

УДК 004.31

ПРОЕКТИРОВАНИЕ КОНВЕЙЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ С УЧЕТОМ ТОПОЛОГИЧЕСКОГО ПРЕДСТАВЛЕНИЯ

И. Е. Тарасов, д.т.н., заведующий лабораторией специализированных вычислительных систем РТУ МИРЭА, Москва, Россия;

orcid.org/0000-0001-6456-4794, e-mail: tarasov_i@mirea.ru

Д. В. Люлява, м.н.с. лаборатории специализированных вычислительных систем РТУ МИРЭА, Москва, Россия;

orcid.org/0009-0009-9623-7777, e-mail: lyulyava@mirea.ru

Н. А. Дуксин, инженер лаборатории специализированных вычислительных систем РТУ МИРЭА, Москва, Россия;

orcid.org/0009-0009-0014-7065, e-mail: duksin@mirea.ru

Рассматривается задача проектирования конвейерного вычислителя для работы в составе цифровой вычислительной системы. Целью работы является исследование влияния архитектуры конвейерного вычислителя на характеристики его топологического представления для проведения оптимизации вычислителя по выбираемым критериям оптимальности. При проектировании высокопроизводительных вычислительных систем важным этапом является архитектурное проектирование и проведение декомпозиции системы. При этом выбор операций для аппаратного ускорения зависит от характеристик ускорителя, получаемого на имеющейся аппаратной платформе. Проектирование на различных уровнях: системном, схемотехническом и топологическом, с одной стороны, позволяет абстрагироваться от деталей реализации и повысить продуктивность разработки, но с другой – переход к деталям реализации уточняет итоговые характеристики проектируемого ускорителя для вычислительной системы и может существенно ухудшить их относительно предварительных ожиданий. Рассмотрены подкласс конвейерных ускорителей вычислений и зависимость их характеристик размера, задержки распространения сигнала и потребляемой мощности от архитектуры системы управления. Выявлена возможность частичной компенсации недостатков архитектур, имеющих простую топологическую реализацию, за счет узлов, обеспечивающих интеграцию конвейера в вычислительную систему. Предложено использовать оптимизацию в дискретном пространстве параметров, для чего используется параметризованное описание конвейера.

Ключевые слова: вычислительная система, конвейер, архитектура, ПЛИС, СБИС, уровень регистровых передач.

DOI: 10.21667/1995-4565-2023-86-86-95

Введение

Одной из задач проектирования специализированных вычислительных систем является их декомпозиция с выделением вычислительных узлов разного типа, предназначенных для решения соответствующих подклассов задач [1]. Признанным архитектурным подходом универсального назначения является процессорное ядро, последовательно выполняющее операции в соответствии с программой, неограниченно модифицируемой в процессе эксплуатации системы. При этом универсальный характер процессорного ядра общего назначения является одновременно и ограничивающим фактором, препятствующим достижению высоких показателей производительности в узкоспециализированных классах алгоритмов. С целью повышения эффективности системы процессорные ядра дополняются аппаратными ускорителями, которые сами по себе могут иметь различные архитектуры. При расширении возможностей процессора аппаратными ускорителями следует учитывать распределение задач между ними и избегать ситуаций, в которых добавленный ускоритель будет дублировать

функции процессора общего назначения или, напротив, эффективно решать крайне узкий класс задач, не востребованный для типичных областей применения проектируемой вычислительной системы. В этой связи имеет смысл рассмотреть расширение возможностей процессоров общего назначения ускорителями в виде конвейеров, предназначенных для решения определенного подкласса задач. Важность сочетания устройств общего назначения и специализированных компонентов отмечается в современных исследованиях как одно из важнейших направлений обеспечения роста производительности вычислений [2].

Конвейерные вычислители представляют собой пример устройства, дополняющего возможности центрального процессора в гетерогенной вычислительной системе. Преимуществом конвейера является потенциальная возможность достижения высокой тактовой частоты, поскольку сложные вычисления разбиваются на отдельные операции, каждая из которых выполняется на соответствующей стадии конвейера. Принципы построения конвейерной схемы хорошо соотносятся с рекомендациями по разработке цифровых схем с технологическими нормами 130 нм и менее.

По мере уменьшения технологических норм в цифровой электронике увеличилась роль следующих негативных эффектов, которые требуют принятия комплекса технических мер для снижения их влияния на проект.

