

Московский физико-технический институт
(государственный университет)
Физтех-школа радиотехники и компьютерных технологий
Кафедра информатики и вычислительной техники

Разработка системы обнаружения и коррекции ошибок кодом BCH для контроллера NAND-Flash памяти

Выполнил: Мотин Н.А. М01-003а
Научный руководитель: к.т.н. Юрлин С.В.
Консультант: Акимов В.А.

Москва 2022

Введение

- NAND-Flash память типов SLC, MLC и TLC широко используется во многих потребительских электронных устройствах, включая смартфоны, планшетные ПК и мобильные интернет-устройства.
- В ходе предыдущей работы был разработан контроллер NAND Flash (NFMC), поддерживающий ONFI интерфейс
- Из-за внутренних процессов NAND Flash, повреждающих сохраненные данные, необходимо защищать информацию с помощью специальных кодов
- Для коррекции ошибок в NAND Flash обычно используются несколько кодов: Hamming, Reed-Solomon, BCH, LDPC

Введение

- Ошибки в NAND Flash начинают появляться при первом использовании микросхемы, и со временем частота их появления увеличивается
- Для каждой модели микросхемы NAND Flash производитель устанавливает требования к количеству исправляемых бит в блоке данных
- Количество циклов записи/стирания гарантируется только при соблюдении требований к количеству исправляемых ошибок
- Под требования к способности исправления ошибок NAND Flash типа TLC подходят только коды BCH и LDPC
- У кода LDPC сложность реализации в аппаратуре заметно превосходит BCH
- BCH коды – это широкий класс циклических кодов, применяемых для защиты информации от ошибок. Отличается возможностью построения кода с заранее определёнными корректирующими свойствами
- Прimitивный двоичный BCH код – BCH код, размер кодового слова которого равен $2^m - 1$
- В данной работе реализуются примитивные двоичные BCH-коды

Цель работы

Обеспечить надёжное хранение данных в NAND Flash типов SLC, MLC, TLC путем разработки системы коррекции ошибок кодом BCH для NFMC (NFMC-BCH)

Требования

- Внедрение системы не должно вносить существенные изменения в архитектуру NFMC
- Скорость работы NFMC-BCH на частоте 1 ГГц должна быть больше скорости чтения NAND Flash
- Система должна иметь возможность программной параметризации кода BCH

Задачи

- Разработка архитектуры NFMC-BCH
- Разработка RTL-описания NFMC-BCH
- Разработка программы-генератора конфигурации NFMC-BCH
- Инженерная верификация разработанного RTL-описания

