

ПЕРСПЕКТИВНЫЕ ПРОЦЕССОРНЫЕ ЯДРА ДЛЯ ПРИМЕНЕНИЯ В СОСТАВЕ СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЕЙ С МАССОВО-ПАРАЛЛЕЛЬНОЙ АРХИТЕКТУРОЙ НА БАЗЕ ПЛИС

Тарасов И.Е., Дуксин Н.А., Дуксина И.И.

Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА - Российский технологический университет» (РТУ МИРЭА), 119454 Россия, г. Москва, проспект Вернадского, дом 78; e-mail: tarasov_i@mirea.ru, duksin@mirea.ru, duksina@mirea.ru

В статье рассматриваются микроархитектуры процессорных ядер, предназначенных для синтеза в программируемых интегральных схемах и применения в составе гетерогенных массово-параллельных вычислительных систем на кристалле. Актуальность работы обусловлена ограничениями дальнейшего роста производительности процессоров общего назначения и необходимостью перехода к специализированным вычислительным архитектурам, оптимизированным под конкретные классы алгоритмов. Целью исследования является анализ перспективных архитектур, синтезируемых управляющих процессорных ядер и оценка влияния их микроархитектурных решений на временные и ресурсные характеристики при использовании в составе массово-параллельных вычислительных систем. В рамках работы рассматриваются требования, предъявляемые к процессорным ядрам вычислительных кластеров, и показаны причины неэффективности традиционных процессоров общего назначения в условиях многократного тиражирования на кристалле. В теоретической части исследуется процессорные архитектуры с минимальной глубиной конвейера, реализующие схемы «выборка-исполнение» и «выборка-декодирование-исполнение». Рассматривается применение механизмов продвижения данных в тракте обработки данных и потоке управления, а также анализируется их влияние на формирование критических путей и задержки при нарушении линейного порядка выполнения команд. Экспериментальные исследования выполнены с использованием системы автоматизированного проектирования Vivado для программируемых логических интегральных схем (ПЛИС) семейств Artix-7 и Virtex Ultrascale+. Проведено сравнение одноктактных, двухтактных и трёхтактных архитектур с bypass и без него. В качестве основной метрики использовалась максимальная достижимая тактовая частота, определяемая методом статически временного анализа. Полученные результаты показывают, что увеличение количества стадий конвейера оказывает более существенное влияние на частотные характеристики процессорного ядра по сравнению с использованием схем продвижения данных, влияние которых ограничено особенностями рассматриваемого класса управляющих процессоров. Результаты исследования могут быть использованы при проектировании специализированных и гетерогенных вычислительных систем на базе программируемых интегральных схем, а также при разработке процессорных ядер для массово-параллельных вычислителей.

Ключевые слова: вычислительная система, процессор, архитектура, ПЛИС, моделирование, продвижение данных, микроархитектура, массово-параллельные вычисления.

DESIGN OF PIPELINED COMPUTING DEVICES CONSIDERING THE TOPOLOGICAL REPRESENTATION

Tarasov I.E., Duksin N.A., Duksina I.I.

The Federal State Budget Educational Institution of Higher Education "MIREA - Russian Technological University" (RTU MIREA), 78, Vernadsky prospect, Moscow, 119454, Russia; e-mail: tarasov_i@mirea.ru, duksin@mirea.ru, duksina@mirea.ru

This paper investigates microarchitectures of processor cores intended for synthesis on field-programmable gate array (FPGAs) and for use within heterogeneous massively parallel computing systems on chip. The relevance of this work is determined by the limitations of further performance scaling of general-purpose processors and the necessity of transitioning toward specialized computing architectures optimized for specific classes of algorithms. The aim of the study is to analyze promising architectures of synthesizable control processor cores and to evaluate the impact of their microarchitecture design decisions on timing and resource characteristics when employed as components of massively parallel computing systems. The paper considers the requirements imposed on processor cores operating within computing clusters and identifies the reason for the inefficiency of conventional general-purpose processors under conditions of large-scale replication on a single chip. The theoretical section examines

