УДК 004.75,004.272

АНАЛИЗ ПОДХОДОВ К РЕАЛИЗАЦИИ НА FPGA ОПЕРАЦИЙ УМНОЖЕНИЯ В ПОЛЕ ГАЛУА

Е.В. ЛИСТОПАД

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 5 октября 2017

Аннотация. Рассмотрены варианты аппаратных реализаций операций умножения в поле Галуа для эффективного решения задач построения специализированных процессоров на базе FPGA, отвечающих строгим требованиям по быстродействию и номинальному значению рабочей тактовой частоты. Предложены подходы к аппаратной реализации таких операций, демонстрирующие различные качественные показатели.

Ключевые слова: умножение в поле Галуа, специализированный процессор, FPGA.

Abstract. The variants of hardware implementations of multiplication operations in the Galois field are considered for efficient solution of the tasks of constructing specialized processors based on FPGA, meeting strict requirements for speed and nominal value of the working clock frequency. Approaches to the hardware implementation of such operations demonstrating various qualitative indicators are proposed.

Keywords: multiplication in the Galois field, specialized processor, FPGA.

Doklady BGUIR. 2018, Vol. 113, No. 3, pp. 5-12
Analysis of possibilities of FPGA-implementation for multiplication operations in the Galois field
E.V. Listopad

Введение

Для решения задачи прототипирования таких классов цифровых устройств, как «встраиваемая система» [1], «система на кристалле» [2], широкое распространение получили IP-ядра – готовые блоки, применяемые для проектирования микросхем и представленные на уровне абстрактного описания [3]. При построении IP-ядер, решающих задачи цифровой обработки сигналов, зачастую возникает необходимость эффективной реализации арифметических операций над элементами поля Галуа. Особый интерес представляет операция умножения в поле, как наиболее требовательная к аппаратным ресурсам FPGA и лежащая в основе более сложных операций в поле. Достоинством любых вычислений в поле Галуа является то, что они допускают параллельную реализацию [8]. Это позволяет рассматривать их как адекватные архитектуре ПЛИС типа FPGA.

Известны различные архитектуры и методы построения универсальных аппаратных умножителей элементов поля, в том числе на базе ПЛИС типа FPGA [4–6]. В данной работе показано, как, применяя известные методы и используя различные варианты аппаратных реализаций операции умножения в поле Галуа, эффективно решать задачи построения цифровых устройств на архитектуре ПЛИС/FPGA.

Исходные данные

Поля Галуа описываются двумя основными параметрами: m и p [7]. Параметр m указывает число двоичных разрядов, использующихся для двоичного представления символа

множества, а также определяет количество элементов множества как 2^m. Таким образом, в поле $GF(2^4)$, где $m=4$, содержится всего 16 элементов, и для двоичного представления каждого из них необходимо четыре двоичных разряда. Параметр *p* (генерирующий полином) указывает порядок, в котором элементы поля Галуа следуют друг за другом. Например, генерирующий полином $p(x)$ для поля $GF(2^4)$ может быть следующим: $p(x)=1+x^3+x^4$. Часто используется представление полинома в виде двоичного числа с разрядностью $m+1$. В данном случае $p=25$ в десятичной системе или 11001 в двоичной, или $1\cdot 2^4+1\cdot 2^3+0\cdot 2^2+0\cdot 2^1+1\cdot 2^0$. Если корень полинома обозначить через *a*, то $a^4=a^3+1$.

Элементы поля $GF(2^4)$ приведены в табл. 1 в трех представлениях:

1) степенное представление, в котором нулевой элемент равен 0, первый равен 1, второй равен *a* и т. д.;

2) полиномиальное представление (в виде многочлена): $x=k_0\cdot 1+k_1\cdot a+k_2\cdot a^2+k_3\cdot a^3$, где $k_0,k_1,k_2,k_3=\{0,1\}$ (старшие разряды справа);

3) бинарное представление или двоичная форма (старшие разряды справа).

