УДК 004.272.2

Применение аппарата модулярной логарифметики для решения специальных задач матричной алгебры

В.М. Амербаев, Е.С. Балака, Д.В. Тельпухов, А.В. Константинов

Федеральное государственное бюджетное учреждение науки Институт проблем проектирования в микроэлектронике Российской академии наук, Ssapra@hotbox.ru

***Аннотация* — Центральным вопросом данной статьи является анализ матричных вычислений над полем *GF(p)*. При этом основной акцент делается на матричном аналоге модулярной арифметики, как основы для распараллеливания операций алгебры матриц, и логарифметике конечного поля *GF(p)*, как основы модулярных вычислений.**

***Ключевые слова* — модулярная логарифметика, логарифм Якоби, логарифм Гаусса, алгебра квадратных матриц.**

I. Введение

В настоящее время в связи с развитием современной техники, информационных и управляющих систем все большее применение находят новые принципы на основе представления данных в модулярной системе счисления (МСС) [1].

В МСС любое целое число представляется в виде набора остатков от деления на выбранные базовые простые модули. Арифметические операции распараллеливаются по каждому модулю, вычисления проводятся независимо по каждому модульному каналу, высокое быстродействие достигается за счет того, что сами операнды малые по величине. Диапазон представления целых чисел в модулярной системе счисления определяется произведением всех ее базовых модулей. Помимо высокого быстродействия, к важным положительным качествам модулярной арифметики относятся высокая точность и надежность вычислений, способность системы контролировать и исправлять ошибки во время выполнения модульных операций (внутренняя самокоррекция). Однако, возникают трудности при реализации так называемых немодульных операций, для выполнения которых необходимо знание цифр операндов по всем разрядам. Что значительно ограничивает лишь область применения МСС, и поэтому такие системы редко реализуются в машинных блоках общего назначения. Для модулярной арифметики представляет интерес класс вычислительных задач с низким процентным составом немодульных операций, а также задачи, где сокращение немодульных операций достигается технически (экономически) допустимым увеличением вычислительного диапазона цифрового устройства. К такому классу задач относятся матричные вычисления, являющиеся ключевыми во многих приложениях.

Следует заметить, что МСС не претендует на статус универсальной системы счисления. Комбинированное применение модулярной арифметики и двоичной системы счисления при построении управляющих систем – наиболее эффективный подход для решения указанного класса специализированных задач.

II. Модулярная логарифметика как основа модулярных вычислений в задачах алгебры матриц

Как уже отмечалось в [2], в модулярной арифметике существует проблема сокращения «накладных расходов» на реализацию модульных операций. Стремление разработчиков снизить накладные расходы, вызванные аддитивным модулярным представлением операндов, привело к идеям перехода к мультипликативному представлению. Традиционным подходом к решению поставленной задачи является отказ от аддитивного представления только для реализации операции умножения по простому модулю p [3]. При таком подходе возникает проблема разбалансировки между затратами на аддитивные и мультипликативные операции. Для того, чтобы все модульные операции приобрели однотипность, Д.А. Поспелов в своей работе [4] предложил каждый вычет $|x|_{p_i}$ представлять парой $\left\langle |x|_{p_i}, \lg_w |x|_{p_i} \right\rangle$. Однако, такой подход к реализации модульных операций требует удвоения регистров операндов.

Алгоритмы матричных вычислений в большинстве своем содержат большое число операций умножений, что приводит к большим временным затратам, если использовать описанные выше подходы к реализации модульных операций. На данный момент в ИППМ РАН нами разрабатывается совершенно новый подход – полный отказ на уровне модульных операций от аддитивных вычислений и переход к мультипликативным. Тем самым, весь

вычислительный процесс строится на базе модулярной логарифметики поля *GF(p)*.

Логарифметика поля *GF(p)* разработана выдающимися математиками К.Ф. Гауссом и Г.Я. Якоби в первой половине XIX века [5],[6]. Процедура изоморфного конструирования операций логарифметики поля *GF(p)*, а также схемы реализаций, а также их оценки аппаратной и временной сложности, основных покомпонентных операций, подробно описаны в работах [7,8]. Существенно, что предложенные конструкции включают в себя процедуры взаимодействия с сингулярными значениями расширенного дискретного логарифма, в отличие от индексной арифметики, рассмотренной в [9].