1. «Темным кремнием» называют эффект, связанный с повышением плотности компонентов на полупроводниковом кристалле, и, как следствие, увеличением тепловыделения микросхемы сверх пределов, допустимых для организации теплоотвода [3, 4]. Важную роль играет тот факт, что речь идет не об отводе тепла от внешних радиаторов, а о невозможности обеспечения достаточного теплового потока от поверхности кристалла до теплоотводящих элементов корпуса. Очевидным путем устранения перегрева микросхемы является перевод части элементов в неактивное состояние, в качестве которого может рассматриваться не только отключение питания, но и отсутствие переключения входных сигналов (switch activity). Однако такой подход ведет к снижению производительности вычислений.

2. Увеличение площади полупроводникового кристалла в сочетании с ростом тактовой частоты усложняет синхронизацию компонентов схемы, что ведет к ее нестабильной работе. Поэтому площадь синхронизируемых компонентов ограничивают, локализуя отдельные тактовые сети, передача данных между которыми осуществляется с помощью специальных узлов ресинхронизации (clock domain crossing). Формируемая таким образом архитектура обозначается GALS (Globally Asynchronous, Locally Synchronous).

3. Понятие «стены интерфейсов» [5] относится к практически наблюдаемому эффекту опережающего роста производительности вычислений по сравнению с ростом производительности внешних интерфейсов (включая подсистемы памяти). Данный эффект может затруднить интеграцию высокопроизводительного вычислительного узла в систему, если требуется передача данных через внешние относительно микросхемы интерфейсы.

Дополнительно следует учитывать иерархический характер уровней проектирования цифровых микросхем, который можно разделить на:

1) уровень программной модели, на котором операции с данными рассматриваются с точки зрения выполняемых преобразований и получаемого результата, причем детали схемотехнической реализации не рассматриваются. Данный уровень наиболее эффективен для исследования и отладки алгоритмов и может быть реализован с помощью языков программирования высокого уровня;

2) уровень регистровых передач (RTL) – реализует электрическую схему с применением языков описания аппаратуры (HDL). По сравнению с программной моделью на этом уровне могут быть выявлены электрические конфликты или несинтезируемые конструкции;

3) топологический уровень представляет размещение компонентов на полупроводниковом кристалле (для СБИС) или конфигурацию логических ресурсов и программируемых соединений для ПЛИС. Данный уровень позволяет также определить размер схемы, задержки распространения сигналов и потребляемую мощность. Формирование перечня размещения

компонентов и схемы их соединения производится в САПР ПЛИС и обозначается термином Implementation. Данная операция является закрытой в большинстве САПР ПЛИС ведущих производителей.

В данной статье рассматривается влияние особенностей топологической реализации конвейерных вычислителей на характеристики получаемой подсистемы с учетом действия перечисленных факторов. Можно отметить, что в процессе разработки вычислительных устройств этап размещения компонентов и трассировки цепей (Place & Route) часто оказывается существенно сложнее с точки зрения вычислительной сложности, чем этап синтеза электрической схемы. Таким образом, результаты, полученные на уровне регистровых передач, могут быть существенно ухудшены на топологическом уровне проектирования, если размещение компонентов происходило без учета их связей. При большом числе компонентов и связей между ними САПР ПЛИС не обеспечивают оптимального размещения компонентов в его строгом математическом смысле, так как подобные алгоритмы относятся к NP-трудным. Важность управления параметрами топологической реализации проекта на базе ПЛИС отмечается, в частности, в руководстве AMD/Xilinx по управлению процессом Implementation в САПР Vivado. Вмешательство разработчика в процесс размещения компонентов проекта обеспечивается добавлением описаний на «языке проектных ограничений» XDC (Xilinx Design Constraints).

Таким образом, при проектировании цифровых устройств возникает задача их совместного анализа на перечисленных уровнях проектирования, причем нижележащие уровни обеспечивают большую детализацию характеристик, а уровень программной модели характеризуется высокой продуктивностью процесса разработки. На примере конвейерных структур рассматривается маршрут проектирования, основанный на ранней оценке характеристик, получаемых для топологического представления схемы, и применении схемотехнических решений на уровне регистровых передач в зависимости от требуемых характеристик производительности, размера и потребляемой мощности.