Таблица 1. **Представление поля Галуа для *m* = 4 и *p* = 25**

Степенное представление	Полиномиальное представление	Бинарное представление
0	0	0000
1	1	1000
a	a	0100
a^2	a^2	0010
a^3	a^3	0001
a^4	$1+a^3$	1001
a^5	$1+a+a^3$	1101
a^6	$1+a+a^2+a^3$	1111
a^7	$1+a+a^2$	1110
a^8	$a+a^2+a^3$	0111
a^9	$1+a^2$	1010
a^{10}	$a+a^3$	0101
a^{11}	$1+a^2+a^3$	1011
a^{12}	$1+a$	1100
a^{13}	$a+a^2$	0110
a^{14}	a^2+a^3	0011

При реализации аппаратных умножителей часто применяется полиномиальное представление элементов поля [5, 6]. Операция умножения элементов поля Галуа выполняется как умножение двух определенных многочленов по модулю третьего многочлена (по которому построены элементы поля).

Реализация аппаратного умножения за 16 шагов

Рассмотрим поле с параметрами $m=16$ и $p=126977$, в котором опишем особенности аппаратной реализации операций умножения. Как видно из параметров поля, операнды

для произведения являются 16-битными. Первый подход предусматривает умножение за 16 шагов. При этом на каждом шаге выполняется умножение на 1 бит операнда и осуществляется приведение по модулю полинома. На рис. 1 приведена универсальная структура IP-ядра.

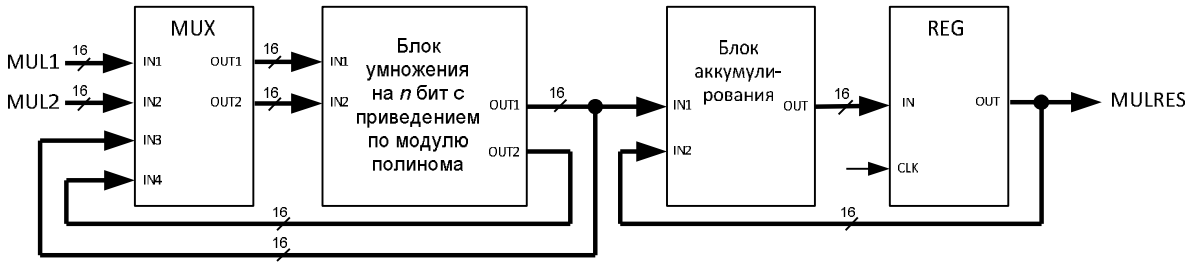

Рис. 1. Универсальная структура IP-ядра, выполняющего умножение за 16 шагов

Были разработаны 3 экспериментальные реализации IP-ядер для данного подхода. Реализация 1 выполняет умножение за 16 тактов, при этом за 1 такт выполняется 1 шаг умножения с приведением (на рис. 1 параметр *n* принять равным 1). Структурная схема IP-ядра для реализации 1 показана на рис. 2.

