

Высокопроизводительный софт-процессор для встраиваемых систем на основе ПЛИС

Е.С. Барков, А.Л. Переверзев, А.М. Силантьев

Национальный исследовательский университет «МИЭТ», г. Москва, 8160209@edu.miet.ru

Аннотация — В статье рассмотрены ограничивающие производительность недостатки микроархитектур популярных 32-разрядных софт-процессоров для встраиваемых систем на основе ПЛИС. Предложена микроархитектура софт-процессора на базе архитектуры RISC-V, лишенная рассмотренных недостатков. Проведена верификация и оценка аппаратных характеристик разработанного софт-процессора. Разработанный софт-процессор имеет на 17% большую производительность по сравнению с рассмотренными решениями.

Ключевые слова — софт-процессор, RISC-V, ПЛИС, FPGA.

I. ВВЕДЕНИЕ

Часто в устройствах на основе ПЛИС применяют микропроцессоры, это обстоятельство привело к появлению систем на программируемом кристалле (СнПК), объединяющем микропроцессор и ПЛИС. Микропроцессор в СнПК может быть реализован на физическом (топологическом) уровне в виде Hard-IP или на программируемой логике ПЛИС в виде Soft-IP, или софт-процессора.

Построение СнПК на базе софт-процессора позволяет сэкономить на конечной стоимости изделия, поскольку такая система может быть реализована на ПЛИС без микропроцессора в виде Hard-IP. Дополнительным преимуществом такого подхода является гибкость в построении системы: на одной

ПЛИС можно разместить настолько много софт-процессоров, насколько позволяет логическая ёмкость ПЛИС; количество же Hard-IP микропроцессоров не подлежит изменению. Однако софт-процессоры имеют на порядок худшую производительность по сравнению с процессорами, выполненными на физическом уровне. Создание высокопроизводительного софт-процессора является актуальной задачей, поскольку позволит улучшить производительность СнПК, построенных на относительно дешёвых ПЛИС.

В настоящей работе приведен анализ ограничивающих производительность недостатков микроархитектур существующих процессоров для ПЛИС, и предложена микроархитектура, лишенная этих недостатков.

Описаны ключевые микроархитектурные особенности и структура процессора, воплощающая предложенную микроархитектуру. В статье приводятся схема верификационного окружения процессора, результаты имплементации, включающие максимальную достижимую тактовую частоту и утилизацию ресурсов ПЛИС, а также оценка микроархитектурной производительности.

В таблице 1 приведены ключевые характеристики наиболее распространенных софт-процессоров [1-7], на основе анализа которых сформулированы следующие требования к разрабатываемому процессору: производительность должна быть более

Таблица 1

Характеристики популярных софт-процессоров

Производитель или архитектура	Ядро	LUT	FF	DSP	Максимальная частота, МГц	Производительность		Открытый исходный код
						CoreMark/МГц	CoreMark	
ARM	Cortex-M1	1564	1171	3	125	1,83	229	нет
ARM	Cortex-M3	5800	1388	3	50	3,45	173	нет
XILINX	MicroBlaze	1607	1471	4	143	2,12	303	нет
INTEL	NIOS II/f	2326	-	-	140	2,29	321	нет
LATTICE	LatticeMico32	1090	951	3	152	1,78	271	да
RISC-V	Orca	1625	934	4	75	1,86	140	да
RISC-V	VexRiscv	1407	949	4	149	2,30	343	да
RISC-V	Taiga	1553	1038	4	115	2,5	288	да
SPARCv8	LEON3	3772	1573	4	110	2,36	260	да

350 CoreMark, а утилизация не более 2500 LUT для ПЛИС Xilinx семейства Artix-7 speedgrade-1.

II. НЕДОСТАТКИ МИКРОАРХИТЕКТУР СУЩЕСТВУЮЩИХ СОФТ-ПРОЦЕССОРОВ

С целью поиска высокопроизводительных микроархитектурных решений проведен анализ популярных 32-разрядных софт-процессоров. Были выявлены следующие микроархитектурные недостатки, ограничивающие производительность:

- малое количество стадий конвейера;
- необходимость простоя конвейера для разрешения конфликта по данным, связанного с инструкцией сдвига;
- большая логическая глубина цепей, связанных с шиной адреса к памяти данных и с шиной данных от памяти данных;
- длительный простой конвейера при выполнении операции умножения без наличия зависимостей по данным.