Теоретическая часть

Рассматриваемые конвейерные вычислительные схемы представляют собой пример практически применимого устройства, которое при этом имеет преимущественно локальные связи между компонентами. Это уменьшает количество факторов, влияющих на характеристики топологического представления, и упрощает анализ применяемых технических решений. Данный вопрос был ранее исследован в [6]. При этом можно отметить, что на базе конвейерных структур обеспечивается решение достаточно востребованных в вычислительных системах задач, например реализация базовых алгоритмов цифровой обработки сигналов [7, 8], поэтому спектр возможных применений конвейеров в составе процессорных систем достаточно широк.

Конвейерная архитектура в целом позволяет снизить влияние перечисленных во введении негативных эффектов. Поточковый характер вычислений позволяет ограничить интерфейсы конвейера входным и выходным наборами данных, а применение FIFO, реализующего очередь, позволяет локализовать тактовый домен конвейерного вычислителя, обеспечив выполнение принципов GALS. Особенностью применения модулей FIFO, используемых в ПЛИС с архитектурой FPGA, является то, что они основаны на двупортовой статической памяти (т.е. «true dual port»), что позволяет использовать независимое тактирование портов от тактового домена системной шины и тактового домена конвейера. Для воспроизведения данного узла в составе СБИС необходимо использование аналогичных по интерфейсам компонентов статической памяти, обеспечивающих два независимых интерфейса с физически раздельными сигналами тактирования.

Что касается уменьшения тепловыделения, то данная задача требует регулирования количества стадий конвейера для изменения задержки распространения сигналов, что опреде-

лит тактовую частоту. Схема интеграции конвейера в вычислительную систему показана на рисунке 1.

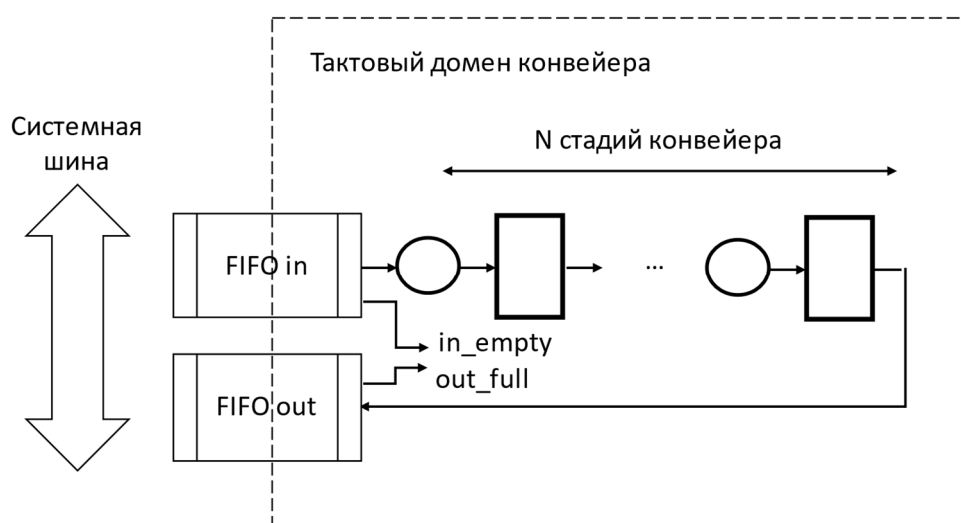


Рисунок 1 – Интеграция конвейерного ускорителя в вычислительную систему с применением FIFO

Figure 1 – Integration of pipeline accelerator into computing system using FIFO

Таким образом, при проектировании конвейера возникают две связанные задачи. Основными задачами являются распределение вычислительных операций по стадиям конвейера и балансировка задержек на отдельных стадиях, поскольку минимальный период тактового сигнала определяется наибольшей задержкой. Дополнительно возникает задача организации управления регистрами конвейера для корректной передачи данных между ними. Поскольку интеграция конвейерного вычислителя в систему производится с помощью входного и выходного FIFO, подразумевается, что конвейер работает с частотой, отличающейся от частоты системной шины, а обмен данными с ней производится нерегулярно. Поэтому сигналы готовности данных для передачи на первую стадию конвейера и готовности выходного FIFO к приему результатов обработки следует считать независимыми и непредсказуемо формирующимися.

Простейшим случаем является использование глобального для конвейера сигнала разрешения данных `clock enable`. Такой сигнал разрешает или блокирует запись результатов обработки в каждый из регистров конвейера, однако пригоден только для потоковой обработки, при которой гарантируются наличие входных данных и готовность выходного устройства к приему результатов.