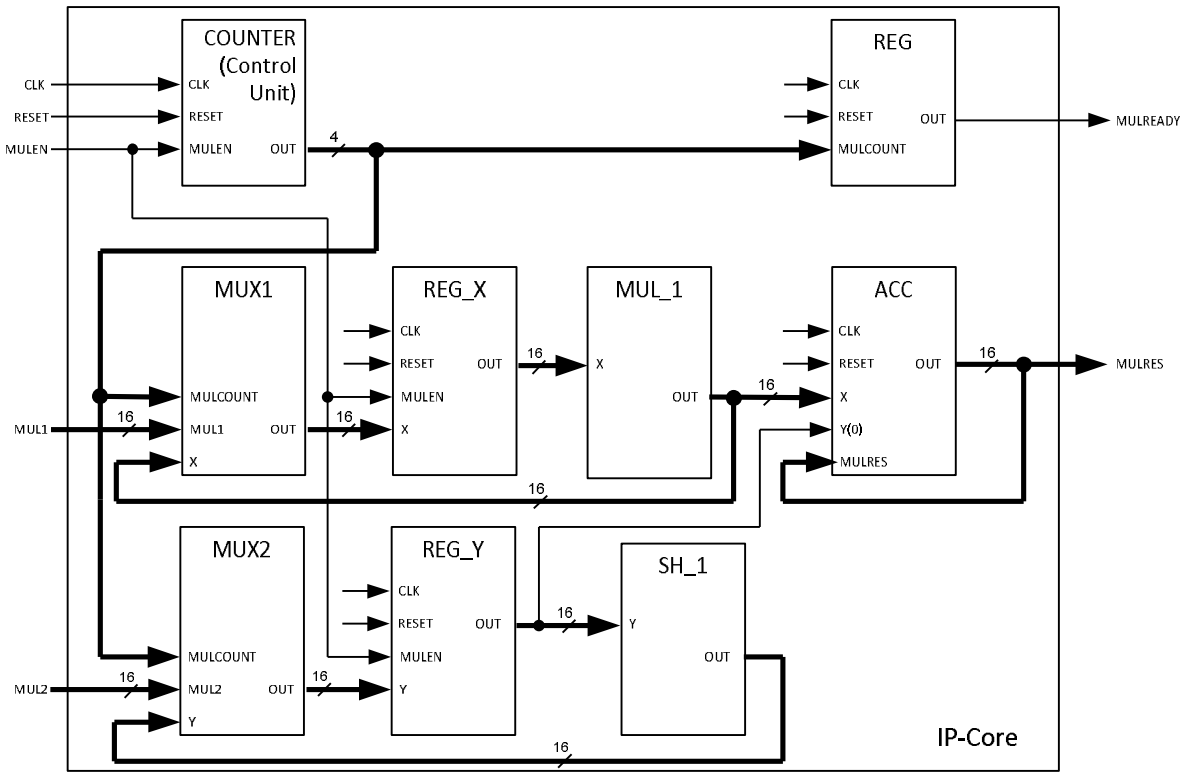

Рис. 2. Структура IP-ядра, выполняющего умножение за 16 тактов

В приведенной реализации в качестве устройства управления выступает четырехразрядный счетчик (*COUNTER*), по сигналам которого выполняется мультиплексирование входных и внутренних значений множителей *MUL1* и *MUL2*, начальная инициализация регистров *X* и *Y*, а также формирование сигнала *MULREADY*, свидетельствующего о готовности результата. Сам процесс умножения выполняется следующим образом.

Вначале происходит загрузка входных значений *MUL1* и *MUL2* в регистры *X* и *Y* соответственно. Далее в блоке *MUL_1* выполняется умножение значения регистра *X* на очередной бит (арифметический сдвиг) за такт с одновременным приведением по модулю полинома (наложение маски). Результат этой операции суммируется с накоплением в блоке *ACC*, если соответствующий бит регистра *Y* равен 1, либо игнорируется, если указанный бит равен 0. Далее выполняется арифметический сдвиг значения в регистре *Y*, чтобы следующий

бит, подлежащий анализу, стал последним в регистре. После этого вычислительный цикл повторяется. Результат вычислительного процесса накапливается в регистре *MULRES* и является валидным в момент активации сигнала *MULREADY*.

Данное IP-ядро было синтезировано средствами Xilinx ISE на FPGA Spartan 6 (XC6SL75). В результате чего было задействовано 17 слайсов на кристалле и достигнута тактовая частота 390 МГц. Производительность данной реализации составила 371,9 Мбит/с.

Реализация 2 выполняет умножение за 8 тактов, при этом за 1 такт выполняется 2 шага умножения с приведением и анализируется 2 бита операнда (на рис. 1 параметр *n* принять равным 2). IP-ядро реализации 2 также было синтезировано средствами Xilinx ISE на FPGA Spartan 6 (XC6SL75). В результате чего было задействовано 24 слайса на кристалле и достигнута тактовая частота 291 МГц. Производительность данной реализации составила 555,0 Мбит/с. Несмотря на меньшую тактовую частоту, производительность реализации оказалась выше за счет уменьшения требуемых тактов процессорного времени с 16 до 8. Количество требуемых аппаратных ресурсов кристалла возросло до 24 слайсов в связи с использованием в реализации дополнительного блока умножения и более сложной структуры блока накопления *ACC*.

Реализация 3 выполняет умножение за 4 такта, при этом за 1 такт выполняется 4 шага умножения с приведением и анализируется 4 бита операнда (на рис. 1 параметр *n* принять равным 4). В результате синтеза IP-ядра реализации 3 было задействовано 55 слайсов на кристалле и достигнута тактовая частота 204 МГц. Производительность данной реализации составила 778,2 Мбит/с. Несмотря на меньшую тактовую частоту, производительность реализации оказалась выше чем у предыдущих за счет уменьшения требуемых тактов процессорного времени до 4. Количество требуемых аппаратных ресурсов кристалла возросло до 55 слайсов в связи с использованием в реализации дополнительных блоков умножения и более сложной структуры блока накопления *ACC*.

