

## **МЕТОДИКА ПРОЕКТИРОВАНИЯ УСТРОЙСТВ КОМПАКТНОГО ТЕСТИРОВАНИЯ**

*Юрьев И.В., Дяченко О.Н.*

*Донецкий национальный технический университет*

В связи с неуклонным ростом сложности и функциональности приборов, а также необходимости диагностики их технического состояния методы компактного тестирования с локализацией ошибок являются одними из наиболее актуальных методов решения задач тестовой диагностики. Достоинством этих методов является значительное сокращение объема информации, необходимой для проведения тестового эксперимента. При традиционном методе компактного тестирования, контроль проводится по принципу "годен - не годен", что является недостаточным в выявлении типа неисправности в отказавшем элементе дискретного устройства. Необходимо и важно определить момент времени, когда на выходе схемы появляется ошибка, то есть локализовать ошибку в выходной тестовой последовательности битов. Данный метод позволяет выявить и определить тип неисправности на основании полученных последовательностей битов, что позволит ускорить диагностику элементов дискретных устройств [1].

Основной задачей локализации ошибок является оптимальное использование информационных характеристик источников и каналов связи для построения и извлечения кодов, обеспечивающих заданную достоверность информации, передаваемой с максимально возможной скоростью и минимально возможной стоимостью передачи сообщений. Стоимость методов локализации является значительное сокращение объема информации, необходимой для проведения тестового эксперимента [1].

Теоретической базой для того, чтобы построить устройство компактного тестирования с локализацией ошибок с эффективной и минимальной избыточностью, которая необходима для выявления и исправления ошибок, возникающих при работе системы и ее элементов, является теория помехоустойчивого кодирования.

### **1. Методика проектирования устройств компактного тестирования (УКТ)**

Методика проектирования устройств компактного тестирования (УКТ) подразумевает следующие этапы:

– анализ объекта диагностики: тип устройства (комбинационный, последовательностный, одновыводной или многовыводной), тип тестирования (зондовое, устройство встроенного контроля), объект диагностики исправный или заведомо неисправный, ошибки вызванные перемежающимися неисправностями, постоянными неисправностями, случайными сбоями, характер распределения ошибок различной кратности;

– анализ тестовых воздействий: исчерпывающее тестирование (комбинационные схемы), псевдослучайное тестирование, синтезированные тесты, случайные тесты и др. Для синтезируемых тестов вероятность ошибки малой кратности максимальна. Для перемежающихся неисправностей вероятность одиночных ошибок максимальна. Для заведомо исправных объектов вероятность одиночного сбоя максимальна, при этом при повторном тестировании выявление может быть разным;

– выбор кода для реализации компактного тестирования. Для случайных сбоев и перемежающихся неисправностей достаточно выбрать код Хэмминга. Для постоянных неисправностей целесообразно использовать коды исправляющие пакеты ошибок (коды Файра или коды Рида-Соломона). Все эти коды относятся к блоковым кодам. Древовидные коды для этих целей неприменимы. Кроме того, неприменимы для этих целей такие коды, как например турбо-коды, которые хотя и являются блоковыми, для них используются методы кодирования, разработанные для древовидных кодов;

– разработка УКТ на основе циклических блоковых кодах. Основа УКТ — декодер выбранного кода без буферного регистра не конвейерной реализации. Для локализации ошибок используется счётчик. Перед сжатием тестовых реакций эталонная сигнатура загружается в генератор синдрома. В некоторых случаях не для укороченных кодов целесообразна альтернативная реализация генератора синдрома (схема деления полинома с внешними сумматорами в цепях обратной связи). В зависимости от длины теста используются декодеры для укороченных или не укороченных кодов;

– получение эталонных компактных оценок. Эталонные компактные оценки рассчитываются либо аналитически, либо формируются с помощью заведомо исправного устройства и кодера выбранного кода. Загрузка эталонной последовательности УКТ может производиться либо параллельным, либо последовательным способом, как до начала тестирования, так и после, главное чтобы выполнялся принцип суперпозиции, то есть при совпадении компактных оценок результат посимвольного сложения ноль, а при несовпадении отличен от нуля.

– анализ характеристик различных реализаций кодов. Ниже приводится анализ характеристики на примере кодов Рида-Соломона.

## 2. Анализ характеристики кодов Рида-Соломона

Для устройств компактного тестирования важны такие характеристики, как корректирующие способности кода, избыточность кода, скорость кода.