При наличии запрещающих сигналов `in_empty` и `out_full`, которые препятствуют передаче данных по конвейеру, следует предусмотреть для каждой из n стадий конвейера сигнал разрешения записи `en`, передающийся между стадиями по мере записи данных в предыдущие стадии. Если сигнал достоверности данных на рассматриваемой стадии обозначить как `valid`, то условие продвижения данных по конвейеру будет выглядеть для каждой стадии следующим выражением на языке Verilog:

$$\text{assign } en[i+1] = \text{valid}[i] \ \& \ (\sim \text{out_full}),$$

где i – номер стадии конвейера

Для первой стадии конвейера условие записи:

$$\text{assign } en[0] = \sim \text{in_empty} \ \& \ (\sim \text{out_full}),$$

Условие записи данных в выходное FIFO:

$$\text{assign } \text{fifo_push} = en[nmax-1] \ \& \ (\sim \text{out_full}).$$

В представленной схеме продвижение данных по конвейеру возможно только при наличии свободного места в выходном FIFO. Однако возможна ситуация, когда при паузах готовности входных данных в конвейере будут присутствовать незаполненные стадии. При остановке конвейера эти стадии могли бы быть заполнены, однако этому препятствует глобальный сигнал запрета на продвижение данных. Для выявления возможности продвижения данных используется комбинация сигналов *valid/ready*, где сигнал *valid* обозначает наличие данных на предыдущей стадии конвейера, а *ready* – готовность следующей стадии к записи. Готовность определяется одним из двух условий.

1. Стадия свободна, т.е. не содержит актуальных данных.
2. Стадия занята, однако следующая за ней стадия готова к приему данных.

Проверка второго условия требует рекурсивной проверки всех стадий конвейера, вплоть до последней, записывающей данные в выходное FIFO. Обычно эта проверка формирует последовательную комбинационную цепочку, которая в длинном конвейере может стать критическим путем, ограничивающим максимальную тактовую частоту. Задержка в такой цепи пропорциональна количеству стадий конвейера, что требует не только оптимизации топологического представления, но и использования схемотехнических приемов по разделению комбинационных цепочек синхронными буферами (т.н. *skid-буфер*) [9, 10].

Схемотехническим способом снижения негативного влияния цепочки проверки готовности является эквивалентное преобразование логического выражения для каждой стадии, которое вместо последовательно соединенных сигналов *ready* объединяет сигналы *valid* для всех последующих стадий и *full* для выходного FIFO. Таким образом, вместо проверки готовности всех последующих стадий достаточно проверить, что хотя бы одна из последующих стадий свободна или FIFO готово к приему данных. Любое из этих условий означает, что цепочка данных впереди текущей стадии продвинется. Можно отметить, что это обеспечивается только в том случае, если все последующие стадии конвейера также реализуют эту логику проверки. На рисунке 2 показана электрическая схема формирования сигнала возможности продвижения данных.

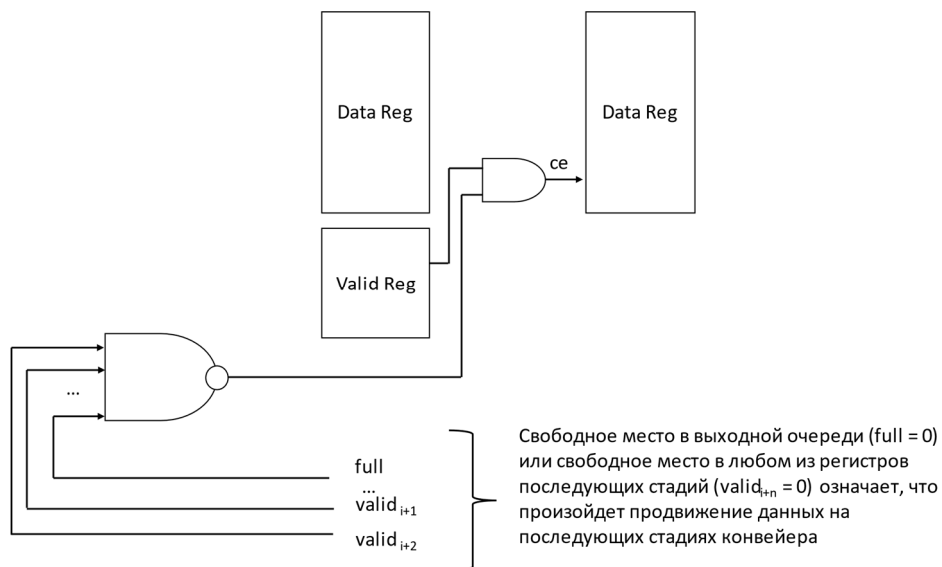


Рисунок 2 – Формирование сигнала возможности продвижения данных
Figure 2 – Generation of a signal for the possibility of data forwarding