Ниже приводится краткий обзор обозначенных микроархитектурных недостатков.

A. Малое количество стадий конвейера

Ядра Cortex-M1 и Cortex-M3 имеют трехстадийный конвейер, показанный на рисунке 1 [8].

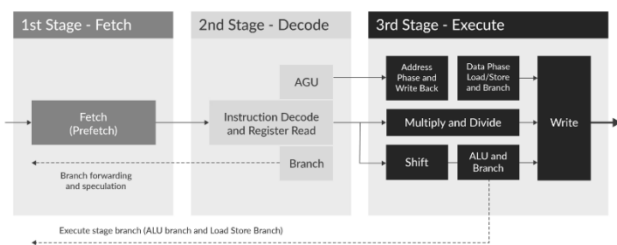


Рис. 1. Конвейер процессоров Cortex-M1 и Cortex-M3

Увеличение количества стадий конвейера позволяет повысить максимально достижимую тактовую частоту, что ведет к улучшению производительности процессора. В то же время добавление новых стадий конвейера увеличивает аппаратные затраты и количество тактов простоя при выполнении инструкций условного и безусловного переходов и разрешении прочих конфликтов, что негативно сказывается на производительности. Таким образом, существует оптимальное с точки зрения производительности количество стадий конвейера, при дальнейшем увеличении которого повышение тактовой частоты не покрывает издержек, связанных с разрешением конфликтов [9]. Беря во внимание это соображение и тот факт, что большинство ядер из таблицы 1, превосходящих по производительности Cortex-M1 и Cortex-M3, имеют большую длину конвейера, можно сделать вывод о том, что производительность ядер Cortex-M1 и Cortex-M3 ограничена в связи с малым количеством стадий конвейера.

B. Простой конвейера при конфликте по данным, связанным с инструкцией сдвига

Распространенным микроархитектурным недостатком является реализация схемы сдвига. Так в процессорах MicroBlaze [10], NIOS II [11], LatticeMico32 [12] и VexRiscv [5] в случае конфликта по данным, связанном с инструкцией сдвига, требуется приостанавливать конвейер на 1 или более тактов. Для иллюстрации проблемы на рисунке 2 ниже показан конвейер процессора LatticeMico32.

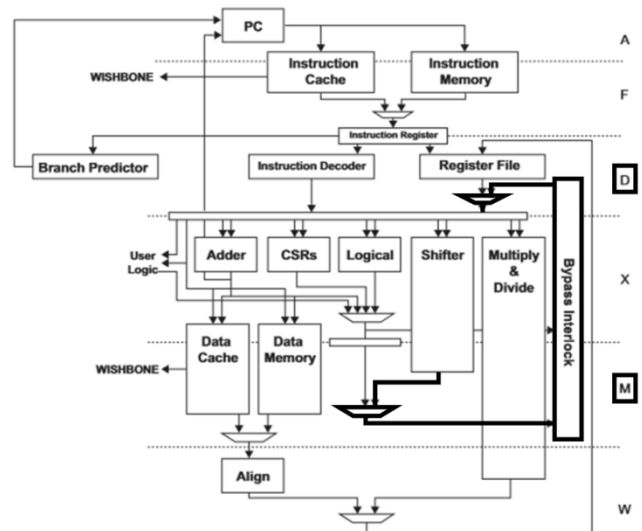


Рис. 2. Простой конвейера для инструкции сдвига

В случае конфликта по данным с инструкцией сдвига требуется задержать зависимую инструкцию в стадии D на один такт, поскольку результат сдвига будет доступен только в стадии M.

