Аппаратная реализация умножителя с использованием группировок разрядов аргументов

О. И. Буренева

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

OIBureneva@etu.ru

А. П. Павлов

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

appavlov@stud.etu.ru

Аннотация. **При проектировании систем цифровой обработки сигналов наиболее востребованным вычислителем является умножитель. В настоящее время разработаны различные архитектуры многоразрядных двоичных умножителей, отличающихся принципами формирования и суммирования частичных произведений, и исследуются возможности улучшения их характеристик. В статье показан вариант реализации умножителя с использованием группировок разрядов аргументов, оценены его характеристики при реализации на ПЛИС компании IntelFPGA, показаны возможности его конвейеризации.**

Ключевые слова: аппаратные ускорители; матричный умножитель; быстрые умножители; умножители FPGA; умножение на группу разрядов; ускорители арифметических операций на ПЛИС

I. Введение

При проектировании систем цифровой обработки сигналов, реализации ML-алгоритмов и нейронных сетей особое внимание уделяется быстродействию арифметических устройств, поскольку в перечисленных задачах арифметические операции используются многократно. Наибольшую актуальность представляют умножители, имеющие решающую роль в эффективном выполнении сложных арифметических операций.

Быстродействие умножителей зависит от нескольких факторов и в первую очередь от разрядности операндов и используемой архитектуры. Некоторые архитектуры могут быть конвейеризированы, что позволяет повысить производительность вычислений при обработке данных, поступающих непрерывно. В настоящее время разработаны различные архитектуры многоразрядных двоичных умножителей, отличающихся принципами получения и суммирования частичных произведений [1]. При формировании частичных произведений реализуются подходы, основанные на работе с отдельными разрядами аргументов и подходы, основанные на использовании группировок. Суммирование частичных произведений осуществляется с применением комбинационных сумматоров, которые соединяются в матричную или древовидную архитектуру.

На основе матричного подхода реализованы умножитель Брауна [2], умножитель Бо-Були [3] и умножитель Пезариса [4]. В них операция умножения сводится к параллельному формированию битов из n-разрядных частичных произведений с последующим их суммированием с помощью матриц сумматоров. Суммирование по древовидной структуре реализовано, например, в умножителе Дадда [5] и умножителе Уоллеса [6]. Древовидная структура отличается меньшим количеством сумматоров: в матричных умножителях для суммирования n частичных произведений требуется n строк сумматоров, а в древовидных схемах количество каскадов сумматоров имеет логарифмическую зависимость $\log_2 n$. Древовидная структура позволяет минимизировать время суммирования частичных произведений, однако является нерегулярной, и поэтому может иметь более сложное описание аппаратной реализации. Серьезное улучшение характеристик умножителей можно получить, оперируя не битами, а группами битов аргументов. Примером работы с группами разрядов является умножитель Бута [7], в котором минимизировано количество частичных произведений за счет умножения на группу разрядов. Каждый вариант умножителя имеет определенные достоинства и области, когда его использование наиболее эффективно [8], и для каждого варианта могут быть разработаны подходы для оптимизации характеристик при ориентации на определенные реализации [9, 10] или конкретные задачи [11].

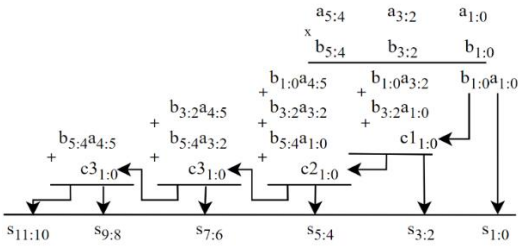
Ряд решений, используемых при проектировании умножителей, заимствован из основ индийской ведической математики – методов, быстрого численного преобразования. Особенности умножения на основе ведического подхода рассмотрены в [12], проведена оценка аппаратных затрат и временных характеристик ведического умножителя по сравнению с другими умножителями (матричным, но основе дерева Уоллеса, умножителя Бута) и показаны достоинства ведического подхода.

Целью данной работы являлась аппаратная реализация устройства умножения на основе комбинации ведического подхода и классического алгоритма умножения с ориентацией на реализацию в базисе программируемых логических интегральных схем.

