Аппаратная реализация реконфигурируемого на лету БЧХ декодера

Э. Э. Гасанов, П. А. Пантелеев, А. П. Соколов, Ю. С. Шуткин

В статье предлагается архитектура реконфигурируемого на лету аппаратного БЧХ декодера.

Ключевые слова: аппаратная реализация, структурные автоматы, помехоустойчивые коды, коды БЧХ.

Введение

Задача аппаратной реализации каких-либо устройств с одной стороны сугубо инженерная, но с другой стороны опирается на фундаментальные результаты теории синтеза управляющих систем. Среди последних работ по этой теории можно выделить [1–19]. Если устройство надо реализовать в виде чипа, то управляющая система, моделирующая чип, — это конечный автомат. Последние работы по теории автоматов можно найти в [20–50]. Фактически задача реализации устройства в виде чипа — это задача синтеза структурного автомата с заданным функционированием. Обзор результатов по этим направлениям можно найти в статье В.Б.Кудрявцева [51]. Помехоустойчивые коды имеют широкий диапазон применений в системах оптической и беспроводной связи, в магнитной записи, в системах хранения данных и т.д. Некоторые последние работы по теории кодирования опубликованы в [52–60].

Бинарные БЧХ коды представляют собой мощный класс помехоустойчивых кодов. При систематическом БЧХ кодировании информационные биты дополняются проверочными битами, и полученная последовательность бит образует кодовые слова. Каждый двоичный БЧХ код с полем расширения $GF(2^m)$ определяется длиной кодового слова n и максимальным числом исправляемых ошибок t. Если в канале связи происходит не более t ошибок, то БЧХ декодер может восстановить исходное кодовое слово.

Большинство аппаратных реализаций БЧХ декодеров имеют дело с фиксированной длиной кодового слова и фиксированным числом исправляемых ошибок. Это означает, что эти параметры фиксируются во время создания экземпляра декодера и не могут быть изменены во время выполнения. Однако в современных системах хранения данных, таких как контроллеры флэш-памяти и во многих других системах требуется поддерживать множество различных длин кодового слова и максимальных чисел исправляемых ошибок в одной конструкции. БЧХ декодеры, используемые в контроллерах таких систем, должны быть конфигурируемыми и время реконфигурации должно быть как можно меньше.

В данной работе мы предлагаем новую реконфигурируемую на лету аппаратную схему БЧХ декодера. Это означает, что изменение конфигурации в этой конструкции может быть сделано за константное число тактов, независящее от длины кодового слова и числа исправляемых ошибок. Данное решение защищено патентом США [61].

Типичное решение

Схема типичной аппаратной реализации БЧХ декодера показана на рис. 1. Декодирование состоит из трех этапов: вычисления синдрома, решения ключевого уравнения и коррекции ошибок. Если этапы реализованы на аппаратном уровне, модуль вычисления синдрома (Syndrome Calculation, SC), получает кодовое слово символ за символом и множество вычислен-

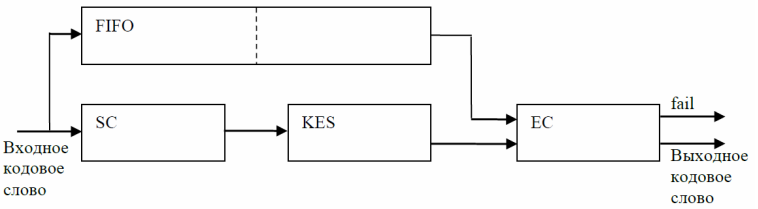

Рис. 1. Схема типичной аппаратной реализации БЧХ декодера

ных синдромов передает модулю решения ключевых уравнений (Key Equation Solver, KES). Модуль KES вычисляет полином локаторов ошибок, корнями которого являются позиции ошибок. Для решения ключевых уравнений часто используют алгоритм Берликемпа-Мэсси (Berlekamp-Massey algorithm, BMA). Модуль KES передает полином локаторов ошибок модулю коррекции ошибок (Error Correction, EC). Входные данные также поступают на модуль FIFO (очередь), который сохраняет данные до их использования в модуле EC. В модуле FIFO может помещаться до двух кодовых слов. На выходе модуля EC получается исправленное кодовое слово. Двоичный выход "fail"модуля EC устанавливается в 1, если число исправленных бит больше, чем максимальное возможное число ошибок.

