*ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ*

*УДК 519.7*

**П.Н. Бибило, Д.А. Городецкий**

# О РЕАЛИЗАЦИИ МОДУЛЯРНЫХ СУММАТОРОВ НА FPGA

*Рассматриваются две структуры модулярных сумматоров. Исследуются параметризованные VHDL-модели, описывающие модулярные сумматоры обоих типов. Приводятся результаты синтеза на микросхемах FPGA модулярных сумматоров для различного числа операндов и различной разрядности складываемых чисел.*

## Введение

Модулярные принципы обработки сигналов находят применение при построении нейропроцессорных систем [1] и цифровых фильтров [2], в криптографии и других областях построения специализированных вычислительных систем [3]. Основной целью применения аппарата модулярной арифметики в таких системах является повышение их быстродействия за счет выполнения арифметических операций. В модулярных вычислительных структурах различают устройства, реализующие модульные и немодульные операции. К немодульным относятся операции преобразования позиционного представления информации, операция деления, к модульным – арифметические операции сложения и умножения. Значительное повышение скорости вычисления указанных арифметических операций достигается за счет разбиения позиционного представления входных операндов на операнды меньшей разрядности с целью их обработки независимо друг от друга.

В работе [1] было указано на перспективность реализации устройств модулярной арифметики на программируемых логических интегральных схемах типа FPGA (Field-Programmable Gate Array – программируемая пользователем вентильная матрица), однако сложность таких реализаций не была оценена. Чтобы реализовать алгоритм модулярного вычисления на FPGA, требуется записать данный алгоритм на языке, являющемся входным языком соответствующей системы автоматизированного синтеза. В качестве языка проектирования модулярных сумматоров был выбран язык VHDL (Very high speed integrated circuits Hardware Description Language), а в качестве системы синтеза – синтезатор LeonardoSpectrum [4].

В настоящей статье рассматриваются два типа параметризованных VHDL-описаний модулярных *N*-операндных сумматоров на синтезируемом подмножестве языка VHDL [4] и приводятся результаты схемных реализаций этих описаний на микросхемах FPGA семейства SPARTAN-II.

## 1. Модели модулярных сумматоров

Как известно [1], любое натуральное число $A = \{A_1, A_2, ..., A_k\}$ из диапазона от 0 до $M-1$ в модулярной арифметике можно представить посредством системы взаимно простых оснований $\{p_1, p_2, ..., p_k\}$, где $M = p_1 \cdot p_2 \cdot ... \cdot p_k$, таким образом, что $A_i \pmod{p_i} = A - \left[\frac{A}{p_i}\right] p_i$ и $i = \overline{1,k}$, где $[C]$ означает округление $C$ до ближайшего целого в меньшую сторону. Перевод из модулярного представления в позиционное производится в соответствии с выражением $A = A_1Y_1 + A_2Y_2 + ... + A_kY_k - rM$, где $r = 0, 1, 2, ...$; $Y_i = \left(\frac{M}{p_i}\right) k_i$, $k_i = \overline{1, p_i}$ и $\frac{Y_i}{p_i} = 1 \pmod{p_i}$.

Рассмотрим структуру устройства модулярной арифметики, предназначенного для выполнения двухоперандной ($N = 2$) арифметической операции $A + B = S$ в модулярной арифметике с системой оснований $M_1 = \{5, 7, 9\}$ (рис. 1).

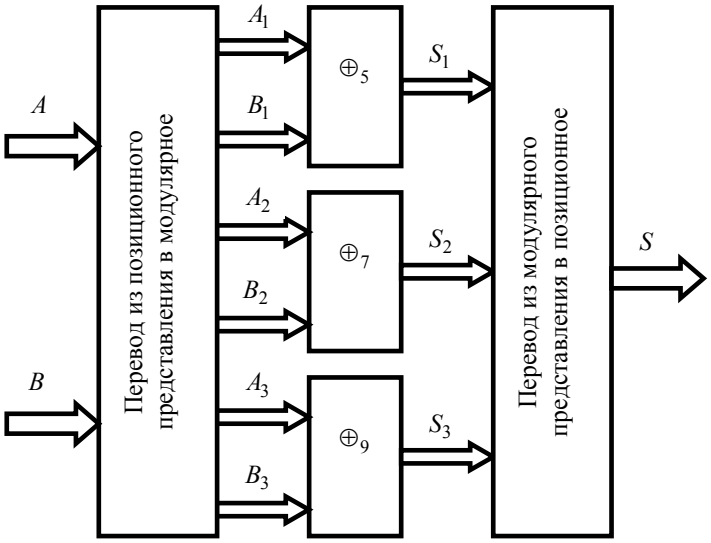


Рис. 1. Структурная схема устройства модулярного сложения с системой оснований $M_1 = \{5, 7, 9\}$