Способность декодера исправить ошибку в УКТ используется для выявления ошибки с целью определения неисправности устройства.

Порождающий полином кода Рида-Соломона, исправляющего  $s$  ошибок, должен содержать  $2s$  корней:

$$\{\alpha_0^j, \alpha_0^{j+1}, \alpha_0^{j+2}, \dots, \alpha_0^{j+2s-1}\},$$

где  $j_0$  – конструктивный параметр.

Как правило,  $j_0$  выбирают равным 1. Тогда множество корней полинома принимает вид  $\{\alpha, \alpha^2, \alpha^3 \dots \alpha^{2s}\}$ .

Для кода Рида-Соломона, исправляющего  $s$  ошибок, порождающий полином имеет следующий вид:

$$RS(X) = (X - \alpha)(X - \alpha^2)(X - \alpha^3) \dots (X - \alpha^{2s}),$$

При таком представлении порождающий полином имеет множество корней  $\{\alpha, \alpha^2, \alpha^3 \dots \alpha^{2s}\}$ .

Сущность помехоустойчивого кодирования заключается во введении в первичные коды избыточности. Поэтому помехоустойчивые коды называют избыточными. Задача помехоустойчивого кодирования заключается в таком добавлении к информационным символам первичных кодов дополнительных символов, чтобы в приемнике информации могли быть найдены и исправлены ошибки. Формула вычисления избыточности имеет вид:  $R=p/n$ , где  $p$  — количество проверочных символов,  $n$ —длина кода. Значение  $p$  вычисляется по следующей формуле  $p=\text{deg}RS(X)=2*s$ .

Длина исправляемого пакета ошибок для последовательного кода без каких-либо ограничений равна  $t=j*b-(b-1)$  для посимвольно перемеженного кода Рида-Соломона поля Галуа  $GF(2^b)$  с параметром перемежения  $j$ .

Схему посимвольно перемеженного кода Рида-Соломона можно получить из схемы исходного кода, вставив дополнительно к каждому элементу памяти  $j-1$  элементов. Например, для поля  $GF(2^3)$  при перемежении с параметром  $j=2$  каждую триаду элементов памяти нужно заменить двумя последовательно включенными триадами.

Чтобы из  $(n, k)$ -кода получить  $(jn, jk)$ -код, выберем из исходного кода  $j$  произвольных кодовых слов и укрупним кодовые слова, чередуя их символы. Если исходный код исправлял произвольный пакет ошибок длины  $d$ , то, очевидно, результирующий код будет исправлять все пакеты ошибок длины  $jd$ . Например, применяя метод перемежения к четырём копиям  $(31, 25)$ -кода, исправляющего пакет ошибок длины 2, получаем  $(124, 100)$  – код, который может исправлять пакет ошибок длины 8 [2].

Предположим, что исходный код порождается полиномом  $g(X)$ . Тогда порождающий полином получаемого перемежением кода равен  $g(X^j)$ . Заметим, что перемежение символов нескольких информационных полиномов с последующим умножением на  $g(X^j)$  даёт то же самое кодовое слово, что и умножение каждого из исходных информационных полиномов на  $g(X)$  с последующим перемежением этих слов  $(n, k)$ -кода.

Избыточность кода и его скорость зависит, прежде всего, от количества исправляемых ошибок, которое задаётся при построении кода.

Для изменения избыточности кода применяют такие подходы:

- 1) изменение параметра  $b$  поля Галуа  $(2^b)$ , на основе которого строится код;
- 2) метод посимвольного перемежения кодов.

Вначале рассмотрим параметры кодов в символах элемента поля Галуа  $GF(2^b)$ .

Для  $s=1$ :  $p=2$ ,  $n=2^b - 1$ ,  $R=p/n=2/(2^b - 1)$ .

Исправляется один  $b$ -битный символ. Чем больше  $b$ , тем меньше избыточность, следовательно, больше скорость кода. Корректирующие возможности и аппаратные затраты увеличиваются. Зависимость избыточности кода от его параметров представлены в таблице 1.

