УДК 004.272.2

Методы ускорения вычислений скалярных произведений векторов в базисе модулярной логарифметики

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

Учреждение Российской академии наук Институт проблем проектирования в микроэлектронике РАН, Ssapra@hotbox.ru

***Аннотация* — В статье приведены основы модулярной логарифметики над полем *GF(p)*. Представлены методы вычисления скалярных произведений векторов в базисе модулярной логарифметики. Синтез методов в базисе ПЛИС показал существенный прирост производительности. За счет значительного упрощения операции умножения логарифметика может успешно использоваться для повышения эффективности реализации арифметических операций в модульных вычислительных системах.**

***Ключевые слова* — модулярная арифметика, дискретный логарифм, логарифметика, логарифм Якоби.**

I. Введение

Проблема повышения быстродействия в системах, функционирующих в реальном времени и в больших динамических диапазонах, может быть решена не только за счет совершенствования технологии, но и за счет распараллеливания вычислительных операций, что влечет за собой разработку методов распараллеливания, как на уровне алгоритмов, так и на уровне машинных кодов. Межразрядные связи позиционной арифметики делают практически невозможным удовлетворения таких требований на уровне машинных кодов. Модулярная арифметика позволяет устранить проблему потери скорости, которую несут арифметические вычисления, что делает ее привлекательной в большинстве задач интенсивных вычислений. Операция вычисления скалярного произведения векторов является неотъемлемой частью арифметического процессора, используемого при решении такого рода задач. 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. Однако, применение модулярной арифметики при построении высокопроизводительных систем, к которым относятся устройства цифровой обработки сигналов, не является широко используемым методом. Несмотря на то, что имеются реальные коммерческие продукты, с применением аппарата модулярной арифметики, эта область является малоизученной. Развитие интегральной схемотехники и использование современных САПР (таких как Synopsys, Cadence) приводит к возможности использования новых методов проектирования как отдельных узлов, реализующих вычислительные операции, так и устройств в целом. Таким образом, существуют способы более эффективного, с точки аппаратных и временных затрат, построения основных блоков, входящих в состав таких устройств с учетом их интегрального исполнения. Что, в свою очередь, приводит к выигрышу по площади и быстродействию для всей системы в целом. Этому вопросу посвящено много работ и защищено немало диссертаций. В настоящее время актуальным вопросом является всестороннее изучение особенностей реализации базовых арифметических операций модулярной арифметики посредством соответствующих операций логарифметики в связи с существующим прогрессом вычислительной техники и необходимостью поиска новых и эффективных вычислительных технологий.

II. Обзор модулярной логарифметики

Рассмотрим дискретный логарифм над полем *GF(p)*.

Определение 1.

Пусть w - образующий элемент поля *GF(p)*. Дискретным логарифмом по основанию w над *GF(p)* будем называть функцию аргумента x $(x \in Z)$, заданную формулой:

$$\lg_w |x|_p = \lambda_p \delta(|x|_p) + ind_w |x|_p \hat{\delta}(|x|_p), \qquad (1)$$

где $|x|_p$ - вычет числа x *mod(p)*; $\delta(|x|_p)$ - функция Дирака; $\hat{\delta}(|x|_p)$ - кофункция Дирака, т.е. $\hat{\delta}(|x|_p) = 1 - \delta(|x|_p)$; $ind_w |x|_p$ - индекс вычета $|x|_p$; λ_p - символ, не являющийся элементом кольца Z_{p-1}.

Положим $\lambda_p = 2^t - 1$. Технологичность такого выбора обусловлена следующим: если p - t-битное чис-

ло, т.е. $2^{t-1} < p < 2^t$, то в роли символа λ_p целесообразно использовать *t*-битную двоичную запись числа $2^t - 1$; поскольку *p* - простое, то справедливо неравенство $p \le 2^t - 1$. Тем самым, число $2^t - 1$ не является символом, изображающим элемент кольца Z_{p-1} для любого простого p ≥ 3. При таком выборе λ_p все значения функции $y = \lg_w |x|_p$ различны и описываются t-битным двоичным кодом.

Областью значений дискретного логарифма (1) является множество $J_p = \{0,1,2,\ldots,\ p-2,\ \lambda_p\}$. Характерными точками из *GF(p)* отображения $\lg_w : GF(p) \to J_p$ при любом *p* и любом выборе *w* являются точки 0, 1, *w*, *p-1*; они, соответственно, отображаются в точки множества J_p: λ_p, 0,1, $\frac{p-1}{2}$.

По построению логарифмическая функция биективно отображает конечное поле *GF(p)* на J_p. Тем самым на J_p генерируется структура конечного поля, изоморфная структуре поля *GF(p)*. Это утверждение служит основой для формирования логарифметики над полем *GF(p)*.

