

**АППАРАТНАЯ РЕАЛИЗАЦИЯ ДЕКОДЕРА КОДА РИДА-СОЛОМОНА**

Стешенко В.Б., Бумагин А.В., Петров А.В., Черныш А.В.

МГТУ им. Н. Э. Баумана.

**Введение.**

Кодирование с исправлением ошибок широко применяется во многих цифровых системах связи для повышения надежности передачи информации по цифровым каналам. Наряду с различными кодами применяется коды Рида-Соломона, являющимися подклассом кодов БЧХ. Несмотря на то, что алгоритмы кодирования и декодирования являются известными, существуют различные варианты аппаратной реализации, влияющие как на быстродействие, так и на качество выполнения алгоритмов.

Цель работы: показать один из вариантов аппаратной реализации декодера кода Рида-Соломона применительно к требованию стандарта EN 300 744 для цифрового телевидения.

**Постановка задачи.**

Стандарт [1] использует укороченный код Рида-Соломона (204,188, t=8), образованный из исходного кода (255, 239, t=8). Он имеет длину 188 байт и позволяет корректировать до 8 ошибочных байт в принятом слове, состоящем из 204 байт.

Генераторный полином кода:

$$g(x) = (x + \lambda^0)(x + \lambda^1)(x + \lambda^2)...(x + \lambda^{15}), \text{ где } \lambda = 02_{HEX}. \quad (1)$$

**Алгоритм декодера.**

Рассмотрим алгоритм декодирования кода Рида-Соломона во временной области. Структурная схема алгоритма приведена на рис. 1 [2].

Компоненты синдрома задаются формулой: 
$$S_j = \sum_{i=0}^{203} Y_i \cdot \lambda^j \quad (2),$$

где  $j=0..15$ ,  $\lambda = 02_{HEX}$  - примитивный элемент поля,  $Y_i$  - символы сообщения. Полиномы локаторов  $f(x)$ , величин ошибок  $\Omega(x)$  находятся из ключевого уравнения:

$$S(x) \cdot f(x) = \Omega(x) \quad (3).$$

Среди методов решения ключевого уравнения (3), широко используется метод Тренча-Берлекэмп-Месси (ТБМ) [2], который задается следующим алгоритмом. Исходные данные: коэффициенты  $S_0, S_1...S_{15}$  полинома  $S(x)$ . Требуется найти полиномы  $f(x)$  и  $\Omega(x)$ . Обозначения:  $i, D, r, L, \delta, A(x), t(x)$  - вспомогательные переменные и полиномы. Начальные условия:  $f(x)=t(x)=1, L=r=0, A(x)=-1, \Omega(x) = 0$ .

1. Вычислить  $D$ : 
$$D = \sum_{j=0}^L f_j \cdot S_{r-j} \quad (4)$$

2. Если  $D \neq 0$  и  $2 \cdot L \leq r$ , то  $\delta = 1$  и  $L = r + 1 - L$  и переходим к шагу 3-а, иначе -  $\delta = 0$   $L$  - не меняет значения, переходим к шагу 3-б.

3-а. Для всех  $i = 0...7$  находим: 
$$f_{i+1} = f_i - D \cdot t_i \quad (5), \quad t_{i+1} = \frac{1}{D} \cdot f_i \quad (6),$$

$$\Omega_{i+1} = \Omega_i - D \cdot A_i \quad (7), \quad A_{i+1} = \frac{1}{D} \cdot \Omega_i \quad (8)$$

3-б. Для всех  $i = 0...7$  находим: 
$$f_{i+1} = f_i - D \cdot t_i \quad (9), \quad t_{i+1} = t_i \quad (10)$$

$$\Omega_{i+1} = \Omega_i - D \cdot A_i \quad (11), \quad A_{i+1} = A_i \quad (12)$$

4. Увеличиваем на  $r$  на единицу:  $r := r + 1$ . Вычисления производятся пока  $r < 16$ .