Таблица 1

Зависимость параметров кода и избыточности кода при постоянном значении  $s=1$

| GF      | p | n   | R     |
|---------|---|-----|-------|
| GF(4)   | 2 | 3   | 0,67  |
| GF(8)   | 2 | 7   | 0,29  |
| GF(16)  | 2 | 15  | 0,13  |
| GF(32)  | 2 | 31  | 0,065 |
| GF(64)  | 2 | 63  | 0,032 |
| GF(128) | 2 | 127 | 0,016 |
| GF(256) | 2 | 255 | 0,008 |

Для  $s=2$ :  $p=4$ ,  $n=2^b - 1$ ,  $R=p/n=4/(2^b - 1)$ . Избыточность в 2 раза больше, чем для  $s=1$ .

Исправляются два  $b$ -битных символа, расположенных в любых двух символах из кодового слова длины  $2^b - 1$ .

Чем больше  $b$ , тем меньше избыточность, следовательно, больше скорость кода.

Корректирующие возможности увеличиваются в  $C_2^n$  раз (по сравнению с  $s=1$ ), а аппаратные затраты увеличиваются незначительно, так как длина кода  $n$  одинакова для  $s=1$  и  $s=2$ . Однако скорость кода в 2 раза меньше, поскольку избыточность кода  $R$  в 2 раза больше.

Рассмотрим избыточность и корректирующие возможности кодов в символах двоичной последовательности  $b$ -битов (применение кодов Рида-Соломона для исправления пакетов ошибок).

Для случая с ограничением характера расположения ошибок получаем такой же результат, как рассмотренный ранее для символов элементов поля Галуа ( $2^b$ ).

Для произвольного расположения пакетов ошибок: для  $s=1$  длина исправляемого пакета  $t=1$  (всего 1 бит). Это можно пояснить следующим образом. При длине пакета  $t=2$  в наихудшем случае один искаженный бит может оказаться в одном принятом символе кодового слова ( $b$  – двоичных символов), а второй – в другом соседнем символе, что равносильно двойной ошибке для кодов Рида-Соломона. Поскольку код построен для исправления одиночной ошибки, то рассмотренная ошибка неисправима.

В случае кодов Рида-Соломона, исправляющего две ошибки ( $s=2$ ) длина произвольного расположенного исправляемого пакета ошибок  $t=b+1$ . Это объясняется тем, что в наихудшем случае при  $t=b+2$  один искаженный бит может оказаться в одном символе кодового слова ( $b$  – двоичных символов),  $b$  искаженных двоичных битов – во втором символе, и еще один искаженный бит – в третьем символе. Таким образом, получили тройную ошибку, которую декодер кода Рида-Соломона не сможет исправить, поскольку построен для кода, исправляющего двойную ошибку. Вместе с тем, любой пакет ошибок искаженных битов длины  $t=b+1$  не сможет расположиться в трех соседних символах принятого кодового слова кода Рида-Соломона. Поэтому, он будет исправим.

Таким образом, для одиночного исправляемого пакета максимальной длины увеличение  $b$  для  $s=1$  не рационально; для  $s=2$  увеличение длины исправляемого пакета незначительно по сравнению с методом перемежения.

Поэтому применение кодов Рида-Соломона, исправляющих одиночные ошибки, для исправления пакетов ошибок рационально только в случае их посимвольного перемежения.

Для произвольного расположения пакетов ошибок максимальной длины: для  $s=1$   $t=j*b-(b-1)$ ; для  $s=2$   $t=2*(j*b-(b-1))$ , где  $j$  — параметр перемежения. Как видно из приведенных выражений зависимости длины исправляемого пакета ошибок, для обоих вариантов кода Рида-Соломона, допускающих синдромное декодирование, она значительно зависит не только от параметра перемежения  $j$ , но также от параметра  $b$  поля Галуа  $GF(2^b)$ .

Избыточность  $R=j*p/j*n=p/n$  – не изменяется, следовательно, скорость кода также не изменяется, аппаратные затраты возрастают в  $j$  раз.

При посимвольном перемежении для  $s=1$  появляются, а для  $s=2$  увеличиваются дополнительные возможности исправления множественных пакетов ошибок. Однако при построении кодов следует ориентироваться на максимальную длину гарантированно исправляемого пакета ошибок, поскольку в большинстве случаев, ошибки сгруппированы в одиночные пакеты [3].