Рассмотрим подробнее соответствующие покомпонентные операции $\lg_w |x_1 \cdot x_2|_p$; $\lg_w |x_1 + x_2|_p$.

Обозначим операцию умножения в логарифметике символом $\boxtimes$, т.е. если $y_1 = \lg_w |x_1|_p$, $y_2 = \lg_w |x_2|_p$, то $y_1 \boxtimes y_2 := \lg_w |x_1 \cdot x_2|_p$.

$$y_1 \boxtimes y_2 = \begin{cases} \lambda_p, \text{если } \left(\delta(y_1 - \lambda_p) \vee \delta(y_2 - \lambda_p)\right) = 1, \\ |y_1 + y_2|_{p-1} \end{cases}$$

Обозначим операцию сложения в логарифметике символом $\boxplus$, т.е. $y_1 \boxplus y_2 := \lg_w |x_1 + x_2|_p$

$$y_1 \boxplus y_2 = \begin{cases} \lambda_p, \text{если } \left(\delta(y_1 - \lambda_p) \wedge \delta(y_2 - \lambda_p)\right) \vee \\ \quad \left(\hat{\delta}(y_1 - \lambda_p) \wedge \hat{\delta}(y_2 - \lambda_p) \wedge \right. \\ \quad \left. \delta\left(|y_2 - y_1|_p - \frac{p-1}{2}\right)\right) = 1, \\ y_1, \text{если } \hat{\delta}(y_1 - \lambda_p) \wedge \delta(y_2 - \lambda_p) = 1, \\ y_2, \text{если } \delta(y_1 - \lambda_p) \wedge \hat{\delta}(y_2 - \lambda_p) = 1, \\ |y_1 + J_w(|y_2 - y_1|_{p-1})|_{p-1} \end{cases}$$

где $J_w(|u|_{p-1}) = \lg_w \left|1 + w^{|u|_{p-1}}\right|_p$ - так называемый логарифм Якоби [7].

Таким образом, вычислительный базис логарифметики по простому модулю *p* составляют три независимых функциональных блока:

- предикаторы сингулярности

$$P(y_1 \boxtimes y_2) = \lambda_p \left(\delta(y_1 - \lambda_p) \vee \delta(y_2 - \lambda_p)\right),$$

$$P(y_1 \boxplus y_2) = y_1 \left(\hat{\delta}(y_1 - \lambda_p) \wedge \delta(y_2 - \lambda_p)\right) +$$
$$+ y_2 \left(\delta(y_1 - \lambda_p) \wedge \hat{\delta}(y_2 - \lambda_p)\right) +$$
$$+ \lambda_p \left(\delta(y_1 - \lambda_p)\right) \wedge \delta(y_2 - \lambda_p) \vee$$
$$\vee \delta\left(|y_2 - y_1|_p - \frac{p-1}{2}\right);$$

- сумматор $\sum_{p-1}$ по *mod(p – 1)*;
- логарифм Якоби $J_w(|u|_{p-1})$

На рис. 1 представлена обобщенная структурная схема вычислительного элемента (ВЭ), реализующая данные операции.

Предикаторы сингулярности проверяют операнды и промежуточные результаты на сингулярность логарифма и в зависимости от результата проверки управляют выходными мультиплексорами для выбора значения результата.

Анализ схемы показывает, что эффективность операции логумножения сравнима с эффективностью реализации модульной операции сложения по *mod(p – 1)*, в то время как эффективность операции логсложения понижается за счет применения двух сумматоров по *mod*(*p* – 1) и таблицы Якоби.

Отметим, что логарифметика конечного поля может быть распространена на кольцо вычетов по составному модулю, опираясь на Китайскую теорему об остатках [5]: логарифметикой кольца вычетов по *mod N*, где $N = p_1 p_2 \ldots p_n$ (p_i - простые), называется арифметика прямого произведения полей $LZ_{p_1} \times LZ_{p_2} \times \ldots \times LZ_{p_n}$, которая изоморфна арифметике кольца $Z_{p_1} \times Z_{p_2} \times \ldots \times Z_{p_n}$. В роли модулей p_1, $p_2,\ldots,p_n$ целесообразно выбирать технологичные модули [6]. В работе [7] приведены сравнительные оценки характеристик функциональных блоков модулярной логарифметики конечного поля *GF*(*p*).

III. Постановка задачи и предлагаемые методы ее решения

Операция вычисления скалярного произведения векторов над полем *GF(p)*:

$$y = \left|\sum_{i=0}^{N-1} a_i x_i\right|_p, \text{ где } p \text{ - простое.}$$

В базисе логарифметики задача нахождения скалярного произведения векторов сводится к задаче нахождения логарифма Гаусса от *N* переменных.

