УДК 004.272.2

В.М. Амербаев, Р.А. Соловьев, Д.В. Тельпухов

РЕАЛИЗАЦИЯ БИБЛИОТЕКИ МОДУЛЬНЫХ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ НА ОСНОВЕ АЛГОРИТМОВ МИНИМИЗАЦИИ ЛОГИЧЕСКИХ ФУНКЦИЙ*

Работа посвящена разработке библиотеки модульных арифметических блоков для модулярных вычислительных структур. Предлагается реализовывать сумматоры и умножители в виде логических функций, минимизированных с использованием современных эвристических алгоритмов. Это позволяет строить эффективные модулярные блоки с точки зрения производительности и занимаемой площади. Кроме того, данный метод является универсальным с точки зрения выбора оснований. А также, в совокупности с предложенным алгоритмом сокращения аппаратных затрат, удалось достигнуть приемлемых значений площади для оснований 5-6 бит. Эффективность предлагаемых методов проверена экспериментально.

Модулярные вычислительные структуры; минимизация логических функций; эвристические алгоритмы; Espresso.

V.M. Amerbaev, R.A. Solovyev, D.V. Telpukhov

LIBRARY IMPLEMENTATION OF MODULAR ARITHMETIC OPERATIONS, BASED ON LOGIC FUNCTIONS MINIMIZATION ALGORITHMS

The paper is devoted to the development of the library of modular arithmetic blocks for modular computing structures. It is proposed to implement adders and multipliers in the form of logical functions, minimized with the use of modern heuristic algorithms. This allows one to build effective modular units in terms of area and performance. In addition, this method is universal in terms of choice of moduli. Besides all, in conjunction with the proposed algorithm of hardware reduction, adequate values of space requirements for 5-6 bit moduli were achieved. The effectiveness of the proposed methods is verified experimentally.

Modular computing structures; logic functions minimization; heuristic algorithms; Espresso.

Введение. Модулярные сумматоры и умножители являются базовыми образующими блоками для всех модулярных спец-устройств. Они являются базовыми элементами, как для модульных каналов, так и для немодульных узлов, определяя основные характеристики реализуемых схем.

В настоящее время задача построения эффективных с точки зрения занимаемой площади и быстродействия двоичных сумматоров и умножителей эффективно решается на уровне современных систем автоматизированного проектирования (САПР). Транслируя Verilog - описание схемы в оптимизированную схему на уровне библиотечных вентилей, САПР использует оптимальные архитектуры арифметических блоков, освобождая разработчика от трудоемкой задачи разработки и построения архитектур арифметических операций. Однако ни один из современных распространенных САПР не поддерживает возможность эффективной реализации модулярных арифметических операций на аппаратном уровне. Для таких задач требуется вручную проектировать архитектуру элементарных схем на разном уровне абстракции.

В данной статье предлагается метод реализации базовых арифметических операций модулярной арифметики на основе современных алгоритмов минимизации логических функций.

* Работа выполнена при поддержке РФФИ (проект № 13-07-00241).

2. Реализация библиотеки арифметических операций модулярной арифметики на основе алгоритмов минимизации логических функций. Спецификой устройств, работающих в базисе модулярной арифметики, является тот факт, что основные арифметические узлы представляют собой малоразрядные блоки. На вход арифметического блока поступают операнды одной разрядности, в то время как выходом блока является число той же разрядности. Учитывая малую разрядность операндов и эффективность современных средств минимизации булевых функций, ниже приведена иллюстрация того, как реализуются арифметические операции модулярной арифметики на основе алгоритмов минимизации логических функций (рис. 1).