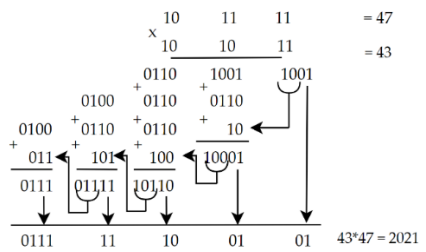
II. Структура умножителя с группировкой разрядов аргументов

В предлагаемой реализации умножителя совмещены два подхода: ведический алгоритм умножения, известный как «вертикально и крестообразно», который представляет собой древний индийский метод умножения, и классический матричный умножитель. Из ведического алгоритма использована идея разбиения процесса умножения на более простые этапы с использованием перекрестного и вертикального сложения, а также разложения чисел на более мелкие компоненты с выполнением арифметических операций над этими компонентами. При этом вопрос последовательности операций не учитывается, так как в предлагаемой аппаратной реализации базовые операции выполняются параллельно на независимом оборудовании.

Рассмотрим схему умножения шестиразрядных чисел A и B, где используются аргументы A = $a_5a_4a_3a_2a_1a_0$ и B = $b_5b_4b_3b_2b_1b_0$, показанную на рис. 1*а*. Для умножения аргументы были разбиты на группы по два разряда. Первоначально умножение начинается с младших групп, в результате чего получаются младшие биты результата (вертикальное умножение). Количество полученных младших бит результата соответствует разрядности группы. Далее младшая группа множимого умножается на следующую группу множителя. Полученный результат добавляется к произведению, полученному в результате умножения младшей группы множителя на следующую более старшую группу множимого (перекрестное умножение). Эта сумма составляет вторую группу бит конечного результата. При таком сложении любой сгенерированный перенос добавляется к частичному произведению, полученному путем умножения старших битовых групп. Такой процесс продолжается до тех пор, пока не будут обработаны все группы аргументов. Пример работы алгоритма при использовании двухразрядных групп аргументов приведен на рис. 1*б*.



а)



б)

Рис. 1. Умножение с группировкой разрядов: а) схема умножения шестиразрядных аргументов с группировкой по два бита; б) пример работы алгоритма

Очевидно, что при использовании вместо группы разрядов одного бита алгоритм будет работать по тем же принципам, что и классический комбинационный матричный умножитель.

Узким местом рассматриваемого умножителя являются операции, учитывающие возможный перенос в старшие группы, показанные в последней строчке матрицы частичных произведений. В случае, если цепочка суммирования остается комбинационной, значительно возрастает задержка получения результата, определяемая количеством столбцов суммирования частичных произведений. Для повышения частоты работы схемы эта цепочка также может быть конвейеризирована, однако это приведет к увеличению ступеней конвейера и увеличению латентности схемы.

Анализ предложенной реализации, связанной с группировкой бит множимого и множителя, показывает, что можно сократить количество частичных произведений увеличением разрядности групп. При этом усложняются умножители, которые в случае однобитного умножения представляют собой простейшие комбинационные схемы, собираемые в базисе логических примитивов, а при умножении многоразрядных чисел превращаются в матричные блоки. Отдельное внимание необходимо уделить организации многоразрядных сумматоров, поскольку они формируют значительную задержку времени вычисления. При небольших разрядностях могут использоваться сумматоры с последовательным переносом, поскольку при малых разрядностях они не вносят существенную задержку в процесс суммирования, при увеличении разрядности актуальными становятся сумматоры с параллельным переносом [13].

III. Аппаратная реализация умножителя с группировкой разрядов аргументов

Для исследования характеристик предложенного умножителя было подготовлено его параметризованное описание на языке проектирования аппаратуры VerilogHDL. При этом параметризация была связана с разрядностью устройства, разрядность групп оставалась неизменной. Проектирование устройства выполнялось с учетом его дальнейшей имплементации в программируемые логические интегральные схемы. Отдельно были рассмотрены варианты реализации сумматоров.