Решение уравнения локаторов (нахождение позиций ошибок) производилось с помощью процедуры Ченя [2, 3], а вычисление величин ошибок - с помощью процедуры Форни [2, 3].

Процедура Ченя предназначена для нахождения позиции ошибки путем вычисления корней полинома:

$$f(x) = f_8 \cdot x^8 + f_7 \cdot x^7 + f_6 \cdot x^6 + f_5 \cdot x^5 + f_4 \cdot x^4 + f_3 \cdot x^3 + f_2 \cdot x^2 + f_1 \cdot x + f_0 \quad (13)$$

Заметим, что  $k$ -й символ является ошибочным, когда 
$$\sum_{i=0}^8 f_i \cdot \lambda^{-i \cdot k} = 0 \quad (14),$$

где  $k$  – позиция символа кодового слова (от 1 до 204). Вычисление позиции ошибки производится путем перебора всех  $k$ , с проверкой тождества (14).

$$\text{Алгоритм Форни описывается формулой: } l_k = -\frac{\Omega(\lambda^{-k})}{f'(\lambda^{-k})} \quad (15),$$

где  $\lambda^{-k}$  – корень полинома (15). Полином величин ошибок задается формулой:

$$\Omega(x) = \Omega_8 \cdot x^8 + \Omega_7 \cdot x^7 + \Omega_6 \cdot x^6 + \Omega_5 \cdot x^5 + \Omega_4 \cdot x^4 + \Omega_3 \cdot x^3 + \Omega_2 \cdot x^2 + \Omega_1 \cdot x + \Omega_0$$

Кодовое слово восстанавливается путем компонентного суммирования принятого кодового слова и вектора ошибок, для чего принятое кодовое слово предварительно задерживается.

Для аппаратной реализации необходимо сформулировать алгоритм определения неисправимых ошибок. Это относится к ситуации, когда число ошибок больше 8-ми и при этом не возможно восстановление кодового слова. Алгоритм определения неисправимой ошибки формулируется следующим образом.

1. Проверить равны ли все синдромы нулю. Если хотя бы один синдром не равен нулю, то ошибки есть.
2. Если ошибки есть, а тождество (14) не выполняется ни для одного символа, значит имеет место неисправимый пакет.

#### **Аппаратная реализация.**

При разработке декодера была выбрана тактовая частота 40 МГц, что обусловлено требуемым темпом передачи данных. Декодер представляет собой конвейерный вычислитель, каждый последующий блок которого начинает работать только после того, как закончит предыдущий. Отсюда принят общий внутренний интерфейс, основанный на выработке стробирующих сигналов. Наибольшее время вычислений затрачивается блоком синдромов, так как необходимо обработать в реальном масштабе времени весь пакет входных данных (204 слова), поступающих с частотой 5 МГц, и, кроме того, последующие блоки производят вычисления за значительно меньшее количество тактов на частоте 40 МГц.

Структурная схема блока синдромов показана на рис. 2. Блок синдромов построен по принципу умножения с накоплением. Для экономии аппаратных ресурсов, вместо 16-ти базовых элементов (сумматор+умножитель), использовано только четыре с мультиплексированием. Данные поступают во входной регистр по сигналу разрешения. В регистрах reg0...reg15 хранятся значения вычисляемых синдромов. Блок управления вырабатывает сигналы управления мультиплексорами, демультимплексорами, а также вырабатывает разрешающий сигнал, по которому значения синдромов записываются в выходной буфер. Последний представляет собой набор регистров. Кроме того, блок управления вырабатывает признак окончания вычислений Data\_out\_enable.

На рис. 3-а показана структурная схема блока ТБМ. Последняя содержит в себе буфер локаторов и буфер величин ошибок. По своей структуре эти буферы идентичны, поэтому на рис. 3б показан буфер локаторов, а все относящееся к буферу величин ошибок на рисунке показано в скобках. Структурная схема отражает алгоритм работы блока ТБМ, который описывается формулами (4-12).

