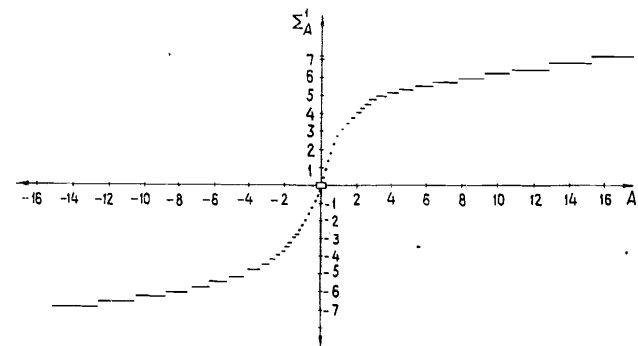


Рис. 1. Соотношение между Σ'_A и A для $-16 \leq A \leq 16$ при $\tau=8$ и $\nu=3$.

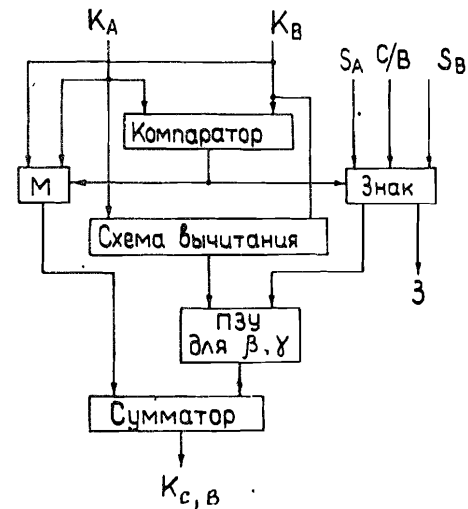


Рис. 2. Блок-схема суммирующего/вычитающего устройства в знаковой логарифмической системе. М- мультиплексор, С/В-признак сложения/вычитания, З - знак результата, $K_{с, в}$ - результат суммы/разности.

Таблица 2

Зависимость K_x от X при $\tau = 16$ и $\nu = 4$

X	K_x	X	K_x
0,875	3,750	3,875 до 4,125	6,000
1,000	4,000	4,250 до 4,500	6,125
1,125	4,125	4,625 до 4,875	6,250
.....	5,000 до 5,375	6,375
2,000	5,000
.....	5,500 до 5,875	6,500
3,000 и 3,125	5,625		

Расчеты показывают, что если для выполнения арифметических операций использовать ШЗУ, а в качестве элементарной операции рассматривать операцию сложения (вычитания) с временем T_c , то получается следующая таблица [7].

Таблица 3

Оценка времени выполнения операций

Операции	Знаковые логарифмы	Обычный метод
Сложение	$4T_c$	T_c
Вычитание	$4T_c$	T_c
Умножение	$2T_c$	$7T_c$
Деление	$2T_c$	$14T_c$

При оценке времени выполнения операций не учитывались задержки в комбинационных схемах, величина которых намного меньше, чем время задержки в ШЗУ.

3. СИСТЕМА СЧИСЛЕНИЯ В ОСТАТОЧНЫХ КЛАССАХ

Используемые в практике системы счисления можно разделить на два класса: позиционные и непозиционные системы счисления. В силу известных свойств наиболее широкое применение нашли позиционные системы счисления, которые, однако, имеют существенный недостаток, связанный с необходимостью учитывать переносы при выполнении арифметических операций. Подобная зависимость ограничивает возможности получения высокого быстродействия. В системе остаточных классов (СОК),

которая является типично непозиционной системой счисления, зависимость между разрядами отсутствует и поэтому имеется возможность их независимой параллельной обработки [10, 11].

Возможность использования СОК для кодирования чисел в машинах впервые была предложена М. Валахом и развита для использования дробных чисел А. Свободой [12]. Поэтому СОК еще называют чешской системой. Эта система получила развитие и в трудах советских ученых [10, 11, 13]. Известно, что любое целое число A можно задать последовательностью его наименьших положительных вычетов (остатков) A_1, A_2, \dots, A_n , которые получаются в результате деления числа A на целые числа P_1, P_2, \dots, P_n .