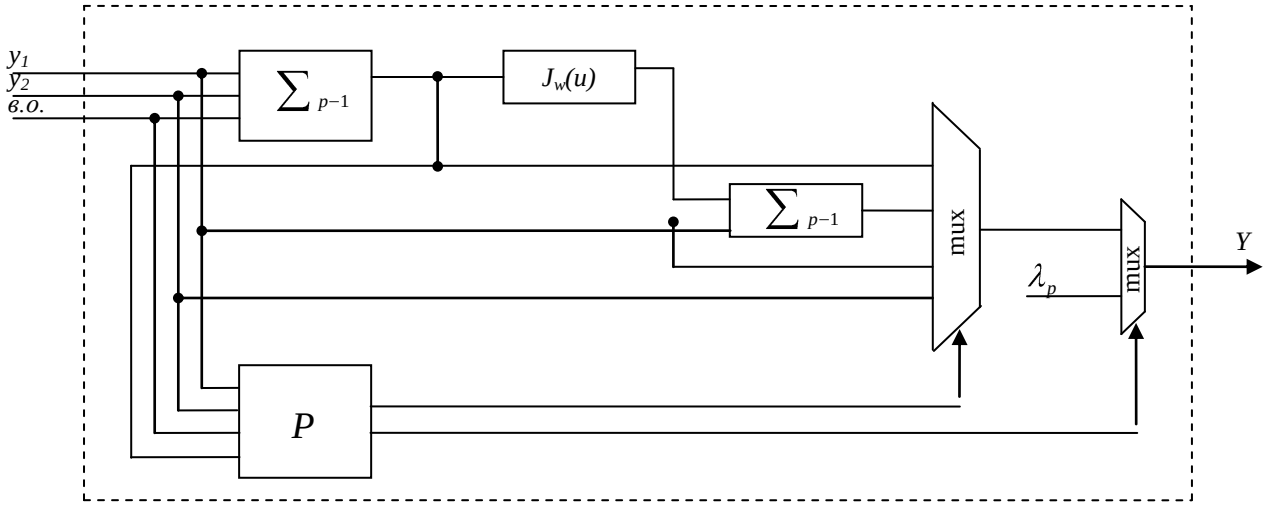

Рис. 1. Обобщенная структура ВЭ по модулю p модулярной логарифметики

Обозначим ее как:

$$G(z_1, z_2, ..., z_N) = \lg_w | \sum_{i=1}^{N} w^{|z_i|_{p-1}} |_p , \qquad (2)$$

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

Традиционным решением вычисления гауссова логарифма от большого числа слагаемых является метод восстановления (потенцирования) каждого слагаемого, их суммирования и последующего логарифмирования результирующей суммы. Типовая структура такого вычислителя представлена на рис. 2:

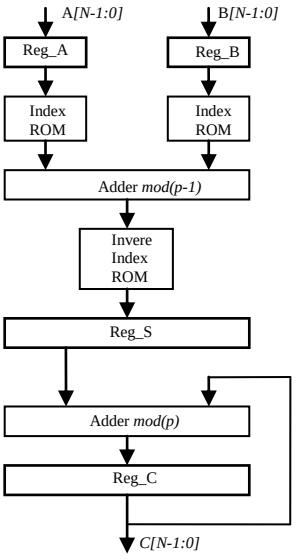

Рис. 2. Типовая структура блока вычисления логарифма Гаусса в модулярной арифметике

Для вычисления скалярного произведения векторов в логарифметике конечного поля *GF(p)*, требуется разработать методы вычисления логарифма Гаусса без промежуточного потенцирования операндов.

Представляя подлогарифмическое выражение уравнения (2) разными способами, можно получить различные варианты решения нахождения логарифма Гаусса.

А. Метод, использующий ассоциативный подход.

Подлогарифмическое выражение уравнения (2) представим в виде:

$$\left| \sum_{i=0}^{N-1} w^{|z_i|_{p-1}} \right|_p = \left| \left| \left| w^{|z_0|_{p-1}} + w^{|z_1|_{p-1}} \right|_p + w^{|z_2|_{p-1}} \right|_p + \ldots \right|_p + w^{|z_{N-1}|_{p-1}} \Bigg|_p$$

Как было показано выше, сложения двух чисел в логарифметике выполняется по правилу:

$$|x + y|_p = \left| w^{\log_w |x|_p} (1 + w^{|\lg_w |y|_p - \lg_w |x|_p|_{p-1}}) \right| =$$

$$= \left| w^{\log_w |x|_p + J_w(|\lg_w |y|_p - \lg_w |x|_p|_{p-1})} \right|_p .$$

Непосредственное вычисление значений логарифма Якоби имеет сложность дискретного логарифма. Табличная реализация логарифма Якоби, в силу технических ограничений на допустимый объем используемых таблиц, ограничена сверху количеством разрядов базисного модуля *p* поля Галуа *GF(p)*. О том, как сократить таблицу Якоби, написано в работе [4].

