

## КОНВЕЙЕРНЫЕ УСТРОЙСТВА НА FPGA

Волошин Д.Н., Зинченко Ю.Е., Дяченко О.Н.  
 Донецкий национальный технический университет  
 кафедра компьютерной инженерии  
 E-mail: [dmitrius.voloshin@gmail.com](mailto:dmitrius.voloshin@gmail.com)

Источник: Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС-2012) / Матеріали III міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених – Донецьк, ДонНТУ – 2012.

### Аннотация:

*Волошин Д.Н., Зинченко Ю.Е., Дяченко О.Н. Конвейерные устройства на FPGA. Рассматриваются основные принципы конвейерной обработки. Кратко описывается архитектура FPGA и особенности построения на её основе конвейерных устройств. Приводятся комбинационные и конвейерные схемы для реализации операций суммирования и умножения. Сравняются результаты их VHDL-синтеза. Делается вывод о целесообразности использования конвейеризации для этих устройств.*

### Введение

С ростом производительности вычислительных средств растут и вычислительная сложность алгоритмов, а также требования по быстродействию со стороны пользователя. Реализовать обработку данных на данный момент можно на основе двух базовых аппаратных технологий: микропроцессорной технике и программируемых логических интегральных схемах (ПЛИС). Обе технологии имеют как преимущества, так и недостатки. К преимуществам микропроцессоров можно отнести простоту разработки устройств на их базе, которая, в сущности, сводится к написанию программного обеспечения. Всё большую популярность приобретает проектирование устройств на базе последней технологии, а именно FPGA (field-programmable gate array). При этом затраты на разработку и сложность значительно возрастают, так как требуется специализированная среда разработки и отладочная плата.

Почему же FPGA технология более привлекательна по сравнению с микропроцессорной технологией? Регулярная структура ПЛИС позволяет строить на основе одной микросхемы комплексы из нескольких устройств и даже несколько независимых устройств. Основным преимуществом ПЛИС является возможность реализации на их базе принципов параллелизма и конвейеризации [6]. Первый принцип заключается в возможности одновременного выполнения нескольких однотипных действий и реализуется в основном в виде дублирования устройств. Второй принцип позволяет разбить выполнение сложной задачи на ряд простых последовательных действий с одновременным совмещением их выполнения во времени. Рассмотрим принцип конвейеризации на основе работы конвейера команд процессора (рис. 1).



Рис. 1. Схема конвейера команд

Рассматриваемый конвейер состоит из пяти ступеней, каждая из которых выполняет определённое действие над командой. После завершения обработки текущей команды на текущем этапе, выполнение команды переходит на следующий этап, а на данный этап обработки поступает следующая команда. Допустим, что каждая ступень конвейера выполняет обработку за один такт, тогда после загрузки конвейера (5 тактов), выполнение команд будет производиться с периодом в один такт. Считая, что обычное (не конвейерное) выполнение команды занимает 5 тактов, имеем ускорение выполнения команд в 5 раз. Имеются, конечно же, ограничения и недостатки данного метода, но все они также решаемы в той или иной степени. Описанный принцип можно использовать и для ускорения выполнения других алгоритмов.

### Конвейерный сумматор

В качестве простейшего примера конвейеризации на ПЛИС рассмотрим построение арифметического беззнакового сумматора. Для анализа реализации этого и всех последующих устройств используются результаты синтеза среды Xilinx 12.4 для семейства микросхем Virtex6 [2]. Для того чтобы сделать вывод об эффективности конвейерной схемы, сначала синтезируем типовую схему комбинационного сумматора с входным и выходным переносами. Следует отметить, что при синтезе данного сумматора среда разработки использует оптимизированную схему суммирования, позволяющую увеличить быстродействие при использовании большего числа входов LUT. Конвейерный вариант данного сумматора будет иметь вид, показанный на рис. 2.