Конфигурируемый БЧХ декодер

Большинство реализаций БЧХ декодеров не позволяют пользователю изменять параметры БЧХ кода, такие как максимальное число ошибок и длина кодового слова. Однако современные приложения кодов БЧХ в контроллерах твердотельных дисков (solid-state disk, SSD) делают необходимым изменение этих параметров во время функционирования. При этом, чтобы достичь быстрой скорости время изменения конфигурации такого контроллера должен быть как можно меньше. Поэтому реконфигурируемые декодеры должны иметь специальный вход (см. рис. 2) под названием "Данные конфигурации". Он состоит из пары

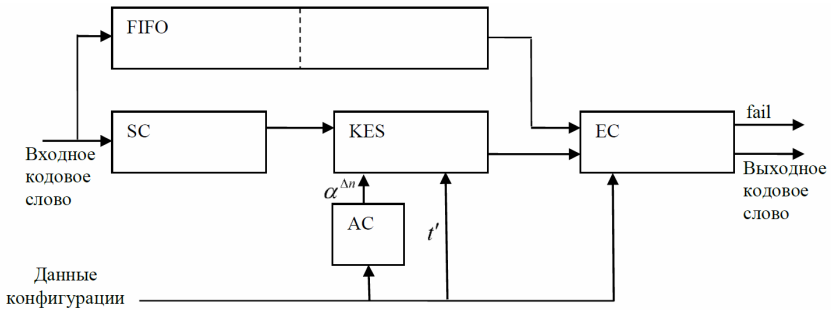

Рис. 2. Схема аппаратной реализации конфигурируемого БЧХ декодера

(n', t'), где n' — текущая длина кодового слова, t' — текущее максимальное число исправляемых ошибок.

Для того, чтобы понять основную задачу в реализации реконфигурируемых БЧХ декодеров мы должны объяснить алгоритм декодирования БЧХ более подробно. Вход БЧХ декодера есть кодовое слово $(c_{n-1}, \ldots, c_0)$, где каждый символ $c_i \in \{0, 1\}$. Мы будем рассматривать его также как полинома кодового слова $c(x) = c_{n-1}x^{n-1} + \ldots + c_1 x + c_0$. Этот полином используется модулем SC, который вычисляет $2t$ синдромов $S_1, S_2, \ldots, S_{2t}$ следующим образом: $S_i = c(\alpha^i)$, $i = 1, 2, \ldots, 2t$, где t — максимальное число ошибок, которое БЧХ код может исправить, а α — примитивный элемент поля расширения $GF(2^m)$, связанного с этим кодом БЧХ. В этом случае длина кодового слова равна $n = 2^m - 1$. Затем эти синдромы приходят к модулю KES, который с помощью алгоритма решения ключевых уравнений (например, BMA) производит полином локаторов ошибок $\Lambda(x)$. Затем модуль EC, используя $\Lambda(x)$, корректирует позиции ошибок в кодовом слове, задержанном модулем FIFO, и сообщает об неудаче, если количество ошибок больше, чем максимально возможное число ошибок t.

Не трудно изменить схему 1, так чтобы она могла обрабатывать БЧХ коды с числом ошибок $t' < t$. Единственное отличие состоит в том, что KES блок должен выполнять $2t'$ итераций вместо $2t$. Но если мы хотим использовать БЧХ код с

другой длиной $n' < n$, то мы должны использовать усеченные коды БЧХ. Это означает, что вместо кодового слова полной длины $(c_{n-1}, \ldots, c_0)$ на вход БЧХ декодера будет поступать усеченное кодовое слово $(c_{n'-1}, \ldots, c_0)$, которое можно рассматривать как кодовое слово полной длины $(c_{n'-1}, \ldots, c_0, 0 \ldots, 0)$ или в полиномиальной форме $c(x) = x^{\Delta n} c'(x)$, где $c'(x) = c_{n'-1}x^{n'-1} + \ldots + c_1 x + c_0$ и $\Delta n = n - n' = 2^m - 1 - n'$. Следовательно, если мы будем использовать стандартную схему для вычисления синдромов то она вместо синдромов $S_i = c(\alpha^i) = \alpha^{i\Delta n} c'(\alpha^i)$, $i = 1, 2, \ldots, 2t$ произведет значения $S'_i = c'(\alpha_i)$. Так что, если мы хотим получить правильные значения синдромов $S_1, S_2, \ldots, S_{2t}$, то мы должны сначала вычислить значения $S'_1, S'_2, \ldots, S'_{2t}$, а затем использовать формулу $S_i = \alpha^{i\Delta n} S'_i$, $i = 1, 2, \ldots, 2t$. Основная проблема состоит в том, что Δn зависит от параметра конфигурации n' — текущей длины кодового слова, и значения $\alpha^{\Delta n}, \alpha^{2\Delta n}, \ldots, \alpha^{2t\Delta n}$ не могут быть вычислены за небольшое фиксированное число тактов, поскольку величина Δn может быть очень большой.