C. Большая логическая глубина цепей, связанных с памятью данных

В LatticeMico32 и Orca [13] наличие пути данных от выхода комбинационной схемы сумматора к шине адреса памяти данных (см. рисунок 3) потенциально ограничивает максимально достижимую тактовую частоту.

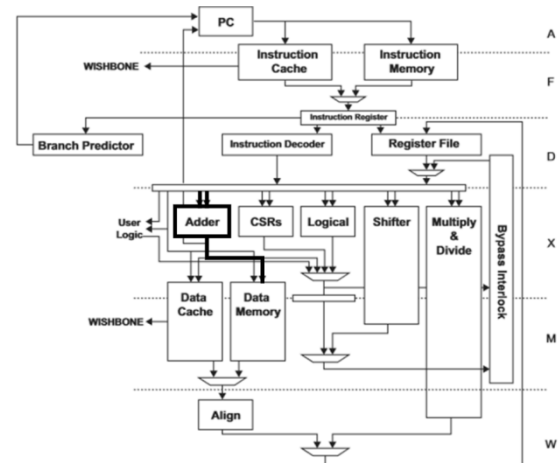


Рис. 3. Сумматор на шине адреса памяти данных

Локальная память данных строится на базе BRAM, а немногочисленные блоки BRAM рассеяны по большой площади кристалла ПЛИС, что вносит существенный вклад в задержку распространения сигнала от процессора до памяти. Так, для реализации памяти данных объемом 64 киббайт в ПЛИС Xilinx семейства Artix-7 используется 16 примитивов RAMB36E1, которых в ПЛИС среднего объема программируемой логики – xc7a35ticsg324-1L – всего 50. Таким образом, инструменту имплементации нужно выбрать 16 из 50 доступных (распределенных по всей площади кристалла) примитивов, и на каждый из них подать сигнал адреса от процессора. По результатам имплементации задержка распространения сигнала от процессора до памяти в среднем составляет 2,45 нс. Добавление сумматора в этот и без того трудный для инструмента имплементации путь лишь усугубляет достижение временных требований.

В процессоре VexRiscv, конвейер которого приведен на рисунке 4, существует путь данных от памяти данных, идущий через схему выравнивания в схему обхода. Задержка распространения данных от блочной памяти весьма велика и для современных ПЛИС Xilinx семейства Artix-7 speedgrade-1 составляет примерно 4,2 нс. Все это приводит к длительному временному пути, ограничивающему максимальную тактовую частоту процессора.

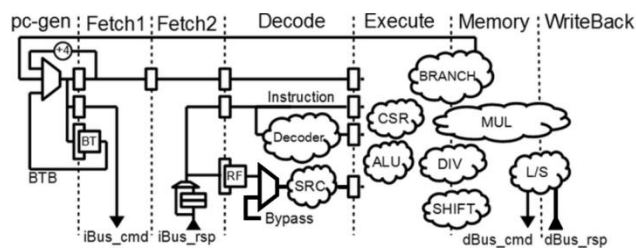


Рис. 4. Длительный путь данных от памяти данных

D. Простой конвейера при выполнении умножения

В процессоре Orca умножитель требует два такта простоя конвейера для вычисления результата умножения: первый такт расходуется на помещение входных операндов в регистры, второй – на запись в соответствующий регистр результата умножения. Это приводит к гарантированному простоя конвейера длительностью в 2 такта при выполнении инструкции умножения вне зависимости от конфликтов по данным.