Если числа P_1, P_2, \dots, P_n попарно простые, то каждой последовательности остатков можно поставить в соответствие только одно число из диапазона $(0, P)$, где $P = \prod_{i=1}^n P_i$ — диапазон, или период повторения чисел, и, наоборот, каждому числу из этого диапазона соответствует только одна последовательность остатков. Эти свойства позволяют рассматривать последовательность остатков как систему счисления. Числа P_1, P_2, \dots, P_n являются основаниями СОК, причем цифра числа A определяется из соотношения

$$A_i = A - \left[\frac{A}{P_i} \right] P_i, \quad \text{где } i = 1, 2, \dots, n,$$

$\left[\frac{A}{P_i} \right]$ — целая часть числа $\frac{A}{P_i}$ и $A_i \leq P_i < 1$.

Пример 1. Рассмотрим следующие основания системы $P_1 = 3; P_2 = 5; P_3 = 7, P = 3 \cdot 5 \cdot 7 = 105$. Представим в СОК число 95.

Имеем $A_1 = 95 - \left[\frac{95}{3} \right] 3 = 95 - 93 = 2$. Аналогично имеем $A_2 = 0$ и $A_3 = 4$. Еще проще: $A_1 = 95:3=2 \pmod{3}; A_2 = 95:5=0 \pmod{5}$ и $A_3 = 95:7=4 \pmod{7}$.

Итак, $95 = (2, 0) 4$.

Прежде чем рассматривать арифметические операции в СОК, приведем способ перевода чисел из СОК в десятичную систему счисления. В соответствии с китайской теоремой об остатках любое число A может быть представлено в виде

$$A = \frac{P}{P_1} B_1 + \frac{P}{P_2} B_2 + \dots + \frac{P}{P_n} B_n - z_A P, \quad (12)$$

где z_A — неопределенное число, B_1, B_2, \dots, B_n — ортогональный базис, и его значение зависит от принятых оснований P_1, P_2, \dots .

P_n . Для нахождения ортогонального базиса проще всего воспользоваться таблицей 4 [12].

Таблица 4

Базисные ортогональные векторы и соответствующие числа для некоторых оснований

Основание	Диапазон	V_1	V_2	V_3	V_4	V_5	V_6
2,3	6	I5	IO	6			
2,3,5	30	I5	IO	6			
3,5,7	IO5	70	2I	I5			
2,3,5,7	2IO	IO5	70	I26	I20		
2,3,5,7,II	23IO	II55	I540	I386	330	2IO	

Метод вычисления ортогонального базиса подробно описан в работе [4].

Пользуясь соотношением [12] и таблицей 4 найдем десятичный эквивалент числа $A=(2,0,4)$. В системе оснований 3,5,7 имеем $A=2 \cdot 70+0 \cdot 2I+4 \cdot I5=2 \cdot I05=200 - z_A \cdot I05 = 200 - I \cdot I05=95$. Величина z_A выбирается такой, чтобы полученное число входило в заданный диапазон, причем величина z_A может быть и равна нулю. Применительно к СП можно заведомо выбрать такой диапазон чисел, при котором $z_A=0$. Арифметические операции в СОК. Как уже отмечалось выше, особенностью СОК является то, что арифметические операции выполняются по каждому основанию независимо друг от друга. Поскольку основания являются относительно небольшими числами, то появляется возможность широкого использования табличных методов решения с помощью ПЗУ.

Пример 2. Сложить число $A=24$ с числом $B=57$. Как и выше, основаниями системы являются $P_1=3$, $P_2=5$, $P_3=7$ и $P=IO5$.

Имеем $A=24=(0,4,3)$; $B=(0,2,I)$.

$$\text{Имеем: } \begin{array}{r} P_1=3 \\ 0 \\ 0 \\ 0 \end{array} \quad \begin{array}{r} P_2=5 \\ 4 \\ 2 \\ I \end{array} \quad \begin{array}{r} P_3=7 \\ 3 \\ I \\ 4 \end{array}$$

Проверка дает $A+B=0 \cdot 70+I \cdot 2I+4 \cdot I5= z_A \cdot I05$. Нетрудно заметить, что $z_A=0$, тогда $A+B=8I$. Операция вычитания в тех случаях, когда ее результат положителен, выполняется вычитанием соответствующих цифр разрядов, причем, всегда в результате приводится наименьший остаток. Если же разность цифр оказалась отрицательной, то берется ее дополнение к основанию. Следует также отметить, что специального символа для знака в СОК не существует, если не считать различных модификаций этой системы.