Можно убедиться, что в показанной на рисунке 2 схеме отсутствуют последовательно соединенные логические вентили. Вместо этого увеличивается количество входов вентиль И-НЕ, который объединяет все сигналы *valid* последующих стадий и сигнал готовности выходного FIFO. Подобное решение также приводит к увеличению задержки распространения сигнала, поскольку многовходовые логические элементы, реализуемые на базе FPGA, приводят к

синтезу узлов из нескольких логических генераторов (LUT), что в целом зависит от семейства FPGA и дополнительных возможностей логических ячеек.

Результаты сравнительного моделирования вариантов конвейеризованного вычислительного устройства

Практический эффект от реализации усложненной схемы проверки готовности показан на рисунке 3. В тестовом воздействии имитируется пауза в записи данных, что показано сигналом `valid_in`, который принимает неактивное состояние на 5 тактов. После этого выходное FIFO переходит в состояние «заполнено», оставаясь в нем 10 тактов. Различие в поведении конвейеров иллюстрируется сигналами `ClockEnable` для конвейера с глобальным разрешением продвижения и `ValidReg` для конвейера с проверкой готовности продвижения данных на каждую стадию. Видно, что во втором случае продвижение данных продолжалось в течение тактов, когда FIFO было неактивно, поскольку из-за паузы в заполнении данных в конвейере оставались незаполненные стадии.

