

**ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

УДК 004.436.2

DOI: 10.17223/19988605/46/12

**С.Е. Рыжова, Е.А. Мыцко, А.Н. Мальчуков****СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕАЛИЗАЦИЙ ДЕКОДЕРОВ БЧХ-КОДА  
С ПАРАМЕТРАМИ (15, 7, 5) НА ПЛИС***Работа выполнена в рамках гранта РФФИ № 18-47-700010 р\_а от 07 июня 2018 г.*

Рассматриваются реализации табличного и циклического методов декодирования для исправления двукратных независимых ошибок с длиной кодового слова 15 бит и длиной информационного блока 7 бит на основе программируемой логической интегральной схемы (ПЛИС). Описаны способы увеличения быстродействия реализуемых устройств за счет применения параллельной структуры и матричного алгоритма деления полиномов. Приведено сравнение по быстродействию различных реализаций декодеров с применением табличного и циклического методов декодирования. Установлено, что при реализации циклического метода минимальное время декодирования входной кодовой комбинации можно уменьшить до 12 нс.

**Ключевые слова:** помехоустойчивый код; код Боуза–Чоудхури–Хоквингема; табличный метод декодирования; циклический метод декодирования; аппаратная реализация; ПЛИС.

В современном мире большие объемы данных передаются по различным каналам связи. В процессе передачи существует вероятность возникновения ошибок в данных, вызванных влиянием окружающей среды. Для обнаружения и исправления таких ошибок используются помехоустойчивые коды. Наиболее часто встречающимися в данной области являются коды Боуза–Чоудхури–Хоквингема (БЧХ) [1, 2], которые относятся к широкому классу циклических кодов и способны обнаруживать и исправлять большое количество независимых ошибок в принимаемой комбинации. Область применения кодов БЧХ охватывает широкий спектр систем передачи данных: системы цифрового телевидения (стандарты DVB-S2X, DVB-C2) [3], единую европейскую систему связи, сотовую и космическую связь, системы автоматизации производства, системы пейджинговой связи (POCSAG, БЧХ-код (31,21)) [4]. Коды БЧХ можно использовать не только для промышленных целей, но и для военных, например для систем кодирования–декодирования информации, передаваемой с беспилотных летательных аппаратов (БПЛА).

Во многих таких стандартах передачи данных применяются коды большой длины (от 1 024 бит), однако коды малой длины (до 31) и с корректирующей способностью до 2–3 бит также могут быть востребованными в системах с передачей небольших пакетов команд или данных, а также при разработке собственных протоколов. Так, коды длиной до 15 бит используются в системах автоматизации производства или в БПЛА [5]. Однако для одного и того же БЧХ-кода существуют разные подходы к реализации, которые будут отличаться по быстродействию и аппаратным затратам, что требует дополнительного исследования.

**1. Обзор существующих алгоритмов**

В системах связи при передаче данных используется два устройства — кодер и декодер, которые находятся на отправляющей и принимающей стороне соответственно. Алгоритмы декодирования имеют свои особенности и различаются по степени сложности реализации. Одними из простых методов декодирования являются: табличный, основанный на применении таблицы предвычисленных

синдромов ошибок, и циклический, основанный на проверке веса остатка от деления кодового слова на образующий полином [6].

### 1.1. Табличный метод декодирования

Наиболее простым в реализации методом декодирования является табличный метод. Он применим как для нециклических, так и для циклических помехоустойчивых полиномиальных кодов. Такие коды строятся на основе образующих полиномов, которые должны удовлетворять следующим условиям: 1) расстояние Хэмминга должно удовлетворять условию  $d_{\min} \geq 2t + 1$ , где  $t$  – корректирующая способность кода; 2) синдромы ошибок кратности  $t$  должны быть уникальными. Основой метода является наличие таблицы, в которой хранятся все возможные значения шаблонов ошибок до кратности  $t$ . Адресами шаблонов ошибок являются синдромы ошибок. Имея такую таблицу, процесс декодирования сводится к следующему алгоритму:

1. Формируются все варианты шаблонов ошибок заданной кратности  $t$ .
2. Осуществляется процесс деления шаблона ошибки на образующий полином с целью получения остатка.
3. Остаток, полученный на этапе 2, является адресом для соответствующего делимого (шаблона ошибки).

Процесс декодирования имеет следующую последовательность этапов:

1. Входная кодовая комбинация делится на образующий полином.
2. По остатку от деления, полученному на этапе 1, выбирается шаблон ошибки из таблицы.
3. Выбранный шаблон суммируется по модулю 2 («исключающее или») с входной кодовой комбинацией.
4. Результатом сложения является исправленное кодовое слово.