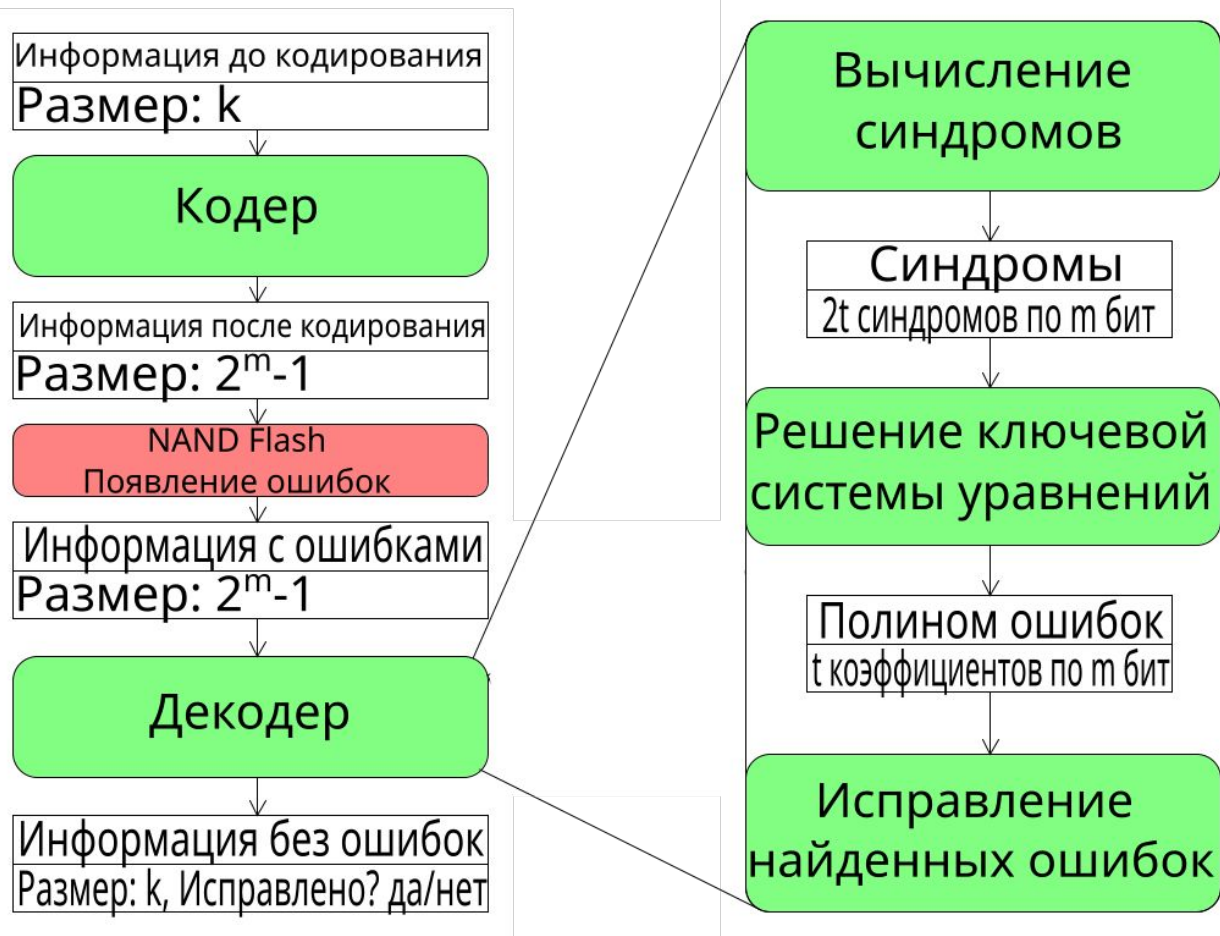
Математическая справка

- Кодирование и декодирование кодом BCH реализуется через операции в поле Галуа
- Полином степени m – многочлен вида $p(x)=p_0+p_1x^1+p_2x^2+\dots+p_mx^m$, где $p_i \in \{0,1\}$
- Поле Галуа степени 2^m - поле состоящее из всех возможных полиномов степени $(m-1)$, в котором определены операции сложения и умножения. В дальнейшем $GF(2^m)$.
- Полином степени m представим в виде битового вектора из его коэффициентов: $(p_0, p_1 \dots p_m)$
- Примитивный полином степени m - неприводимый полином, для которого (x^n-1) делит его только для $n \geq 2^m-1$
- Умножение в $GF(2^m)$ - умножение полиномов по модулю примитивного полинома степени m .
- Сложение в $GF(2^m)$ - xor битовых векторов
- Деление полиномов с остатком представляется как умножение вектора на матрицу

Разработка архитектуры NFMC-ВСН

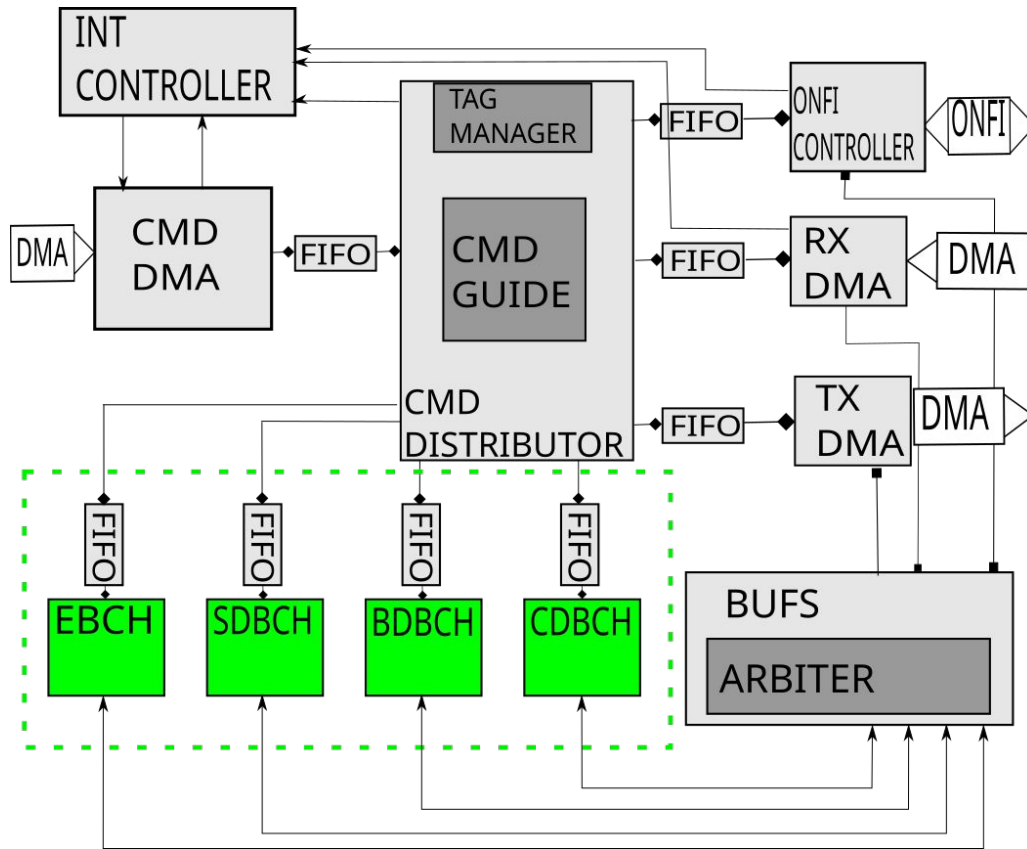
Основные параметры ВСН кода:

- m – степень 2ки поля Галуа в котором построен код
- $n = 2^m - 1$ – размер кодового слова (бит)
- t – количество исправляемых ошибок
- k - максимальный размер кодируемой информации (бит)



Разработка архитектуры NFMC-BCH

Интеграция модулей BCH в NFMC



- EBCH - кодер BCH
- SDBCH - первая стадия декодера BCH
- BDBCH - вторая стадия декодера BCH
- CDBCH - третья стадия декодера BCH
- CMD DMA — модуль чтения команд по DMA в fifo.
- CMD DISTRIBUTOR — модуль распределения команд.
- TAG MANAGER — менеджер таблицы тэгов.
- CMD GUIDE — модуль разделения команд по функциям.
- RX DMA — модуль чтения данных по DMA.
- TX DMA — модуль записи данных по DMA.
- ONFI CONTROLLER — контроллер ONFI
- BUFS – система буферов
- ARBITER – приоритетный арбитр доступа к буферам.
- INT CONTROLLER – контроллер прерываний

Разработка RTL-описания NFMC-VCH.

Кодер

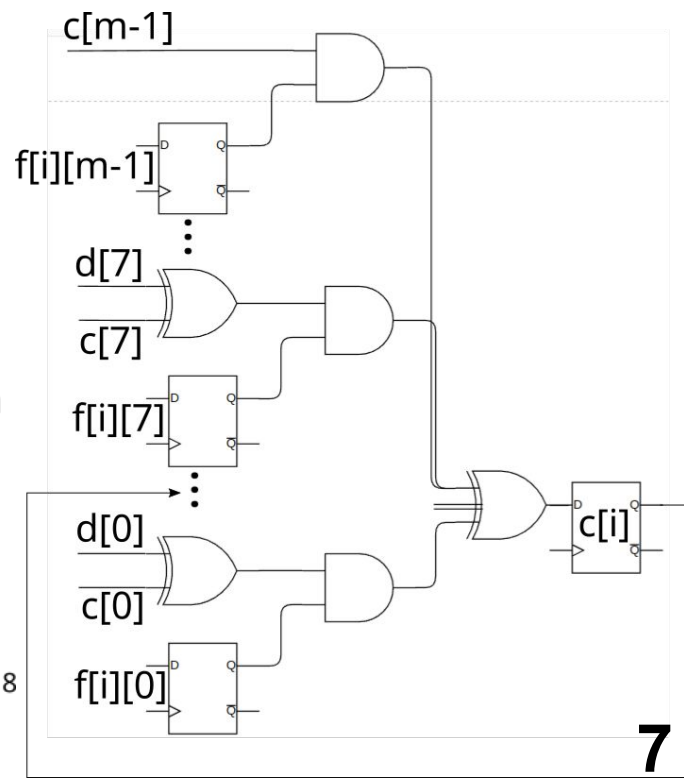
- Задача кодера - посчитать дополнительные биты, которые записываются в NAND-Flash вместе с данными
- Дополнительные биты это остаток от деления информации на полином BCH кода
- Байты данных последовательно подаются в кодер, где умножаются на матрицу
- В NFMC-VCH матрица кодера заполняется программно

$$C_{i+1} = F^8(C_i + D[8*(i+1):i*8])$$
$$C_0 = 0$$

- m - количество добавочных бит
- p - полином BCH кода
- F - матрица побитного умножения
- F^8 - матрица побайтного умножения
- c - накапливаемые дополнительные биты
- d - входные данные
- f - значения ячеек матрицы F^8