III. ПРЕДЛОЖЕНИЯ ПО ПРОЕКТИРОВАНИЮ ПРОЦЕССОРА

Устранения вышеуказанных недостатков можно добиться путем учета следующих соображений:

- для устранения простоя конвейера в случае конфликта по данным, связанным с исполнением операции сдвига, можно реализовать полностью комбинационную

схему сдвига, выход которой передается в схему обхода;

- логическую глубину цепи, связанной с шиной адреса к памяти данных, можно ограничить путем разнесения задачи генерации адреса и обращения к памяти по разным стадиям;
- логическую глубину цепи, связанной с шиной данных от памяти данных, можно ограничить путем разнесения задачи приема данных от памяти и их подачи в конвейер (схему обхода) по разным стадиям;
- уменьшения простоя конвейера в случае умножения можно достичь с помощью конвейерного, т.е. многостадийного исполнения операции умножения.

С учетом выше изложенного сформулируем требования к разрабатываемому софт-процессору:

- архитектура: RV32IM;
- поддержка одного привилегированного режима – Machine Mode;
- гарвардская архитектура – независимые интерфейсы памяти данных и инструкций;
- поддержка прерываний и исключений;
- поддержка регистров контроля и статуса (CSR);
- реализация инструкции сдвига без простоя конвейера при конфликте по данным;
- разделение генерации адреса для памяти данных и обращения к памяти данным по разным стадиям;
- разделение приема данных от памяти данных и их подачу в конвейер по разным стадиям;
- конвейерная реализация инструкций умножения.

Было решено использовать архитектуру RISC-V ввиду ее открытости, модульной организации и перспективности внедрения в отечественную микроэлектронную промышленность.

IV. ОПИСАНИЕ ПРЕДЛОЖЕННОЙ МИКРОАРХИТЕКТУРЫ

На рисунке 5 приведена структурная схема разработанного софт-процессора. Ключевыми микроархитектурными особенностями предложенного решения являются:

- конвейер из шести стадий;
- статический предсказатель условных переходов (алгоритм VTFTNT);
- исполнение команды JAL в стадии F – 1 такт простоя;
- исполнение команды условного перехода в случае успешного предсказания вызывает от 0

до 1 такта простоя, в случае неудачного – от 3 до 4 тактов;

- исполнение команды JALR – 3 такта простоя;
- операция сдвига реализована на комбинационной схеме, расположенной внутри ALU;
- компактная схема деления: 35 тактов простоя, в случае деления на ноль – 3 такта;
- конвейерный умножитель, реализованный на базе 4-х DSP блоков ПЛИС: 1 такт простоя.

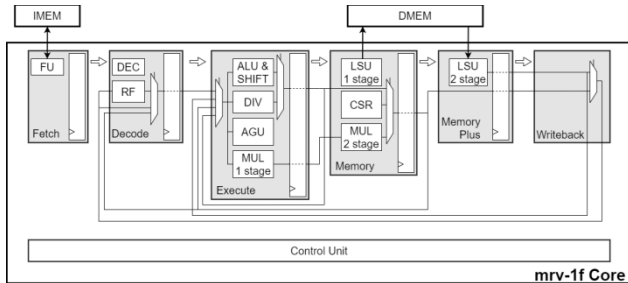


Рис. 5. Конвейер предложенного софт-процессора

Инструкция сдвига выполняется в АЛУ и не вызывает простоя конвейера при конфликте по данным. Для этого в составе АЛУ реализована комбинационная схема сдвига. Как и все инструкции, выполняющиеся на АЛУ, инструкция сдвига не вызывает простоя конвейера при конфликте по данным.

Команды умножения и обращения к памяти данных конвейеризованы. Умножение начинается в стадии Execute, завершается в Memory, а результат умножения доступен в Memory Plus (см. рисунок 6).