Преимуществами данного метода являются простота его реализации и отсутствие необходимости выполнения циклических сдвигов кодовой комбинации, в отличие от циклического метода декодирования.

Основным недостатком данного метода является то, что при увеличении длины входной кодовой комбинации и корректирующей способности кода таблица будет расти согласно выражению  $2^k n$  где  $k$  – длина контрольного блока сообщения;  $m$  – длина информационного блока сообщения;  $n$  – длина всей кодовой комбинации ( $n = m + k$ ).

### 1.2. Циклический метод декодирования

Коды Боуза–Чоудхури–Хоквингема относятся к классу циклических кодов, поэтому для их декодирования можно применить циклический метод.

Алгоритм, реализующий циклический метод, заключается в следующем:

1. Производится деление по модулю 2 входной кодовой комбинации на образующий многочлен с целью получения остатка от деления.
2. Проверка условия, что вес остатка ( $w$  – количество единиц в комбинации) должен быть меньше или равен количеству исправляемых декодером ошибок ( $w \leq t$ ).
3. Если условие не выполняется, то необходимо осуществить циклический сдвиг входной кодовой комбинации влево на 1 бит и вернуться к пункту 1, иначе – перейти к пункту 4.
4. Если условие  $w \leq t$  выполняется, то необходимо произвести суммирование кодовой комбинации с остатком и получить исправленную кодовую комбинацию.
5. Если кодовая комбинация была циклически сдвинута влево, то необходимо исправленную комбинацию циклически сдвинуть на такое же количество бит вправо.
6. Если осуществлено  $n - 1$  циклических сдвигов, но условие  $w \leq t$  не выполнено, то регистрируется неисправимая ошибка.





Рис. 2. Структурная схема реализации табличного метода декодирования  
 Fig. 2. Structural scheme of implementation of the tabular method of decoding

По результатам тестирования для выполнения процесса декодирования необходимо 36 тактов с частотой 275,2 МГц (реализация № 1 в таблице), из которых 30 тактов – это процедура деления входной кодовой комбинации на образующий полином. За 6 тактов происходит выбор необходимого синдрома ошибки из таблицы, записанной в память, и сложение выбранного шаблона с кодовой комбинацией. Данная реализация требует 65 логических элементов и 1 792 бита памяти.

## 2.2. Реализация циклического метода с классическим делением полиномов

Декодер, основанный на циклическом методе декодирования, с реализацией ЛРОС для выполнения процесса деления имеет четкую последовательную структуру. Переходы от одного этапа декодирования к другому происходят по строго заданным условиям при помощи управляющих сигналов разрешения работы конкретного блока. На рис. 3 приведена структурная схема декодера с применением циклического метода декодирования. Остаток от деления входной кодовой комбинации на образующий полином вычисляется по схеме, представленной на рис. 1.

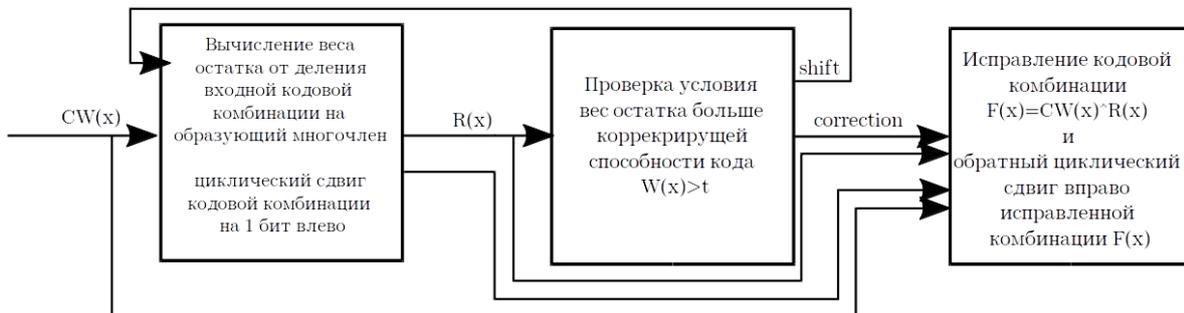


Рис. 3. Структурная схема декодера с применением циклического метода декодирования  
 Fig. 3. The block diagram of the decoder with the application of a cyclic method of decoding

Такой декодер с последовательной структурой имеет большое время работы, однако занимает малое количество ресурсов. В процессе декодирования выделяется три основных этапа:

- процесс деления входной кодовой комбинации на образующий полином и циклический сдвиг кодовой комбинации на 1 бит влево;
- процесс подсчета веса полученного остатка и проверка условия  $w \leq t$ ;
- сложение кодовой комбинации с соответствующим остатком и циклический сдвиг исправленной кодовой комбинации на необходимое количество бит вправо.

