

Экспериментальное исследование влияния конвейеризации на быстродействие и аппаратные затраты алгоритма помехоустойчивого кодирования Хэмминга

С.Д. Редькин¹, Г.И. Качаева², М.Т. Эфендиев¹, А.А. Федотенков¹

¹Национальный исследовательский университет «Высшая школа экономики»,

¹101000, г. Москва, ул. Мясницкая, д. 20, Россия,

²Дагестанский государственный технический университет,

²367015, г. Махачкала, проспект Имама Шамиля 70, Россия,

Резюме. Цель. В работе рассмотрено влияние конвейеризации на эффективность реализации кода Хэмминга на ПЛИС. Конвейеризация рассматривалась как метод повышения скорости работы модуля, реализующего алгоритм помехоустойчивого кодирования. **Метод.** Эмпирическим путем был произведен анализ скорости работы и аппаратных требований различных реализаций кода Хэмминга на ПЛИС. При проведении исследования рассматривались ресурсы, которые потребляются на ПЛИС, а также максимальная тактовая частота, при которой модуль будет работать корректно. **Результат.** Конвейеризация позволила значительно оптимизировать реализацию кода Хэмминга. **Вывод.** Конвейеризация является эффективным методом оптимизации реализации кода Хэмминга при корректном подборе глубины конвейера.

Ключевые слова: конвейерная архитектура, ПЛИС, код Хэмминга.

Для цитирования: С.Д. Редькин, Г.И. Качаева, М.Т. Эфендиев, А.А. Федотенков. Экспериментальное исследование влияния конвейеризации на быстродействие и аппаратные затраты алгоритма помехоустойчивого кодирования Хэмминга. Вестник Дагестанского государственного технического университета. Технические науки. 2026; 53(2):104-109. DOI:10.21822/2073-6185-2026-53-2-104-109

Experimental study of the influence of pipelining on performance and hardware costs of Hamming error-correcting code algorithm

S.D. Redkin¹, G.I. Kachaeva², T. M. Efendiev¹, A.A. Fedotenkov¹

¹National Research University Higher School of Economics,

¹20 Myasnitskaya St., Moscow 101000, Russia,

²Daghestan State Technical University,

²70 I. Shamil Ave., Makhachkala 367015, Russia

Abstract. Objective. The paper investigates the impact of pipelining on the efficiency of Hamming code implementation in FPGA devices. Pipelining is considered as a method to increase the speed of operation for modules implementing error-correcting algorithms. **Method.** An empirical analysis was conducted to evaluate both the performance speed and hardware requirements of various implementations of Hamming codes on FPGAs. During this study, attention was given to resources consumed by the FPGA and the maximum clock frequency at which the module operates correctly. **Result.** Pipelining has significantly optimized the implementation of Hamming codes. **Conclusion.** Pipelining proves to be an effective optimization technique when properly selecting pipeline depth for the Hamming code realization.

Keywords: Pipeline architecture, FPGA, Hamming code.

For citation: S.D. Redkin, G.I. Kachaeva, M.T. Efendiev, A.A. Fedotenkov. Experimental study of the influence of pipelining on performance and hardware costs of Hamming error-correcting code algorithm. Herald of Daghestan State Technical University. Technical Sciences. 2026;53(2):104-109. (In Russ) DOI:10.21822/2073-6185-2026-53-2-104-109

Введение. При проектировании цифровых устройств распространение получила конвейерная архитектура. Конвейеризацию часто рассматривают как метод повышения производительности системы. Конвейерная архитектура нашла применение в криптографических приложениях, например, при реализации шифра AES на FPGA [1, 2] в целях повышения его производительности, в реализации алгоритмов компьютерного зрения [3], в микропроцессорах [4]. Конвейерная архитектура позволяет увеличить максимальную тактовую частоту, на которой может работать система на FPGA. Так например, конвейерные реализации схемы аутентификации Elephant продемонстрировали наибольшую тактовую частоту среди других реализаций Elephant [5]. Преимуществом конвейера является потенциальная возможность достижения высокой тактовой частоты, поскольку сложные вычисления разбиваются на отдельные операции, каждая из которых выполняется на соответствующей стадии конвейера [6, 7].

