

СРАВНЕНИЕ СХЕМОТЕХНИЧЕСКИХ РЕАЛИЗАЦИЙ УСТРОЙСТВ ДЕКОДИРОВАНИЯ ПОМЕХОУСТОЙЧИВОГО ПОЛИНОМИАЛЬНОГО КОДА, ИСПРАВЛЯЮЩЕГО НЕЗАВИСИМЫЕ ОШИБКИ

А.И. Левандовский
 Научный руководитель: А.Н. Мальчуков
 Томский политехнический университет
 e-mail: ail5@tpu.ru

Введение

В работе рассмотрены схемотехнические реализации устройств декодирования помехоустойчивого полиномиального кода, исправляющего независимые ошибки табличным методом и методом матрички. Методы использованы для схемотехнической реализации устройств, исправляющих $t = 1-4$ ошибок в кодовых словах с длиной информационного сообщения $m = 1$.

Схемотехническая реализация устройств декодирования табличным методом

Реализация устройства декодирования табличным методом основывается на нахождении синдромов ошибки – остатков деления кодовых слов длины n на образующий полином g [1]. Полученные синдромы ошибок сравниваются с остатками деления возможных шаблонов ошибок на тот же образующий полином. При нахождении ошибки на соответствующем логическом элементе образуется исправляющий символ, который добавляется в исправляющее слово.

Для исправления до $t = 4$ ошибок (включительно), при длине информационного сообщения $m = 1$, длина кодового слова составляет $n = 9$. Структурная схема устройства декодирования, исправляющего до четырех независимых ошибок, реализующегося табличным методом представлена на рисунке 1.



Рис. 6. Структурная схема устройства декодирования исправляющего до четырех независимых ошибок, реализующегося табличным методом

Блок определения синдрома ошибки является самым простым и содержит всего 8 логических элементов исключающего ИЛИ и 8 инверторов. Этот блок реализует нахождение остатка от деления кодового слова CW на образующий полином g . Блок формирования исправляющей комбинации напротив является самым трудоемким и содержит 255 логических элементов И на 8 входов. Каждый из логических элементов отвечает за обработку од-

ного синдрома ошибки и формирование исправляющего бита. Таким образом, блок обнаруживает 9 однократных, 36 двукратных, 84 трехкратных и 126 четырехкратных ошибок. Исправляющие биты передаются в блок формирования исправляющей комбинации. Блок состоит из 9 логических элементов ИЛИ на 93 входа. Каждый элемент производит коммутацию исправляющих битов для формирования исправляющей комбинации. На выходе блока образуется исправляющая комбинация FC . Кодовое слово CW и исправляющая комбинация FC подаются на вход блока исправления ошибки. В блоке соответствующие разряды попадают на 9 логических элементов исключающего ИЛИ. На выходе элемента образуется исправленное кодовое слово.

Для схемы с такими параметрами возможны два варианта кодовых слов: $CW_1 = 000000000$, $CW_2 = 111111111$. Представленная схема была реализована и промоделирована на исправление $t = 1-4$ ошибок в САПР Quartus II 9.1sp2. Результаты моделирования представлены на рисунке 2.

CodeWord	000000000	000010000	000101000	011000100	100010011
FixedCW	000000000				
CodeWord	111111111	111101111	111010111	100111011	011101100
FixedCW	111111111				

Рис. 2. Результаты моделирования схемы реализованной табличным методом и исправляющей $t = 1-4$ ошибок

На вход схемы последовательно подаются кодовые слова CodeWord: без ошибок, с 1, 2, 3, 4-мя ошибками. На выходе схемы образуется исправленное кодовое слово FixedCW. На рисунке 2 рассмотрены примеры для CW_1 и CW_2 . Из результатов моделирования видно успешное обнаружение и исправление до 4 ошибок. Обнаружение ошибок большей кратности невозможно, поскольку в используемых кодовых словах количество контрольных бит $k = 8$, что даёт информационную ёмкость на 256 различных событий: отсутствие ошибки и 255 вариантов ошибок, описанных выше.

Схемотехническая реализация устройств декодирования методом матрички

При помощи табличного метода, для кодового слова длины $n = 9$ реализован элемент, исправляющий однократную ошибку. На его основе построено устройство, исправляющее до 4 независимых ошибок. Структурная схема устройства декодирования, реализованного методом матрички, представлена на рисунке 3.