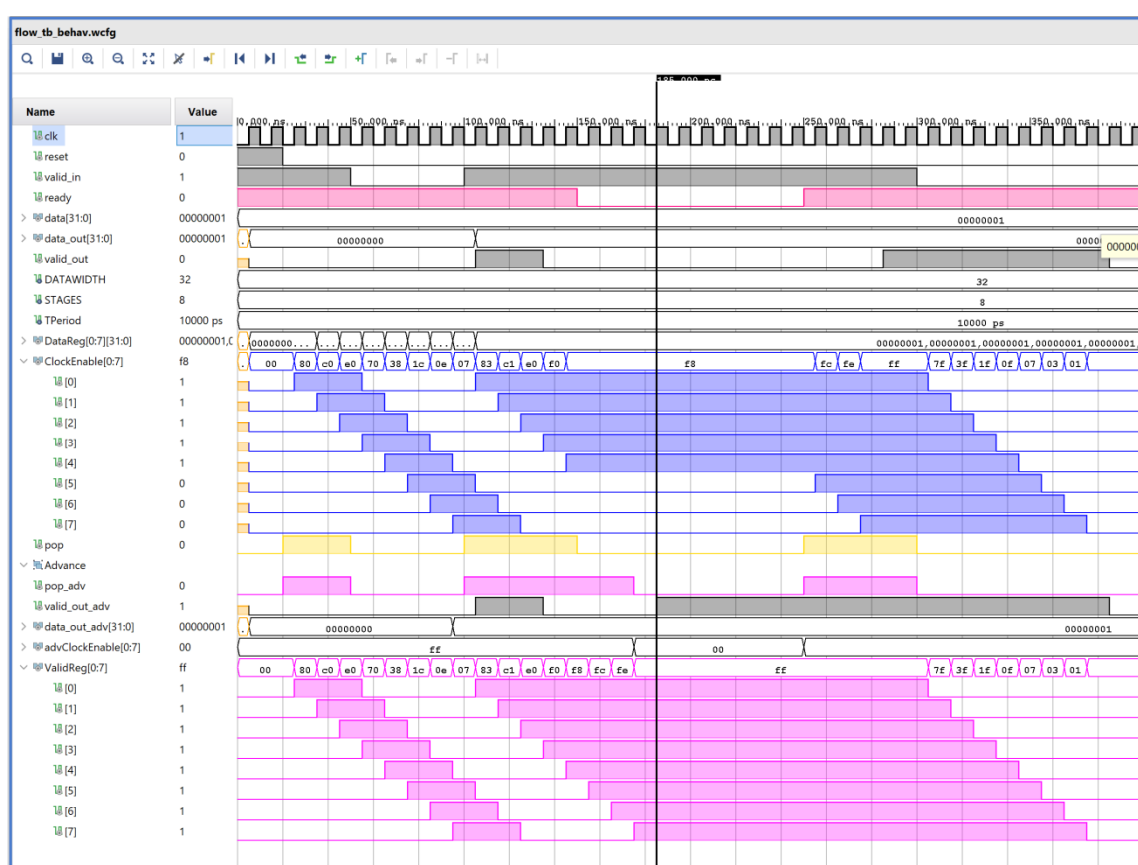


Рисунок 3 – Сравнительное моделирование поведения конвейера в зависимости от реализации продвижения данных в освободившиеся стадии
Figure 3 – Comparative modeling of pipeline behavior depending on the implementation of data forwarding to freed stages

Приведенное сравнение показывает, что усложненная схема проверки готовности стадий конвейера обеспечивает продвижение данных при условии, что:

- 1) выходное FIFO не готово к приему данных;
- 2) при заполнении конвейера из входного FIFO имелись паузы, которые привели к появлению незаполненных стадий конвейера.

Можно утверждать, что данная ситуация может быть исключена из практики, если в системе реализован достаточный объем FIFO или алгоритмы работы других подсистем предусматривают своевременный прием данных из выходного FIFO, устраняя ситуации, при которых оно могло бы блокировать продвижение данных.

Кроме того, если тактовая частота конвейера выше, чем частота чтения данных из выходного FIFO, факт дополнительного простоя конвейера не приведет к паузе в чтении данных из FIFO, поскольку более высокая частота конвейера позволит своевременно продвинуть данные до последней стадии, пока производится чтение из головы FIFO.

Таким образом, при выборе архитектуры управления конвейером следует учитывать как особенности его топологической реализации, так и системные аспекты проектируемого вычислительного устройства. В данной работе предлагается использовать схемы управления с более простой топологической реализацией, компенсируя их недостатки в продвижении данных за счет узлов и алгоритмов системной интеграции – применения FIFO достаточной разрядности и алгоритмов потокового заполнения входного и чтения данных из выходного FIFO.

В процессе проектирования конвейерного вычислителя регулируемые параметрами являются:

- тип схемы управления;
- количество стадий;
- глубина FIFO.

Поскольку точные значения получаемых характеристик зависят от используемой аппаратной платформы и особенностей размещения конвейера на кристалле, сложно указать адекватные аналитические выражения для определения этих характеристик. Вместо этого следует использовать метод оптимизации в дискретном пространстве параметров, так как качественное влияние перечисленных факторов является прогнозируемым.

Экспериментальные исследования топологической реализации вариантов конвейеризованного вычислительного устройства

Для экспериментальной проверки предположений, сформулированных в ходе исследования, был реализован конвейерный вычислительный модуль, использующий перечисленные в статье схемы управления. Измерения проводились для трёх выше рассмотренных вариантов построения схемы управления конвейером в ПЛИС XC7VX1140TFLG1930-1 серии Virtex-7 компании Xilinx/AMD. Расчет производится в автоматическом режиме, при этом сбор данных о результатах изменений производился программным модулем на языке Tcl.