Опираясь на КТО (китайская теорема об остатках), имеем: пусть $p_1, p_2, ... p_s$ – попарно взаимно-простые натуральные числа и

$$P = p_1 \cdot p_2 \cdot ... \cdot p_s, P_i = \frac{P}{p_i} \ (1 \le i \le r).$$

Любая матрица $X \in M_n(Z_p)$ единственным образом представима в форме: $X = \left| \sum_{k=1}^{r} \left| X \right|_{p_k} \cdot P_k^{-1} \right|_{p_k} \cdot P_k \Bigg|_P$. Не трудно показать, что справедливы следующие утверждения:

а) $\forall A, B \in M_n(Z)$ и $\forall p \in \mathrm{N} \ (p \ne 0,1)$ имеет место $\left| A \pm B \right|_p = \left| \left| A \right|_p \pm \left| B \right|_p \right|_p$; $\left| A \cdot B \right|_p = \left| \left| A \right|_p \cdot \left| B \right|_p \right|_p$ $\left| A * B \right|_p = \left| \left| A \right|_p * \left| B \right|_p \right|_p$, где $*$ - операция тензорного произведения.

б) Функция $y = \left| x \right|_m$, определенная на $M_n(Z)$, гомоморфно отображает кольцо $M_n(Z)$ в кольцо $M_n(Z_p)$.

Любая матрица $X \in M_n(Z_p)$, где $P = p_1 \cdot p_2 \cdot ... \cdot p_s$, единственным образом разложима в полиадический ряд

$$X = A_1 + A_2 p_1 + A_3 p_1 p_2 + ... + A_r p_1 ... p_{r-1},$$

где $A_1 = X_{p_1}, A_k = \left\| \frac{x}{p_1 ... p_{k-1}} \right\|_{pk} \ 2 \le k \le r.$

Соответствующую матричную интерпретацию имеют все немодульные операции модулярной арифметики. Таким образом, модулярная арифметика является эффективным инструментом распараллеливания матричных вычислений.

III. Алгоритмы и схемы реализаций базовых операций матричных вычислений

Матричные вычисления строятся на иерархии базовых операций линейной алгебры: скалярное произведение векторов (DOT); линейная комбинация векторов (SAXPY); умножение матрицы на вектор (GAXPY). Все эти операции могут быть описаны в алгоритмическом виде. Для записи алгоритмов воспользуемся стилизированной версией языка Matlab.

Скалярное произведение векторов (DOT) – одна из базовых операций линейной алгебры, состоящая из скалярных операций сложения и умножения: по векторам $x, y \in V_n(Z_p)$ необходимо вычислить $c = x^T y$:

for $i = 1:n$

$$\left| c \right|_p = \left| c + \left| x(i) y(j) \right|_p \right|_p$$

end

Сложность такого алгоритма – *O(n)* (объем работы линейно зависит от размерности векторов).

В базисе логарифметики задача нахождения скалярного произведения векторов сводится к задаче нахождения логарифма Гаусса от N переменных. Обозначим ее как:

$$G(z_1, z_2, ..., z_N) = \lg_w | \sum_{i=1}^{N} w^{|Z_i|_{p-1}} |_p,$$

где $\left| z_i \right|_{p-1} = \left| \lg_w \left| x_i \right|_p + \lg_w \left| y_i \right|_p \right|_{p-1}$.

Методы вычисления гауссова логарифма, а также оценки соответствующих аппаратных решений, детально рассмотрены в работах [10,11].

Другой вариант построения алгоритма матрично-векторного умножения основывается на операции SAXPY (*scalar alpha x plus y*): по векторам $x, y \in V_n(Z_p)$ и скаляру $\alpha \in Z_p$ необходимо вычислить $z = \alpha x + y$:

for $i = 1:n$

$$\left| z_i \right|_p = \left| \left| \alpha x(i) \right|_p + y(i) \right|_p$$

end

Сложность SAXPY имеет тот же порядок, что и скалярное произведение, *O(n)*. Отличие ее состоит в том, что она возвращает не скаляр, а вектор.

Таким образом, вычисление z_i ($1 \le i < n$) компоненты вектора результата операции необходимо проводить по схеме: пусть $\alpha' = \lg_w \left| \alpha \right|_p$,

$x'_i = \lg_w |x_i|_p$, $y'_i = \lg_w |y_i|_p$, $z'_i = \lg_w |z_i|_p$. Имеем:

$$z'_i = \left| |\alpha' + x'_i|_{p-1} + J_w\left(\left| y'_i - |\alpha' + x'_i|_{p-1} \right|_{p-1} \right) \right|_{p-1} .$$

Типовая архитектура блока, реализующего данную операцию, представлена на рис. 1.