A. Имплементация сумматоров частичных произведений

В качестве базовых решений сумматоров рассматривались: сумматор с последовательным переносом, сумматор с параллельным переносом и реализация суммирования с использованием библиотеки схемных решений, используемых в САПР Quartus II. Основу умножителя с группировкой по два разряда составляют параметризованные сумматоры. Разрядность результата суммирования двух аргументов определяется суммой разрядов аргументов +1 для учета возможного переноса из старшего разряда. Максимальная разрядность результатов суммирования фрагментов частичных произведений получается в центральном

столбце матрицы частичных произведений и определяется следующим образом: 2*n+(m/n–1)+1, где m – разрядность множителя, n – разрядность группы. Первое слагаемое обусловлено разрядностью произведения группы множимого на группу множителя; второе слагаемое обусловлено необходимостью расширения разрядности при суммировании m/n чисел; суммирование с единицей объясняется необходимостью суммирования переноса, сформированного при суммировании младшей группы. В табл. 1 показаны результаты сравнения имплементации четырехразрядных сумматоров в микросхему FPGA Cyclone III: 1 – с последовательным переносом; 2 – с параллельным переносом; 3 – реализация суммирования с использованием библиотеки схемных решений.

ТАБЛИЦА I. ХАРАКТЕРИСТИКИ КОМБИНАЦИОННЫХ СУММАТОРОВ

Тип сумматора **Параметр**	1	2	3
Максимальное время задержки, нс	10,124	10,468	10,741
Количество логических ячеек	7	9	5
Количество использованных LUT -- 4-х входовых -- 3-х входовых -- <=2 входов	 2 4 1	 5 2 2	 0 4 1

Анализ результатов показывает, что схема с ускорением переноса практически не дает эффекта с точки зрения быстродействия. Это обусловлено тем, что схема параллельной генерации переносов сложна и за счет многоступенчатости вносит задержку, соизмеримую со временем срабатывания одноразрядного сумматора. При увеличении разрядности соотношение время/аппаратные затраты для разных вариантов сумматоров сохраняется.

B. Имплементация умножителей

RTL представление 6-ти битного комбинационного умножителя с группировкой по два разряда, сгенерированное системой проектирования Quartus II, приведено на рис. 2. Для сборки умножителя использованы 9 умножителей 2 x 2 (на схеме обозначены m_2x2) и 7 параметризованных сумматоров (обозначены add_n_bit). Параметризация сумматоров по разрядности понадобилась для того, чтобы учесть разные разрядности при суммировании столбцов групп частичных произведений. Максимальное количество аргументов суммирования для рассматриваемой схемы – 4, максимальная разрядность сумматора – 6 с представлением результата 7 битным кодом.

Для корректной оценки частоты работы комбинационного умножителя с использованием утилиты TimeQuest Timing Analyzer, позволяющей выявлять частоту тактового сигнала на основании самой длинной комбинационной цепи в схеме, на входе и выходе устройства были установлены регистры защелки. Эти регистры не несут функциональной нагрузки, поэтому, несмотря на сообщение в файлах отчетов (*.rpt) о их использовании, их можно не учитывать при подсчете общих затрат на реализацию схемы.

Корректность работы умножителя проверена в ходе модельных экспериментов с использованием системы моделирования ModelSim Altera, на рис. 3 показана временная диаграмма, полученная в процессе моделирования. Для обеспечения максимального тестового покрытия использовался случайный генератор входных сигналов. Также в тестовое окружение был встроен контроллер, который формировал произведение (rezult_model) путем умножения аргументов и сравнивал полученный результат с выходным сигналом тестируемого умножителя. При несовпадении ожидаемого и полученного значения формировался сигнал ошибки (error). Для компенсации временного сдвига рассчитанного результата введены дополнительные сигналы (rezult_model1 и rezult_model2), задержанные во времени на один и два такта соответственно.

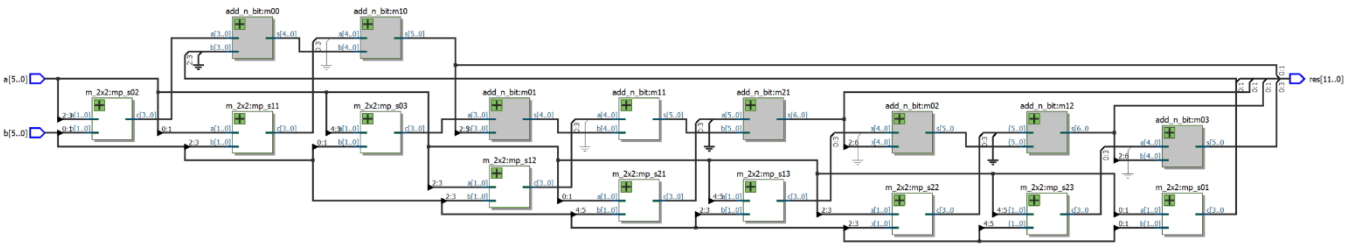


Рис. 2. RTL схема 6-ти битного комбинационного умножителя с группировкой по два разряда