Реализация аппаратного умножения за 2 шага

Если главной особенностью первого подхода являлась реализация операции приведения по модулю полинома после каждой операции умножения на очередной бит, то во втором подходе операция приведения по модулю полинома выполняется после всех операций умножения на бит (т. е. после 16 операций). Таким образом, второй подход предусматривает умножение за 2 шага (рис. 3): непосредственно умножение 16-битных операндов с получением 32-битного промежуточного результата и приведение его по модулю полинома к 16-битному результату.

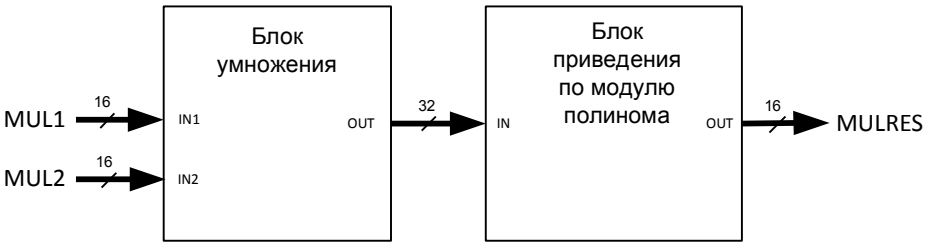

Рис. 3. Универсальная структура IP-ядра, выполняющего умножение за 2 шага

Реализация 4 выполняет полное умножение за 1 такт, при этом под полным умножением будем понимать умножение арифметическое с приведением по модулю полинома. Структурная схема IP-ядра для реализации 4 показана на рис. 4. В приведенной реализации вычисления осуществляются двумя асинхронными блоками (умножения и приведения по модулю полинома). Известно, что при 16-разрядных операндах арифметическое умножение является 32-разрядным. При этом для обеспечения данной разрядности в блоке умножения присутствуют 2 блока, идентичные по структуре: один для вычисления младших 16 бит умножения, другой – для старших. Каждый из таких блоков представляет собой 16 логических функций, каждая из которых вычисляет один бит произведения. Следует отметить, что указанные логические функции выполняют элементарные логические операции (AND, XOR), однако имеют различное число входов (от 2 до 32).

Блок приведения по модулю полинома определенным образом модифицирует младшую часть полученного произведения с учетом старшей его части. После модификации в данном блоке результат умножения поступает на выходной регистр, обеспечивающий фиксацию результата и синхронную работу всего IP-ядра реализации 4.

В результате синтеза данного IP-ядра было задействовано 50 слайсов на кристалле и достигнута тактовая частота 184 МГц. Производительность данной реализации составила 2807,6 Мбит/с. Несмотря на меньшую тактовую частоту, производительность реализации оказалась выше чем у предыдущих за счет выполнения всех вычислений за 1 такт процессорного времени.

Реализация 5 выполняет полное умножение за 2 такта. При этом на первом такте выполняется арифметическое 32-разрядное умножение, на втором – приведение по модулю полинома. Структурная схема IP-ядра для реализации 5 показана на рис. 5.