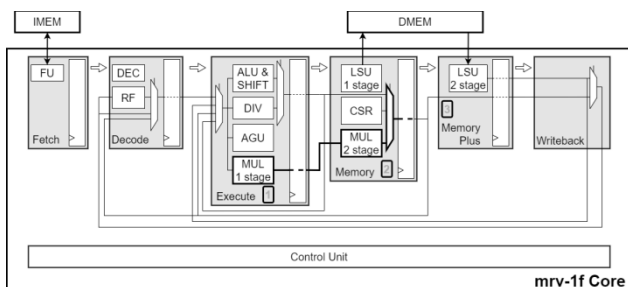


Рис. 6. Исполнение инструкции умножения

Для обращения к памяти адрес формируется внутри стадии Execute, запрос к памяти инициируется в стадии Memory, ответ от памяти принимается в стадии Memory Plus и только в стадии Writeback считанное из памяти значение попадает в схему обхода (см. рисунок 7). Такая реализация позволяет избежать сумматора на шине адреса памяти данных (как у LatticeMico32) и длинного временного пути в схеме обхода, связанного с шиной данных памяти данных (как у VexRiscv).

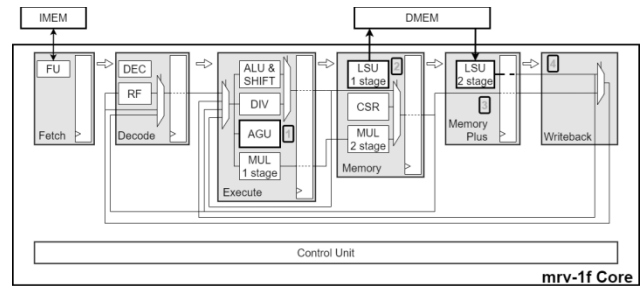


Рис. 7. Исполнение инструкции обращения к памяти данных

V. ВЕРИФИКАЦИЯ

Разработанный процессор был верифицирован с помощью тестового окружения на базе UVM с использованием генератора ассемблерных программ Google RISC-V-DV и эталонного симулятора архитектуры команд Spike более чем на пяти тысячах различных ассемблерных программах (см. рисунок 8).

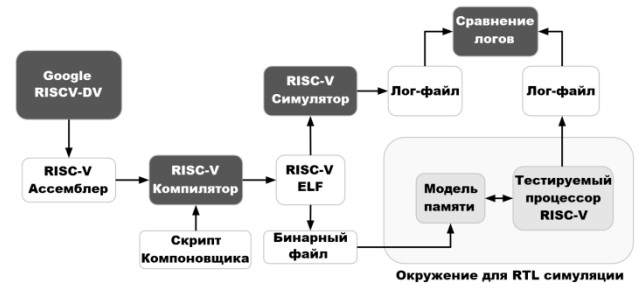


Рис. 8. Схема верификационного окружения

VI. ХАРАКТЕРИСТИКИ АППАРАТНОЙ РЕАЛИЗАЦИИ

На рисунке 9 приведены занимаемые процессором ресурсы ПЛИС по результатам имплементации в ПЛИС Xilinx семейства Artix-7 speedgrade-1. Как видно из сравнения с другими решениями процессор занимает достаточно много LUT-ов и триггеров; утилизация больше только у LEON3, NIOS II/f и Cortex-M3.