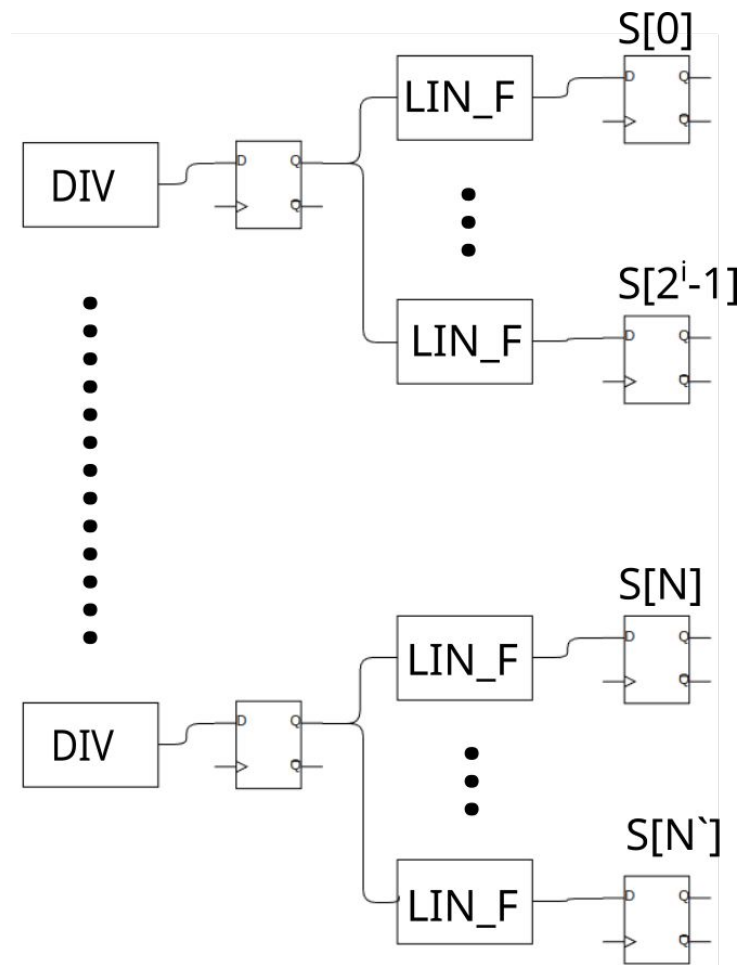
$$F = \begin{matrix} \begin{matrix} \uparrow \\ m \\ \downarrow \end{matrix} & \begin{bmatrix} p_{m-1} & 1 & 0 & \dots & 0 \\ p_{m-2} & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_1 & 0 & 0 & \dots & 1 \\ p_0 & 0 & 0 & \dots & 0 \end{bmatrix} \\ & \begin{matrix} \leftarrow \\ m \\ \rightarrow \end{matrix} \end{matrix}$$

$$F^8 = \begin{matrix} \begin{matrix} \uparrow \\ m \\ \downarrow \end{matrix} & \begin{bmatrix} p'_{m-1} & p''_{m-1} & 0 & \dots & 0 \\ p'_{m-2} & p''_{m-2} & 1 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p'_8 & p''_8 & 0 & 0 & 1 \\ p'_7 & p''_7 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p'_0 & p''_0 & 0 & 0 & 0 \end{bmatrix} \\ & \begin{matrix} \leftarrow \\ 8 \\ \rightarrow \end{matrix} \end{matrix} \quad \begin{matrix} \uparrow \\ 8 \\ \downarrow \end{matrix}$$



Разработка RTL-описания NFMC-VCH. Декодер. Первая стадия(Вычисление синдромов)