Рис. 3. Структурная схема устройства декодирования, реализованного методом матрички

Блок исправления однократной ошибки реализован табличным методом и содержит в себе 27 логических элементов: 8 элементов исключающего ИЛИ и 8 инверторов для формирования синдрома ошибки кодового слова CW , 9 элементов И для формирования исправляющего бита в соответствующем разряде, 1 элемент ИЛИ на 9 входов для формирования флага исправления и 1 элемент исключающего ИЛИ на два 9-ти разрядных слова для формирования исправленного кодового слова $FCW1$.

Блок исправления двукратной ошибки состоит из 9 коммутирующих блоков с одним инвертором в каждом, в которых происходит принудительное исправление соответствующего разряда, и 9 элементов исправляющих однократную ошибку. В блоке используется приоритетный шифратор, формирующий адрес для мультиплексора, который выдает соответствующий вариант кодового слова с двумя исправленными ошибками $FCW2$. Исправление двукратной ошибки осуществляется 243 логическими элементами. Блок исправления трехкратной ошибки построен схожим образом и содержит 9 элементов, исправляющих двукратную ошибку. Исправление трехкратной ошибки осуществляется 2187 логическими элементами. Соответственно, блок исправления четырехкратной ошибки содержит 19683 элемента, исправляющих четырехкратную ошибку.

Флаги исправления с каждого блока подаются на приоритетный шифратор, формирующий адрес для мультиплексора, на который подаются исправленные кодовые слова с каждого блока. Мультиплексор организует выдачу итогового исправленного кодового слова FCW .

Несмотря на большее количество логических элементов, по сравнению со схемой, реализованной табличным методом, метод матрички подразумевает многократное использование элементов, исправляющих ошибки от 1 до 3. К тому же, возможность использования элементов в виде блоков, а не схем значительно упрощает реализацию.

Представленная схема была реализована и промоделирована на исправление $t = 1-4$ ошибок в САПР Quartus II 9.1sp2. Результаты моделирования представлены на рисунке 4.

cw	00000000	01000000	00101000	10010100	11000011
fcw	00000000	00000000	00000000	00000000	00000000
cw	11111111	10111111	11010111	01101011	00111100
fcw	11111111	11111111	11111111	11111111	11111111

Рис. 4. – Результаты моделирования схемы, реализованной методом матрички и исправляющей $t = 1-4$ ошибок

На вход схемы последовательно подаются кодовые слова cw : без ошибок, с 1, 2, 3, 4-мя ошибками. На выходе схемы образуется исправленное кодовое слово fcw . На рисунке 2 рассмотрены примеры для CW_1 и CW_2 . Из результатов моделирования видно успешное обнаружение и исправление до 4 ошибок.

Сравнение реализованных схем

Схемы, реализованные табличным методом и методом матрички, сравниваются по трём показателям: количество логических ячеек на ПЛИС, задержка самой длинной цепочки элементов (tpd) и трудоёмкость. Результаты сравнения представлены в таблице 1.

Таблица 1. Результаты сравнения схем

Метод реализации	t	Кол-во лог. ячеек, шт	tpd, нс	Трудоёмкость, чел.-час
Табличный	1	4	11.027	0.25
	2	15	15.111	0.5
	3	43	18.793	4
	4	95	22.388	10
Матричка	2	41	15.834	0.5
	3	18	16.771	1
	4	16	22.755	3

Метод матрички с ростом t начинает занимать значительно меньше логических ячеек и остается достаточно простым в схемотехнической реализации за счет многократного использования нескольких простых элементов, что также уменьшает вероятность возникновения ошибок в схеме при сборке. По показателям tpd оба метода почти эквивалентны.

Заключение

По результатам реализации и тестирования схем, метод матрички показывает такую же эффективность в исправлении независимых ошибок, как и табличный метод. При этом метод матрички позволяет использовать меньше логических ячеек на ПЛИС и затрачивать намного меньше времени на схемотехническую реализацию устройств декодирования помехоустойчивого полиномиального кода, исправляющего независимые ошибки.

Список использованных источников

1. Морелос–Сарагоса Р. Искусство помехоустойчивого кодирования: методы, алгоритмы, применение: учебное пособие / М.: Техносфера, 2006. – 320 с.