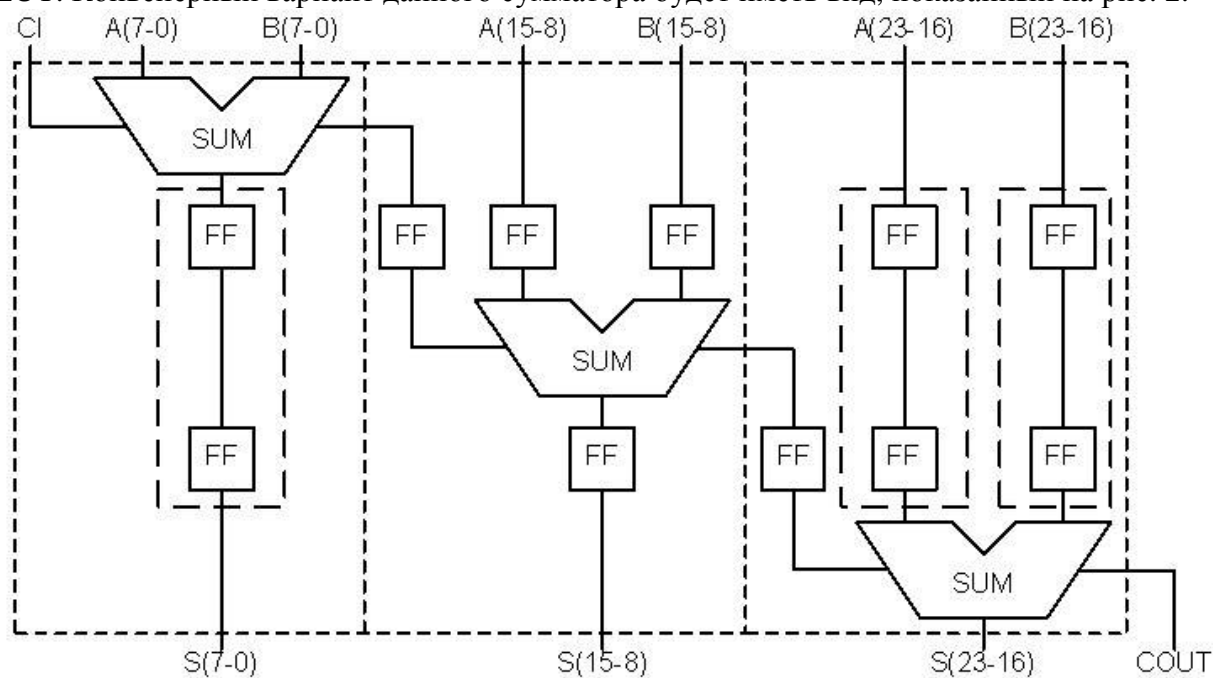


Рис. 2. Схема конвейерного сумматора

Теперь, для реализации сумматора на  $n$  разрядов при  $m$  ступенях конвейера необходимо те же  $n$  LUT и дополнительные регистры. Суммарное количество

$$(m-1) * \left(\frac{3}{2} * n + 1\right)$$

необходимых для реализации триггеров равно . Из полученной формулы можно сделать вывод, что результирующие затраты триггеров значительно зависят от разрядности сумматора, чем от количества ступеней конвейера. Рассмотренная схема является типовой схемой конвейеризации и вместо сумматоров может быть использована любая комбинационная схема. Следует иметь в виду, что схема является синхронной, все триггера срабатывают по фронту сигнала синхронизации. Для упрощения цепи синхронизации и асинхронного сброса на схеме не указаны. Реализуем рассмотренную схему в структурном стиле, на основе параметризованных компонентов, в виде сумматора с цепочками регистров на входе и выходе [3]. Следует отметить, что при длине цепочки 2 и более, средство синтеза

автоматически реализует цепочку на основе встроенного сдвигового регистра, что значительно уменьшает аппаратные затраты. Также необходимо помнить, что триггера, входящие в состав базовой ячейки также могут быть использованы в данной схеме. Результаты синтеза рассмотренного конвейерного сумматора в сравнении с комбинационным сумматором приведены в таблице 1.

Таблица 1 – Результаты синтеза сумматоров

Разрядность, n	Число ступеней, m	Комбинационный		Конвейерный	
		Затраты, LUTs	Задержка, ps	Затраты, LUTs	Задержка, ps
512	2	512	9189	513	5151
	4			1539	2975
	8			1799	1887
256	2	256	4837	257	2975
	4			771	1887
	8			903	1343
	16			975	1071
	32			1023	935

Из полученной таблицы можно сделать некоторые важные выводы. При двух ступенях конвейера, регистры практически полностью реализуются на тех же логических ячейках, на которых реализованы сумматоры. Затем использование ресурсов резко возрастает, а при количестве ступеней 8, 16, 32 изменяется незначительно, это связано с использованием встроенных сдвиговых регистров [1].