По результатам тестирования декодера были определены характеристики, оценивающие декодер: количество тактов, необходимое для максимального количества сдвигов кодовой комбинации; максимальная частота работы устройства; количество использованных логических элементов. Полная процедура декодирования производится за 627 тактов с частотой 192,8 МГц (реализация № 2 в таблице). Каждая процедура деления входной кодовой комбинации на образующий полином осуществляется за 30 тактов. Процедура сложения сдвинутой кодовой комбинации с остатком и обратный циклический сдвиг на 14 бит влево выполняется за 27 тактов. Наименее трудоемкими из всех являются 15 процедур

подсчета веса остатка от деления входной кодовой комбинации на образующий полином, каждая из которых выполняется за 10 тактов. Данная реализация требует 211 логических элементов ПЛИС.

### 2.3. Реализация циклического метода, с параллельной структурой и классическим алгоритмом деления полиномов

Одной из возможных модификаций для достижения быстродействия устройства является распараллеливание структуры декодера. На рис. 4 представлена структурная схема для реализации циклического метода декодирования с распараллеливанием промежуточных процессов декодирования: деление входной кодовой комбинации на образующий полином и подсчет весов остатков, полученных в процессе деления.

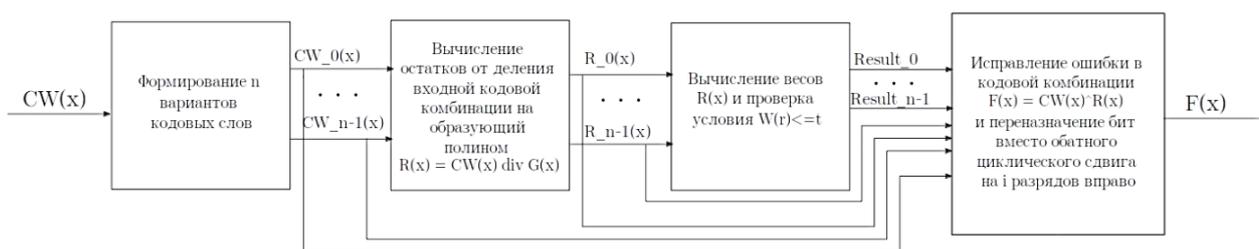


Рис. 4. Структурная схема для реализации циклического метода декодирования с параллельной структурой  
Fig. 4. Block diagram for realization of cyclic decoding method with parallel structure

В отличие от предыдущей реализации, каждый циклический сдвиг кодового слова обрабатывается параллельно. Вместо выполнения циклического сдвига в результате невыполнения условия  $w \leq t$  производится одновременное деление всех сдвигов кодовой комбинации на образующий полином. Для реализуемого кода с длиной кодовой комбинации  $n = 15$  реализовано 15 вычислительных блоков, задействованных в процессе деления, и 15 вычислительных блоков, в которых производится подсчет весов для остатков от деления, полученных на предыдущем шаге, и проверка условия  $w \leq t$ .

Такая модификация реализации позволяет уменьшить время работы декодера с 627 тактов при частоте 257,2 МГц до 45 тактов с частотой 179,3 МГц (реализация № 3 в таблице). Процесс деления для одного сдвига при этом выполняется за 30 тактов, однако в данном случае все кодовые слова обрабатываются параллельно. Остальные этапы декодирования, включающие подсчет весов, проверку условия и исправление кодовой комбинации, выполняются за 15 тактов. Для увеличения быстродействия декодера, разрабатываемого по предложенной структуре, необходимо сократить время выполнения операции деления. Для решения данной задачи можно применить матричный алгоритм деления полиномов.

### 2.4. Применение матричного алгоритма деления полиномов

Результаты тестирования декодеров показывают, что выполнение циклического метода декодирования с использованием параллельной структуры осуществляется за 45 тактов. Согласно закону Амдала [16], при распараллеливании циклического метода декодирования не может осуществляться быстрее, чем за 30 тактов, поэтому для достижения наилучшего быстродействия необходимо заменить классический алгоритм деления полиномов матричным алгоритмом [17], который позволит выполнять деление за 1–2 такта.