Помехоустойчивое кодирование играет важную роль в обеспечении надежной передачи данных в условиях наличия шума и помех в канале связи [8]. Современные программируемые логические интегральные схемы (FPGA) предоставляют уникальную возможность для реализации эффективных методов коррекции ошибок, таких как коды Хэмминга, RS-коды и LDPC-кодирование [9]. Аппаратная гибкость FPGA и возможность поддержки параллельных вычислений делают их идеальным решением для создания высокопроизводительных систем коррекции ошибок. За счет глубокой конвейеризации и оптимального распределения вычислительных ресурсов FPGA достигают высоких скоростей обработки данных при минимальном уровне энергопотребления, что особенно актуально для устройств с ограниченной мощностью питания. Применение FPGA находит свое отражение в широком спектре отраслей, включая телекоммуникации, авиационно-космическое оборудование и промышленные автоматизированные системы, где необходимы повышенные уровни отказоустойчивости и устойчивости к внешним воздействиям.

Коды Хэмминга представляют собой семейство кодов коррекции ошибок, пришедших на смену простому повторяющемуся кодированию [10]. Коды Хэмминга представляют собой линейные блочные коды, способные исправлять одну или обнаруживать две ошибки [11, 12]. Коды Хэмминга обеспечивают целостность и надежность данных в различных коммуникационных системах: компьютерные сети, устройства хранения данных, цифровые каналы связи [13, 14]. В источнике [15] приведен пример формирования контрольных векторов кодов Хэмминга с использованием следующего алгоритма:

1. Все разряды кода нумеруются десятичными числами, начиная с единицы.
2. Каждое десятичное число преобразуется в двоичное.
3. Нумеруются контрольные разряды по степени числа 2, которая их образует.
4. Для получения значения i -го контрольного разряда выполняется операция суммирования по модулю $M=2$ тех информационных разрядов, в двоичном эквиваленте номеров которых на i -м месте справа стоит единица.

Реализация кода Хэмминга на Verilog может быть полезна при разработке цифровых устройств, спроектированных на основе ПЛИС. Эффективная реализация кода Хэмминга на FPGA приобретает особую значимость ввиду доступной стоимости и высокой гибкости данной платформы, что позволяет существенно сократить потери данных при передаче и повысить общую надежность коммуникационных систем [16]. При подобном проектировании возможны обстоятельства, когда имеется необходимость повышения скорости работы устройств. Для данной оптимизации может быть рассмотрена конвейеризация.

Постановка задачи. Задача исследования заключается в экспериментальной оценке влияния глубины конвейера на производительность и аппаратные затраты реализации кода Хэмминга на ПЛИС. Необходимо определить оптимальное соотношение между глубиной конвейера и эффективностью реализации, обеспечивающей максимальное повышение быстродействия при минимальных затратах ресурсов микросхемы.

Методы исследования. Влияние конвейеризации рассматривалось на примере трех реализаций кода Хэмминга на языке Verilog. Первая реализация предполагает кодирование за один такт, в то время как другие две реализации предполагают использование конвейера глубины 2 и глубины 3. Структура конвейера продемонстрирована на рис. 1 и на рис. 2. Каждый из реализованных модулей, использующих в себе конвейер, работает по положительному фронту тактового сигнала. Для верификации работоспособности разработанных модулей были созданы тестовые окружения на языке Verilog.



Рис. 1 - Структура конвейера глубины 2

Fig. 1 - Structure of a depth pipeline with two stages



Рис. 2 - Структура конвейера глубины 3

Fig. 2 - Structure of a depth pipeline with three stages

Обсуждение результатов. При помощи yosys [17, 18] был произведен синтез цифровой схемы реализованной на Verilog. На текущий момент yosys считается эталонным программным средством логического синтеза с открытым исходным кодом и используется повсеместно [19]. При помощи yosys также были определены потребляемые ресурсы: количество LUT (look-up table) и количество использованных триггеров на кристалле.

Информация о количестве данных примитивов может быть использована для оценки площади занимаемой синтезированной схемой на ПЛИС [20]. LUT с К-входами позволяет реализовать любую логическую функцию от К-переменных [21]. Использование LUT подразумевает предварительное вычисление результатов операций для всех возможных входных данных [22]. Информация о максимальной тактовой частоте была получена инструментом nextpnr-ice40 [23] для ПЛИС модели ICE40-LP8K.

Стоит отметить, что результаты, полученные при помощи данного инструмента носят теоретический характер, т.к. не учитывают физических ограничений, которые накладываются на реальные ПЛИС, т.е. при размещении на ПЛИС тактовая частота может отличаться от той, что указана nextpnr-ice40.

Рассмотрение полученных в ходе эксперимента с измерением аппаратных ресурсов и максимальных тактовых частот для различных модулей показывает, что конвейеризация глубины 2 и реализация без конвейеризации демонстрируют одинаковые максимальные тактовые частоты, при том что реализация с конвейером потребляет больше ресурсов (табл. 1).