### Конвейерный умножитель

Помимо операций сложения и вычитания, на FPGA возможна также реализация устройств, выполняющих умножение и деление. Схема комбинационного умножителя приведена на рис. 3.

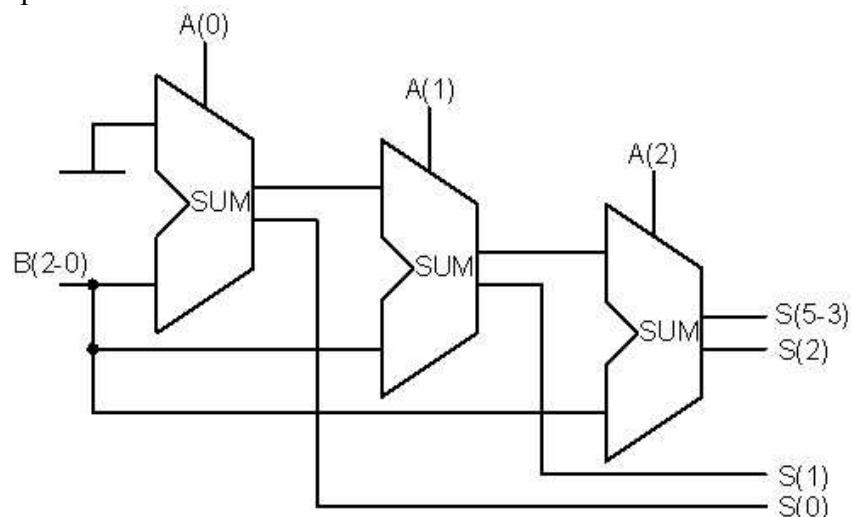


Рис. 3. Схема комбинационного умножителя

Принцип работы схемы основан на последовательном анализе бит первого множителя. Если бит, равен единице, то к частичному произведению прибавляется второй множитель. Время срабатывания и производительность данной схемы можно значительно увеличить, если обеспечить конвейеризацию вычислений [4],[7]. Схема усовершенствованного устройства показана на рис. 4. Как и в случае с конвейерным сумматором, на схеме не приведены цепи синхронизации и асинхронного сброса. Суммарные затраты триггеров на реализацию данной схемы равны  $3 * n * (n - 1)$ . Из

данной формулы можно сделать вывод, что дополнительные затраты на конвейеризацию, так же как и основные затраты на построение сумматоров ( $n^2$ ), растут пропорционально квадрату разрядности умножителя. Синтезируем рассмотренную схему, результаты синтеза приведены в таблице 2.