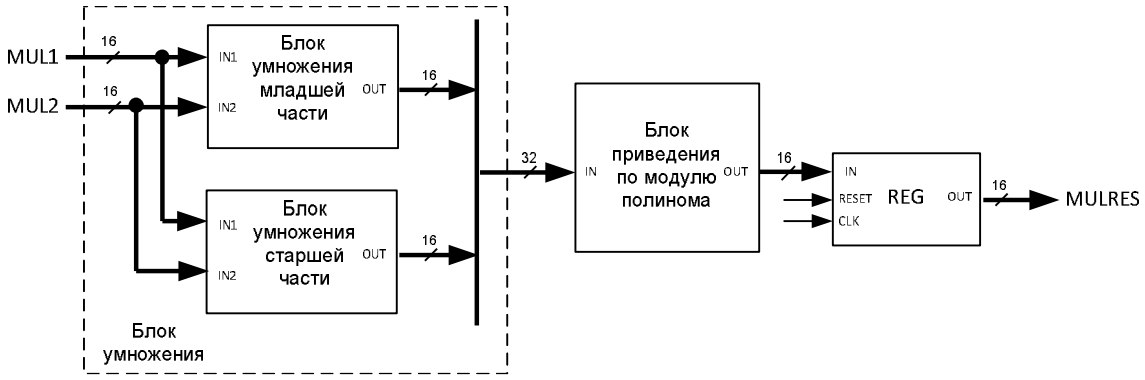

Рис. 4. Структура IP-ядра, выполняющего умножение за 1 такт

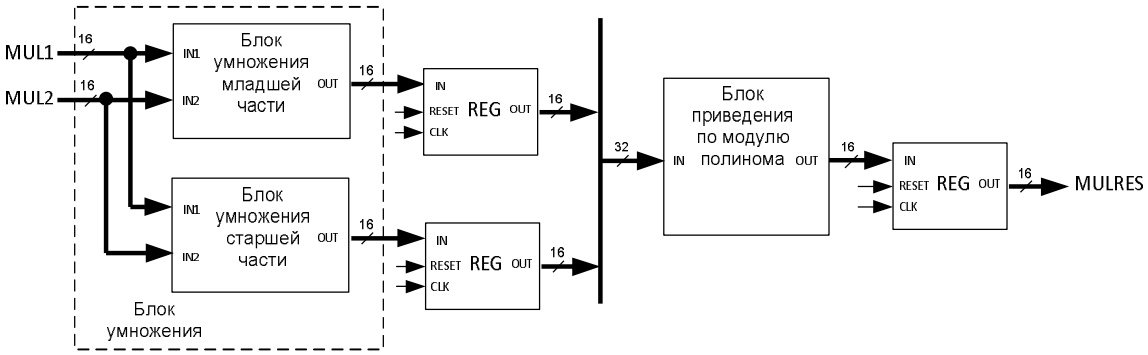

Рис. 5. Структура IP-ядра, выполняющего умножение за 2 такта

В приведенной реализации вычисления осуществляются теми же блоками, что и в реализации 4. Структура IP-ядра данной реализации предусматривает наличие дополнительных регистров между основными вычислительными блоками. Такая структура обеспечивает получение произведения в поле Галуа за 2 такта процессорного времени и снижение сложности логических структур в данной схеме за счет сокращения критического пути схемы при ее разделении регистрами.

В результате синтеза IP-ядра реализации 5 было задействовано 46 слайсов на кристалле и достигнута тактовая частота 221 МГц. Производительность данной реализации составила 1686,1 Мбит/с. Несмотря на большую тактовую частоту, производительность реализации оказалась ниже чем у предыдущей за счет выполнения вычислений за 2 такта процессорного времени.

Реализация аппаратного умножения с оптимизацией под кристалл

В реализации 6 была произведена попытка разделить асинхронную вычислительную часть, состоящую из блоков умножения и приведения по модулю полинома, дополнительными регистрами более эффективно, чем это было сделано в реализации 5. Усматривается возможность оптимизировать структуру блоков умножения таким образом, чтобы разрядность логических функций не превышала 6 и соответствовала структуре слайсов, имеющихся

на базовом кристалле FPGA. Такая возможность заключается в установке дополнительных регистров внутри блоков умножения после первого уровня логики в каждом из блоков логических функций. На рис. 6 приведена структурная схема реализации 6.

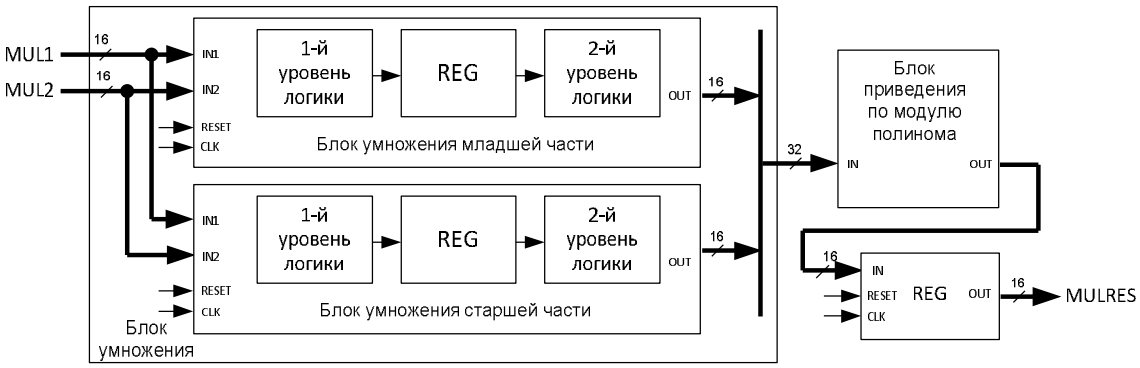

Рис. 6. Структура усовершенствованного IP-ядра, выполняющего умножение за 2 такта