processor architectures with minimal pipeline depth implementing “fetch-execute” and “fetch-decode-execute” schemes. The application of data forwarding mechanisms in both the datapath and the control flow path is analyzed, and their influence on critical path formation and latency caused by violations of linear instruction execution order is discussed. Experimental studies were conducted using the Vivado computer-aided design environment for FPGA devices of the Artix-7 and Virtex UltraScale+ families. A comparative evaluation of single-cycle, two-stage and three-stage pipeline architectures with and without bypass mechanisms was performed. The primary evaluation metric was the maximum achievable clock frequency determined through static timing analysis. The obtained results demonstrate that increasing the number of pipeline stages has a significantly greater impact on the achievable frequency characteristics of the processor core than the use of data forwarding schemes, whose influence is limited by the specific features of the considered class of control processors. The results demonstrate that increasing the number of pipeline stages has a significantly greater impact on the achievable frequency characteristics of the processor core than the use of data forwarding schemes, whose influence is limited by the specific features of the considered class of control processors. The results of this study can be applied in the design of specialized and heterogeneous FPGA-based computing systems, as well as in the development of processor cores for massively parallel computing architectures.

Keywords: computing system, pipeline, architecture, FPGA, bypass, parallel computing.

Введение

Разработка специализированных вычислительных комплексов включает в себя этапы декомпозиции аппаратного обеспечения с выделением подсистем, предназначенных для ускорения вычислений определенных типов. Развитие архитектур процессоров общего назначения было ограничено на рубеже 1990-2000-х годов технологическими ограничениями полупроводниковой технологии, и позже, около 2010 года, законом Амдала, что в совокупности привело к прекращению значимого роста производительности, определяемой экстенсивным улучшением технических характеристик цифровых компонентов и их количеством. На этом этапе получили распространение ускорители вычислений на базе графических процессоров в режиме выполнения вычислений общего назначения (GPGPU – General-Purpose on GPU), а также специализированные вычислительные устройства с оригинальной архитектурой, реализуемые на базе программируемых логических интегральных схем (ПЛИС).

Специализированные устройства на базе ПЛИС целесообразно применять в условиях, когда ни процессоры общего назначения, ни GPU не отвечают в полной мере требованиям к производительности ввиду несоответствия их аппаратной архитектуры общей структуре алгоритмов. Например, GPU, несмотря на большое количество вычислительных узлов, производит вычисления по модели SIMD (Single Instruction, Multiple Data), т.е. использует один и тот же поток команд для всех элементов вычислительного кластера. В этой связи для ПЛИС представляется перспективной архитектура класса MIMD (Multiple Instruction, Multiple Data) по классификации Флинна [1]. Кроме собственно вычислительных ядер, в состав системы могут входить управляющие процессоры и процессоры для управления периферийными устройствами. Таким образом, специализированный вычислитель на базе ПЛИС имеет в своем составе разнородные процессоры, ориентированные на выполнение алгоритмов соответствующих им типов.

В данной статье рассматриваются архитектуры процессорных ядер, которые могут являться основой гетерогенных вычислительных систем с массовым параллелизмом операций на кристалле.

Теоретическая часть

К процессорам, работающим в составе массово-параллельной вычислительной системы, предъявляются требования, отличающиеся от требований к процессорам общего назначения. Можно перечислить ряд причин, обуславливающих это:

1. Избыточность ресурсов, обеспечивающая процессору общего назначения дополнительные функциональные возможности при решении заранее неопределенного круга задач, приводит к неоправданным расходам аппаратных ресурсов при многократном тиражировании по полупроводниковому кристаллу компонентов, которые заведомо не будут применяться для выбранного целевого подкласса задач.

2. Иерархическая система доступа к памяти, характерная для процессора общего назначения, должна использоваться не для процессорных элементов вычислительного кластера, а для всего кластера в целом (например, с доступом через выделенный управляющий процессор). Это связано с ограниченными возможностями доступа к памяти через внешние интерфейсы.

3. Развитая система прерываний и сложная шинная архитектура характерна для организации доступа к набору периферийных устройств, в том числе к заранее неизвестному перечню контроллеров, интегрируемых на полупроводниковый кристалл для совместной работы с процессорным ядром. Поэтому процессорные ядра для микроконтроллеров и одноплатных компьютеров часто используют системные шины с поддержкой ожидания

устройств, арбитража, преобразований интерфейса с помощью мостов (bridge) и т.п. Эти возможности избыточны для процессорного ядра, работающего в составе матрицы вычислительных элементов с заранее известным и небольшим набором периферийных устройств.