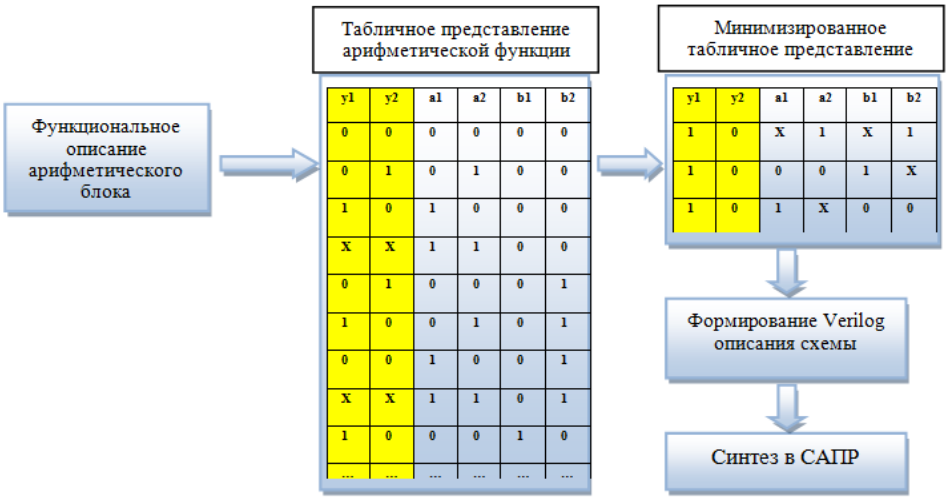

Рис. 1. Схема предлагаемого маршрута проектирования модульных арифметических блоков

Теоретически, любую арифметическую функцию можно представить в табличной форме. Входные операнды побитово представляются в виде аргументов логической функции, а результат выполнения операции в виде выходов логической функции. Затем, для эффективной реализации этой функции, необходимо воспользоваться алгоритмом минимизации булевых функций. Ранее для этих целей использовали Карты Карно и метод Куайна-Мак-Класки [1]. Они дают точный результат минимизации, однако они не очень удобны для практического применения из-за экспоненциального роста сложности от битности входных данных. В настоящее время, производственными стандартами де-факто являются: эвристический алгоритм Espresso [2] и его производные. После минимизации, происходит трансляция таблицы в схему на языке описания аппаратуры, и последующий синтез в базисе библиотеки стандартных ячеек некоторого САПР.

Данный подход применим для реализации любого арифметического блока, как двоичного, так и модулярного. Однако, необходимо понимать, что эвристические алгоритмы минимизации булевых функций эффективны для сравнительно небольшого количества аргументов. Считается, что эффективность этих методов ограничена 16 битами на входные переменные, что в случае бинарной операции (два входа у логического блока) означает ограничение до 8 бит на каждый вход. С этой точки зрения, данный метод эффективен для реализации арифметических функций в модулярных вычислительных структурах, и не может быть с той же эффективностью использован в полноразрядных двоичных арифметических блоках.

Для оценки эффективности предлагаемого метода, были созданы автоматизированные генераторы, выполняющие все шаги, необходимые для получения Verilog описания арифметической операции по методу минимизации логических функций Espresso. В качестве графической оболочки для алгоритма Espresso, была использована свободно распространяемое программное средство Logic Friday [3]. На САПР Synopsys Design Compiler в базисе стандартных ячеек Nangate, были проведены тесты модулярных сумматоров и умножителей, выполненных с помощью различных подходов на модулях из диапазона [3, 150).

Рассмотрим результаты синтеза модулярных сумматоров. Кроме предлагаемого метода был реализован традиционный модулярный сумматор. Результаты сумматоров в пределах модулей восьми бит, представлены на рис. 2.