Общая структура первого типа сумматоров, назовем их *алгоритмическими* (ALG), состоит из двух последовательно соединенных блоков: блока сложения операндов и блока, реализующего аппаратное деление, т. е. выполняющего операцию нахождения наибольшего неотрицательного вычета – остатка. Данный блок функционирует в соответствии с выражением $S \pmod{p_i} = S_i' - \left[\frac{S_i'}{p_i}\right] p_i$, где $p_i \in \{p_1, p_2, ..., p_k\}$; $S_i' = X_1 + X_2 + ... + X_N$ – результат сложения до модулярного преобразования и $S \pmod{p_i} = X_1 + X_2 + ... + X_N$ – результат сложения операндов по модулю $p_i$. В качестве примера приведем структурную схему алгоритмического сумматора двух операндов по модулю пять (рис. 2). На входы блока сложения «+» поступают значения двух операндов $A$ и $B$, а на его выходе формируется результат их сложения $S'$, который, в свою очередь, поступает на вход блока модулярного преобразователя «(mod 5)», на выходе которого формируется результат $S$ модулярного сложения. В соответствии с выбранным модулем $p_i = 5$ операнды $A, B$ и $S$ принимают значение из множества {0, 1, 2, 3, 4}.

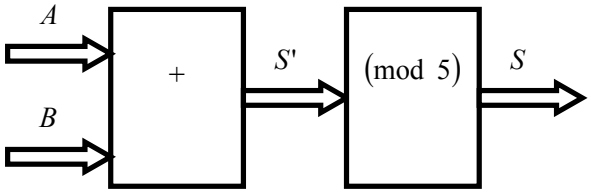


Рис. 2. Структура алгоритмического сумматора двух операндов по модулю пять

Так как для синтезатора LeonardoSpectrum и других подобных синтезаторов операция получения остатка от деления одного целого числа на другое целое число является несинтезируемой [4] (данная операция не реализуется соответствующей логической подсхемой), основной проблемой при разработке VHDL-описания модулярных сумматоров первого типа стала проблема представления операции деления на синтезируемом подмножестве языка VHDL. Существует ряд алгоритмов быстрого деления, например, представленных в работах [5–7]. Для модели ALG алгоритмического сумматора был выбран синтезируемый VHDL-алгоритм

[5] по результату сравнения данного алгоритма с лучшим алгоритмом (non-restoring algorithm) из экспериментально изученных в работе [6]. В [6] приведены сложности реализации алгоритма (non-restoring algorithm) для FPGA семейства Virtex II Pro. Алгоритм из [5] также был реализован в LeonardoSpectrum на FPGA данного семейства и позволил получить реализацию меньшей сложности и большего быстродействия для диапазона, когда число бит в двоичном представлении делимого числа не превышает 16. Например, при использовании алгоритма, представленного в [6], будет получена схема устройства, состоящая более чем из 200 CLB и 200 триггеров, в то время как алгоритм [5] приводит к схеме из 76 CLB и без триггеров, быстродействие которой в три раза выше. Конфигурируемый логический блок CLB (Configurable Logic Block) – базовый программируемый элемент различных семейств FPGA, в состав которого входят две таблицы состояний LUT (Look-Up Table), реализующие логические функции от ограниченного числа (четыре либо пять) переменных. Структуры CLB для FPGA различных семейств приведены в [8].