Принцип матричного деления заключается в применении операции умножения по модулю 2 кодового слова на предвычисленную матрицу остатков от деления полиномов от  $x^0$  до  $x^n$  на образующий полином. Такой подход позволяет получить формулу для вычисления остатка, которая на ПЛИС реализуется при помощи комбинационной схемы и будет выполняться за 1 такт, что позволит

увеличить быстродействие декодера. На рис. 5 приведена матрица для BCH-кода (15, 7, 5), которая использовалась в дальнейших реализациях.

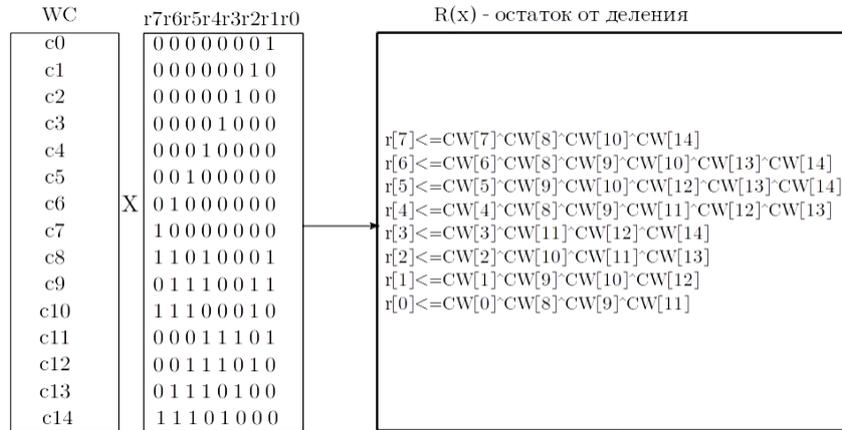


Рис. 5. Структурная схема матричного алгоритма для BCH-кода (15, 7, 5)  
 Fig. 5. Block diagram of matrix algorithm for BCH-code (15, 7, 5)

### 2.5. Модификация декодеров для увеличения быстродействия

Для полноценного сравнительного анализа реализованных устройств необходимо модифицировать все предшествующие реализации декодеров, заменив при этом классический алгоритм деления полиномов на матричный алгоритм. Это позволит сравнивать быстродействие декодеров при изменении одного вычислительного блока. Так, при реализации табличного метода декодирования с матричным делением полиномов (реализация № 4 в таблице) время работы сократилось с 36 тактов при частоте 275,2 МГц до 7 тактов с частотой 275,2 МГц, из которых за 1 такт происходит процесс деления входной кодовой комбинации на образующий полином и за 6 тактов – выбор шаблона ошибки из таблицы и исправление кодовой комбинации.

Также стоит отметить, что при реализации последовательной структуры циклического метода декодирования с применением матричного деления полиномов (реализация № 5 в таблице) время работы устройства сократилось с 627 тактов с частотой 192,8 МГц до 192 тактов с частотой 200,1 МГц. Каждая из 15 процедур деления выполняется за 1 такт. Процедура подсчета веса остатка от деления выполняется за 10 тактов, а процедура сложения сдвинутого кодового слова с соответствующим остатком и обратный циклический сдвиг реализуются за 27 тактов.

Реализация параллельной структуры с матричным алгоритмом деления (реализация № 6 в таблице) также позволяет увеличить быстродействие устройства с 45 тактов с частотой 179,3 МГц до 16 тактов с частотой 174,1 МГц, из которых 1 такт необходим на процесс деления и 15 тактов – на два оставшихся этапа декодирования.

Из сравнения видно, что замена одного алгоритма позволяет увеличить быстродействие декодера, при этом преодолеть предел в 30 тактов, который нельзя преодолеть распараллеливанием системы. Однако достигнуть улучшения характеристик декодера можно не только за счет применяемых алгоритмов, но и за счет особенностей ПЛИС.

### 2.6. Реализация асинхронного циклического метода декодирования с параллельной структурой и матричным делением полиномов

При проектировании схем с применением языков описания аппаратуры возможно использование двух подходов:

- подход с использованием секвенциальной логики;
- подход с использованием комбинационной логики.

Секвенциальная логика [18] представляет собой логику памяти устройств, и цифровые устройства формируются с учетом предыдущих значений. Декодеры, реализованные ранее, используют секвенциальную логику проектирования.

Последняя представленная реализация декодера использует не только все описанные выше модификации, позволяющие решить проблему низкого быстродействия при реализации циклического метода декодирования. Она также спроектирована на основе комбинационной логики. Схемы, построенные на комбинационной логике, используют текущие значения с входов для формирования значений на выходе. Такой подход позволит увеличить быстродействие декодера за счет отсутствия задержек, необходимых для установки сигналов в триггерах, которые используются в секвенциальной логике. Также данная модификация позволит уменьшить количество используемых декодером ресурсов.