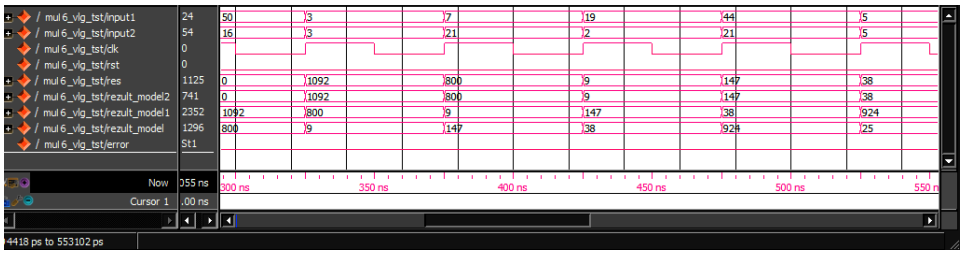


Рис. 3. Временная диаграмма работы умножителя с группировкой разрядов множимого и множителя

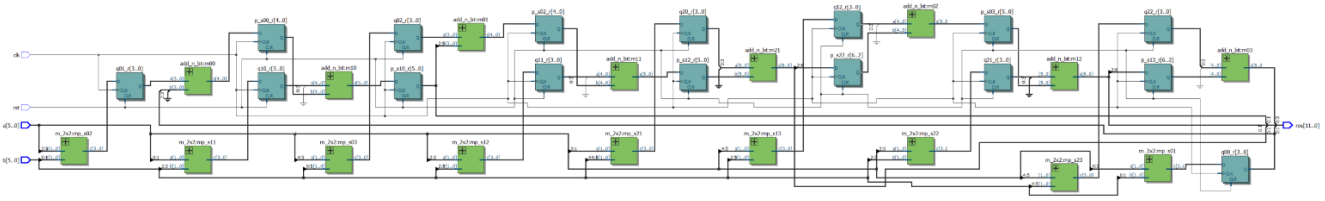


Рис. 4. RTL схема 6-ти битного конвейерного умножителя с группировкой по два разряда

Дополнительно была исследована возможность конвейеризации предложенного умножителя. Количество ступеней конвейера было определено в соответствии с количеством строк в центральном столбце матрицы групп частичных произведений. Допустимо и иное расположение регистров для уменьшения количества ступеней, что очевидно приведет к снижению максимально возможной частоты. RTL схема 6-ти битного конвейерного умножителя показана на рис. 4.

Временные характеристики оценивались с использованием моделей Slow1 и Slow2, соответствующих худшим случаям 1200 мВ при 85° С и 1200 мВ при 0° С. В таблице приведена максимальная частота Fmax, рассчитанная САПР Quartus II без учета ограничений конкретной микросхемы и превышающая значения Restricted Fmax. Характеристики разработанных устройств представлены в табл. 2 для предложенного комбинационного (1) и конвейерного (2) умножителей.

ТАБЛИЦА II. ХАРАКТЕРИСТИКИ УМНОЖИТЕЛЕЙ

Тип умножителя / **Параметр**	1	2
Fmax, MHz (Slow1)	139,98	418,76
Fmax, MHz (Slow2)	156,74	472,59
Количество логических ячеек	102	98
Количество комбинационных ячеек	90	98
Количество использованных LUT -- 4-х входовых -- 3-х входовых -- <=2 входов	 68 10 12	 56 18 24
Количество логических элементов -- в нормальном режиме	 90	 98
-- в арифметическом режиме	0	0
Количество регистров	24	69

Анализ данных табл. 2 показывает, что конвейеризация умножителя даже при фиксации каждого частичного произведения не увеличивает аппаратные затраты. Это объясняется тем, что в составе логических ячеек использованной ПЛИС имеется как комбинационная часть, так и триггер, в котором и фиксируется промежуточный результат, при этом количество ячеек не увеличивается, производительность же возрастает кратно.