По активному уровню сигнала Write\_enable синдромы из блока синдромов записываются во входной буфер синдромов, и, кроме того, начинает работать блок счетчиков с вычитателем. Блок счетчиков с вычитателем служит для управления работой всего блока ТБМ. Входной мультиплексор выбирает из буфера синдромов требуемый синдром  $S_{r-j}$ , который необходим для вычисления D по формуле (4). Параметр L вычисляется в п.2 алгоритма, но исследования показали, что для вычисления D, его можно задать равным 8, что упрощает аппаратную реализацию. Значения  $f_j$  поступают из буфера локаторов.

Вычислитель величины  $\delta$  реализует пункт 2 алгоритма, параметр L является его внутренним сигналом. В зависимости от величины  $\delta$  происходит вычисления по пункту 3-а или 3-б. Отличие в вычислениях состоит в вычислении коэффициентов полиномов t и A. Реализация этого отличия произведена с помощью мультиплексора (см. рис. 3-б), который в зависимости от значения  $\delta$  присваивает значения коэффициентов t(A) на текущем шаге равным предыдущим значениям либо значениям, получаемому согласно формулам (7,9). Операция 1/D реализована с помощью ПЗУ (ROM).

Вычитание в формулах (5, 7, 9, 11) реализуется с помощью поразрядного суммирования по модулю 2 (так как операция производится в поле Галуа), а все операции умножения – умножителями в поле Галуа (см.рис. 3-а). Хранение вычисляемых локаторов (величин ошибок) осуществляется в сдвиговых регистрах RG1, а коэффициентов t(A) – в сдвиговых регистрах RG2 (см. рис.3-б). Сдвиг необходим для того, чтобы в формулах (5-12) производить вычисление текущих величин в зависимости от предшествующих. По сигналу reset слова в регистрах принимают значения, показанные на рисунке. Сдвиговый регистр содержит в себе 9 слов по 8 разрядов, сдвиг производится всего слова целиком.

Кроме того, буфер локаторов (величин ошибок) содержит дешифраторы и мультиплексоры, управление которыми производится блоком счетчиков с вычитателем.

С помощью выходного счетчика вычисленные полиномы локаторов и величин ошибок поступают на блоки Ченя, Форни, при этом вырабатывается сигнал data\_out\_enable, который подтверждает правильность и готовность данных.

Несмотря на то, что описанный алгоритм ТБМ имеет последовательную структуру, аппаратно он реализуется параллельно. Для этого на первом шаге ( $r=0$ ) производятся вычисления  $D$  и это значение заносится в промежуточный регистр. Все дальнейшие вычисления производятся с этим значением и параллельно вычисляется  $D$  для следующего шага, которое потом записывается в этот же промежуточный регистр, и т.д.

На рисунке 4 представлена структурная схема блоков Ченя-Форни, коррекции ошибок, контроля.

Блок Ченя реализует формулу (14). По сигналам data\_out\_enable, count\_9, вырабатываемыми блоком ТБМ, коэффициенты полинома  $f$  записываются в регистр. Счетчик от 1 до 204 считает позицию символа и выдает сигнал enable, который разрешает загрузку рассчитываемых значений  $\alpha$  в регистры. Таким образом, с помощью умножения с накоплением, реализуется операция возведения в степень чисел  $\alpha$ . Здесь же происходит перемножение коэффициентов  $f$  с получаемыми значениями  $\alpha^k$ . Далее, после выполнения суммирования, производится сравнение рассчитываемого значения  $f'(\alpha^{-k})$  с нулем. Если  $f'(\alpha^{-k}) = 0$ , то вырабатывается сигнал признака ошибки (sign\_error).

Блок Форни по своей структуре похож на блок Ченя, он производит вычисления по формуле (15). Для сокращения аппаратных ресурсов, при вычислениях можно использовать значения рассчитываемых величин блока Ченя. Операция деления в поле Галуа реализована использованием ПЗУ. В результате вычислений блок Форни выдает в блок коррекции величину ошибки  $l_k$ .