### 3. Описание работы функциональной схемы УКТ

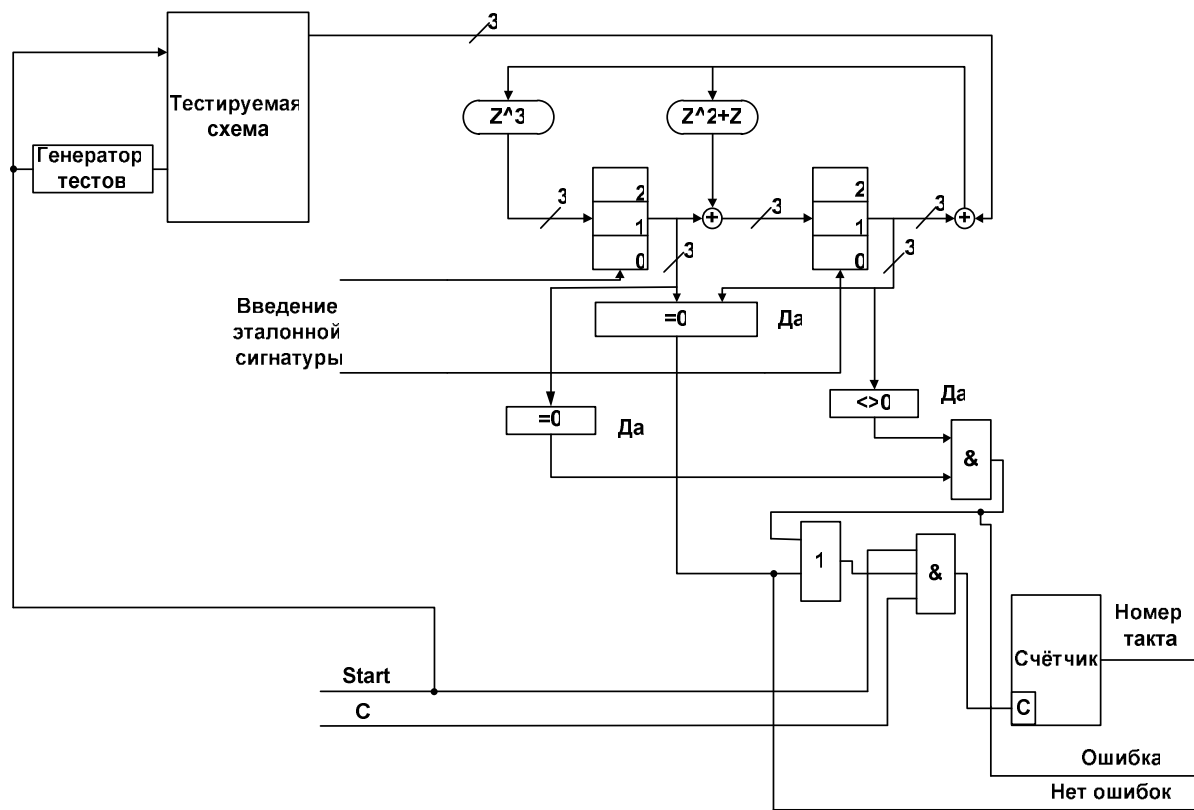


Рисунок 1 — Функциональная схема УКТ основанная на декодере кода Рида-Соломона для поля Галуа GF(8)

Работа схемы на рисунке 1 начинается с ввода эталонной сигнатуры в элементы памяти генератора синдрома Рида-Соломона, затем по сигналу Start запускается генератор тестов и тестируемая схема. Генератор тестов посылает тестовую последовательность на тестируемую схему. Реакция тестируемой схемы подаётся на генератор синдрома Рида-Соломона, где происходит деление на порождающий полином и посимвольное сложение с эталонной сигнатурой. Затем происходит проверка битов с генератора синдрома Рида-Соломона на ноль, если да (все нули), то выдаётся сигнал нет ошибок. Иначе при обнаружении в одном блоке элементов памяти нулевого значения, а в другом не нулевого значения в таком порядке, как показано на схеме, то выдаётся сигнал об ошибке и счётчик прекращает свою работу на номере того такта, когда произошла ошибка.

#### Литература:

- [1] Дяченко О.Н. Компактное тестирование запоминающих устройств с локализацией пакетов ошибок // Электрон. моделирование.- 1996.- 18, №6.- С.43-48.
- [2] Блейхут Р. Теория и практика кодов, контролирующих ошибки / Пер. с англ. - М.: Мир, 1986. - 576 с.
- [3] Юрьев И.В., Дяченко О.Н. Влияние параметров поля Галуа и перемежения на избыточность кода Рида-Соломона // Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС та КМ-2010) - 2010. - С. 164-168.