Пример. Найти разность чисел $B-A=57-24=33$.

$$\text{Имеем: } \begin{array}{r} P_1=3 \\ 0 \\ 0 \\ 0 \end{array} \quad \begin{array}{r} P_2=5 \\ 2 \\ 4 \\ 3 \end{array} \quad \begin{array}{r} P_3=7 \\ I \\ 3 \\ 5 \end{array}$$

Проверка дает $A-B=0 \cdot 70+3 \cdot 2I+5 \cdot I5= z_A \cdot I05=I38 - z_A \cdot I05=33$ ($z_A=I$). Умножение двух чисел выполняется путем поразрядного умножения сомножителей.

Пример. Умножить число $A=I9=(I,4,5)$ на $B=4=I,4,4$.

$$\begin{array}{r} P_1=3 \\ I, \\ I, \\ I, \end{array} \quad \begin{array}{r} P_2=5 \\ 4, \\ 4, \\ I, \end{array} \quad \begin{array}{r} P_3=7 \\ 5 \\ 4 \\ 6 \end{array}$$

Проверка дает $A \cdot B=I \cdot 70+I \cdot 2I+6 \cdot I5= z_A \cdot I05=I8I - z_A \cdot I05=76$ ($z_A=I$). Для представления отрицательных чисел и выполнения операций над ними применяется простой способ разбиения диапазона чисел на два поддиапазона, один из которых используется для изображения положительных, а другой — для отрицательных чисел. С этой целью в систему оснований нередко вводится число $P_0=2$. Тогда общий диапазон будет

$$P_0 P = P_0 \prod_{i=1}^n P_i = P_0 P_1 P_2 \dots P_n.$$

В этом случае можно оперировать с числами, которые удовлетворяют условию

$$0 < |A| < P.$$

Инверсный код числа A обозначается как $[A]_{инв}$ [13], и образуется в соответствии с формулой

$$[A]_{инв} = \begin{cases} A & \text{если } A \geq 0 \\ 2P+A & \text{если } A < 0. \end{cases}$$

В инверсном коде положительные числа представляются в поддиапазоне от 0 до P , а отрицательные — от P до $2P$.

Между цифрой числа A , представляемого цифрами a_0, a_1, \dots, a_n , и числом A в инверсном коде, представляемом цифрами a'_0, a'_1, \dots, a'_n , существуют соотношения $a_0+a'_0=P_0$, $a_1+a'_1=P_1$ и т.д.

Пример. $P_0=2$, $P_1=3$, $P_2=5$, $P_3=7$, $P=3 \cdot 5 \cdot 7=IO5$.

Представить число $A=-59$ в инверсной форме. $59=I,2,4,3$, тогда $A=(I,I,I4)$.

При сложении инверсных кодов чисел результат получается также в инверсном коде, а операция вычитания сводится к операции сложения, если цифры вычитаемого инвертировать.

Пример. Выполнить сложение и вычитание чисел $A=19$ и $B=-25$;
 $P_0=2, P_1=3, P_2=5, P_3=7$. $A_{инв}=(1,1,4,5)$, $B=(1,1,0,4)$;
 $A+B_{инв}=(1,1,4,5)+(1,2,5,3)=(0,0,4,1)$,
 $A-B_{инв}=(1,1,4,5)+(1,1,0,4)=(2,2,4,2)$.
 Используя таблицу I, можно проверить, что $19+(-25)=-6$, или
 $A+B_{инв}=0 \cdot 105+0,70+4 \cdot 126+1 \cdot 120-2 \cdot 10=-6$ (принимаем $Z_A=3$).
 Аналогично имеем: $A-B_{инв}=44=2 \cdot 105+2 \cdot 70+4 \cdot 126+2 \cdot 120=1094-5 \cdot 210$
 (здесь $Z_A=5$).

Существенным является тот факт, что для получения инверсного кода произведения необходимо выполнить умножение инверсных кодов сомножителей. Известна также так называемая искусственная форма представления отрицательных чисел в СОК [10]. Однако инверсная форма представления чисел более предпочтительна, так как при сложении в инверсном коде требуется меньше элементарных операций, чем в искусственной форме.

Рассмотренные выше примеры характеризуют следующие основные достоинства СОК.

- Независимость образования разрядов чисел и возможность их параллельной обработки (без учета переносов).
- Малоразрядность остатков, представляющих число, что позволяет широко использовать быстродействующие ПЗУ для выполнения таких операций, как сложение, вычитание и умножение методами табличной арифметики.

Однако СОК имеет ряд существенных недостатков, которые ограничивают область применения.