По результатам моделирования работы декодера (реализация № 7 в таблице) на процесс декодирования необходим всего 1 такт. Это меньше, чем при реализации табличного метода декодирования с матричным делением полиномов, которому необходимо 7 тактов для выполнения всего алгоритма. Также в данном случае отсутствует необходимость в подключении памяти ПЛИС для хранения таблицы шаблонов ошибок. Недостатком такой реализации является низкая максимальная частота работы устройства.

### 3. Результаты

Таблица содержит результаты тестирования всех реализованных вариантов декодеров. Приводятся показатели по следующим характеристикам:

- количество тактов для декодирования кодовой комбинации;
- Max Frequency — максимальная частота, на которой способно работать устройство;
- $T$  – минимальное время, необходимое для полного декодирования входного кодового слова;
- Logic Cells – количество занимаемых логических элементов ПЛИС;
- Memory Bits – объем занимаемой памяти в битах.

Для проведения эксперимента выбрана ПЛИС семейства Altera Cyclone III.

**Сводная таблица параметров при различных подходах реализации декодера БЧХ-кода (15,7,5)**

| №  | Тип реализации           | Тип деления       | Количество тактов выполнения | Max Frequency, МГц | T (время работы), нс | Logic Cells, (%) | Memory Bits, (%) |
|----|--------------------------|-------------------|------------------------------|--------------------|----------------------|------------------|------------------|
| 1. | Табличный метод          | ЛРОС              | 36                           | 275,2              | 131                  | 65 (1%)          | 1 792 (1%)       |
| 2. | Последовательная         | ЛРОС              | 627                          | 192,8              | 3 252                | 211 (4%)         | –                |
| 3. | Параллельная синхронная  | ЛРОС              | 45                           | 179,3              | 251                  | 1 770 (34%)      | –                |
| 4. | Табличный метод          | Матричное деление | 7                            | 275,2              | 25                   | 35 (< 1%)        | 1 792 (1%)       |
| 5. | Последовательная         | Матричное деление | 192                          | 200,1              | 959                  | 202 (4%)         | –                |
| 6. | Параллельная синхронная  | Матричное деление | 16                           | 174,1              | 91                   | 1 549 (30%)      | –                |
| 7. | Параллельная асинхронная | Матричное деление | 1                            | 87                 | 12                   | 446 (9%)         | –                |

По результатам тестирования, приведенным в таблице, можно отметить, что при реализации на ПЛИС матричное деление полиномов является более быстродействующим в сравнении с классической реализацией деления через ЛРОС. Также можно заметить, что табличный метод декодирования является наилучшим вариантом как по быстродействию, так и по количеству используемых логических элементов. Однако в отличие от остальных реализаций декодеров необходимо обязательное подключение дополнительной памяти ПЛИС для хранения таблицы шаблонов ошибок размером 1 792 бита.

Поскольку размер таблицы будет увеличиваться в зависимости от длины кодовой комбинации и количества исправляемых ошибок в соответствии с формулой  $2^k n$ , то при увеличении длины кодового слова объем требуемой памяти будет возрастать.

В отличие от табличного метода декодирования при реализации циклического метода нет необходимости в подключении памяти ПЛИС, однако количество тактов для выполнения процедуры декодирования и количество логических ячеек будет больше (реализации № 2 и 3), чем при реализации табличного метода.

Также можно отметить, что реализация матричного метода деления полиномов имеет преимущества перед классическим делением при помощи ЛПРОС как по количеству тактов на всю процедуру декодирования, так и по количеству логических элементов (реализации № 1 и 4; 2 и 5; 3 и 6).

Из всех реализаций декодер, построенный на асинхронной параллельной структуре циклического метода декодирования с применением матричного алгоритма декодирования (реализация № 7), показывает лучший результат по времени выполнения процесса декодирования.

### Заключение

В процессе исследований методов декодирования кодов BCH рассмотрены и реализованы табличный и циклический методы декодирования. Разработанные декодеры модифицированы путем изменения последовательной структуры циклического метода декодирования на параллельную, наиболее подходящую для реализации на ПЛИС. За счет изменения структуры декодера количество тактов для декодирования уменьшилось с 627 при частоте 192 МГц до 45 при частоте 179 МГц, что эквивалентно минимальному времени полного декодирования входного кодового слова 3,2 мкс и 251 нс соответственно, при этом самый длинный этап декодирования – деление полиномов – выполняется за 35 тактов.