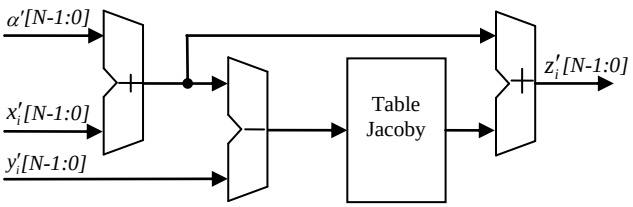

Рис. 1. Типовая архитектура блока вычисления операции SAXPY в модулярной логарифметике по модулю *p*

Замечание. Табличный метод реализации логарифма Якоби существенно влияет на рост аппаратных затрат при увеличении разрядности операндов. Ниже представлено тождество, позволяющее сократить объемы таблиц вдвое, за счет использования дополнительной логики:

$$J_w(|u|_{p-1}) = \left| |u|_{p-1} + J_w\left(|(p-1) - u|_{p-1} \right) \right|_{p-1} .$$

Следующий вариант построения алгоритма матрично-векторного умножения основывается на операции, называемой модификация матрицы внешним произведением: пусть $A \in M_n(Z_p)$, $x, y \in V_n(Z_p)$, тогда $A := A + xy^T$, т.е. элементы матрицы A перевычисляются по формуле: $a_{ij} = a_{ij} + x_i y_i$.

В логарифметике задача сводится к вычислениям логарифма Гаусса от двух переменных, с модификацией одного их слагаемых:

пусть $a'_{ij} = \lg_w |a_{ij}|_p$, $x'_i = \lg_w |x_i|_p$, $y'_i = \lg_w |y_i|_p$, тогда

$$a'_{ij} = \left| a'_{ij} + J_w\left(\left| x'_i + y' \right|_{p-1} - a'_{ij} \Big|_{p-1} \right)_i \right|_{p-1} .$$

Способы построения модульных логсумматоров и умножителей, а также соответствующие оценки аппаратных и временных затрат, подробно рассмотрены в работе [12].

Модификация матрицы внешним произведением векторов занимает большое место в традиционных формулировках многих важных матричных алгоритмов, большинство которых можно переформулировать так, что доминирующей становится операция GAXPY (general Ax plus y): по векторам $x, y \in V_n(Z_p)$ и $A \in M_n(Z_p)$ необходимо вычислить $z = y + Ax$:

$z = y;$

for $j = 1 : n$

$$|z|_p = \left| z + |x(j)A(:, j)|_p \right|_p$$

end

Таким образом, в векторе z накапливается сумма, значение которой обновляется последовательностью операций SAXPY: пусть $x'(j) = \lg_w |x(j)|_p$, $A'(:, j) = \left\{ \lg_w |a_{1j}|_p, \lg_w |a_{2j}|_p, ..., \lg_w |a_{nj}|_p \right\}$, $z' = \left\{ \lg_w |z_{11}|_p, \lg_w |a_{12}|_p, ..., \lg_w |a_{1n}|_p \right\}$, тогда

$$z' = \left| z' + J_w\left(\left| x'(j) + A'(:, j) \right|_{p-1} - z' \Big|_{p-1} \right) \right|_{p-1}$$

Базовая структура такого вычислителя на модулярной логарифметической основе представлена на рис. 2.

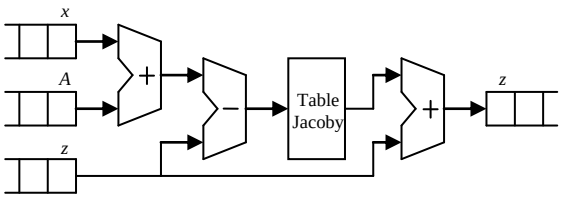

Рис. 2. Базовая структура модульного канала вычислителя перемножения матриц с использованием операции GAXPY

IV. Оценки быстродействия и аппаратных затрат

На базе рассмотренных алгоритмов, были разработаны компьютерные модели типовых устройств, выполняющих рассматриваемый ряд операций матричной алгебры. На языке Perl созданы автоматизированные генераторы функциональных представлений для создания высокоуровневого Verilog описания блоков вычисления операций матричной алгебры в двоичной (BNS), модулярной (RNS) и модулярной логарифметической (RLNS) арифметиках. Из расчета размеров матриц и входной разрядности данных, программа рассчитывает необходимый динамический диапазон для вычислений, что гарантирует сохранность результата операции в заданном диапазоне; проводит выбор наиболее технологичных модулей базового диапазона [13], исходя из критерия равнобитности используемых модулей системы. Результатом работы генераторов является набор файлов формата Verilog (.v) для

дальнейшей загрузки в систему автоматизации проектирования. Структурный синтез проводился средствами САПР Synopsys Synplify в базисе ПЛИС Altera Stratix II EP2S15F484C3. Симуляция и верификация Verilog проектов проводилась средствами ModelSim Mentor Graphics. Быстродействие схемы определяется тактовой частотой (MHz), сложность реализации измеряется числом адаптивных логических блоков табличного типа (ALUT, Adaptive Look Up Table). Вычислительные эксперименты проводились на входных данных, разрядностью 16 бит и 24 бита, размерности матрицы n=10. Результаты вычислительных экспериментов представлены в табл. 1, 2.

