СРАВНЕНИЕ СХЕМОТЕХНИЧЕСКИХ РЕАЛИЗАЦИЙ ДЕКОДИРОВАНИЯ ПОМЕХОУСТОЙЧИВОГО ПОЛИНОМИАЛЬНОГО КОДА, ИСПРАВЛЯЮЩЕГО ПАКЕТНЫЕ ОШИБКИ M=2, P=2

Н.С. Саликов
Научный руководитель: А. Н. Мальчуков
Томский политехнический университет
nss23@tpu.ru

Введение

В наше время передача данных без сбоев с одного устройства на другое имеет важнейшее значение. Сбои возникают при воздействии внешней среды. Для борьбы с такими явлениями используется, в том числе помехоустойчивое кодирование.

Частными случаями результатов сбоев являются пакетные ошибки.

В данной статье описана реализация трех алгоритмов помехоустойчивого кодирования, также проведено сравнение по быстродействию, затратам на реализацию, трудоемкости.

Табличный метод при m=2, p=2

Самым простым из алгоритмов помехоустойчивого кодирования является табличный метод [1].

Самый простой алгоритм помехоустойчивого кодирования является табличным методом.

При m=2 и p=2, имеется 4 варианта кодового слова: 000000, 010101, 101010, 111111.

Поступившее кодовое слово длиной 6, в котором 2 информационных символа и 4 проверочных, делится на образующий полином 10101. Деление реализуется через логический элемент (далее ЛЭ) “искл. ИЛИ”, cw[0] и cw[4] образуют r[0], cw[1] и cw[5] r[1], cw[2] и cw[4] r[2], cw[3] и cw[5] r[3]. Так формируется синдром ошибки.

После этого синдром ошибки сверяют с шаблонами. Происходит это посредством ЛЭ “4И” и ЛЭ “НЕ”. r[0] подается на ЛЭ “4И”, r[1], r[2], r[3] подаются на этот элемент через ЛЭ “НЕ”. Так формируется сверка с первым шаблоном 0001. Результатом будет 0 либо 1, хранящийся в cc[0]. Таким же образом происходит и с шаблонами 0010, 0100, 1000, 0101, 1010, 0011, 0110, 1100, 0010, 1111, 1011, результаты сверки которых хранятся в cc[1], cc[2], … , cc[11], соответственно.

Далее происходит группировка по разрядам. Результаты сверки cc[0], cc[6] и cc[11] подаются на ЛЭ “3ИЛИ”. Также образуются и другие группы: cc[1], cc[6], cc[7]; cc[2], cc[7], cc[8]; cc[3], cc[8], cc[9]; cc[4], cc[9], cc[10]; cc[5], cc[10], cc[11], результаты которых хранятся в check[1], check[2], check[3], check[4] и check[5], соответственно. После этого check[0] и cw[0] подаются на ЛЭ “искл. ИЛИ”, на выходе fcw[0] которого хранится разряд, который был исправлен, в случае повреждения. Аналогичная процедура происходит для check[1] и cw[1], check[2] и cw[2], check[3] и cw[3], check[4] и cw[4], check[5] и cw[5]. Результаты хранятся в fcw[1], fcw[2], fcw[3], fcw[4], fcw[5], соответственно.

Если ошибка есть, но является неисправимой, в схеме выполняется проверка на подобные явления. Для этого r[0], r[1], r[2], r[3] подаются на ЛЭ “4ИЛИ”, а check[0], check[1], check[2], check[3], check[4], check[5] на ЛЭ “6ИЛИ”. Результаты двух операций подаются на ЛЭ “искл. ИЛИ”, выходом является Error. Если на выходе 0, значит, слово не имеет ошибок, либо схема может исправить результаты сбоев, иначе ошибка является неисправимой, на выходе 1.