В дальнейшем потребуется устройство деления чисел, число разрядов которых в двоичном представлении не превышает шести, поэтому для эксперимента был выбран алгоритм [5]. Для больших диапазонов делимых чисел, например для делимого 32-разрядного числа, алгоритм, предложенный в [6], приводит к схеме меньшей сложности, но все-таки проигрывает по времени выполнения операции деления алгоритму из [5]. Однако стоит отметить, что в устройствах, реализованных на модулярных принципах обработки информации, длины двоичных представлений модулей не превышают 8 бит [3].

Другой тип модулярного сумматора, который назовем *адресным* (ADR), вместо блока аппаратного деления содержит дешифратор DC и таблицу всевозможных значений, которые может принимать сумма операндов по модулю $p_i$, – всего таблица содержит $N(p_i - 1)$ строк для $N$ -операндного сумматора, осуществляющего сложение по модулю $p_i$.

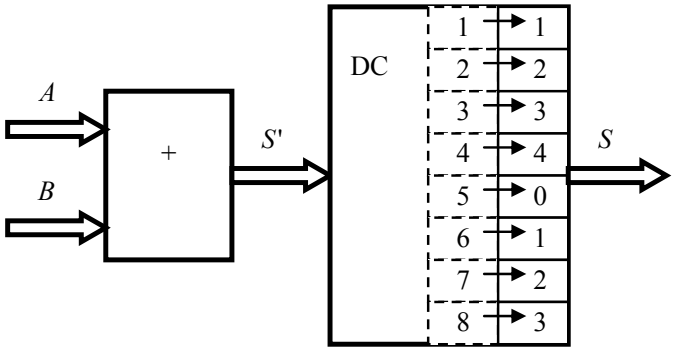


Рис. 3. Структура адресного сумматора двух операндов ($N = 2$) по модулю пять

На рис. 3 результат сложения $A + B = S'$ преобразуется в соответствующее значение по модулю пять $A + B = S \pmod 5$, которое содержится по адресу $S'$. Как и в предыдущем случае, операнды $A, B$ и $S$ принимают значение из множества $\{0, 1, 2, 3, 4\}$.

### 2. Результаты синтеза на FPGA алгоритмических и адресных сумматоров

Эксперимент по схемной реализации VHDL-описаний сумматоров типа ALG и ADR был проведен в синтезаторе LeonardoSpectrum, в качестве целевых использовались микросхемы FPGA семейства SPARTAN-II [8]. Авторами ставилась цель реализовать на FPGA блоки модулярного сложения, которые занимают центральное положение на рис. 1, схемы преобразования чисел в модулярное представление (и обратно в позиционное) на FPGA не реализовывались. Алгоритмические и аппаратные способы реализации преобразования в модулярное представление и обратно описаны, например, в [1, 3, 9], в [10] в качестве элементной базы для осуществления таких преобразований использовались FPGA.

В эксперименте были использованы системы модулей, характеристики которых приведены в табл. 1.

Таблица 1

Характеристики выбранных систем модулей

| Система модулей $M = \{p_1, p_2, ..., p_k\}$ | Диапазон представимых чисел | Число разрядов в позиционном представлении | Число разрядов каждого из оснований системы модулей: *значения оснований* $\rightarrow$ *число разрядов* |
|---|---|---|---|
| $M_1 = \{5,7,9\}$ | [0, 314] | 9 | $5,7 \rightarrow 3;\ 9 \rightarrow 4$ |
| $M_2 = \{15,17,31,37\}$ | [0, 292484] | 19 | $15 \rightarrow 4;\ 17,31 \rightarrow 5;\ 37 \rightarrow 6$ |
| $M_3 = \{7,13,15,29,31,59,61\}$ | [0, 4416458864] | 33 | $7 \rightarrow 3;\ 13,15 \rightarrow 4;$ $29,31 \rightarrow 5;\ 59,61 \rightarrow 6$ |
| $M_4 = \{17,19,23,25,27,29,31\}$ | [0, 4508102924] | 33 | $17,19,23,25,27,29,31 \rightarrow 5$ |

Результаты экспериментального синтеза модулярных сумматоров приведены в табл. 2, где жирным шрифтом отмечены лучшие решения – схемы меньшей сложности и меньшей задержки. Стоит отметить, что схемы обоих типов сумматоров являются комбинационными и не содержат триггеров.