Эквивалентность максимальных тактовых частот объясняется тем, что реализованные схемы продемонстрировали максимальные тактовые частоты, который определил nextpnr-ice40 для ПЛИС. Реализация с конвейером глубины 3 и вовсе оказалась существенно менее эффективной чем модуль без конвейеризации. Объяснением данных результатов является то, что код Хэмминга 7,4 является вычислительно простым алгоритмом

и введение конвейера оказывает отрицательный эффект на эффективность работы и, в обстоятельствах проведенного эксперимента, не может быть рассмотрен как метод оптимизации.

Рассмотрим тактовую частоту вышеприведенных реализаций на случай когда входные данные составляют 64 бит.

Таблица 1. Показатели различных реализаций кода Хэмминга
Table 1. Performance of various Hamming code implementations

	Без конвейеризации Without pipelining	Конвейеризация глубины 2 Depth 2 Pipelining	Конвейеризация глубины 3 Depth 3 Pipelining
Максимальная тактовая частота, МГц Maximum clock frequency	424.99	424.99	267.02
Общее количество примитивов Total number of primitives	3	20	25
Количество D-триггеров Number of D-triggers	2	16	21
Количество LUT4 Number of LUT4s	1	4	4

Предполагается, что увеличение входных данных усложнит работу системы, следовательно снижая максимальную тактовую частоту. Анализ результатов, приведенных в табл. 2 показывает, что при увеличении входных данных конвейеризация оказывает положительный эффект и является оптимальным выбором увеличения производительности.

Таблица 2. Показатели различных реализаций кода Хэмминга
Table 2. Performance of various Hamming code implementations

	Без конвейеризации Without pipelining	Конвейеризация глубины 2 Depth 2 Pipelining	Конвейеризация глубины 3 Depth 3 Pipelining
Максимальная тактовая частота, МГц Maximum clock frequency	117.99	424.99	112.5
Общее количество примитивов Total number of primitives	308	234	299
Количество D-триггеров Number of D-triggers	206	142	207
Количество LUT4 Number of LUT4s	102	92	92

Конвейер глубины 2 показал максимальную тактовую частоту гораздо более высокую чем решение без конвейеризации. При этом количество ресурсов на модуль с конвейером глубины 2 необходимо меньше чем решение без конвейеризации. Модуль с конвейеризацией глубины 3 показал примерно равные результаты с модулем без конвейеризации по максимальной тактовой частоте.

Вывод. Применение метода конвейеризации способно повысить эффективность исполнения модулей кода Хэмминга при значительных объемах входных данных, обуславливающих высокую вычислительную нагрузку. При большом объеме входных данных решение без конвейеризации и конвейерная архитектура глубины 3 продемонстрировали одинаковые максимальные тактовые частоты.

Конвейерная архитектура глубины 3 оказалась на 72% производительней чем реализация без конвейера. Важную роль играет структурная организация проектируемого устройства и характер выполняемых операций. Производительность конвейерной системы лимитируется продолжительностью наиболее ресурсоемкого этапа, определяющего максимальную частоту тактирования.

Библиографический список:

1. Hasija T. et al. A survey on performance analysis of different architectures of AES algorithm on FPGA // Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021. – 2023. – P. 39-54.
2. Kumar T.M. et al. A low area high speed FPGA implementation of AES architecture for cryptography application // Electronics. – 2021. – Vol. 10. – No. 16. – P. 2023.
3. Kreowsky P., Stabernack B. A full-featured FPGA-based pipelined architecture for SIFT extraction // IEEE access. – 2021. – Vol. 9. – P. 128564-128573.
4. Дашкевич Е. А. Особенности архитектур и режимов работы современных микропроцессоров // Вестник науки и образования. – 2018. – Т. 2. – №. 5 (41). – С. 37-39.
5. Konstantopoulou E., Athanasiou G., Sklavos N. Review and Analysis of FPGA and ASIC Implementations of NIST Lightweight Cryptography Finalists//ACM Computing Surveys. 2025;57(10):1-35.
6. Тарасов И.Е., Люлява Д.В., Дуксин Н.А. Проектирование конвейерных вычислительных устройств с учетом топологического представления // Вестник Рязанского государственного радиотехнического университета. – 2023. – № 86. – С. 86-95.
7. Kobaylo A.S. Architectural Features of Real-Time Computing Systems. Integrated Pipelined Computers // Proceedings of BSTU. Series3:Physical and Mathematical Sciences and Informatics. 2017;9(200):129–132.
8. Костюков А.С. и др. Помехоустойчивое кодирование в современных форматах связи // Вестник Воронежского государственного технического университета. – 2019. – Т. 15. – №. 2. – С. 132-138.
9. Hailes P. et al. A survey of FPGA-based LDPC decoders // IEEE Communications Surveys & Tutorials. – 2015. – Vol. 18. – No. 2. – P. 1098-1122.
10. Khan S.A. et al. Protocols and mechanisms to recover failed packets in wireless networks: History and evolution // IEEE Access. – 2016. – Vol. 4. – P. 4207-4224.
11. Золотарёв В.В., Овечкин Г.В. Помехоустойчивое кодирование // Методы и алгоритмы. Справочник–М.: Горячая линия–Телеком. – 2004.
12. Almazrouei K., Alnajjar K.A. Error-correcting Codes in Communication Systems // 2024 International Wireless Communications and Mobile Computing (IWCMC). – 2024. – P. 1-6.
13. Sabbry N.H., Levina A. Hamming code: An analysis of its reliability and efficiency in computer networks // 2023 International Conference Automatics and Informatics (ICAI). – IEEE, 2023. – P. 356-361.
14. Aliev Y., Ivanova G., Borodzhieva A. Design and Development of Interactive Software Models for Teaching Coding Theory: A Case Study on Hamming Codes—General Algorithm // Applied Sciences. – 2025. – Vol. 15. – No. 8. – P. 4231.
15. Ефанов Д.В., Блюдов А.А. Коды Хэмминга и их обнаруживающие способности в схемах функционального контроля // Информатика и системы управления. – 2012. – №. 2. – С. 100-111.
16. Umarathy K. et al.FPGA based implementation of hamming encoder and decoder. Micro-Electronics and Telecommunication Engineering: Proceedings of 4thICMETE 2020.Singapore:Springer Singapore, 2021:245-251.
17. YosysHQ. Yosys Open SYnthesis Suite [Электронный ресурс]. – 2025. – Режим доступа: https://yosys.readthedocs.io/_/downloads/en/latest/pdf/ (дата обращения: 09.10.2025).
18. Jafarof N., Kent K. B. Enhancing the VTR Flow: Integration of ABC9 via Yosys for Better Technology Mapping and Optimization /2024 International Workshop on Rapid System Prototyping (RSP). IEEE, 2024. – P. 56-62.
19. Гаврилов С.В. и др. Разработка доверенных средств проектирования ИС в базисе гетерогенных ПЛИС // Труды Института системного программирования РАН. – 2023. – Т. 35. – №. 5. – С. 107-126.
20. Камкин А.С. и др. Сравнение инструментов высокоуровневого синтеза и конструирования цифровой аппаратуры //Труды Института системного программирования РАН. 2022. Т.34. – №.5. – С.7-22.
21. Чочаев Р.Ж. и др. Модели и методы анализа структуры коммутационных ресурсов ПЛИС Обзор // Известия высших учебных заведений. Электроника. – 2020. – Т. 25. – №. 5. – С. 410-422.
22. McLoone M., McCanny J.V. Rijndael FPGA implementations utilising look-up tables //Journal of VLSI signal processing systems for signal, image and video technology. – 2003. – Vol. 34. – No. 3. – P. 261-275.
23. YosysHQ/nextpnr: nextpnr portable FPGA place and route tool [Электронный ресурс]. – 2025. – Режим доступа: <https://github.com/YosysHQ/nextpnr> (дата обращения: 15.09.2025).

References:

1. Hasija T., et al. A survey on performance analysis of different architectures of AES algorithm on FPGA. Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021. 2023;39-54.
2. Kumar T.M., et al. A low area high speed FPGA implementation of AES architecture for cryptography application. *Electronics*. 2021; 10(16):2023.
3. Kreowsky P., Stabernack B. A full-featured FPGA-based pipelined architecture for SIFT extraction. *IEEE Access*. 2021; 9:128564-128573.