IV. ЗАКЛЮЧЕНИЕ

Представленные результаты исследования умножителя с группировкой разрядов аргументов показывают, что при определенных условиях организации вычислений можно получить преимущества по времени выполнения операции. Предложенная реализация имеет регулярную структуру как с точки зрения формирования частичных произведений, так и с точки зрения организации суммирования, что позволяет ее конвейеризировать, увеличивая тем самым производительность. Алгоритм легко распараллеливается, что позволяет реализовать его на базе ПЛИС. Дальнейшие исследования могут быть связаны с определением оптимальных характеристик групп и оптимальной конвейеризацией с точки зрения аппаратных затрат и производительности.

СПИСОК ЛИТЕРАТУРЫ

[1] Mi Lu, Arithmetic and Logic in Computer Systems, 1rd ed., Wiley-Interscience, 2004.

[2] B. Neeraja and R. S. P. Goud, "Design of an Area Efficient Braun Multiplier using High Speed Parallel Prefix Adder in Cadence," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2019, pp. 1-5, doi: 10.1109/ICECCT.2019.8869307.

[3] R. Gayathri, Kumar Ajith, P. Balaji, P. Magesh, V. Sridhar,"VLSI Design of Approximate Baugh-Wooley Multiplier for Image Edge Computing," International Journal of Advanced Research in Science, Communication and Technology, vol. 3(8), pp. 28-33, April 2023, doi: 10.48175/IJARSCT-9537 28.

[4] J. Stohmann and E. Barke, "A universal Pezaris array multiplier generator for SRAM-based FPGAs," Proceedings International Conference on Computer Design VLSI in Computers and Processors, Austin, TX, USA, 1997, pp. 489-495, doi: 10.1109/ICCD.1997.628913.

[5] S. Chanda, K. Guha, S. Patra, A. Karmakar, L. M. Singh and K. Lal Baishnab, "A 32-bit Energy Efficient Exact Dadda Multiplier," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, pp. 1-4, doi: 10.1109/I2CT45611.2019.9033535.

[6] S. Nagaraj, K. Thyagarajan, D. Srihari and K. Gopi, "Design and Analysis of Wallace Tree Multiplier for CMOS and CPL Logic," 2018 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), Chennai, India, 2018, pp. 006-010, doi: 10.1109/ICCPEIC.2018.8525224.

[7] Booth, A.D. A signed binary multiplication technique. Q. J. Mech. Appl. Math. 1951, 4, 236–240.

[8] M. Chinbat et al., "Performance Comparison of Finite Field Multipliers for SM2 Algorithm based on FPGA Implementation," 2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, 2020, pp. 69-72, doi: 10.1109/ASID50160.2020.9271714.

[9] S. Raveendran, P.J. Edavoor, Y. B. N. Kumar and M. H. Vasantha, "Inexact Signed Wallace Tree Multiplier Design Using Reversible Logic," in IEEE Access, vol. 9, pp. 108119-108130, 2021, doi: 10.1109/ACCESS.2021.3100892.

[10] Y.-J. Chang, Y.-C. Cheng, S.-C. Liao and C.-H. Hsiao, "A Low Power Radix-4 Booth Multiplier with Pre-Encoded Mechanism," in IEEE Access, vol. 8, pp. 114842-114853, 2020, doi: 10.1109/ACCESS.2020.3003684.

[11] O. Bureneva, S. Mironov, "Fast FPGA-Based Multipliers by Constant for Digital Signal Processing Systems," Electronics, vol 12, 605, 2023, doi: 10.3390/electronics12030605.

[12] Якунин А.Н., Аунг Мьо Сан Повышение скорости работы многоразрядного двоичного умножителя // Проблемы разработки перспективных микро- и наноэлектронных систем. 2018. Выпуск 2. С. 149-155. doi:10.31114/2078-7707-2018-2-149-155.

[13] Угрюмов Е.П. Цифровая схемотехника: учеб. пособие для вузов. 3-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2010. 816 с.: ил. ISBN 978-5-9775