Тем самым, структурная схема алгоритма вычисления логарифма Гаусса большой арности, использующая метод ассоциативности (сокращенно СПВа1), представлена на рис. 3.

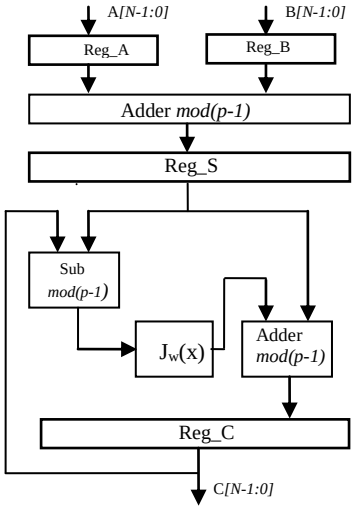

Рис. 3. Структурная схема одного модульного канала блока СПВа1

Б. Метод, основанный на обобщенном преобразовании Горнера.

Подлогарифмическое выражение уравнения (2) примет вид:

$$\left|\sum_{i=1}^{N} w^{|Z_i|_{p-1}}\right|_p = \left|w^{|z_1|_{p-1}}\left|1+w^{|z_{21}|_{p-1}}\right|\ldots\left|1+w^{|z_{N,N-1}|_{p-1}}\right|_p\right|_p\ldots\Big|_p$$

где $|Z_{i,i-1}|_{p-1} = |z_i - z_{i-1}|_{p-1} = |z_i + (p - z_{i-1})|_{p-1}$, $i \geq 2$.

Такой подход позволяет получить конвейерную структуру (сокращенно СПВа2) с предварительной обработкой данных. Подробнее о этом методе написано в [4]. Структурная схема данного алгоритма представлена на рис. 4.

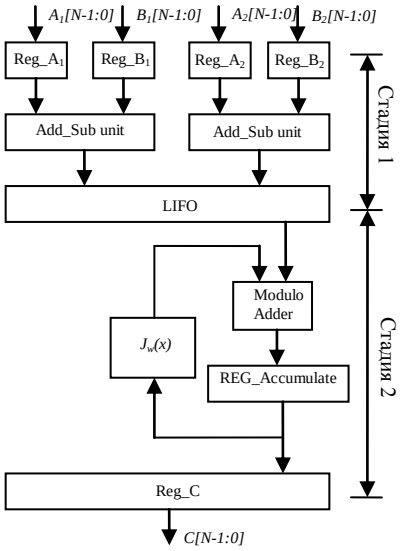

Рис. 4. Структурная схема одного модульного канала блока СПВа2

IV. Результаты синтеза

Структурный синтез проводился средствами САПР Synopsys Synplify в базисе ПЛИС Altera Stratix II EP2S15F484C3. Симуляция и верификация VHDL проектов проводились средствами ModelSim Mentor Graphics. Быстродействие схемы определяется тактовой частотой, сложность реализации измеряется числом адаптивных логических блоков табличного типа (ALUT, Adaptive Look Up Table).

Таблица 1

Характеристики устройств вычисления скалярных произведений векторов

	Clock, Mhz	ALUT
Позиционный вариант	168	387
Модулярный вариант	376	254
СПВа1	394	323
СПВа2	333	555

V. Заключение

Результаты синтеза предложенных алгоритмов показывают, что счет значительного упрощения операции умножения логарифметика может успешно использоваться для повышения эффективности реализации арифметических операций в модульных вычислительных системах.

Литература

[1] M. G. Arnold, Method and Apparatus for Fast Logarithmic Addition and Subtraction, U. S. Patent 5337266, Aug. 9. 1994.

[2] A.P. Preethy, D. Radhakrishnan, RNS-based logarithmic adder, IEE Proc. Computer and Digital Techniques. 2000. v. 147. no. 4. p. 283-287.

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

[4] В.М. Амербаев, Е.С. Балака Анализ и синтез алгоритмов вычисления гауссовых логарифмов N слагаемых над полем Галуа GF(p) // Электроника. Известия вузов. - 2009. (в печати).

[5] И.М. Виноградов Основы теории чисел. – Изд. 6-е. – М.: Наука, 1972. – 167 с.

[6] В.М. Амербаев, Д.В. Тельпухов, А.В. Константинов Бивалентный дефект модулярных кодов. Выбор технологических модулей, понижающих бивалентный дефект // Проблемы разработки перспективных микро- и наноэлектронных систем – 2008. Сб. научных трудов / под общ. ред. А.Л. Стемпковского. – М.: ИППМ РАН, 2008. – С. 462–465.

[7] Амербаев В.М., Малашевич Д.Б. Анализ эффективности реализации модульных операций индексной модулярной арифметики // Электроника. Известия вузов. - 2009. (в печати).