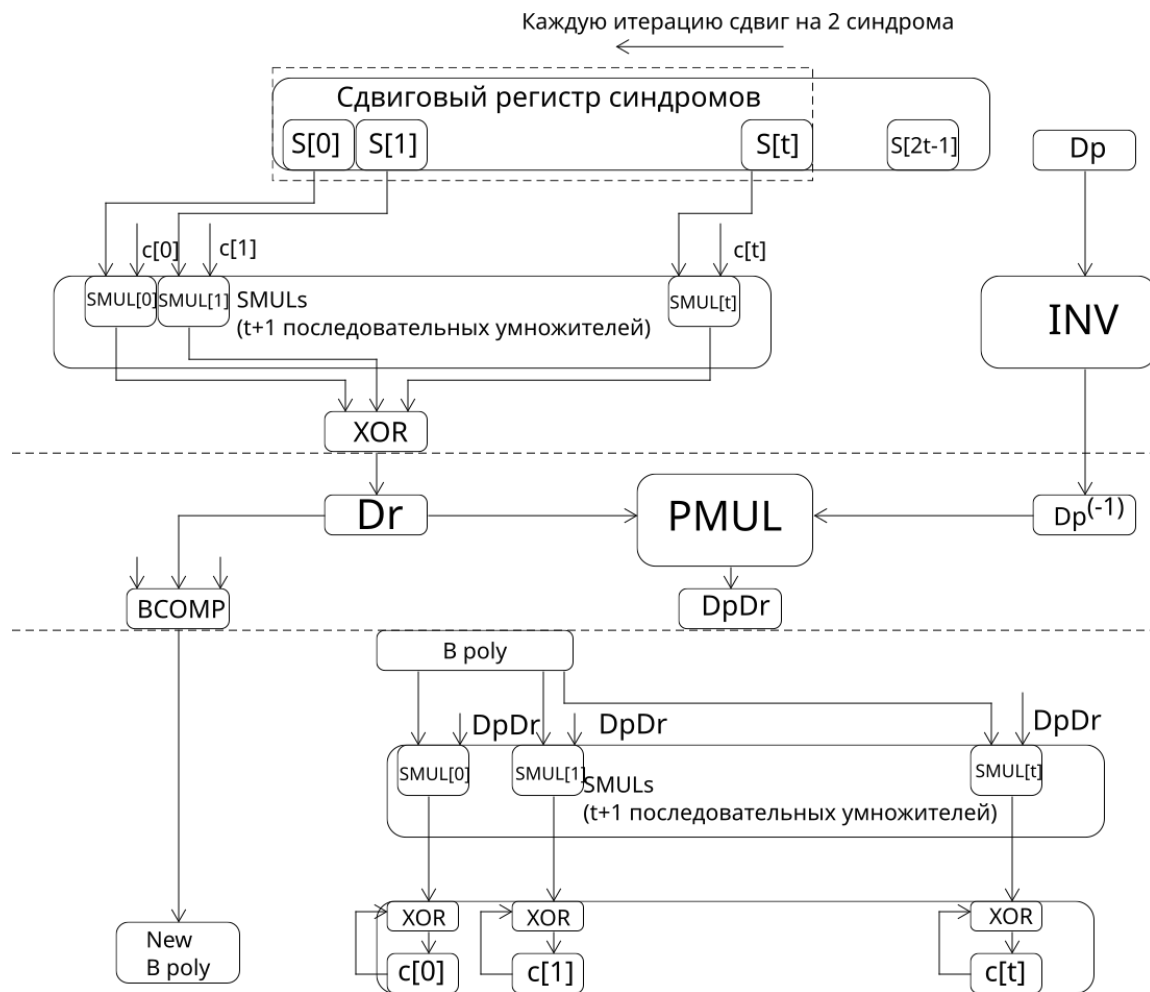
- Каждый бит синдрома линейно формируется из битов остатка от деления информации на соответствующий минимальный полином VCH кода
- Количество делителей и то, от какого делителя берется функция конкретного синдрома, это аппаратные параметры, вычисляемые утилитой генерации аппаратных параметров
- Матрицы деления и линейных функций задаются программно



Разработка RTL-описания NFMC-VCH.

Декодер. Вторая стадия(Решение ключевой системы уравнений)