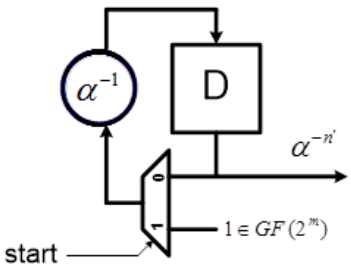

Рис. 3. Схема Альфа Калькулятора

Основная идея данной работы заключается в вычислении значения $\alpha^{\Delta n}$ одновременно с вычислением синдромов $S'_1, S'_2, \ldots, S'_{2t}$, а затем последовательном выполнении умножений $S_1 = \alpha^{\Delta n} S'_1, S_2 = \alpha^{2\Delta n} S'_2, \ldots, S_{2t} = \alpha^{2t\Delta n} S'_{2t}$, по мере того как значения $S_1, S_2, \ldots, S_{2t}$ используются в KES блоке. Для того, чтобы упростить вычисление $\alpha^{\Delta n}$, заметим, что $\alpha^{\Delta n} = \alpha^{2^m - 1 - n'} = \alpha^{-n'}$ так как $\alpha^{2^m - 1} = 1$ в поле $GF(2^m)$. Так что для того, чтобы вычислить $\alpha^{-n'}$, мы можем использовать константт-

ный умножитель в поле $GF(2^m)$, который выполняет умножение на α^{-1}. Модуль, который выполняет эти вычисления, называется Альфа Калькулятором (Alpha Calculator, AC) и реализован, как показано на рис. 3. Если сигнал start $= 1$, то вычисления запускаются. Модуль AC работает одновременно с модулем SC (см. рис. 2) и в конце вычислений он производит значение $\alpha^{-n'}$.

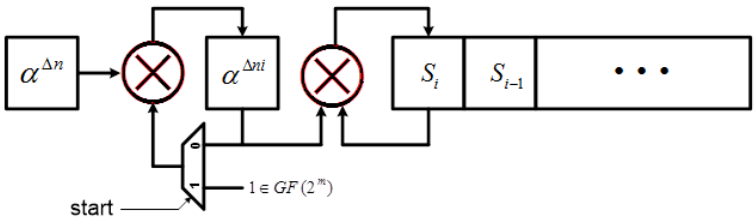

Рис. 4. Схема обновления синдромов

После того, как значение $\alpha^{-n'}$ получено, оно идет к KES блоку (см. рис.2). В нашей реализации модуля KES значения синдромов $S_1, S_2, \ldots, S_{2t}$ используются последовательно. На первой итерации используется только S_1, на второй — только S_1, S_2, и т.д. Поэтому, мы имеем достаточно времени, чтобы вычислить все значения $S_1 = \alpha^{\Delta n} S'_1, S_2 = \alpha^{2\Delta n} S'_2, \ldots, S_{2t} = \alpha^{2t\Delta n} S'_{2t}$, используя только два умножителя в поле $GF(2^m)$, как показано на рис. 4. На этом рисунке сигнал "start"выводится из блока SC и если start $= 1$, то это означает, что модули SC и AC закончили свои расчеты и модуль KES должен начать работать.

Список литературы

[1] Калачев Г.В. Порядок мощности плоских схем, реализующих булевы функции // Дискретная математика, 2014, том 26:1, 49-–74.

[2] Годнева А.В. Умножение с параметром и его применение в криптографии // Интеллектуальные системы. 2014. 18:1. 61-74.

[3] Плетнев А.А. Информационно-графовая модель динамических баз данных и ее применение// Интеллектуальные системы. — 2014. Т. 18, Вып. 1. — С. 111–140.

[4] Перпер Е.М. Нижние оценки временной и объёмной сложности задачи поиска подслова // Дискретная математика, 2014, том 26:2, 58–70.

[5] Калачев Г.В. Нижние оценки мощности плоских схем, реализующих частичные булевы операторы // Интеллектуальные системы. 2014. 18:2. 279–322.

[6] Половников В.С. Особенности нейронных схем Мак-Каллока-Питтса над полем рациональных чисел // Интеллектуальные системы. 2014. 18:2. 331–336.