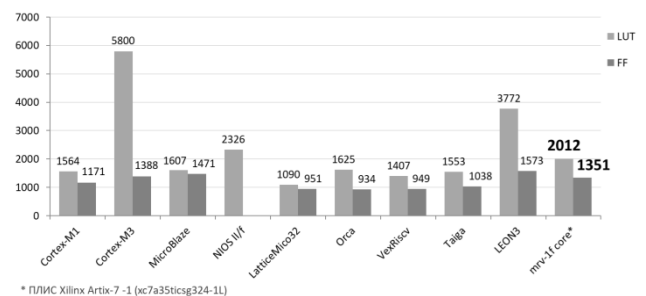
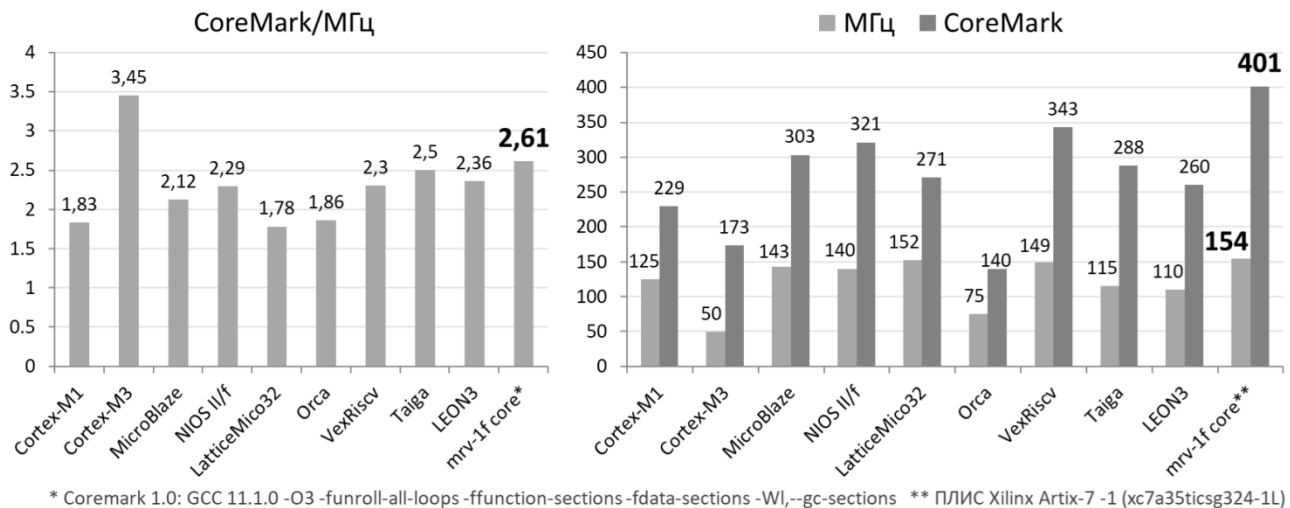


Рис. 9. Утилизация ресурсов ПЛИС для аппаратной реализации процессора

Однако при этом процессор обладает одним из самых высоких показателей по микроархитектурной производительности (измеряемой в CoreMark/МГц, на рисунке 10 слева) и имеет наибольшую тактовую



* Coremark 1.0: GCC 11.1.0 -O3 -funroll-all-loops -ffunction-sections -fdata-sections -Wl,--gc-sections ** ПЛИС Xilinx Artix-7 -1 (xc7a35ticsg324-1L)

Рис. 10. Оценка производительности процессора

частоту и производительность (измеряемую в CoreMark, на рисунке 10 справа). Для оценки тактовой частоты использовалась та же ПЛИС, что и для оценки занимаемых процессором ресурсов.

Можно оценить эффективность используемых процессором ресурсов ПЛИС путем деления его производительности на количество занимаемых им LUT-ов. Полученный показатель CoreMark/LUT показан на рисунке 11.

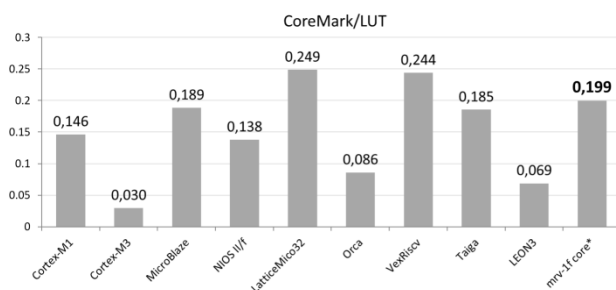


Рис. 11. Эффективность используемых процессором ресурсов ПЛИС

VII. ВЫВОДЫ

В рамках данной работы был разработан и верифицирован софт-процессор на базе архитектуры RISC-V для встраиваемых систем на программируемой логике. Разработанный софт-процессор имеет на 17% большую производительность по сравнению с рассмотренными решениями. Сравнение по утилизации ресурсов ПЛИС показывает большие аппаратные затраты разработанного процессора по сравнению с другими процессорами на архитектуре RISC-V. Такая разница объясняется разницей в полноте поддержки стандартов RISC-V, например, в схеме обработки прерываний и наборе служебных регистров, что имеет рекомендательный характер. В дальнейшем планируется дальнейшее улучшение производительности и утилизации за счет интеграции динамического предсказания переходов, оптимизации блока регистров CSR и счетчиков производительности.