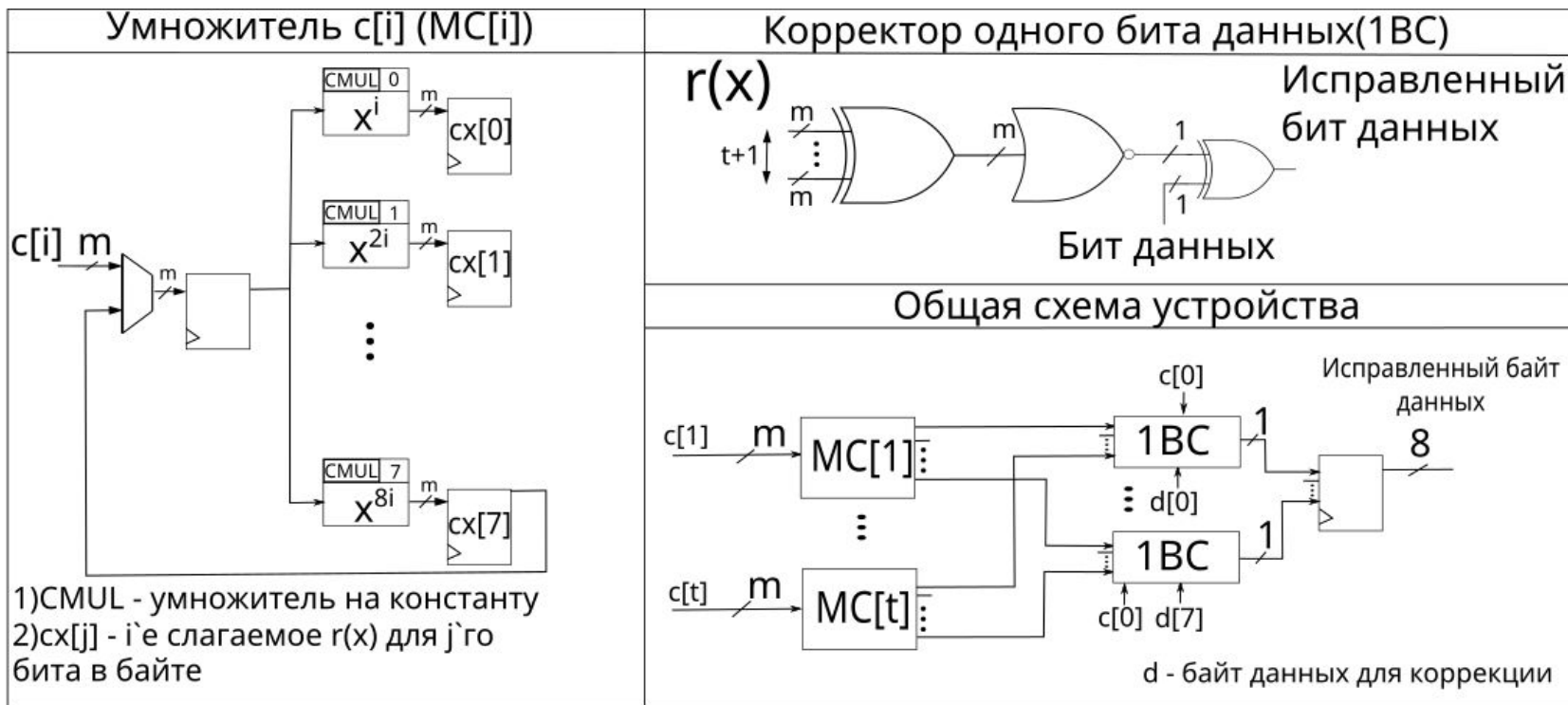
- Необходимо найти коэффициенты полинома ошибок $c(x) = c_0 + c_1*x + \dots + c_t*x^t$ используя найденные на прошлой стадии синдромы $S_j (1 \leq j < 2t)$
- Коэффициенты находятся по алгоритму BMA(Berlekamp-Massey Algorithm)
- В BMA коэффициенты полинома ошибок $c(x)$ находятся за $t-1$ рекурсивных итераций.
- Степень полинома $c(x)$ это и есть количество ошибок в данных
- S - синдром BCH
- D_p - показатель расхождения в начале итерации
- D_r - новый показатель расхождения
- SMUL - последовательный умножитель в поле Галуа
- INV - нахождение обратного в поле Галуа
- PMUL - параллельный умножитель в поле Галуа
- $D_p(-1)$ - инверсия показателя расхождения
- $D_p D_r$ - результат умножения D_r на $D_p(-1)$
- BCOMP - вычислитель нового полинома поправки
- B poly - полином поправки
- c - искомым полином ошибки



Разработка RTL-описания NFMC-VCH.

Декодер. Третья стадия(Исправление найденных ошибок)

- Позиции ошибочных бит находятся по алгоритму Chien search
- Для каждого бита данных вычисляется выражение вида: $r(x) = c_0 \text{ XOR } c_1 * x^1 \text{ XOR } \dots \text{ XOR } c_t * x^{j^t}$, j - номер итерации ($j=0,1,\dots,k-1$)
- Если на итерации i $r(x)=0$, то бит данных на позиции $n-i-1$ требует исправления (n -размер данных)
- Данные пропускаются через устройство коррекции побайтно