4. Dashkevich E.A. Features of architectures and operating modes of modern microprocessors. *Bulletin of Science and Education*. 2018; 2(5 (41)):37-39. (In Russ)
5. Konstantopoulou E., Athanasiou G., Sklavos N. Review and Analysis of FPGA and ASIC Implementations of NIST Lightweight Cryptography Finalists. *ACM Computing Surveys*. 2025; 57(10):1-35.
6. Tarasov I.E., Lyulyava D. V. Design of Pipelined Computing Devices with Consideration of Topological Representation. (In Russ)
7. Kobaylo A.S. Architectural Features of Real-Time Computing Systems. Integrated Pipelined Computers. *Proceedings of BSTU. Series 3: Physical and Mathematical Sciences and Informatics*. 2017;9(200):129-132.
8. Kostyukov A.S., et al. Error-Resistant Coding in Modern Communication Formats. *Bulletin of Voronezh State Technical University*. 2019;15(2):132-138. (In Russ)
9. Hailes P., et al. A survey of FPGA-based LDPC decoders. *IEEE Communications Surveys & Tutorials*. 2015; 18(2):1098-1122.
10. Khan S.A., et al. Protocols and mechanisms to recover failed packets in wireless networks: History and evolution. *IEEE Access*. 2016; 4: 4207-4224.
11. Zolotaryov V.V., Ovechkin G.V. Error-Resistant Coding. Methods and Algorithms. Reference Book. – Moscow: Goryachaya Liniya–Telecom, 2004. (In Russ)
12. Almazrouei K., Alnajjar K.A. Error-correcting Codes in Communication Systems. *2024 International Wireless Communications and Mobile Computing (IWCMC)*. 2024;1-6.
13. Sabbry N.H., Levina A. Hamming code: An analysis of its reliability and efficiency in computer networks. *2023 International Conference Automatics and Informatics (ICAI)*. IEEE, 2023: 356-361.
14. Aliev Y., Ivanova G., Borodzhieva A. Design and Development of Interactive Software Models for Teaching Coding Theory: A Case Study on Hamming Codes—General Algorithm. *Applied Sciences*. 2025;15(8):4231.
15. Efanov D.V., Blyudov A.A. Hamming Codes and Their Detecting Capabilities in Functional Control Circuits. *Informatics and Control Systems*. 2012; 2:100-111. (In Russ)
16. Umapathy K., et al. FPGA based implementation of hamming encoder and decoder. *Micro-Electronics and Telecommunication Engineering: Proceedings of 4th ICMETE 2020*. Singapore: Springer Singapore, 2021:245-251.
17. YosysHQ. Yosys Open SYnthesis Suite [Electronic resource]. – 2025. – Access mode: <https://yosys.readthedocs.io/en/latest/pdf/> (date of request: 09.10.2025).
18. Jafarof N., Kent K.B. Enhancing the VTR Flow: Integration of ABC9 via Yosys for Better Technology Mapping and Optimization. *2024 International Workshop on Rapid System Prototyping (RSP)*. IEEE, 2024: 56-62.
19. Gavrilov S.V., et al. Development of Trusted Design Tools for Integrated Circuits Based on Heterogeneous FPGAs. *Proceedings of the Institute for System Programming of the Russian Academy of Sciences*. 2023; 35(5): 107-126. (In Russ)
20. Kamkin A.S., et al. Comparison of High-Level Synthesis and Digital Hardware Design Tools. *Proceedings of the Institute for System Programming of RAS*. 2022; 34(5):7-22. (In Russ)
21. Chochaev R.Zh., et al. Models and Methods for Analyzing the Structure of FPGA Switching Resources: A Review. *News of Higher Education Institutions. Electronics*. 2020; 25(5):410-422. (In Russ)
22. McLoone M., McCanny J.V. Rijndael FPGA implementations utilising look-up tables. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*. 2003; 34(3):261-275.
23. YosysHQ/nextpnr: nextpnr portable FPGA place and route tool [Electronic resource]. – 2025. – Access mode: <https://github.com/YosysHQ/nextpnr> (date of request: 15.09.2025).

Сведения об авторах:

Редькин Святослав Дмитриевич, студент; sdredkin@edu.hse.ru
Гюльханум Ибадулаховна Качаева, кандидат экономических наук, доцент, заведующая кафедрой информационной безопасности и программной инженерии; providetc@mail.ru
Эфендиев Мурад Тимурович, студент; mtfendiev@edu.hse.ru
Федотенков Алексей Александрович, студент; aafedotenkov@edu.hse.ru

Information about authors:

Svyatoslav D. Redkin, Student; sdredkin@edu.hse.ru
Gulhanum I. Kachaeva, Cand. Sci. (Econom.), Assoc. Prof., Head of the Department of Information Security and Software Engineering; providetc@mail.ru
Murad T. Efendiev, Student; mtfendiev@edu.hse.ru
Aleksey A. Fedotenkov, Student; aafedotenkov@edu.hse.ru

Конфликт интересов/Conflict of interest.

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/Received 20.10.2025.

Одобрена после рецензирования/Revised 24.12.2025.

Принята в печать/Accepted for publication 26.04.2026.