[7] Kalachev G.V. Order of power of planar circuits implementing Boolean functions // Discrete Mathematics and Applications. — 2014. Volume 24, Issue 4, Pages 185–205.

[8] Гасанов Э.Э., Ефремов Д.В. Фоновый алгоритм решения двумерной задачи о доминировании // Интеллектуальные системы. 2014. 18:3. 133–158.

[9] Сытдыков Т.Р. Линейный алгоритм построения деревьев разводки сигнала // Интеллектуальные системы. 2014. 18:3. 175–202.

[10] Шуткин Ю.С. Моделирование схемных управляющих систем // Интеллектуальные системы. 2014. 18:3. 253–261.

[11] Лебедев А.А. О задачах оптимального распределения ресурсов и проверки устойчивости для схем функциональных элементов в k-значной логике // Интеллектуальные системы. 2014. 18:4. 95-118.

[12] Perper E.M. Lower bounds of temporal and spatial complexity of the substring search problem // Discrete Mathematics and Applications. — 2014. Volume 24, Issue 6, Pages 373–382.

[13] Перпер Е.М. Порядок сложности задачи поиска в множестве слов вхождений подслова // Интеллектуальные системы. 2015. 19:1, 99–116.

[14] Плетнев А.А. Динамическая база данных, допускающая параллельную обработку произвольных потоков запросов // Интеллектуальные системы. 2015. 19:1, 117-142.

[15] Плетнев А.А. Логарифмическая по сложности параллельная обработка автоматами произвольных потоков запросов к динамической базе данных // Интеллектуальные системы. 2015. 19:1, 171–212.

[16] Плетнев А.А. Нижняя оценка на область видимости автомата, обрабатывающего произвольный поток запросов к динамической базе данных // Интеллектуальные системы. 2015. 19:4, 117–151.

[17] Сытдыков Т.Р. Построение деревьев разводки сигнала // Интеллектуальные системы. 2015. 19:4, 211–246.

[18] Курганов Е.А. О глубине аппаратной реализации блочного шифра Кузнечик // Интеллектуальные системы. 2016. 20:1, 61–78.

[19] Плетнев А.А. Минимально возможный по степени ветвления информационный граф с радиусом видимости один, обрабатывающий произвольный поток запросов к динамической базе данных // Интеллектуальные системы. 2016. 20:1, 223–254.

[20] Пархоменко Д.В. Порожденные автоматами р-языки // Дискретная математика, 2014, том 26:1, 96–102.

[21] Александров Д.Е. Эффективные методы реализации проверки содержания сетевых пакетов регулярными выражениями // Интеллектуальные системы. 2014. 18:1. 37–60.

[22] Титова Е.Е. Конструирование движущихся изображений клеточными автоматами // Интеллектуальные системы. 2014. 18:1. 153–180.

[23] Бабин Д.Н. Частотные регулярные языки // Интеллектуальные системы. 2014. 18:1. 205–210.

[24] Иванов И.Е. О некоторых свойствах автоматов с магазинной памятью // Интеллектуальные системы. 2014. 18:1. 243–252.

[25] Часовских А.А. Проблема А-полноты линейно-автоматных функций над конечным полем // Интеллектуальные системы. 2014. 18:1. 253–258.

[26] Parkhomenko D.V. Automata generated p-languages // Discrete Mathematics and Applications. — 2014. Volume 24, Issue 4, Pages 207–212.

[27] Часовских А.А. Условия полноты линейно-р-автоматных функций // Интеллектуальные системы. 2014. 18:3. 203–252.

[28] Александров Д.Е. Об оценках автоматной сложности распознавания класса регулярных языков // Интеллектуальные системы. 2014. 18:4. 121–146.

[29] Дементьев В.М. О звездной высоте регулярного языка и циклической сложности минимального автомата // Интеллектуальные системы. 2014. 18:4. 159–166.

[30] Кучеренко И.В. О минимизации монофункциональных классов бинарных клеточных автоматов с неразрешимым свойством обратимости // Интеллектуальные системы. 2014. 18:4. 171–242.

[31] Якимец К.К. Об инвариантности характеристик конфигураций однородных структур // Интеллектуальные системы. 2014. 18:4. 271–282.

[32] Гасанов Э.Э. Прогнозирование периодических сверхсобытий автоматами // Интеллектуальные системы. 2015. 19:1. 23–34.

[33] Иванов И.Е. О сохранении периодических последовательностей автоматами с магазинной памятью с однобуквенным магазином // Интеллектуальные системы. 2015. 19:1, 145–160.