- По виду числа невозможно определить его величину.
- Нет простых способов определения знака числа и выхода результатов вычисления за пределы диапазона.
- Ограниченность действия системы областью целых положительных чисел.
- Исключается возможность получения приближенного значения результата и его округления.
- Нет простых схем для сравнения амплитудных значений и выполнения операции деления.

В разное время появился ряд работ, направленных на устранение недостатков СОК и ее применение для построения цифровых фильтров [14-26]. Однако эти усовершенствования, в свою очередь, приводят к усложнению аппаратных средств.

4. МЕТОДЫ РЕАЛИЗАЦИИ ОПЕРАЦИИ УМНОЖЕНИЯ В СОК

Наиболее известны два метода умножения по модулю P_i : с помощью табличного поиска при малом P_i (5 бит и менее) и методом

индексного счисления при большом P_i (6 бит и более). Эффект выполнения операции умножения в СОК методом табличного поиска можно показать на следующем примере [26]. Если для прямого умножения табличным методом двух 8-разрядных чисел в диапазоне 0-65525 требуется таблица, содержащая 434660 комбинаций, то в СОК с модулями 3,5,7,8, 11 и 13 в диапазоне 0-120119 требуется 580 комбинаций.

Если модуль P_i - простое число, то индексное счисление позволяет получить очень экономичную реализацию умножителя с помощью метода, аналогичного методу вычисления логарифмов [10,27]. Этим методом можно пользоваться в тех случаях, когда величину P можно разложить на простые числа.

Так, на рис. 3 приведена схема умножителя по модулю $183=61 \times 3$. Поскольку числа 61 и 3 простые, то возможно индексное счисление по разложенным модулям, хотя вычисление по модулю 3 можно выполнять, не прибегая к индексам. Для этой схемы требуется всего 4 ПЗУ на 4 уровня. Аналогично можно построить схему умножителя по модулю $249=83 \times 3$.

Пример. Умножить с помощью индексного счисления два числа $A=5$ и $B=6$ по модулю $P=61$. Находим индексы величин, входящих в произведение, приняв первообразный корень для $P=61$ равным 7. Из таблиц [10]

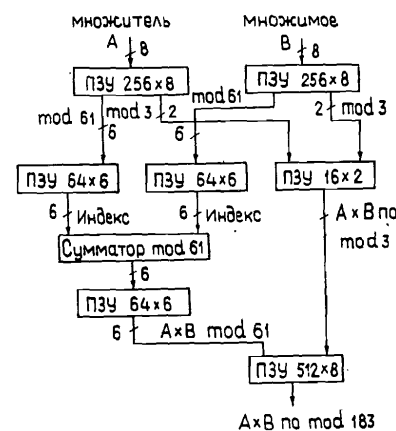


Рис. 3. Блок-схема умножителя по модулю 183 по методу индексного счисления.

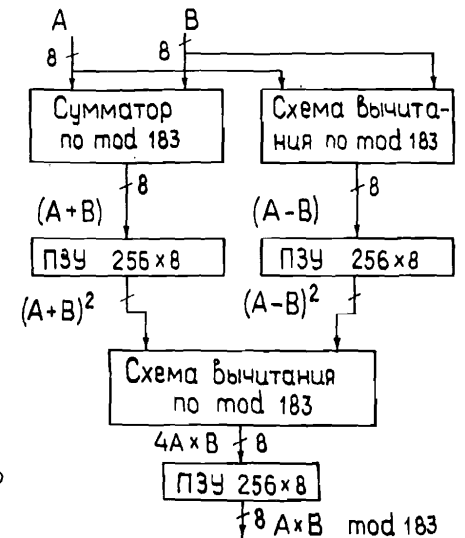


Рис. 4. Экономичная и быстродействующая схема умножителя по модулю 183.

находим $ind\ 5=58, ind\ 6=43$. Находим сумму индексов $ind\ AxB=43+58=101\ mod\ (6I-1)$. По таблицам $[10]$ находим антииндекс $4I: N(4I)=30$. Таким образом, $5x6=30$.

Кроме того, в системе СОК эффективно реализуется схема умножения, которая работает по принципу квадратичного умножителя:

$$A \times B = \frac{(A+B)^2 - (A-B)^2}{4}. \quad (13)$$

На рис. 4 приведена схема умножителя по модулю 183. Наличие в ней всего трех ППЗУ на два уровня говорит о том, что это — наилучшая реализация по быстродействию и стоимости [27].