Значение числа LUT ( $S_{LUT}$ ), которое превышает 2352 (1176 CLB), свидетельствует о том, что схема такого устройства для сложения не может быть размещена даже на самом сложном кристалле XC2S200 семейства SPARTAN-II. Однако в силу независимости сложения по каждому из модулей такое устройство может быть размещено на нескольких кристаллах [3] либо надо выбирать более сложные FPGA. Оба рассмотренных типа сумматоров были также реализованы на микросхемах FPGA семейства VIRTEX, полученные результаты коррелируют с приведенными в табл. 2 для семейства Spartan-II.

Таблица 2

Результаты схемной реализации на FPGA модулярных сумматоров

| Система модулей $M = \{p_1, p_2, ..., p_k\}$ | Число операндов $N$ | Тип сумматора | | | |
|---|---|---|---|---|---|
| | | ALG | | ADR | |
| | | $S_{LUT}$ | $t$, *ns* | $S_{LUT}$ | $t$, *ns* |
| $M_1 = \{5,7,9\}$ | 2 | **60** | **17** | 74 | 18 |
| | 7 | **188** | 27 | 321 | **21** |
| | 11 | **287** | 34 | 539 | **23** |
| | 16 | **383** | 34 | 781 | **23** |
| | 32 | **692** | 36 | 3155 | **26** |
| $M_2 = \{15,17,31,37\}$ | 2 | **130** | 23 | 175 | **16** |
| | 7 | **349** | 42 | 763 | **20** |
| | 11 | **502** | 52 | 1244 | **24** |
| | 16 | **634** | 52 | 1832 | **22** |
| | 32 | **1117** | 61 | 5162 | **25** |
| $M_3 = \{7,13,15,29,31,59,61\}$ | 2 | **217** | 23 | 481 | **14** |
| | 7 | **608** | 44 | 2030 | **20** |
| | 11 | **876** | 52 | 3290 | **25** |
| | 16 | **1107** | 53 | 4723 | **23** |
| | 32 | **1950** | 63 | 9527 | **25** |
| $M_4 = \{17,19,23,25,27,29,31\}$ | 2 | **153** | 15 | 444 | **14** |
| | 7 | **467** | 29 | 1885 | **18** |
| | 11 | **692** | 37 | 3020 | **22** |
| | 16 | **888** | 37 | 4432 | **20** |
| | 32 | **1609** | 46 | 8989 | **23** |

Примечание: $N$ – число операндов сумматора; ALG – сумматоры алгоритмического типа; ADR – сумматоры адресного типа; $S_{LUT}$ – число программируемых ячеек LUT микросхемы SPARTAN-II, требуемых для размещения модулярного сумматора; $t$, *ns* – задержка схемы модулярного сумматора.

Для двухоперандных ($N = 2$) алгоритмических сумматоров ALG было проведено также сравнение с известными реализациями на микросхемах FPGA Virtex XCV400E по указанным в [9] модулям $p_1 = 7, 13, 29, 59$. Проведенный с помощью LeonardoSpectrum синтез показал полное преимущество разработанной модели ALG модулярных сумматоров как по сложности, так и по быстродействию получаемых логических схем по сравнению с реализациями «на логике», приведенными в [9, табл. 1].

Проведенный эксперимент позволяет сделать следующие выводы:

1. По мере роста числа $N$ операндов увеличивается разница в быстродействии алгоритмических и адресных сумматоров. Эта разница увеличивается с увеличением как разрядности операндов, так и разрядности внутри системы модулей.

2. В результате экспериментов было установлено также, что переходы к RTL-описаниям модулярных сумматоров и оптимизация этих описаний с помощью программ раздельной минимизации систем булевых функций в классе ДНФ и диаграмм двоичного выбора (аналогично тому, как это было сделано в работе [11]) позволяли получать схемы большего быстродействия по сравнению со схемами, получаемыми из исходных параметризованных VHDL-описаний сумматоров.

3. С увеличением разрядности складываемых чисел, превышающей 32 бита, применение системы оснований $M_4$ с модулями одинаковой разрядности (см. правый столбец табл. 1) приводит к реализации устройства для сложения, функционирующего с большей скоростью, чем устройство, реализованное в системе $M_3$ модулей с различной разрядностью. Устройство, реализованное в системе модулей $M_4$, является менее сложным по сравнению со своим аналогом, реализованным в $M_3$.