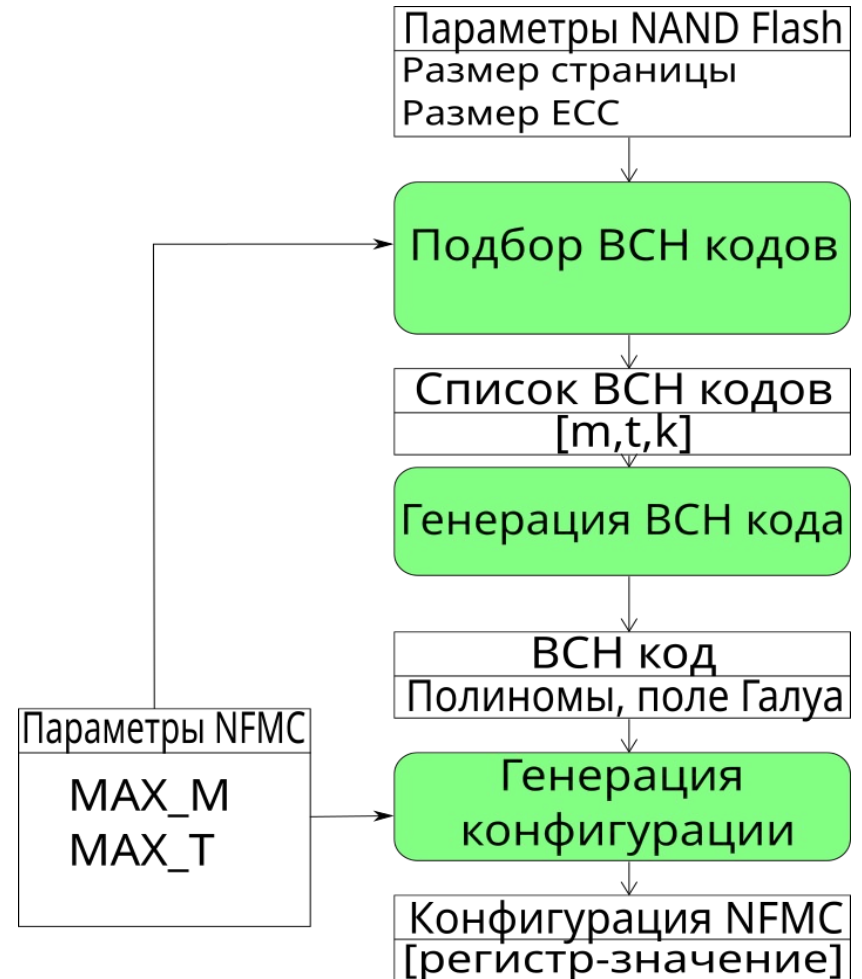
Параметры реализованного RTL-описания NFMC-BCH

Скорость работы и способность коррекции ошибок NFMC-BCH удовлетворяют требованиям NAND Flash типов SLC,MLC и TLC

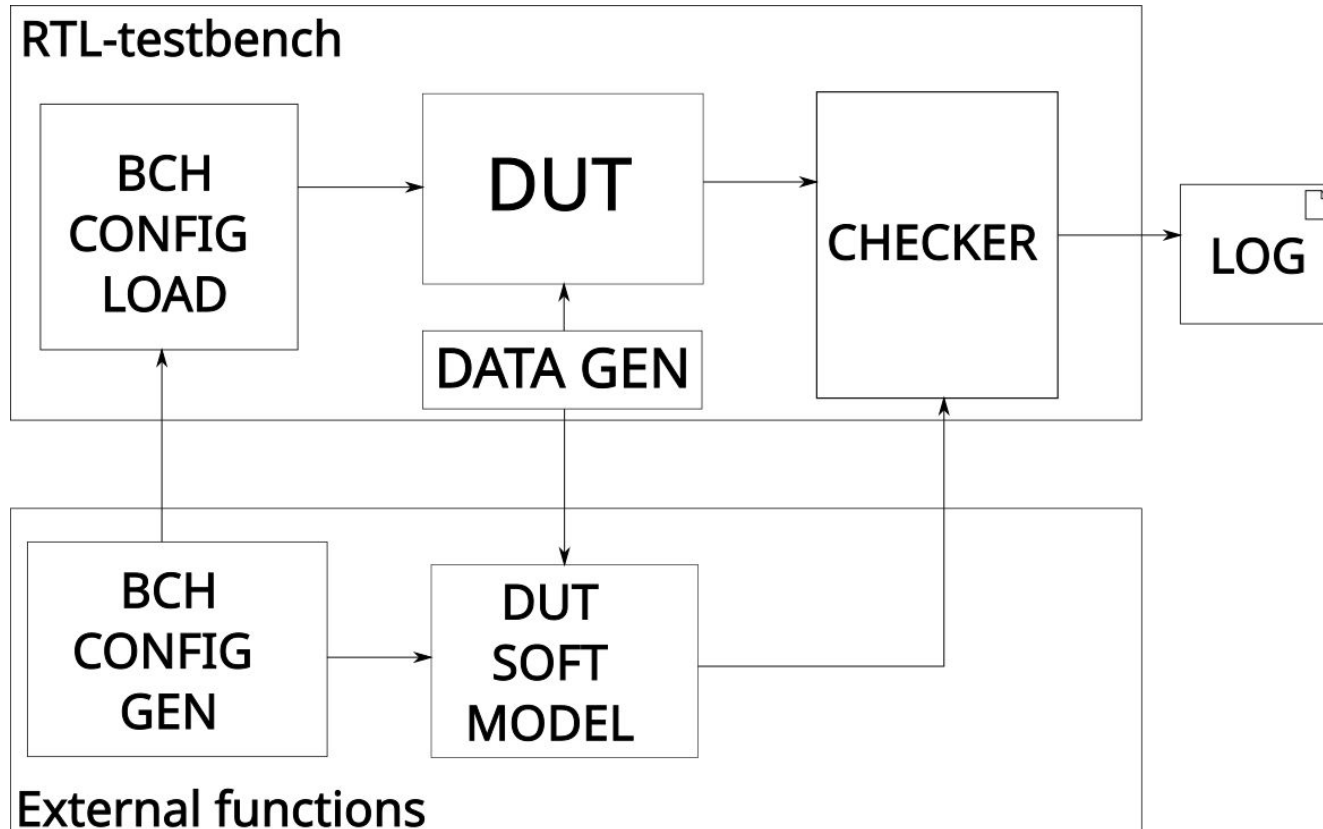
Название NAND Flash	Размер данных страницы NAND Flash(байт)	Требуемое количество исправляемых ошибок (кол-во ошибок/размер блока) в NAND Flash	NFMC-BCH кол-во исправляемых ошибок m - степень 2ки поля Галуа t - кол-во исправляемых ошибок в блоке split - кол-во блоков на которые разбита страница	Скорость чтения из NAND Flash	NFMC-BCH скорость работы блоков на частоте 1 ГГц
MX30LF1208AA (SLC)	2048+64	1 бит/ 528 байт	34 бита/2048 байт (8 бит/528 байт) m = 15, t = 34, split = 1	30ns/byte 33,3 МБ/с	EBCH: 1 байт/такт, 1000МБ/с SDBCH: 1 байт/такт, 1000МБ/с BDBCH: 1530 тактов/страницу, 1300МБ/с CDBCH: 1 байт/такт, 1000МБ/с
MT29F32G08CB AAA (MLC)	4096+218	12 бит/539 байт	109 бит/ 4096 байт (14 бит/539 байт) m = 16, t = 109, split = 1	20ns/byte 50 МБ/с	EBCH: 1 байт/такт, 1000МБ/с SDBCH: 1 байт/такт, 1000МБ/с BDBCH: 5232 тактов/страницу, 1270МБ/с CDBCH: 1 байт/такт, 1000МБ/с
UT81NDQ512G8 T (TLC)	16,384+2208	60 бит/1162 байт	278 бит / 4096 байт (78 бит/1162 байт) m = 16, t = 278, split = 4	20ns/byte 50 МБ/с	EBCH: 1 байт/такт, 1000МБ/с SDBCH:1 байт/такт, 1000МБ/с BDBCH: 13.344*4 тактов/страницу, 350МБ/с CDBCH: 1 байт/такт, 1000МБ/с