Первый циклический метод при m=2, p=2

Синдром ошибки для данного метода реализуется через ЛЭ “искл. ИЛИ” cw[0] и cw[4] образуют r[0], cw[1] и cw[5] r_0[1], cw[2] и cw[4] r_0[2], cw[3] и cw[5] r_0[3]. Для сдвига на 1: cw[5] и cw[3] образуют r_1[0], cw[0] и cw[4] r_1[1], cw[1] и cw[3] r_1[2], cw[2] и cw[4] r_1[3]. И так далее для каждого последующего сдвига.

Далее синдромы ошибок сверяют с шаблонами 0001, 0010, 0100, 1000, 0011, 0110, 1100 посредством ЛЭ “4И”. r_0[0] подается на ЛЭ “4И”, r_0[1], r_0[2], r_0[3] подаются на этот элемент через ЛЭ “НЕ”. Результаты хранятся в cc_0[6..0] (сверка с 0001 хранится в cc_0[0], с 0010 в cc_0[1]и так далее). Для каждого сдвига повторяют данную процедуру, результаты хранятся в cc_1[6..0], cc_2[6..0], …, cc_5[6..0].

Результаты сверок группируются поразрядно следующим образом: на ЛЭ “11ИЛИ” подается cc_0[0], cc_0[4], cc_1[1], cc_1[4], cc_1[5], cc_2[2], cc_2[5], cc_2[6], cc_3[3], cc_3[6]. Выход check[0] данного элемента подается на ЛЭ “искл. ИЛИ” вместе с cw[0]. На выходе fcw[0] хранится разряд, который был исправлен, в случае повреждения. Аналогичным образом cc_0[1], cc_0[4], cc_0[5], cc_1[2], cc_1[5], cc_1[6], cc_2[3], cc_2[6], cc_5[0], cc_5[4] для check[1], который подается вместе с cw[1] на ЛЭ “искл. ИЛИ”, где выходом является fcw[1], и так далее.

Для проверки на неисправимость ошибки r_0[0], r_1[0], r_2[0], r_3[0], r_4[0], r_5[0] подаются на ЛЭ “6ИЛИ”, а check[0], check[1], check[2], check[3], check[4], check[5] на ЛЭ “6ИЛИ-НЕ”. Результаты двух операций подаются на ЛЭ “И”, выходом является Error. Если на выходе 0, значит, слово не имеет ошибок, либо схема может исправить результаты сбоев, иначе ошибка является неисправимой, на выходе 1.

Второй циклический метод при m=2, p=2

Во втором варианте нахождение синдрома ошибки происходит аналогично первому варианту, однако далее идет отказ от сверки синдрома с шаблонами ошибки. Вместо этого $r_i[0]$, $r_i[1]$, $r_i[2]$, $r_i[3]$ подается на два приоритетных шифратора [2].

Таблица 1. Истинность приоритетных шифраторов

1-ый HPRI					
$r_i[3]$	$r_i[2]$	$r_i[1]$	$r_i[0]$	$a_{0,i}[1]$	$a_{0,i}[0]$
1	X	X	X	1	1
0	1	X	X	1	0
0	0	1	X	0	1
0	0	0	1	0	0
2-ой HPRI					
$r_i[0]$	$r_i[1]$	$r_i[2]$	$r_i[3]$	$a_{1,i}[1]$	$a_{1,i}[0]$
X	X	X	0	0	1
1	X	X	0	1	0
0	1	X	1	0	0
0	0	1	1	1	0

Таким образом, на выходе 1-го приоритетного шифратора будет номер левой границы ошибки в двоичном коде $a_{0,i}[1]$ $a_{0,i}[0]$, 2-го – правой $a_{1,i}[1]$ $a_{1,i}[0]$ (i означает количество сдвигов).

Номера границ подаются на сумматор, где из кода правой границы вычитается код левой границы. На выходе результат вычитания в двоичном коде, так мы находим длину ошибки. Далее результат сравнивается с допустимой длиной 2 минус 1 ошибки через компаратор. Если длина меньше либо равна допустимой, то на выходах check[0], check[1], …, check[5] единица, иначе ноль.