Для распределения ресурсов кластера при выполнении вычислительных задач необходимо производить мониторинг состояния отдельных процессорных элементов, своевременную загрузку программ и служебных данных, а также обеспечивать обмен данными между источником (например, процессором общего назначения) и кластером. Поскольку эти задачи существенно отличаются от типичных задач, подлежащих ускорению, для их решения целесообразно разработать отдельное управляющее ядро. Архитектура вычислительного кластера гетерогенной вычислительной системы с массовым параллелизмом операций показана на рис. 1.

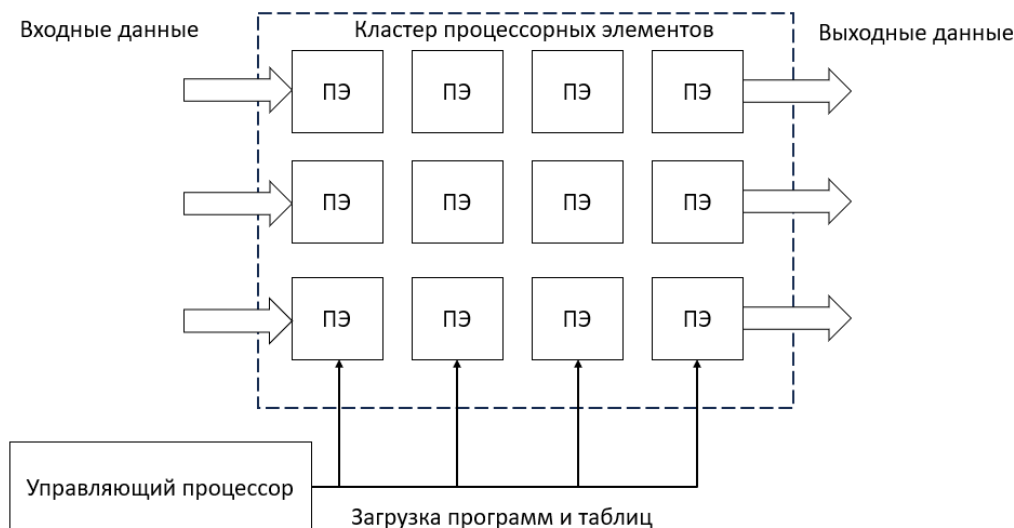


Рисунок 1 – Архитектура вычислительного кластера гетерогенной вычислительной системы с массовым параллелизмом операций в независимых процессорных элементах (ПЭ)

Архитектуры процессорных ядер для работы в составе вычислительных кластеров

Для процессорных ядер, работающих в составе ПЛИС, можно рассматривать несложные архитектуры, ограничивающиеся схемами «выборка-исполнение» или «выборка-декодирование-исполнение». Дополнительно в простых архитектурах применимо продвижение данных, как для тракта обработки данных, так и для потока управления.

На рисунке 2 показана архитектура простого процессорного ядра, содержащего блок регистров Reg, арифметико-логическое устройство АЛУ в тракте данных, а также тракт управления, содержащий счетчик команд PC (Program Counter) и память программ.

В схеме, показанной на рис. 2, отсутствуют зависимости по данным или структурные конфликты в тракте обработки данных, поскольку обновление регистров данных происходит по одноктактной схеме. В то же время, для тракта управления наблюдается задержка в чтении команды в один такт при нарушении линейного порядка выполнения программы. Это связано с тем, что вычисление нового значения счетчика команд происходит одновременно с вычислением нового значения для регистра назначения в тракте данных. После записи нового значения в счетчик команд потребуется еще один такт для чтения команды с нового адреса. Такое поведение характерно для процессоров, использующих конвейеризацию, даже при минимальной глубине конвейера, и решением проблемы является пропуск такта с целью обновления команды или введение «слота задержки» (delay slot), в виде выполнения команды за адресом, с которого произошел переход.

Альтернативным вариантом является введение схемы продвижения данных для потока управления. В этом случае, как показано на рис. 3, адресом памяти программ является не выход регистра счетчика команд, а новое значение, которое должно быть записано в этот регистр по фронту ближайшего тактового сигнала.