Для достижения наилучшего быстродействия классический алгоритм деления полиномов заменен на матричный, что позволило уменьшить время декодирования при последовательной структуре декодера с 627 тактов до 192 при частоте 200 МГц (959 нс). При параллельной структуре с применением матричного алгоритма деления полиномов декодирование кодового слова осуществляется за 16 тактов при частоте 174 МГц (91 нс). Применение асинхронной реализации циклического метода декодирования с матричным делением и параллельной структурой декодера позволяет увеличить быстродействие работы устройства до 12 нс, что в ~ 2 раза быстрее, чем при реализации табличного метода с матричным делением.

### ЛИТЕРАТУРА

1. Золотарёв В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы : справочник. М. : Горячая линия-Телеком, 2004. 126 с.
2. Морелос-Сарагоса Р.М. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М. : Техносфера, 2005. 320 с.
3. Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital transmission system for cable systems (DVB-C2) // DVB Document A138. 2010. P. 111.
4. Протокол POCSAG. Протоколы пейджинговой системы связи. URL: [https://studopedia.ru/2\\_56965\\_protokol-POCSAG.html](https://studopedia.ru/2_56965_protokol-POCSAG.html) (дата обращения: 23.05.2018)
5. Слюсар В. Передача данных с борта БПЛА: стандарты НАТО // Электроника: наука, технология, бизнес. 2010. № 3. С. 81–86.
6. Кодирование информации: двоичные коды : справочник / под ред. Н.Г. Березюка. Харьков : Вища школа, 1978. 251 с.
7. Поляков А.К. Языки VHDL И VERILOG в проектировании цифровой аппаратуры. М. : Солон-Пресс, 2003. 305 с.
8. Prakash G., Muthamizhse I. FPGA Implementation of Bose Chaudhuri Hocquenghem Code (Bch) Encoder and Decoder for Multiple Error Correction Control // International Journal of Innovative Research in Science, Engineering and Technology. 2016. V. 5, is. 3. P. 4467–4473.
9. Panda A., Sarik S., Awasthi A. FPGA implementation of encoder for (15, k) binary BCH code using VHDL and performance comparison for multiple error correction control // International Conference on Communication Systems and Network Technologies (CSNT-2012).
10. Rohith S., Pavithra S. FPGA Implementation of (15,7) BCH encoder and decoder for text message // International Journal of Research in Engineering and Technology. 2013. V. 2, is. 9. P. 209–214.

11. Hiremath M., Devi M. A novel method implementation of a FPGA using (n, k) binary BCH code // International Journal of Research in Engineering Technology and Management. 2014. Special is. P. 1–8.
12. Sunita M.S., Chiranth V., Akash H.C., Bhaaskaran V.S.K. Pipeline architecture for fast decoding of bch codes For nor flash memory // ARPN Journal of Engineering and Applied Sciences. 2015. V. 10, No. 8. P. 3397–3404.
13. Yathiraj H., Hiremath R. Implementation of BCH code (n, k) encoder and decoder for multiple error correction control // International Journal of Computer Science and Mobile Applications. 2014. V. 2, is. 5. P. 45–54.
14. Mohammed S., Abdulsada H. Design and implementation of 2 bits BCH error correcting codes using FPGA // Journal of Telecommunications. 2013. V. 19, is. 2. P. 11–17.
15. Lakhendra K. FPGA implementation of (15, 7) BCH encoder and decoder for audio message // International Journal of Engineering Sciences & Research Technology (IJESRT). 2014. No. 3 (8). P. 407–413.
16. Закон Амдала и будущее многоядерный процессоров. URL: <https://www.osp.ru/os/2009/04/9288815/> (дата обращения: 23.05.2018).
17. Буркатовская Ю.Б., Мальчуков А.Н., Осокин А.Н. Быстродействующие алгоритмы деления полиномов в арифметике по модулю два // Известия Томского политехнического университета. Разд. Автоматика. Вычислительная техника. 2006. Т. 309, вып. 1. С. 19–24
18. Харрис Д.М., Харрис С.Л. Цифровая схемотехника и архитектура компьютера. 2-е изд. Нью Йорк : Morgan Kaufman, 2013. 1662 с.

Поступила в редакцию 16 мая 2018 г.