ЛИТЕРАТУРА

- [1] Arm Cortex-M Processor Comparison Table [Электронный ресурс]. — URL: <https://developer.arm.com/documentation/102787/latest> (дата обращения: 25.08.2022)
- [2] MicroBlaze Benchmarks Part 1 — CoreMark Performance [Электронный ресурс]. — URL: <https://www.jblopen.com/microblaze-benchmarks-part-1-coremark-performance/> (дата обращения: 25.08.2022)
- [3] NIOS II, Processor Performance Benchmarks [Электронный ресурс]. — URL: <https://www.intel.com/content/www/us/en/products/details/fpga/nios-processor/ii.html> (дата обращения: 25.08.2022)
- [4] Włostowski T., Serrano J., Vaga F. Developing distributed hard-real time software systems using fpgas and soft cores // 15th International Conference on Accelerator and Large Experimental Physics Control Systems, Melbourne, Australia, 2015, pp.ТННА2101, doi: 10.18429/JACoW-ICALEPCS2015-ТННА2101
- [5] VexRiscv CPU Core [Электронный ресурс]. — URL: <https://github.com/SpinalHDL/VexRiscv> (дата обращения: 25.08.2022)
- [6] Heinz C., Lavan Y., Hofmann J., Koch A. A Catalog and In-Hardware Evaluation of Open-Source Drop-In Compatible RISC-V Softcore Processors // 2019 International Conference on ReConfigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 2019, pp. 1-8, doi: 10.1109/ReConFig48160.2019.8994796.
- [7] LEON3, Excel sheet for SOC area estimation [Электронный ресурс]. — URL: <https://www.gaisler.com/index.php/downloads/leongrlib?task=view&id=156> (дата обращения: 25.08.2022)
- [8] Arm Cortex-M3 Processor Datasheet [Электронный ресурс]. — URL: <https://developer.arm.com/documentation/102831/0100/?lang=en> (дата обращения: 25.08.2022)
- [9] Харрис, Д. М. Цифровая схемотехника и архитектура компьютера RISC-V / Д. М. Харрис, С. Л. Харрис ; под редакцией А. Ю. Романова ; перевод с английского В. С. Яценкова. — Москва : ДМК Пресс, 2022. — С. 548-549 — ISBN 978-5-97060-961-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/241166> (дата обращения: 25.08.2022).

- [10] MicroBlaze Processor Reference Guide, Ch. 2: MicroBlaze Architecture, Pipeline Architecture, Five Stage Pipeline [Электронный ресурс]. — URL: <https://docs.xilinx.com/v/u/en-US/ug984-vivado-microblaze-ref> (дата обращения: 25.08.2022)
- [11] NIOS II Processor Reference Guide, NIOS II Core Implementation Details, NIOS II/f Core, Execution Pipeline, Pipeline Stalls [Электронный ресурс]. — URL: <https://www.intel.com/content/www/us/en/docs/programmable/683836/current/pipeline-stalls-29473.html> (дата обращения: 25.08.2022)
- [12] LatticeMico32 Processor Reference Manual, Instruction Set, Instruction Descriptions, sl [Электронный ресурс]. — URL: https://www.latticesemi.com/view_document?document_id=48881 (дата обращения: 25.08.2022)
- [13] Vectorblox ORCA RISC-V CPU Core [Электронный ресурс]. — URL: <https://github.com/riscveval/orca-1> (дата обращения: 25.08.2022)