При использовании показанной схемы тактовая частота несущественно уменьшается из-за невозможности перераспределить задержку распространения сигнала путем перемещения регистра PC по кристаллу. Вместе с этим наблюдается устранение штрафов из-за нарушения линейного порядка выполнения команд.

Экспериментальные исследования

Характеристики процессорных ядер с рассмотренными архитектурами были оценены в САПР Vivado для ПЛИС семейства Artix-7.

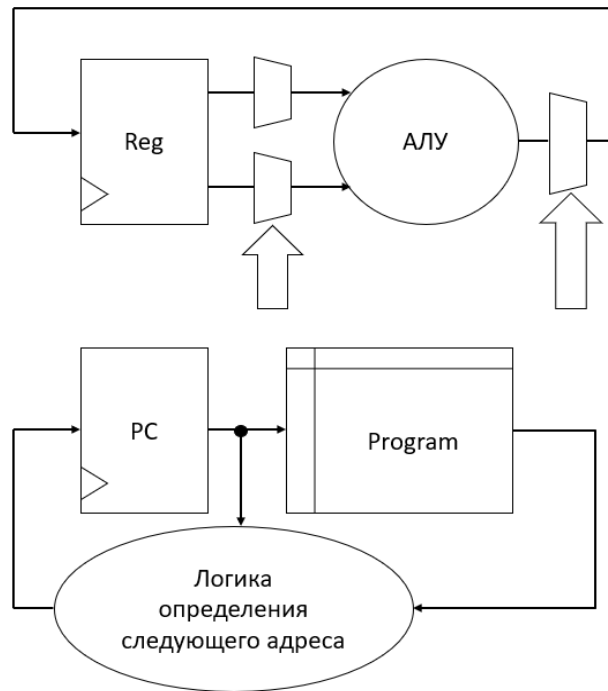


Рисунок 2 – Архитектура простого процессорного ядра

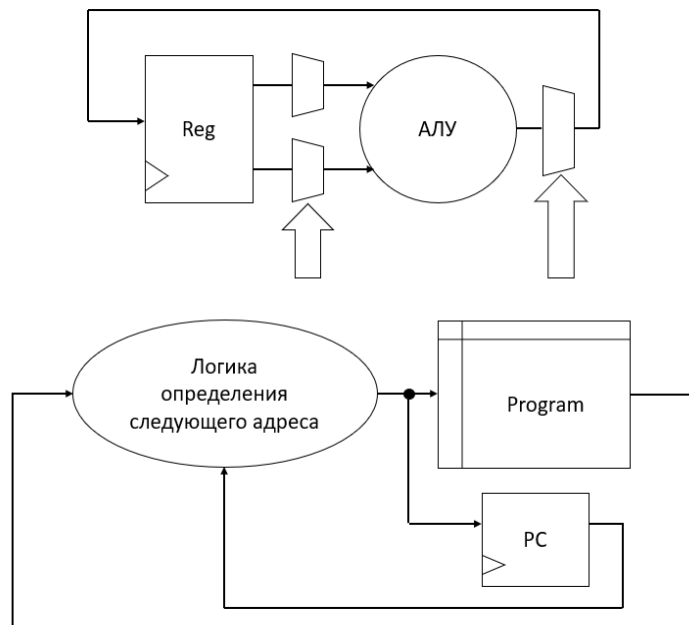


Рисунок 3 – Иллюстрация к вопросу продвижения данных в потоке управления

Для повышения репрезентативности результатов эксперименты выполнены на ПЛИС, реализованных по различным технологическим нормам. В качестве представителя 28-нм техпроцесса использовано семейство Xilinx Artix-7, тогда как семейство Virtex UltraScale+ применялось как представитель более современного техпроцесса в рамках поколения UltraScale+, выполненного по технологической норме 16-нм FinFET.

Экспериментальное исследование направлено на количественную оценку влияния стадийности конвейера и наличия трактов продвижения данных на предельную тактовую частоту синтезируемого процессорного ядра. В рамках работы проведён ряд сравнительных экспериментов, включающих анализ трёхтактной конвейерной архитектуры с реализацией продвижения данных и без него, а также сопоставление одноктактной архитектуры, с применением механизма bypass, с двухтактной архитектурой, разделённой на стадии выборки инструкции (fetch) и исполнения (execute).