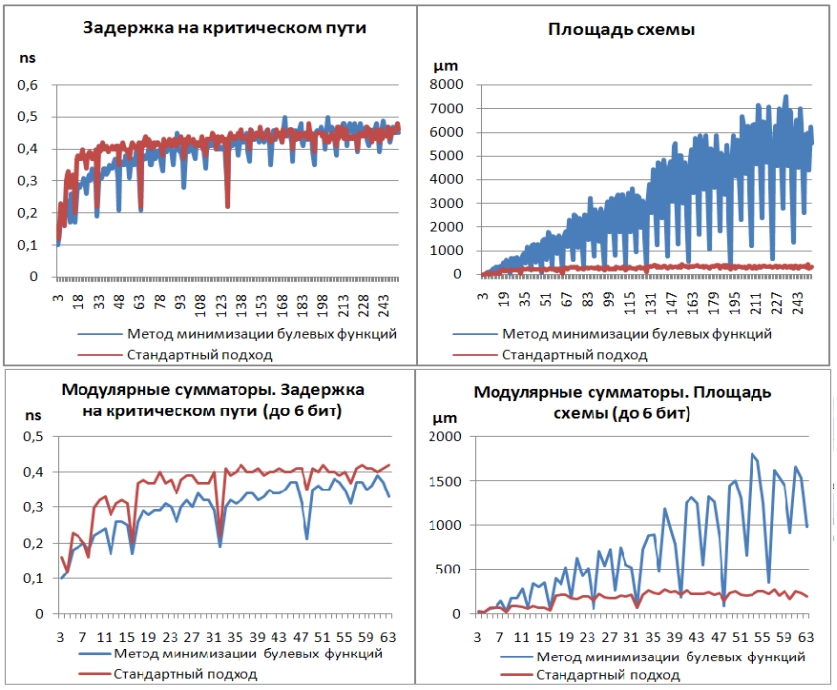

Рис. 2. Результаты синтеза модульных сумматоров

Как мы можем видеть, преимущество в производительности свыше 6 бит практически пропадает в то время, как площадь возрастает в десятки раз. Таким образом, эксперимент подтвердил тезис о том, что предлагаемый метод построения модулярных арифметических блоков эффективен для малоразрядных модулей в диапазоне до 6 бит.

Теперь обратимся к модулярным умножителям. Традиционно, модулярные умножители являются более сложными устройствами, чем модулярные сумматоры, в то время как предлагаемый метод на основе минимизации логических функций совершенно не зависит от выбора арифметической функции, что позволяет ожидать более существенные преимущества по сравнению с предыдущим экспериментом.

Для следующего эксперимента были реализованы 3 схемы модулярных умножителей: индексный модулярный умножитель [4], модулярный умножитель на базе формулы разности квадратов [5], модулярный умножитель на основе метода минимизации логических функций.

Очевидным достоинством предложенного подхода является широта его применимости. В отличие от метода разности квадратов, который реализуется только для нечетных оснований, и тем более от индексного метода, который применим лишь для простых модулей, предлагаемый метод может быть реализован для любого целого числа.

Сравнительные графики были построены для простых оснований в диапазоне [3, 150), а также в диапазоне [3, 32]:

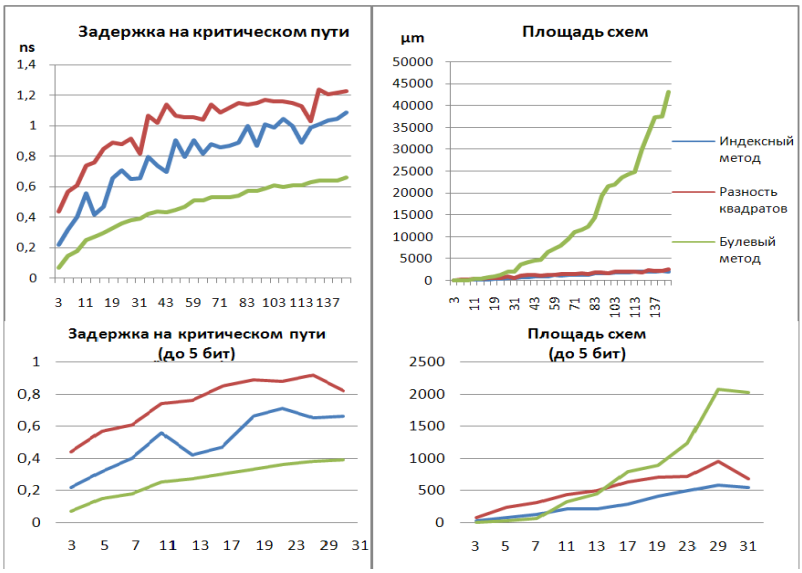

Рис. 3. Результаты синтеза модульных умножителей

Результаты синтеза для модулей в пределах 5 бит говорят о значительном преимуществе предлагаемого метода, относительно традиционных.

2. Минимизация площади. Полученные результаты показали, что с увеличением модуля для модулярных сумматоров и умножителей быстро растет площадь устройства, даже с учетом использования современных средств синтеза. По этой причине был разработан эвристический алгоритм [6], для генерации Verilog, который позволяет сохранить скорость работы и при этом уменьшить площадь устройства.