[34] Летуновский А.А. Выразимость линейных автоматов относительно расширенной суперпозиции // Интеллектуальные системы. 2015. 19:1, 161–170.

[35] Александров Д.Е. Об оценках мощности некоторых классов регулярных языков // Дискретная математика, 2015, том 27:2, 3–21.

[36] Часовских А.А. Проблема полноты для класса линейно-автоматных функций // Дискретная математика, 2015, том 27:2, 134–151.

[37] Гербуз В.Г. О связи функций автомата и автоматной функции // Интеллектуальные системы. 2015. 19:2, 135–142.

[38] Миронов А.М. Критерий реализуемости функций на строках вероятностными автоматами Мура с числовым выходом. 2015. 19:2, 175–186.

[39] Терехина И.Ю. Модель невлияния для квантовых автоматов // Интеллектуальные системы. 2015. 19:2, 209–216.

[40] Васильев Д.И. О стабилизации одной динамической системы, связанной с автоматным моделированием миграционных процессов // Интеллектуальные системы. 2015. 19:3, 27–38.

[41] 31. Бабин Д.Н., Летуновский А.А. О возможностях суперпозиции, при наличии в базисе автоматов фиксированной добавки из булевых функций и задержки // Интеллектуальные системы. 2015. 19:3, 71–78.

[42] 33. Бабин Д.Н. Автоматы с суперпозициями, пример нерасширяемости до предполного класса // Интеллектуальные системы. 2015. 19:3, 87–94.

[43] Гасанов Э.Э. Мастихина А.А. Прогнозирование общерегулярных сверхсобытий автоматами // Интеллектуальные системы. 2015. 19:3, 127–154.

[44] Иванов И.Е. Нижняя оценка на максимальную длину периода выходной последовательности автономного автомата с магазинной памятью // Интеллектуальные системы. 2015. 19:3, 175–194.

[45] Часовских А.А. Критериальные системы в классах линейно-автоматных функций над конечными полями // Интеллектуальные системы. 2015. 19:3, 195–207.

[46] Алешин С.В., Пантелеев П. А. Конечные автоматы и числа // Дискретная математика, 2015, том 27:4, 3–20.

[47] Миронов А.М. Основные понятия теории вероятностных автоматов // Интеллектуальные системы. 2015. 19:4, 75–116.

[48] Alexandrov D.E. Cardinality estimates for some classes of regular languages // Discrete Mathematics and Applications. — 2015. Volume 25, Issue 6, Pages 323–337.

[49] Орлов В.А. О конечных автоматах с максимальной степенью различимости состояний // Интеллектуальные системы. 2016. 20:1, 213–222.

[50] Chasovskikh A.A. Completeness problem for the class of linear automata functions // Discrete Mathematics and Applications. — 2016. Volume 26, Issue 2, Pages 89–104.

[51] Кудрявцев В.Б. Кафедра математической теории интеллектуальных систем (МаТИС) // Интеллектуальные системы. 2014. 18:2. 5-30.

[52] Дергач П.С. Об однозначности алфавитного декодирования общерегулярных сверхязыков // Дискретная математика, 2014, том 26:1, 32-–48.

[53] Рыков Д.О. О правильных семействах функций, используемых для задания латинских квадратов // Интеллектуальные системы. 2014. 18:1. 141–152.

[54] Дергач П.С. О каноническом регулярном представлении s-тонких языков // Интеллектуальные системы. 2014. 18:1. 211–242.

[55] Плаксина И.А. Построение параметрического семейства многомерных латинских квадратов // Интеллектуальные системы. 2014. 18:2. 323–330.

[56] Тожибаева М.Э. Верхняя оценка минимального расстояния квазициклических низкоплотностных кодов // Интеллектуальные системы. 2014. 18:2. 337–343.

[57] Dergach P.S. On uniqueness of alphabetical decoding of ?-regular languages // Discrete Mathematics and Applications. Volume 24, Issue 3, Pages 139—152.

[58] Дергач П.С. О проблеме вложения допустимых классов // Интеллектуальные системы. 2015. 19:2, 143–174.

[59] Дергач П.С. О двух размерностях спектров тонких языков // Интеллектуальные системы. 2015. 19:3, 155–174.

[60] Шульгина Е.А. Оценка параметров бирегулярных двудольных графов // Интеллектуальные системы. 2016. 20:1, 225–263.

[61] Panteleev P. A., Gasanov E. E., Neznanov I. V., Sokolov A. P., Shutkin Yu. A. Reconfigurable BCH decoder. United States Patent: 8,621,329, December 31, 2013.