В качестве основной метрики использовалась максимальная достижимая тактовая частота, определяемая по

результатам статического временного анализа после завершения этапа трассировки. Результаты экспериментов были усреднены в силу применения алгоритма имитации отжига при размещении и трассировки. Результаты экспериментов представлены на рисунке 4.

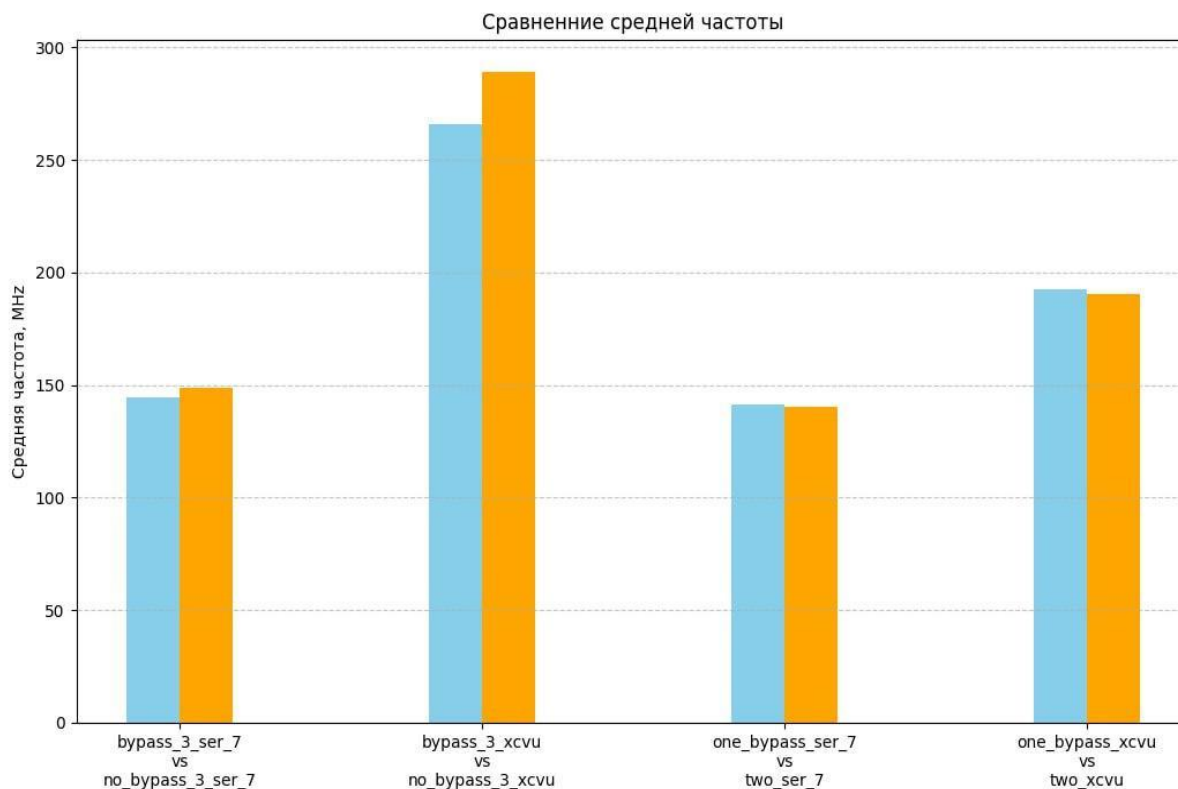


Рисунок 4 – Экспериментальные сравнения средней частоты

На графике отображено усреднённое значение тактовой частоты для каждого из исследуемых вариантов архитектуры. Анализ полученных данных показывает, что использование продвижения данных в трёхтактной архитектуре в ряде случаев приводит к незначительному снижению предельной тактовой частоты за счёт увеличения длины критических путей. Вместе с тем, в пределах статистической погрешности, обусловленной стохастическим характером алгоритмов размещения и имитации отжига, наблюдаются случаи, когда реализация с продвижением данных демонстрирует сопоставимые либо незначительно лучшие значения тактовой частоты.