Ryzhova S.E., Mytsko E.A., Malchukov A.N. (2018) COMPARATIVE ANALYSIS OF BCH-CODES DECODERS IMPLEMENTATIONS WITH PARAMETERS (15, 7, 5) USING FPGA COVARIATES. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 46. pp. 98–107

DOI: 10.17223/19988605/46/12

In this paper we consider the possibility of implementing various decoder of the Bose-Chaudhuri-Hocquingham code (BCH-codes) using field-programmable gate array (FPGA). BCH codes are most common error-correcting codes that are used in many data transmission systems for detecting and correcting errors. BCH-codes with short-length of codewords and with a small number of correcting bits are used in automation systems of production. The code that implemented in the article has the following parameters:

- $n = 15$  – the length of the input code combination;
- $m = 7$  – the length of the information block;
- $k = 8$  – the length of the message control block;
- $t = 2$  – the number of errors, that code corrects;
- $d = 5$  – the maximum Hamming distance between two combinations;
- $g(x) = 111010001$  – the generating polynomial.

The paper describes implementations of two simplest decoding methods: a tabular, that based on the application of a table with precalculated errors syndromes; and cyclic, that based on the checking of the weight of remainder of the division of the codeword by the generating polynomial.

The simplest implementation is the tabular decoding method. The basis of this method is the table, which stores all possible values of error patterns up to the multiplicity of  $t$ . Addresses of error patterns are error syndromes. Having such the table, the decoding process is reduced to selecting a pattern from the table at the address, which is equal to the remainder of the division of the input codeword by the generating polynomial. The error pattern that was selected from the table is summed up modulo two with the input codeword. The corrected codeword arrives to the output of the decoder. The main drawback of this method it is the necessary to build the table and connect additional FPGA resources to store this table.

The second decoding method, which is easy to implement, is a cyclic method. This decoding method was designed for codes with the cyclicity property, and it suitable for BCH-codes. This method based on the checking the condition  $w \leq t$  of the remainder of the division the input codeword by the generating polynomial.

For the first implementations of these methods, the process of division the input codeword by the generating polynomial is performed using a linear feedback register (LROS), and have a clear sequential structure. The maximum number of cycles required for the decoding process in devices, which was implemented based on table decoding methods is 36 cycles with frequency 275.2 MHz. And for decoder based on cyclic decoding method it is necessary 627 cycles with frequency 192.8.

The first modification was the using of the parallel structure is used instead of the sequential one, for the decoder based on the cyclic decoding method, to improve the performance. In this case, the number of cycles decreased from 627 and frequency 192.8 MHz to 45 cycles with frequency 179.3 MHz.

In order to make it possible to overcome the limit of the number of cycles, in accordance with Amdahl's law, was using a modification at the level of the used algorithms. For that, the classical procedure of polynomials division was replaced by the matrix division. It allows to perform the process of division the input codeword by the generator polynomial in 1 - 2 cycles. Such a replacement allowed reducing the number of cycles in implementations from 36 cycles with frequency 275.2 to 7 cycles with frequency 275.2 MHz; from 627 cycles with frequency 192.8 MHz to 192 cycles with frequency 200.1 MHz and in the end from 45 cycles with frequency 179.3 MHz to 16 cycles with frequency 174.1 MHz.

The latest modification was a changing of the design logic of devices. It was used for the decoder which based on the cyclic decoding method, had the parallel structure and matrix division of polynomials. For that decoder which based on a combinational logic it is enough the 1 cycle for decoding procedure.

In this article are given a summary table of results by the main characteristics of implemented devices, and it also has a full description of all the values, which are given in that table.

Keywords: error-correction codes; CBH-codes; cyclic decoding methods; table decoding methods, hardware-based devices; FPGA.

*RYZHOVA Svetlana Evgenievna* (Post-graduate student of Division for Information Technologies of School of Computer Science and Robotics of National Research Tomsk Polytechnic University, Russian Federation).

E-mail: r.svet93@yandex.ru

*MYTSKO Evgeniy Alekseevich* (Post-graduate student of Division for Information Technologies of School of Computer Science and Robotics of National Research Tomsk Polytechnic University, Russian Federation).

E-mail: evgenrus70@mail.ru

*MALCHUKOV Andrey Nikolayevich* (Candidate of Technical Sciences, Associate Professor of Division for Information Technologies of School of Computer Science and Robotics of National Research Tomsk Polytechnic University, Russian Federation).