Далее $r_0[0]$ и cw[0], $r_0[1]$ и cw[1], $r_0[2]$ и cw[2], $r_0[3]$ и cw[3] подаются на четырёхразрядный ЛЭ “искл. ИЛИ”. Для сдвига на 1: $r_1[0]$ и cw[5], $r_1[1]$ и cw[0], $r_1[2]$ и cw[1], $r_1[3]$ и cw[2]. Аналогично для остальных сдвигов. На выходе ЛЭ получается $xor_0[3..0]$, $xor_1[3..0]$, ..., $xor_5[3..0]$.

Затем на шестиразрядный ЛЭ “И” подаются пары: $xor_0[0]$ и check[0], $xor_0[1]$ и check[0], $xor_0[2]$ и check[0], $xor_0[3]$ и check[0], cw[4] и check[0], cw[5] и check[0], на выходе $fcw_0[5..0]$, соответственно. Для сдвига на 1: $xor_1[1]$ и check[1], $xor_1[2]$ и check[1], $xor_1[3]$ и check[1], cw[3] и check[1], cw[4] и check[1], $xor_1[5]$ и check[1], на выходе $fcw_1[5..0]$, соответственно. Аналогично для остальных сдвигов.

Последним этапом является подача $fcw_0[5..0]$, $fcw_1[5..0]$, $fcw_2[5..0]$, $fcw_3[5..0]$, $fcw_4[5..0]$, $fcw_5[5..0]$ на шестиразрядный ЛЭ “ИЛИ”, на выходе которого fcw[5..0].

Для обнаружения неисправимой ошибки $r_0[0]$, $r_0[1]$, $r_0[2]$, $r_0[3]$ подаются на ЛЭ “4ИЛИ”, check[0], check[1], check[2], check[3], check[4], check[5] подаются на ЛЭ “6ИЛИ-НЕ”. Выходы данных ЛЭ подаются на ЛЭ “И”. На выходе Error получается единица в случае неисправимой ошибки. Результаты работы схем представлены на Рис. 1.

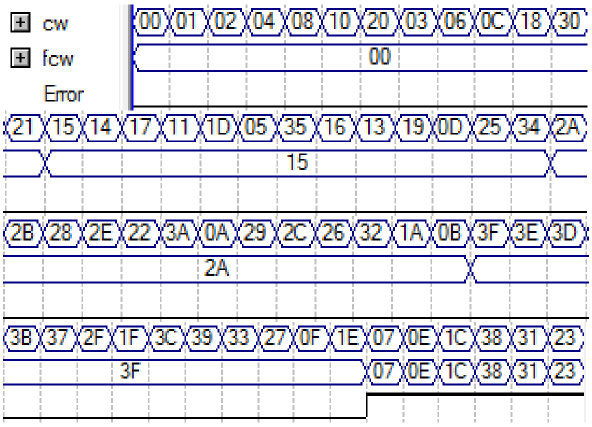

Рис. 1. Результаты работы схем

Таблица 2. Показатели методов

Название метода	Табл. метод	1-ый цикл. метод	2-ой цикл. метод
Tpd, нс	17.482	18.907	19.618
Кол-во ЛЭ, шт.	19	18	40
Трудозатраты, ч	1,5	1	1

Заключение

Табличный метод наиболее производительный. Наименее затратный в финансовом плане 1-ый циклический метод, 2-ой циклический метод является самым дорогостоящим. По времени на реализацию табличный метод уступает циклическим.

Список использованных источников

1. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса; перевйод В.Б. Афанасьева. – М.: Техносфера, 2005. – 320 с.
2. Тронин С.Н. Краткий конспект лекций по теории кодирования: учебное пособие / С.Н. Тронин. – Казань: Казанский университет, 2017 – 36 с.