Таблица 1

Результаты синтеза для 16-битных данных

	dot		saxpy		gaxpy	
	MHz	ALUT	MHz	ALUT	MHz	ALUT
BNS	168	387	290	33	287	83
RNS	375	314	385	68	373	109
RLNS	437	325	452	75	450	127

Таблица 2

Результаты синтеза для 24-битных данных

	dot		saxpy		gaxpy	
	MHz	ALUT	MHz	ALUT	MHz	ALUT
BNS	157	426	282	47	280	97
RNS	335	394	365	83	353	134
RLNS	413	415	433	105	437	142

Полученные результаты показывают, что использование логарифметики повышает эффективность реализации операции матричной алгебры в модульных вычислительных системах. Усложнение реализации операции логсложения оказывает не столь значимое влияние по сравнению с упрощением операции логумножения, из чего следует, что данный подход будет эффективен при реализации алгоритмов модулярной арифметики алгебры матриц.

V. Заключение

Аппарат модулярной логарифметики может быть эффективно использован при построении специализированных конвейерных вычислителей для задач матричной алгебры с целью повышения производительности устройства в целом. Интегральное исполнение устройств, с применением аппарата модулярной логарифметики, позволяет гибко подходить к реализации основных модулярных вычислительных процедур, применяя специализированные методы уменьшения площади или увеличения быстродействия в зависимости от требований, предъявляемых к ним.

Преимущества и особенности модульной логарифметики позволяют ставить новые задачи оптимального структурного проектирования специализированных вычислительных устройств алгебры матриц.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант №12-07-00199-а).

Литература

[1] Червяков Н.И. Методы и принципы построения модулярных нейрокомпьютеров. Сайт http://www.computer-museum.ru/, 2005.

[2] Амербаев В.М., Корнилов А.И., Стемпковский А.Л. Модулярная логарифметика – новые возможности для проектирования модулярных вычислителей и преобразователей // IV Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем – 2010»: сб. научн. тр. / под общ. ред. А.Л. Стемпковского. М.: ИППМ РАН, 2010. С. 368 – 373.

[3] Preethy A.P. and Radhakrishnan D. A 36-bit Balanced Moduli MAC Architecture // 42nd Midwest Symp. on Circuits and Systems (MWSCAS99). Las Cruces, NM. Aug. 1999. Vol. 1. P. 380 – 383.

[4] Поспелов Д.А. Арифметические основы вычислительных машин дискретного действия. М.: Высш. шк., 1970.

[5] Лидл Р., Нидеррайтер Г. Конечные поля: в 2 т. / под общ.ред. В.И. Нечаева. М.: Мир, 1988.

[6] Математический Энциклопедический словарь. М.: Сов. Энциклопедия, 1988. С. 141, 330.

[7] Arnold M.G. The residue logarithmic number system: theory and implementation // 17th IEEE Symp. on Computer Arithmetic, ARITH-17 2005. P. 196 – 205.

[8] Амербаев В.М., Балака Е.С., Константинов А.В., Тельпухов Д.В. Методы ускорения вычислений скалярных произведений векторов в базисе модулярной логарифметики // IV Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем – 2010»: сб. научн. тр. / под общ. ред. А.Л. Стемпковского. М.: ИППМ РАН, 2010. С. 378 – 381.

[9] Preethy A.P., Padhakrishnan D. An RNS based logarithmic adder // JEEE Proceeding, Computer and Digital Techniques. July, 2000. V. 147. Issue 4. P. 283 - 296.

[10] Амербаев В.М., Балака Е.С. Анализ и синтез алгоритмов вычисления гауссовых логарифмов большого числа слагаемых над полем Галуа GF(p) // Изв. ВУЗов. Электроника. 2010. №4. С. 64 – 69.

[11] Балака Е.С., Тельпухов Д.В. Принципы построения специализированного вычислителя для задач матричной алгебры с применением параллельной арифметики // Нейрокомпьютеры: разработка и применение. 2010. №9. С. 46 – 49.

[12] Амербаев В.М., Малашевич Д.Б. Анализ эффективности реализации модульных операций индексной модулярной арифметики // Известия вузов. Электроника. 2009. № 6(80).