E-mail: jgs@tpu.ru

#### REFERENCES

1. Zolotarev, V.V. & Ovechkin, G.V. (2004) *Pomekhoustoychivoe kodirovanie. Metody i algoritmy* [Error-correcting coding. Methods and algorithms]. Moscow: Goryachaya liniya-Telekom.
2. Morelos-Saragosa, R.M. (2005) *Iskusstvo pomekhoustoychivogo kodirovaniya. Metody, algoritmy, primeneniye* [The Art of Error Correcting Coding]. Translated from english by V. Afanasiev. Moscow: Tekhnosfera.
3. Digital Video Broadcasting (DVB). (2010) *Frame structure channel coding and modulation for a second generation digital transmission system for cable systems (DVB-C2)*. DVB Document A138. 2010. pp. 111.
4. Anon. (n.d.) *Protokol POCSAG. Protokoly peydzhingovoy sistemy svyazi* [Protocol POCSAG. Protocols of paging communication system]. [Online] Available from: [https://studopedia.su/2\\_56965\\_protokol-POCSAG.html](https://studopedia.su/2_56965_protokol-POCSAG.html). (Accessed: 23rd May 2018)
5. Slyusar, V. (2010) *Peredacha dannykh s borta BPLA: standarty NATO* [Data transmission from aboard the UAV: NATO standards]. *Elektronika: nauka, tekhnologiya, biznes – Electronics: Science, Technology, Business*. 3. pp. 81–86.
6. Bereznyuk, N.G. (ed.) (1978) *Kodirovanie informatsii: dvoichnye kody* [Information coding: Binary codes]. Kharkov: Vishcha shkola.
7. Polyakov, A.K. (2003) *Yazyki VHDL I VERILOG v proektirovanii tsifrovoy apparatury* [Languages VHDL and VERILOG in the design of digital equipment]. Moscow: Solon-Press.
8. Prakash, G. & Muthamizhse, I. (2016) FPGA Implementation of Bose Chaudhuri Hocquenghem Code (Bch) Encoder and Decoder for Mul-tiple Error Correction Control. *International Journal of Innovative Research in Science, Engineering and Technology*. 5(3). pp. 4467–4473. DOI: 10.15680/IJRSET.2016.0503253
9. Panda, A., Sarik, S. & Awasthi, A. (2012) FPGA implementation of encoder for (15, k) binary BCH code using VHDL and performance comparison for multiple error correction control. *International Conference on Communication Systems and Network Technologies (CSNT-2012)*. May 12–13, 2012.
10. Rohith, S. & Pavithra, S. (2013) FPGA Implementation of (15,7) BCH encoder and decoder for text message. *International Journal of Research in Engineering and Technology*. 2(9). pp. 209–214. DOI: 10.15623/ijret.2013.0209031
11. Hiremath, M. & Devi, M. (2014) A novel method implementation of a FPGA using (n, k) binary BCH code. *International Journal of Research in Engineering Technology and Management*. Special is. pp. 1–8.
12. Sunita, M.S., Chiranth, V., Akash, H.C. & Bhaaskaran, V.S.K. (2015) Pipeline architecture for fast decoding of bch codes For nor flash memory. *ARN Journal of Engineering and Applied Sciences*. 10(8). pp. 3397–3404.
13. Yathiraj, H. & Hiremath, R. (2014) Implementation of BCH code (n, k) encoder and decoder for multiple error correction control. *International Journal of Computer Science and Mobile Applications*. 2(5). pp. 45–54.
14. Mohammed, S. & Abdulsada, H. (2013) Design and implementation of 2 bits BCH error correcting codes using FPGA. *Journal of Telecommunications*. 19(2). pp. 11–17.
15. Lakhendra, K. (2014) FPGA implementation of (15, 7) BCH encoder and decoder for audio message. *International Journal of Engineering Sciences & Research Technology (IJESRT)*. 3(8). pp. 407–413.
16. Chernyak, L. (2009) *Zakon Amdala i budushchee mnogoyadernyy protsessorov* [Amdahl's law and the future of multi-cores processors]. [Online] Available from: <https://www.osp.ru/os/2009/04/9288815/>. (Accessed: 23rd May 2018).
17. Burkatovskaya, Yu.B., Malchukov, A.N. & Osokin, A.N. (2006) *Bystrodeystvuyushchie algoritmy deleniya polinomov v arifmetike po modulyu dva* [The high-speed algorithms of the polynomials division in arithmetic module two]. *Izvestiya Tomskogo politekhnicheskogo universiteta. Razd. Avtomatika. Vychislitel'naya tekhnika – Bulletin of Tomsk Polytechnic University*. 309(1). pp. 19–24
18. Kharris, D.M. & Kharris, S.L. (2013) *Tsifrovaya skhemotekhnika i arkhitektura komp'yutera* [Digital Design and Computer Architecture]. Translated from English. 2nd ed. New York: Morgan Kaufman.