Таким образом, было выполнено усовершенствование предыдущей реализации, учитывающее природу строения кристалла ПЛИС (6-входовые элементы LUT кристалла Xilinx Spartan 6). В реализации была применена двухступенчатая схема умножения. На первой ступени логические операции были описаны в виде 6-входовых логических элементов, которым при синтезе были поставлены в соответствие 6-входовые элементы LUT на кристалле. На второй ступени описаны логические операции над результатами первой ступени вычислений и реализовано приведение по модулю полинома. В результате синтеза IP-ядра реализации 6 было задействовано 48 слайсов на кристалле и достигнута тактовая частота 279 МГц. Производительность данной реализации составила 2128,6 Мбит/с. Следует отметить, что данная реализация обеспечила более высокую тактовую частоту и производительность, чем реализация 5, что свидетельствует об эффективности предложенного решения. При этом увеличение данных показателей практически не повлияло на количественные характеристики используемых ресурсов кристалла (слайсов).

Результаты исследований

В ходе исследований были выполнены аппаратные реализации операций умножения в поле Галуа с параметрами $m = 16$ и $p = 126977$ с применением одного из двух подходов. Первый подход предусматривает реализацию операции приведения по модулю полинома после каждой операции умножения на очередной бит. С применением данного подхода были выполнены 3 реализации, характеристики которых приведены в табл. 2.

Таблица 2. **Характеристики реализаций, построенных с применением умножения за 16 шагов**

Вариант	Количество тактов	Ресурсы FPGA, Slices	Частота, МГц	Производительность, Мбит/с
Реализация 1	16	17	390	371,9
Реализация 2	8	24	291	555,0
Реализация 3	4	55	204	778,2

Ключевой особенностью данных реализаций является использование 16-разрядной арифметики, так как приведение по модулю полинома осуществляется после каждой элементарной операции умножения, предотвращая тем самым потенциальную возможность переполнения или выхода за пределы арифметики. Благодаря данной особенности удалось обеспечить достаточно высокие тактовые частоты данных реализаций и их относительно низкую требовательность к ресурсам кристалла. В то же время реализации демонстрируют относительно невысокую производительность ввиду выполнения вычислений в течение нескольких (4–16) тактов процессорного времени.

Второй подход предусматривает реализацию операции приведения по модулю полинома единожды после 32-разрядного арифметического умножения. С применением данного подхода были также выполнены 3 реализации, характеристики которых приведены в табл. 3.

Таблица 3. **Характеристики реализаций, построенных с применением умножения за 2 шага**

Вариант	Количество тактов	Ресурсы FPGA, Slices	Частота, МГц	Производительность, Мбит/с
Реализация 4	1	50	184	2807,6
Реализация 5	2	46	221	1686,1
Реализация 6	2	48	279	2128,6

В реализациях, построенных с применением второго подхода, используется 32-разрядная арифметика для выполнения арифметического умножения 16-разрядных операндов. Данная группа реализаций демонстрирует относительно низкую требовательность к ресурсам кристалла (сопоставимую с реализациями первой группы) и довольно высокую производительность, достигаемую, в том числе, за счет выполнения вычислительных операций за малое количество (1–2) тактов процессорного времени.

Следует отметить, что выбор оптимальной реализации в качестве IP-ядра в полной мере зависит от аппаратных требований и ограничений той системы, в которую такое IP-ядро необходимо встраивать.

Заключение

Рассмотрены особенности аппаратной реализации операций умножения элементов поля Галуа. Предложены подходы к эффективной реализации данных операций на FPGA. С применением каждого из подходов было выполнено по три экспериментальные аппаратные реализации операций умножения в поле Галуа с целью оптимизации скорости их вычисления. Описаны принципы построения каждой из реализаций. Основные усилия при оптимизации были сконцентрированы на уменьшении количества тактов процессорного времени, требуемых для получения результата, и повышении рабочей тактовой частоты аппаратных реализаций за счет уменьшения критического пути схемы различными способами. Дана оценка результатов тестирования как для каждой отдельной реализации, так и для основных предложенных подходов. Описаны достоинства и недостатки отдельных реализаций.