ЗАКЛЮЧЕНИЕ

Требование к быстродействию и надежности в работе СП в больших спектрометрах физики высоких энергий и других областях науки и техники приводит к необходимости изучения и практического использования малоизвестных, но эффективных математических алгоритмов, которые адекватны современной полупроводниковой технологии. Если сравнивать два рассмотренных выше метода выполнения арифметических операций, то можно отметить следующее.

Знаковые логарифмы являются более универсальной системой счисления по сравнению с СОК. Обладая всеми качествами обычных логарифмов, система знаковых логарифмов имеет два весьма ценных качества: возможность представления отрицательных чисел и допустимость выполнения операций сложения и вычитания непосредственно над логарифмами чисел без перевода промежуточного результата обратно в логарифмы. Однако для практического применения знаковых логарифмов необходимо иметь таблицы, так как в литературе они приведены для малого диапазона чисел (± 16).

Составление таких таблиц (типа I-2) с помощью ЭВМ не представляет особых затруднений. Используя современные ППЗУ с числом адресных входов 8-9 бит, можно получить вычисления с точностью до единицы младшего разряда целого числа. Варьируя величину ϵ , можно менять точности вычислений.

Система СОК может успешно использоваться в тех случаях, когда необходимо вычислять с высокой скоростью выражения типа $A \times B \times C$ или $\sum_{i=1}^n A_i X_i$ и т.д. над целыми числами. Как показано в работах [28, 29], СОК эффективно реализуется с помощью оптических элементов.

В настоящее время как знаковые логарифмы, так и СОК используются для построения цифровых фильтров, для обработки изображений, в компьютерной томографии и проч.

ЛИТЕРАТУРА

1. Verkerk C. Special purpose processors. CERN-preprint, DD/74/27, CERN, Geneva, 1974.
2. Никитюк Н.М. Фильтрация цифровой информации на линии с электронной вычислительной машиной в физике высоких и средних энергий. - ПТЭ, 1983, № 2, с.7.
3. Guzik Z., Basiladze S.G. - Nucl. Instrum and Meth., 1974, v.144, N.1, p.83.
4. Fischer B.E. Processor for position sensitive detectors. Nuclear Instrum. and Meth., 1977, vol.141, No.1, p173.
5. Hogan G.E. - Nucl. Instrum. and Meth., 1978, v.165, N 1, p. 7.
6. Frenkel A., Martellotti G., Prolo E., Tusi A. A hardware processor for a missing energy trigger. Nucl. Instrum. and Meth, 1986, vol.A.245, No 1, p.45.
7. Swartzlander E.E. The Sing/Logarithm number system. IEEE Trans. on Comput., 1975, v. C-24, N 12, p.1238.
8. Swartzlander E.E. Gilbert B.K. Arithmetic ultra-high-speed tomography. IEEE Trans.on Comput., 1980, v. C-29, N 5, p.341.
9. Swartzlander E.E. Satish Chandra D.V., Troy Nagle H. a.o. Sign/Logarithm arithmetic for FFT implementation. IEEE Trans. on Comput., 1983, v.C-32, No 6, p.526.
10. Акушкин И.Я., Юдицкий Д.Н. Машинная арифметика в остаточных классах. "Сов.Радио", М., 1968, 439 с.
11. Торгашев В.А. Система остаточных классов и надежность ЦВМ. "Сов.Радио", М. 1973, 115 с.
12. Свобода А. Развитие вычислительной техники в Чехословакии. Система остаточных классов. Кибернетический сборник. "Мир", М., 1964, № 8, с.115.
13. Пенюгалов Н.А. Некоторые вопросы применения системы остаточных классов в электронных цифровых вычислительных машинах. Сб. Цифровая вычислительная техника и программирование, "Сов.Радио", М., 1968, с.38.
14. Banerji D.K. Brozozowski J.A. Sign detection in residue number systems, IEEE Trans.on Comput., vol.C-18, No.4, 1969, p.313
15. Rao R.N., Trenan A.K. Binary logic for residue arithmetic using magnitude index. IEEE Transaction on Comput.vol.C-19, No.8, 1970, p.752.
16. Soderstand M.A. Multipliers for residue -number-arithmetic digital filters. Electronic letters, vol.13, No.6, p.164.
17. Jenkins W.K. Techniques for residue to-analog conversion for residue-encoded digital filters. IEEE Trans.on circuits and systems, vol.-CAS-25, No.7, 1978, p.555.