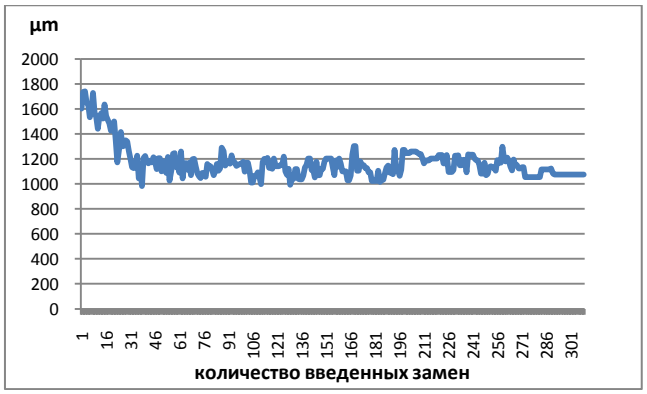

Рис. 4. Зависимость площади сумматора по модулю p = 71 от количества введенных замен

Алгоритм работает по следующей схеме. Для всех пар входов *a* и *b* ищутся пары логических термов вида (*a***b*), (*a**~*b*), (~*a***b*), (~*a**~*b*) в логическом выражении для каждого из выходов устройств. Если какая-то из этих 4 пар встречается более *N* раз, то она заменяется на отдельную переменную *GEN_a_b*, а пара исключается из последующего поиска. В дальнейших итерациях алгоритма переменная *GEN_a_b* участвует наравне с другими входами схемы. Алгоритм повторяется до тех пор, пока не останется пар логических термов встречающихся более *N* раз.

В данном случае *N* – является параметром эвристики и по нашим наблюдениям дает хорошие результаты для экспериментальных данных при *N* = 30.

Как мы можем видеть из графика, в случае модуля 71, таким способом удаётся сократить площадь более чем в 1,5 раза.

Заключение. В статье исследован метод реализации арифметических операций в модулярной арифметике, основанный на минимизации булевых функций современными эвристическими методами. Были проведены эксперименты, доказывающие эффективность предлагаемого подхода для модулей в диапазоне 5-6 бит. Также, на примере алгоритма минимизации площади, было показано, что возможности по оптимизации не исчерпаны, и исследования по данной тематике целесообразно продолжить.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *McCluskey Jr., Edward J.* Minimization of Boolean Functions // Bell Systems Technical Journal. – 1956. – Vol. 35. – P. 1417-1444.
2. *Rudell, Richard L.* Multiple-Valued Logic Minimization for PLA Synthesis // Research Report, Memorandum No. UCB/ERL M86-65 (Berkeley) 1986.
3. http://sontrak.com/.
4. *Omondi A., Premkumar B.* Residue Number Systems: Theory and Implementation. London: Imperial College Press 2007. – 312 p.
5. *Soderstrand M.A.* A new hardware implementation of modulo adders for residue number systems // Proceedings, 26th Midwest Symposium on Circuits and Systems. – 1983. – P. 412-415.
6. http://ru.wikipedia.org/wiki/Эвристический_алгоритм.

Статью рекомендовал к опубликованию д.т.н., профессор А.Л. Глебов.

Амербаев Вильжан Мавлютинович – Федеральное государственное бюджетное учреждение науки Институт проблем проектирования в микроэлектронике Российской академии наук (ИППМ РАН); e-mail: ippm@ippm.ru; 124365, Москва, Зеленоград, ул. Советская, 3; тел.: +74997299890; отдел методологии вычислительных процедур; д.т.н.; г.н.с.

Соловьев Роман Александрович – e-mail: turbo@ippm.ru; тел.: +74997299890; отдел методологии вычислительных процедур; руководитель отдела; к.т.н.

Тельпухов Дмитрий Владимирович – e-mail: nofrost@inbox.ru; отдел методологии вычислительных процедур; м.н.с.; к.т.н.

Amerbaev Viljan Mavlutinovich – The Institute for Design Problems in Microelectronics of the Russian Academy of Science (IPPM RAS); e-mail: ippm@ippm.ru; 3, Sovetskaya street, Zelenograd, Moscow, 124365, Russia; phone: +74997299890; department of computing procedure methodology; chief researcher; dr. of eng. sc.

Solovyev Roman Alexandrovich – e-mail: turbo@ippm.ru; department of computing procedure methodology; head of department; cand. of eng. sc.

Telpukhov Dmitry Vladimirovich – e-mail: nofrost@inbox.ru; department of computing procedure methodology; junior researcher; cand. of eng. sc.