High-performance Soft Processor for Embedded FPGA-based Systems

E.S. Barkov, A.L. Pereverzev, A.M. Silantiev

National Research University of Electronic Technology, Moscow

Abstract — The article discusses the performance-limiting disadvantages of the microarchitecture of popular 32-bit soft processors for embedded FPGA-based systems. A microarchitecture of a soft processor based on the RISC-V architecture is proposed, devoid of the considered disadvantages. Verification and evaluation of the hardware characteristics of the developed soft processor was carried out. The developed soft processor has 17% higher performance compared to the solutions considered.

Keywords — soft processor, RISC-V, FPGA.

REFERENCES

- [1] Arm Cortex-M Processor Comparison Table — URL: <https://developer.arm.com/documentation/102787/latest> (access date: 25.08.2022).
- [2] MicroBlaze Benchmarks Part 1 — CoreMark Performance — URL: <https://www.jbopen.com/microblaze-benchmarks-part-1-coremark-performance/> (access date: 25.08.2022).
- [3] NIOS II, Processor Performance Benchmarks — URL: <https://www.intel.com/content/www/us/en/products/details/fpga/nios-processor/ii.html> (access date: 25.08.2022).
- [4] Włostowski T., Serrano J., Vaga F. Developing distributed hard-real time software systems using fpgas and soft cores // 15th International Conference on Accelerator and Large Experimental Physics Control Systems, Melbourne, Australia, 2015, pp. THHA2I01, doi: 10.18429/JACoW-ICALPEPCS2015-THHA2I01.
- [5] VexRiscv CPU Core — URL: <https://github.com/SpinalHDL/VexRiscv> (access date: 25.08.2022).
- [6] Heinz C., Lavan Y., Hofmann J., Koch A. A Catalog and In-Hardware Evaluation of Open-Source Drop-In Compatible RISC-V Softcore Processors // 2019 International Conference on ReConfigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 2019, pp. 1-8, doi: 10.1109/ReConFig48160.2019.8994796.
- [7] LEON3, Excel sheet for SOC area estimation — URL: <https://www.gaisler.com/index.php/downloads/leongrplib?task=view&id=156> (access date: 25.08.2022).
- [8] Arm Cortex-M3 Processor Datasheet — URL: <https://developer.arm.com/documentation/102831/0100/?lang=en> (access date: 25.08.2022).
- [9] Harris, D. M. Cifrovaya skhemitexhnika i arhitektura kompyutera RISC-V (Digital Design and Computer Architecture) / D. M. Harris, S. L. Harris ; pod redakciej A. YU. Romanova ; perevod s anglijskogo V. S. YAcenkova. — Moskva : DMK Press, 2022. — С. 548-549 — ISBN 978-5-97060-961-3. — Tekst : elektronnyj // Lan' : elektronno-bibliotecnaya sistema. — URL: <https://e.lanbook.com/book/241166> (data obrashcheniya: 25.08.2022).
- [10] MicroBlaze Processor Reference Guide, Ch. 2: MicroBlaze Architecture, Pipeline Architecture, Five Stage Pipeline — URL: <https://docs.xilinx.com/v/u/en-US/ug984-vivado-microblaze-ref> (access date: 25.08.2022).
- [11] NIOS II Processor Reference Guide, NIOS II Core Implementation Details, NIOS II/f Core, Execution Pipeline, Pipeline Stalls — URL: <https://www.intel.com/content/www/us/en/docs/programmable/683836/current/pipeline-stalls-29473.html> (access date: 25.08.2022).
- [12] LatticeMico32 Processor Reference Manual, Instruction Set, Instruction Descriptions, sl — URL: https://www.latticesemi.com/view_document?document_id=48881 (access date: 25.08.2022).
- [13] Vectorblox ORCA RISC-V CPU Core — URL: <https://github.com/riscveval/orca-1> (access date: 25.08.2022).