18. Baranieska A, Jullien G.A. On decoding techniques for residue number system realizations of digital signal processing hardware. IEEE Transaction on circuits and systems, vol.CAS-25, No.11, 1978, p.935.
19. Ben-Dau Tseng, Jullien G.A., Miller W.C. Implementation of FFT structures using the residue number system. IEEE Trans.on Comput, vol.C-28, No.11, 1979, p.831.
20. Chao H. Huang, Peterson D.G., Rauch H.E. a.o. Implementation of fast digital processor using the residue number system. IEEE Trans. on circuits and systems, vol. CAS-28, No.1, 1981, p.32.
21. Huang C.H. A fully parallel mixedradix conversion algorithm for residue number applications. IEEE Trans. on comput., vol.C-32, No.4, 1983, p.398.
22. Taylor P.J. An overflow-free residue multiplier. IEEE Trans.on Comput., vol.C-32, No.5, 1983, p.501.
23. Ulman Z.D. Sign detection and implicit conversion of numbers in residue arithmetic. IEEE Trans. on Comput.,vol.C-32, No.6,1983, p.590.
24. Thu Van Vu, Efficient implementations of the Chinese remainder theorem for sign detection and residue decoding. IEEE Trans on Comput., vol. C-34, No.7, 1985, p.646.
25. Chakzaborti N.B., Soundarazajan J.S., Reddy A.L.N. An implementation of mixed-radix conversion for residue number applications. IEEE Trans. on Comput., vol.C-35 No.8, 1986, p.762.
26. Guest C.C., Mirsaleni M.M., Gaylord T.K. Residue number system truth-table look-up processing-moduli selection an logical minimization.IEEE Trans.on Comput., vol. C-33, No.10, 1984, p.927.
27. Содерстренд М.А., Верниа К. Недорогой быстродействующий умножитель по модулю P_i и его применение в арифметических устройствах на основе системы счисления в остаточных классах. ТИИЭР, т.68, №4, 1980, с.114
28. Miller D.D., Polky J.N. A residue number system implementation of the LMS algorithm using optical waveguide circuits. IEEE Trans. on Comput., vol.C-32, No.11, p.1013.
29. Huang A., Tsunoda J., Goodman J.W., a.o. Optical computation using residue arithmetic. Applied Optics, vol.18,1979, No.2, p.149.

Рукопись поступила в издательский отдел
10 апреля 1987 года.

Никитюк Н.М.

P11-87-234

Быстрые и экономичные алгоритмы для специализированных процессоров в спектрометрах физики высоких энергий. Знаковые логарифмы и система остаточных классов

Описаны два быстрых и экономичных алгоритма, которые были исследованы и использованы для построения специализированных процессоров /СП/ в спектрометрах физики высоких энергий, - знаковые логарифмы и система счисления в остаточных классах /СОК/. Характерной особенностью знаковых логарифмов является возможность выполнения операций сложения и вычитания логарифмов чисел и возможность представления чисел с отрицательным знаком. В СОК очень быстро, без затрат времени на переносы между разрядами чисел, выполняются операции сложения, вычитания и умножения над целыми числами. Обе малоизвестные системы счисления имеют то преимущество, что они эффективно реализуются табличными методами решения с помощью ПЗУ. С этой целью с помощью ЭВМ созданы специальные таблицы для выполнения арифметических операций над 12-разрядными двоичными числами. Приводится ряд примеров, иллюстрирующих эффективность рассмотренных алгоритмов.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1987

Перевод автора

Nikitjuk N.M.

P11-87-234

Fast and Economical Algorithms for Special Purpose Processors Used in High-Energy Physics Spectrometers. Sign Logarithms and a Residue System

Two fast and economical algorithms are described which can be effectively used for the construction of special purpose processors for high-energy physics spectrometers, sign logarithms and a residue system. The characteristic property of the sign logarithms is the possibility of executing addition and subtraction operations of the logarithms of numbers and the possibility of presenting numbers with negative sign. The operations addition, subtraction and multiplication over whole numbers are carried out in the residue system very fast, without spending time for transfer between digits of numbers. Both, not well-known, number systems have the advantage that they are effectively executed by the table method with the help of PROM. For this purpose special tables made by computer are created for the execution of arithmetical operations over 12-digits biner numbers. Several examples which illustrate the efficiency of the considered algorithms are given.

The investigation has been performed at the Laboratory of High Energies, JINR.

Preprint of the Joint Institute for Nuclear Research, Dubna 1987