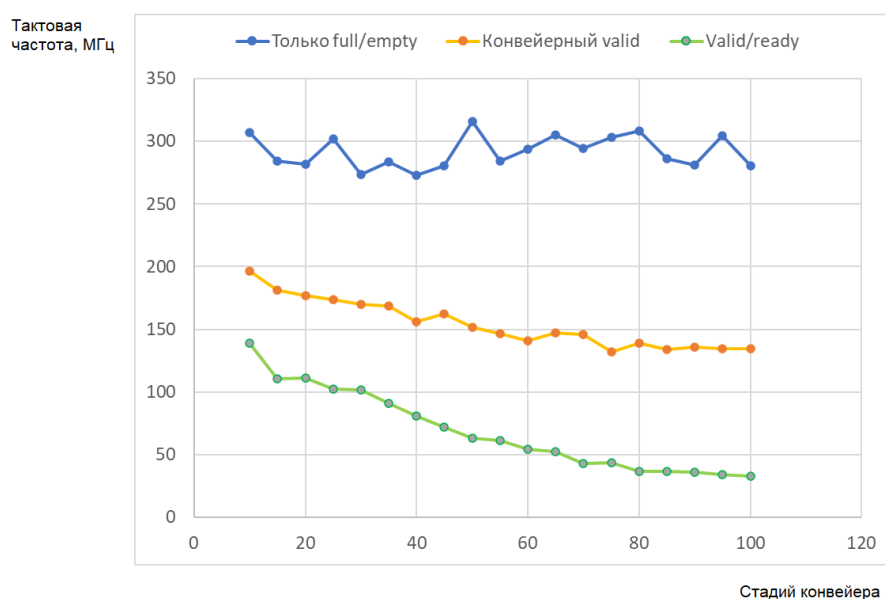


Рисунок 4 – Зависимость тактовой частоты конвейерного вычислителей от количества стадий конвейера при различных схемах управления продвижением данных
Figure 4 – Dependence of clock frequency of pipeline computers on the number of pipeline stages for various data flow control schemes

Рисунок 4 наглядно демонстрирует, что при глобальном управлении продвижением данных снижение частоты при увеличении длины конвейера практически не наблюдается, а вариации тактовой частоты обусловлены изменениями топологии трассировочных линий, что для больших тактовых частот приводит к показанным вариациям при отсутствии топологических проектных ограничений (area constraints). Наличие дополнительных схем, управляющих продвижением данных по конвейеру, приводит к снижению тактовой частоты из-за увеличения длины проводников, передающих управляющие сигналы. При этом снижение частоты ярко выражено для схемы с продвижением данных по мере освобождения стадий, что связано с увеличением количества входов в логических вентилях, формирующих сигнал разрешения записи в начальные стадии конвейера, которые проверяют состояние всех последующих стадий.

Конкретные численные значения параметров зависят от семейства ПЛИС, однако качественный вид зависимостей сохраняется. Для разработанного модуля не требуется адаптация RTL-описания к семейству ПЛИС, что позволяет использовать его для предварительной оценки характеристик топологического представления в процессе проектирования конвейерных ускорителей вычислений.

Заключение

Рассмотрены архитектуры конвейерных ускорителей для работы в составе вычислительных систем. Полученные результаты показывают, что усложнение логики работы системы управления продвижением данных упрощает алгоритмы обмена данными с ускорителем, однако приводит к существенному снижению тактовой частоты конвейера по мере увеличения количества стадий. Вместе с этим увеличение количества стадий конвейера приводит к повышению достижимой тактовой частоты для тракта данных. Это формирует задачу оптимизации конвейерного ускорителя в дискретном пространстве параметров, набор которых предложен в работе.

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема № FSFZ-2022-0004 Архитектуры специализированных вычислительных комплексов, методики, алгоритмы и инструменты проектирования цифровых вычислительных устройств).

Библиографический список

1. **Rowen C.** Engineering the Complex Soc: Fast, Flexible Design with Configurable Processors. Prentice Hall, 2004. ISBN 0-13-145537-0.
2. **Hennessy J.L., Patterson D.A.**: A new golden age for computer architecture: Domain-specific hardware/software co-design, enhanced security, open instruction sets, and agile chip development. In Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), Los Angeles, CA, USA, 1–6 (June 2018). doi:10.1109/ISCA.2018.00011.
3. **Kanduri, Anil & Rahmani, Amir M. & Liljeberg, Pasi & Hemani, Ahmed & Jantsch, Axel & Tenhunen, Hannu.** (2017). A Perspective on Dark Silicon. 10.1007/978-3-319-31596-6_1.
4. **Esmailzadeh Hadi & Blem Emily & Amant Renée & Sankaralingam Karthikeyan & Burger Doug.** Dark Silicon and the End of Multicore Scaling // ACM Sigarch Computer Architecture News. 2011. 32. pp. 365-376. 10.1145/2024723.2000108
5. **Hennessy J.L., Patterson D.A.** 2017 Computer Architecture. 6th Edition. A Quantitative Approach (The Morgan Kaufmann Series in Computer Architecture and Design) 936 p
6. **Tarasov I.E.**: Architecture of a Specialized VLSI for Serial Digital Signal Processing. In Proceedings: 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency, SUMMA 2021, 3, pp. 457-460 (2021)
7. **Varma R. & Subbarao M. & Varma D. Ramesh & Raju G.** High-Throughput VLSI Architectures for VLSI Signal Processing. 2021. 10.1007/978-981-15-3828-5_37.
8. **Tarasov I.E., Potekhin D.S.** Real-time kernel function synthesis for software defined radio and phase-frequency measuring digital systems. Russ. Technol. J. 2018, 6, 41–54, doi:10.32362/2500-316X-2018-6-6-41-54. (In Russian)