Список литературы

1. Зотов В.Ю. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx. М.: Горячая линия – Телеком, 2006. 520 с.
2. Немудров В., Мартин Г. Система на кристалле. Проектирование и развитие. М.: Техносфера, 2004. 2016 с.
3. Петровский Ал.А., Станкевич А.В., Петровский А.А. Быстрое проектирование систем мультимедиа от прототипа. Минск: Бестпринт, 2011. 410 с.
4. Захаров В.М., Нурутдинов Ш.Р., Шалагин С.В. Аппаратная реализация умножения элементов поля Галуа на программируемых микросхемах архитектуры FPGA // Вест. КГТУ им. А.Н. Туполева. 2001. № 1. С. 36–41.
5. Reyhani M.A., Hasan M.A. Low Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over GF(2^m) // IEEE Transaction on Computers. 2004. Vol. 63. № 8.
6. José L.I. Low Latency GF(2^m) Polynomial Basis Multiplier // IEEE Transaction on Circuits and Systems. 2011. Vol. 58. № 5.
7. Поляков А., Мехди Т., Незхат Т. Библиотека VERILOG описаний арифметических операций в поле Галуа //Современная электроника. 2007. № 5. С. 46–49.
8. Шалагин С.В. Реализация цифровых устройств в архитектуре ПЛИС/FPGA при использовании распределенных вычислений в полях Галуа: моногр. Казань: Изд-во КНИТУ-КАИ, 2016. 228 с.

References

1. Zotov V.Ju. Proektirovanie vstraivaemyh mikroprocessornyh sistem na osnove PLIS firmy Xilinx. M.: Gorjachaja linija – Telekom, 2006. 520 s. (in Russ.)
2. Nemudrov V., Martin G. Sistema na kristalle. Proektirovanie i razvitie. M.: Tehnosfera, 2004. 2016 s. (in Russ.)
3. Petrovskij Al.A., Stankevich A.V., Petrovskij A.A. Bystroe proektirovanie sistem mul'timedia ot prototipa. Minsk: Bestprint, 2011. 410 s. (in Russ.)
4. Zaharov V.M., Nurutdinov Sh.R., Shalagin S.V. Apparatnaja realizacija umnozhenija jelementov polja Galua na programmiruemyh mikroshemah arhitektury FPGA // Vest. KGTU im. A.N. Tupoleva. 2001. № 1. S. 36–41. (in Russ.)

5. Reyhani M.A., Hasan M.A. Low Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over GF(2m) // IEEE Transaction on Computers. 2004. Vol. 63. № 8.
6. José L.I. Low Latency GF(2m) Polynomial Basis Multiplier // IEEE Transaction on Circuits and Systems. 2011. Vol. 58. № 5.
7. Poljakov A., Mehdi T., Nezhat T. Biblioteka VERILOG opisanij arifmeticheskih operacij v pole Galua // Sovremennaja jelektronika. 2007. № 5. S. 46–49. (in Russ.)
8. Shalagin S.V. Realizacija cifrovyh ustrojstv v arhitekture PLIS/FPGA pri ispol'zovanii raspredelennyh vychislenij v poljah Galua: monogr. Kazan': Izd-vo KNITU-KAI, 2016. 228 s. (in Russ.)

Сведения об авторе

Листопад Е.В., аспирант кафедры электронных вычислительных средств Белорусского государственного университета информатики и радиоэлектроники.

Information about the author

Listopad E.V., PG student of the computer engineering department of Belarusian state university of informatics and radioelectronics.

Адрес для корреспонденции

220013, Республика Беларусь,
г. Минск, ул. П. Бровки, 6,
Белорусский государственный
университет информатики и радиоэлектроники
тел. +375-17-293-89-46;
e-mail: listopadev@bsuir.by
Листопад Евгений Валерьевич

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovka st., 6,
Belarusian state university
of informatics and radioelectronics
tel. +375-17-293-89-46;
e-mail: listopadev@bsuir.by
Listopad Evgeni Valerievich