**Заключение**

В статье рассматриваются два вида параметризованных VHDL-описаний модулярных сумматоров. В результате экспериментальных исследований установлено, что каждое из параметризованных VHDL-описаний устройств модулярного сложения имеет свои достоинства. Так, алгоритмическую модель ALG целесообразно использовать, если требуется схемная реализация на FPGA меньшей аппаратной сложности; адресную модель ADR, если требуется более быстродействующая реализация, при этом преимущество в быстродействии увеличивается с увеличением динамического диапазона складываемых чисел. При реализации на FPGA модулярных устройств для сложения чисел больших разрядностей (превышающих 32) более эффективным является использование системы модулей с одинаковой разрядностью.

**Список литературы**

1. Модулярные параллельные вычислительные структуры нейропроцессорных систем / Н.И. Червяков [и др.]. – М. : Физматлит, 2003. – 288 с.

2. Семенов, М.Ю. Применение аппарата модулярной арифметики для построения фильтра с конечной импульсной характеристикой / М.Ю. Семенов, В.С. Калашников, О.В. Ласточкин // Известия вузов. Электроника. – 2005. – № 3. – С. 46–50.

3. Стемпковский, А.Л. Особенности реализации устройств с цифровой обработкой сигналов в интегральном исполнении с применением модулярной арифметики / А.Л. Стемпковский, А.И. Корнилов, М.Ю. Семенов // Информационные технологии. – 2004. – № 2. – С. 2–9.

4. Бибило, П.Н. О несинтезируемых конструкциях языка VHDL / П.Н. Бибило // Современная электроника. – 2008. – № 5. – С. 68–71.

5. Erokhin, V.V. Single-clock division algorithm / V.V. Erokhin // OpenCores [Electronic resource]. – Mode of access : http://opencores.com. – Date of access : 15.12.2010.

6. Sorokin, N. Implementation of high-speed fixed-point dividers on FPGA / N. Sorokin // Servicio de diffusión de la creación intellectual. – Mode of access : http://sedici.unlp.edu.ar. – Date of access : 15.12.2010.

7. Fast Division Algorithm with a Small Lookup Table / P. Hung [et al.] // Scientific Literature Digital Library and Search Engine. – Mode of access : citeseerx.ist.psu.edu. – Date of access : 17.12.2010.

8. Кузелин, О.М. Современные семейства ПЛИС фирмы Xilinx : справочное пособие / О.М. Кузелин, Д.А. Кнышев, Ю.В. Зотов. – М. : Горячая линия – Телеком, 2004. – 440 с.

9. Стрекалов, А.В. Реализация базовых блоков нейрокомпьютера, функционирующего в системе счисления в остаточных классах на программируемых логических интегральных схемах / А.В. Стрекалов, Ю.А. Стрекалов // Инфокоммуникационные технологии. – 2006. – Т. 4, № 2. – С. 76–81.

10. Стрекалов, Ю.А. Реализация устройства преобразования из системы счисления в остаточных классах в позиционную систему счисления на программируемых логических интегральных схемах с использованием табличных схем / Ю.А. Стрекалов, Н.И. Червяков // Инфокоммуникационные технологии. – 2006. – Т. 4, № 1. – С. 72–76.

11. Бибило, П.Н. Автоматизированный синтез устройств модулярной арифметики: может ли САПР заменить изобретателя / П.Н. Бибило, Д.А. Городецкий // Автоматика и вычислительная техника. – 2009. – № 2. – С. 15–27.

**Поступила 13.10.10**

*Объединенный институт проблем*
*информатики НАН Беларуси,*
*Минск, Сурганова, 6*
*e-mail: bibilo@newman.bas-net.by*

**P.N. Bibilo, D.A. Gorodecky**

## ON THE IMPLEMENTATION OF MODULAR ADDERS ON FPGA

Two structures of modular adders are considered. Parameterized VHDL-models for both types of modular adders are investigated. The results of synthesis of modular adders on FPGA for various numbers and various digit capacity of input operands are shown.