Существенно больший вклад в изменение частотных характеристик вносит количество стадий. Переход к трёхтактной конвейерной архитектуре обеспечивает сокращение критической задержки на каждой из стадий, что неминуемо приводит к повышению максимально достижимой тактовой частоты вне зависимости от наличия продвижения данных.

Следует отметить, что эксперименты проводились на системе команд, не предполагающей выполнения сложных арифметических операций, что характерно для управляющих процессорных ядер. Ввиду этого интенсивность возникновения конвейерных зависимостей ограничена, а влияние схем продвижения данных на временные характеристики архитектуры дополнительно снижается.

Заключение

В статье рассмотрены архитектуры процессорных ядер, предназначенных для работы в составе ПЛИС в качестве независимо программируемых процессорных элементов вычислительного кластера, а также в качестве управляющих процессоров такого кластера. Вместо подхода, направленного на увеличение количества стадий конвейера с его усложнением, рассмотрены варианты простых процессорных архитектур, достоинством которых является отсутствие зависимостей по данным и следующая из этого предсказуемость потактового исполнения команд. Проведенные экспериментальные исследования продемонстрировали влияние аппаратной архитектуры процессорного ядра на достижимые значения тактовой частоты.

Список литературы

1. 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
2. Jungeblut, Thorsten & Puttmann, Christoph & Dreesen, R. & Pormann, Mario & Thies, M. & Kastens, U. (2010). Resource efficiency of hardware extensions of a 4-issue VLIW processor for elliptic curve cryptography. *Advances in Radio Science*. 8. 295-305. 10.5194/ars-8-295-2010. <https://clck.ru/3Sr67V>
3. Yan, Cheah & Fahmy, Suhaib & Kapre, Nachiket. (2015). On Data Forwarding in Deeply Pipelined Soft Processors. 181-189. 10.1145/2684746.2689067.
4. Lev Denisov, Andrea Galimberti, Daniele Cattaneo, Giovanni Agosta, Davide Zoni, Design-time methodology for optimizing mixed-precision CPU architectures on FPGA, *Journal of Systems Architecture*, Volume 155, 2024, 103257, ISSN 1383-7621, <https://doi.org/10.1016/j.sysarc.2024.103257>.
5. Дуксин, Н. А. Управление реконфигурируемой кластерной системой программируемой логики / Н. А. Дуксин // *Высокопроизводительные вычислительные системы и технологии*. – 2025. – Т. 9, № 1. – С. 44-49. – EDN IEWFVS.
6. Щепухин, Д. О. Вопросы конфигурирования массово-параллельной гетерогенной вычислительной системы / Д. О. Щепухин // *Высокопроизводительные вычислительные системы и технологии*. – 2025. – Т. 9, № 1. – С. 57-61. – EDN XHCRAR.
7. Дуксин, Н. А. Проектирование вычислительных кластеров программируемой логики / Н. А. Дуксин // *Вестник Рязанского государственного радиотехнического университета*. – 2025. – № 92. – С. 92-99. – DOI 10.21667/1995-4565-2025-92-92-99. – EDN MEUDYC.
8. Fog A. Modern processor design: Fundamentals of superscalar processors. — CreateSpace Independent Publishing Platform, 2016. — 793 p. — ISBN 978-8793237999.
9. Богачёв К. Ю. Архитектура процессоров : учебные материалы [Электронный ресурс]. — М., 2019.
10. Цилькер Б. Я., Орлов С. А. Организация ЭВМ и систем : учебник для вузов. — 5-е изд., перераб. и доп. — СПб. : Питер, 2019. — 688 с. — ISBN 978-5-4461-1064-2.
11. A. A. Ab Rahman, A. N. Prihozhy, and M. Mattavelli, "Pipeline synthesis and optimization of FPGA-based video processing applications with CAL," *Journal of Signal Processing Systems*, vol. 63, no. 2, pp. 231–244, 2011. doi: 10.1186/1687-5281-2011-19.
12. Yi Q., Li H., Zhang Y., Wang K. FPGA-based accelerator for neural networks computation with flexible pipeline architecture // *arXiv.org*. 2021.
- 13.7 Series FPGAs Clocking Resources User Guide (UG472). URL: https://docs.amd.com/v/u/en-US/ug472_7Series_Clocking (Дата обращения: 28.01.2026).
14. AC701 Evaluation Board for the Artix-7 FPGA User Guide (UG952). URL: <https://docs.amd.com/v/u/en-US/ug952-ac701-a7-eval-bd> (Дата обращения: 28.01.2026).
15. Vivado Design Suite User Guide: Dynamic Function eXchange (UG909). URL: <https://docs.amd.com/t/en-US/ug909-vivado-partial-reconfiguration/Introduction-to-Dynamic-Function-eXchange> (Дата обращения: 28.01.2026).