В блоках Ченя, Форни можно сократить количество умножителей, путем введения в схему мультиплексоров.

Блок коррекции служит для исправления ошибки. Принцип его работы состоит в покомпонентном суммировании величины ошибки и задержанного сигнала. Когда на выходе блока Ченя появляется активный уровень сигнала sign\_error, мультиплексор на сумматор выдает величину ошибки. В противном случае происходит суммирование задержанного сигнала с нулем. Задержка входного сигнала реализуется с помощью двухбанковой памяти ОЗУ, использование которой необходимо для обеспечения непрерывности поступления кодовых слов.

Блок контроля служит для формирования сигнала неисправимой ошибки. Активный уровень этого сигнала появляется в случае, когда хотя бы один из синдромов не равен нулю и, при этом, на выходе блока Ченя не сформировался активный уровень сигнала sign\_error.

#### **Основные выводы и результаты.**

В результате проделанной работы была синтезирована принципиальная схема декодера Рида-Соломона, создана топология кристалла по технологии 0.25 микрон, был изготовлен тестовый кристалл.

Исходные файлы, описывающие поведенческое описание схемы декодера, были написаны на языке VHDL, синтез принципиальной схемы производился синтезатором Synopsys Design Analyzer на базе специализированной станции SUN, топологическое проектирование выполнено с использованием пакета Silicon Ensemble.

После синтеза принципиальной схемы, согласно файлу-отчета Design Analyzer, площадь кристалла без учета связей между элементами и контактных площадок составила  $701530 \mu m^2$ . Площадь кристалла после создания топологии с учетом связей между элементами составила  $1440632 \mu m^2$ .

Проведенные испытания тестового кристалла показали, что последний способен работать с частотой поступления данных 10 МГц, с тактовой частотой 80 МГц. При этом использовался специально изготовленный стенд, включающий в себя персональный компьютер, плату ввода-вывода данных из компьютера, и плату сопряжения на основе ПЛИС ф. Altera семейства АРЕХ20К.

#### **Литература**

1. EN 300 744 v1.1.2 (1997-08) European Standard (Telecommunications series). Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television.
2. Муттер В.М. Основы помехоустойчивой телепередачи информации. Л, Энергоатомиздат, 1990.
3. Дж. Кларк, мл, Дж. Кейн. Кодирование с исправлением ошибок в системах цифровой связи. М. Радио и Связь, 1987.

One of the versions of Rid – Solomon decoder equipment implementation for digital TV according to requirement of EN 300 744 standard is considered in this work.

Algorithm consists of following modules: computer of syndromes, Trench-Berlekamp-Massy (TBM), Chien, Forney, error correction and control. The TBM module is most difficult calculating module.

Owing to insertion of intermediate register and write in this register intermediate results and initial value, TBM algorithm is realized in parallel structure, what increased his computation speed. Instead of 16 base units (sumimator + multiplier) was used only 4 base unit with multiplexing in computer of syndromes what reduce hardware. The procedure Forney use for calculating values of Chien module. Owing to insertion in circuit of multiplexers was reduced number of multipliers in Chien and Forney modules.

Results in this work are synthesis of Rid – Solomon decoder circuit, topology design of integral circuit for 0.25  $\mu m$  process, manufacturing of test integral circuit.

The source files, which describe decoder circuit, was wrote in VHDL, synthesis made in Synopsis Design Analyzer, topology design made in Silicon Ensemble.

The area of design according to report file of Design Analyzer has 701530  $\mu m^2$ . The area of design after topology is 1440632  $\mu m^2$ .

The tests of integral circuit show, what decoder works on during data frequency is 10 MHz and clock frequency is 80 MHz. Tests of integral circuit was on facilities, which include personal computer, special board for data input-output from computer, conjugating board on basis FPGA (Altera) of family APEX20K.