Реализованная программа-генератор конфигурации системы NFMC-VCH

- Основная структура для хранения данных в NAND Flash – страница. Её размер разбивается на 2 части:
 - Размер хранимых данных
 - Размер дополнительной информации для исправления ошибок
- В генераторе реализованы режимы формирования параметров для кодирования страницы целиком и по частям
- Подбор BCH кодов - выбор из заранее вычисленных BCH кодов подходящих под параметры страницы и NFMC. Проверяются все возможные разбивки страницы на блоки
- Генерация BCH кода - построение поля Галуа и основного и минимальных полиномов BCH кода
- Генерация конфигурации - вычисление значений всех конфигурационных регистров NFMC-VCH



Инженерная верификация разработанного RTL-описания



- RTL-testbench – тестовое окружение блока NFMC-BCH на SystemVerilog
- External functions – внешние функции на языке программирования C, подключенные к тестовому окружению через DPI
- BCH CONFIG LOAD – модуль настройки окружения и тестируемого блока на BCH код
- DUT – RTL тестируемого блока
- DATA GEN – генератор тестовых векторов
- CHECKER – модуль проверки результата работы тестируемого блока на совпадение с результатом программной модели.
- LOG – логи тестирования
- BCH CONFIG GEN – программа генерации значений конфигурационных регистров NFMC-BCH
- DUT SOFT MODEL – программная модель тестируемого блока

Результаты

- Разработана архитектура NFMC-BCH
- Разработано RTL-описание NFMC-BCH
- Разработана программа-генератор конфигурации NFMC-BCH
- Проведена инженерная верификация разработанной системы
- Обеспечено надёжное хранение данных в NAND Flash типов SLC, MLC, TLC. Достигнутая способность коррекции ошибок удовлетворяет требованиям NAND Flash типов SLC, MLC, TLC
- Удовлетворено требование о том, что скорость работы BCH на частоте 1 ГГц должна быть больше скорости чтения NAND Flash. Скорость работы системы на частоте 1 ГГц на 2 порядка выше скорости чтения из NAND Flash.