9. http://fpgacpu.ca/fpga/Pipeline_Skid_Buffer.html (дата обращения 10/10/2023)

10. <https://chipmunklogic.com/digital-logic-design/designing-skid-buffers-for-pipelines/> (дата обращения 10/10/2023)

UDC 004.31

DESIGN OF PIPELINE COMPUTING DEVICES CONSIDERING TOPOLOGICAL REPRESENTATION

I. E. Tarasov, Dr. in technical sciences, Head of the Laboratory, RTU MIREA, Moscow, Russia;
orcid.org/0000-0001-6456-4794, e-mail: tarasov_i@mirea.ru

D. V. Lulyava, junior researcher, RTU MIREA, Moscow, Russia;
orcid.org/0009-0009-9623-7777, e-mail: lyulyava@mirea.ru

N. A. Duksin, engineer, RTU MIREA, Moscow, Russia;
orcid.org/0009-0009-0014-7065, e-mail: duksin@mirea.ru

The problem of designing a pipeline computer to operate as part of a digital computing system is considered. The aim is to study the influence of pipeline computer architecture on the characteristics of its topological representation in order to optimize a computer according to selected optimality criteria. When designing high-performance computing systems, an important stage is the architectural design and decomposition of the system. In this case, the choice of operations for hardware acceleration depends on the characteristics of the accelerator obtained on existing hardware platform. Design at various levels: system, circuit and topological allows, on the one hand, to abstract from implementation details and increase development productivity, but on the other hand, the transition to implementation details clarifies the final characteristics of the designed accelerator for a computing system and can significantly worsen them relative to preliminary expectations. The article discusses a subclass of pipeline computing accelerators and the dependence of their size characteristics, signal propagation delay and power consumption on control system architecture. The possibility of partial compensation for the shortcomings of architectures that have simple topological implementation has been identified, at the expense of nodes that ensure pipeline integration into computing system. The authors propose to use optimization in discrete parameter space, for which a parameterized pipeline description is used.

Keywords: computing system, pipeline, architecture, FPGA, VLSI, register transfer level.

DOI: 10.21667/1995-4565-2023-86-86-95

References

1. **Rowen C.** *Engineering the Complex Soc: Fast, Flexible Design with Configurable Processors*. Prentice Hall, 2004. ISBN 0-13-145537-0.
2. **Hennessy J.L., Patterson D.A.** A new golden age for computer architecture: Domain-specific hardware/software co-design, enhanced security, open instruction sets, and agile chip development. In *Proceedings of the 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, Los Angeles, CA, USA, 1-6 (June 2018). doi:10.1109/ISCA.2018.00011.
3. **Kanduri, Anil & Rahmani, Amir M. & Liljeberg, Pasi & Hemani, Ahmed & Jantsch, Axel & Tenhunen, Hannu.** (2017). A Perspective on Dark Silicon. 10.1007/978-3-319-31596-6_1.
4. **Esmailzadeh Hadi & Blem Emily & Amant Renée & Sankaralingam Karthikeyan & Burger Doug.** Dark Silicon and the End of Multicore Scaling. *ACM Sigarch Computer Architecture News*. 2011, 32, pp.365-376. 10.1145/2024723.2000108
5. **Hennessy J. L., Patterson D.A.** *2017 Computer Architecture*. 6th Edition. A Quantitative Approach (The Morgan Kaufmann Series in Computer Architecture and Design) 936 p.
6. **Tarasov I.E.:** Architecture of a Specialized VLSI for Serial Digital Signal Processing. In *Proceedings: 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency*, SUMMA 2021, 3, pp. 457-460 (2021).

7. **Varma R. & Subbarao M. & Varma D. Ramesh & Raju G.** *High-Throughput VLSI Architectures for VLSI Signal Processing*. 2021. 10.1007/978-981-15-3828-5_37.
8. **Tarasov I.E., Potekhin D.S.** Real-time kernel function synthesis for software defined radio and phase-frequency measuring digital systems. *Russ. Technol. J.* 2018, 6, 41–54, doi:10.32362/2500-316X-2018-6-6-41-54 (In Russian).
9. http://fpgacpu.ca/fpga/Pipeline_Skid_Buffer.html (data obrashcheniya 10/10/2023)
10. <https://chipmunklogic.com/digital-logic-design/designing-skid-buffers-for-pipelines/> (data obrashcheniya 10/10/2023)