References

1. 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
2. Jungeblut, Thorsten & Puttmann, Christoph & Dreesen, R. & Pormann, Mario & Thies, M. & Kastens, U. (2010). Resource efficiency of hardware extensions of a 4-issue VLIW processor for elliptic curve cryptography. *Advances in Radio Science*. 8. 295-305. 10.5194/ars-8-295-2010. <https://clck.ru/3Sr67V>
3. Yan, Cheah & Fahmy, Suhaib & Kapre, Nachiket. (2015). On Data Forwarding in Deeply Pipelined Soft Processors. 181-189. 10.1145/2684746.2689067.
4. Lev Denisov, Andrea Galimberti, Daniele Cattaneo, Giovanni Agosta, Davide Zoni, Design-time methodology for optimizing mixed-precision CPU architectures on FPGA, *Journal of Systems Architecture*, Volume 155, 2024, 103257, ISSN 1383-7621, <https://doi.org/10.1016/j.sysarc.2024.103257>.
5. .Duksin, N. A. Upravlenie rekonfiguriruemoy klasternoj sistemoj programmiruemoj logiki / N. A. Duksin // *Vysokoproizvoditel'nye vychislitel'nye sistemy i tekhnologii*. – 2025. – Т. 9, № 1. – С. 44-49. – EDN IEWFVS.
6. .Shchepuhin, D. O. Voprosy konfigurirovaniya massovo-parallel'noj geterogennoj vychislitel'noj sistemy / D. O. SHCHepuhin // *Vysokoproizvoditel'nye vychislitel'nye sistemy i tekhnologii*. – 2025. – Т. 9, № 1. – С. 57-61. – EDN XHCRAR
7. Duksin, N. A. Proektirovanie vychislitel'nyh klasterov programmiruemoj logiki / N. A. Duksin // *Vestnik*

Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta. – 2025. – № 92. – S. 92-99. – DOI 10.21667/1995-4565-2025-92-92-99. – EDN MEUDYC.

8. . Fog A. Modern processor design: Fundamentals of superscalar processors. — CreateSpace Independent Publishing Platform, 2016. — 793 p. — ISBN 978-8793237999.

9. Bogachev, K. Yu. *Processor Architecture: Teaching Materials*. Moscow, 2019.

10. Tsilker, B. Ya., Orlov, S. A. *Computer Organization and Systems*. 5th ed. Saint Petersburg: Piter Publishing, 2019. 688 p.

11. A. A. Ab Rahman, A. N. Prihozhy, and M. Mattavelli, “Pipeline synthesis and optimization of FPGA-based video processing applications with CAL,” *Journal of Signal Processing Systems*, vol. 63, no. 2, pp. 231–244, 2011. doi: 10.1186/1687-5281-2011-19.

12. Yi Q., Li H., Zhang Y., Wang K. FPGA-based accelerator for neural networks computation with flexible pipeline architecture // arXiv.org. 2021.

13.7 Series FPGAs Clocking Resources User Guide (UG472). URL: https://docs.amd.com/v/u/en-US/ug472_7Series_Clocking (accessed: 28.01.2026).

14. AC701 Evaluation Board for the Artix-7 FPGA User Guide (UG952). URL: <https://docs.amd.com/v/u/en-US/ug952-ac701-a7-eval-bd> (accessed: 28.01.2026).

15. Vivado Design Suite User Guide: Dynamic Function eXchange (UG909). URL: <https://docs.amd.com/r/en-US/ug909-vivado-partial-reconfiguration/Introduction-to-Dynamic-Function-eXchange> (accessed: 28.01.2026).