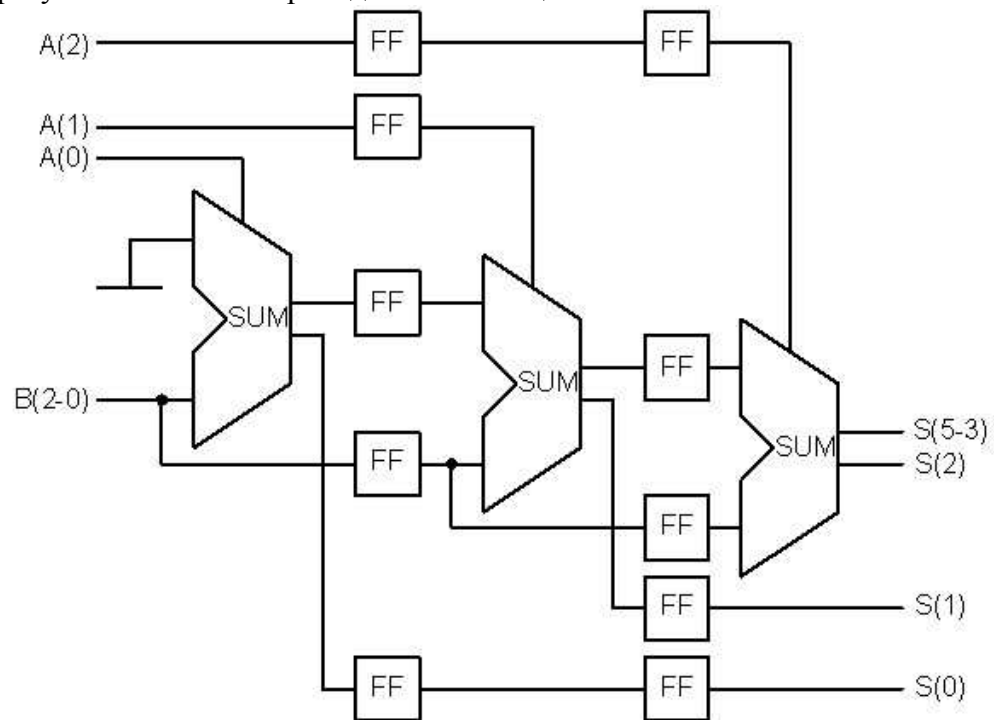


Рис. 4. Схема конвейерного умножителя

Таблица 2 – Результаты синтеза умножителей

Разрядность, n	Комбинационный		Конвейерный		Прирост аппаратных затрат
	Быстродействие, ps	Затраты, LUTs	Быстродействие, ps	Затраты, LUTs	
8	7755	64	935	90	41%
16	16144	256	1071	314	23%
32	32824	1024	1343	1146	12%

Из полученной таблицы можно сделать вывод, что быстродействие полученной схемы незначительно зависит от разрядности, а дополнительные аппаратные расходы на конвейеризацию с ростом разрядности стремятся к нулю. Это можно объяснить использованием в схеме входящих в состав логической ячейки триггеров и сдвигового регистра.

### Выводы

В данной работе были рассмотрены основные принципы конвейеризации и их использование. Следует отметить, что использование данного метода обработки возможно не только при построении конвейерной обработки команд в микропроцессорных устройствах, но и при разработке практически любых устройств. Главной задачей, при реализации конвейерного варианта устройства, является обеспечение независимости работы текущей ступени конвейера на данном такте от результатов работы предыдущей ступени на этом же такте. Для реализации этого требования чаще всего можно использовать регистровые схемы. Были разработаны конвейерные схемы сумматора и умножителя, рассчитаны дополнительные затраты регистровых элементов при реализации этих схем в базисе дискретных элементов.

Также, в работе были рассмотрены особенности структуры логической ячейки FPGA и её эффективного использования для построения устройств с конвейерной архитектурой [5]. В результате реализации разработанных ранее схем на FPGA были

получены их синтезируемые VHDL-описания. На основе результатов синтеза можно сделать следующие выводы:

- Прирост быстродействия схемы при конвейеризации зависит от числа ступеней конвейера и соотношения быстродействий комбинационной и регистровой части и стремится к значению  $t/m$ , где  $t$  – время работы исходной схемы,  $m$  – число ступеней конвейера. То есть, конвейеризация устройств с большим временем срабатывания ступени, даст большой прирост по быстродействию;
- Прирост аппаратных затрат на реализацию конвейерного метода обработки зависит от соотношения затрат на реализацию комбинационной части и регистровой части и стремится к нулю, то есть, при конвейеризации устройств с большими затратами LUT на реализацию ступени конвейера, дополнительные аппаратные затраты снижаются, так как используются триггера из логических ячеек комбинационной части;
- Конвейеризация операции суммирования требует значительно больших затрат запоминающих элементов и даёт меньший прирост по быстродействию по сравнению с конвейерным вариантом умножения.

Рассмотренные в этой работе принципы могут быть также использованы при построении более сложных устройств для работы в следующих сферах: кодирование, шифрование, аппаратная сортировка и архивация, цифровая обработка сигналов.

#### **Список источников**

1. Virtex-6 FPGA Configurable Logic Block. User Guide. UG364(v1.2) February 3, 2012.
2. Virtex-6 Family Overview. Product Specification. DS150(v2.4) January 19, 2012.
3. XST User Guide for Virtex-6, Spartan-6, and 7 Series Devices. UG687(v13.4) January 18, 2012.
4. Alex Panato, Sandro Silva, Flavio Wagner, Marcelo Johan, Ricardo Reis, Sergio Bampi. Design of Very Deep Pipelined Multiplier for FPGAs.
5. Oswaldo Cadenas, Graham Megson. A clocking technique for FPGA pipelined designs. Journal of System Architecture 50 (2004) 687-696.
6. Keshab K. Parhi. Pipelining and Parallel Processing.
7. Mathew Wojko. Pipelined Multipliers and FPGA Architecture.