

Шифраторы, дешифраторы и кодопреобразователи

Одними из простейших комбинационных устройств являются шифраторы и дешифраторы. Их основным назначением является изменение кода, в котором представлено входное число. Также дешифраторы позволяют обнаружить заданные комбинации сигналов на своем входе и запустить на выполнение другие компоненты системы. Шифраторы, в свою очередь, на основании номера активного входа генерируют заданные комбинации на выходе.

Несмотря на свою простоту, шифраторы и дешифраторы широко применяются при проектировании цифровых устройств различного назначения.

Дешифратор (decoder)

Дешифратор – это комбинационное цифровое устройство, которое преобразует N -разрядный двоичный, троичный или m -ичный код в m^N -ичный одноединичный (унитарный) код, где m – основание системы счисления. Логический сигнал активен на том выходе, порядковый номер которого соответствует двоичному, троичному или m -ичному коду.

Двоичный дешифратор (см. рис. 1) работает по следующему правилу. Пусть дешифратор имеет N входов, на которые подаётся двоичное слово x_0, x_1, \dots, x_{N-1} , тогда на выходах дешифратора формируется код, разрядности меньшей или равной 2^N , где разряд, номер которого равен десятичному представлению входного слова, становится активным (принимает значение логической единицы, логического нуля или переводится в высокоимпедансное состояние – отключается, что зависит от конкретной реализации дешифратора), все остальные разряды неактивны. Максимально возможная разрядность выходного слова равна 2^N . Такой дешифратор называется полным. Если часть входных наборов не используется, то число выходов меньше 2^N , и дешифратор является неполным.

Пример таблицы истинности для двухвходового дешифратора представлен в таблице 1.

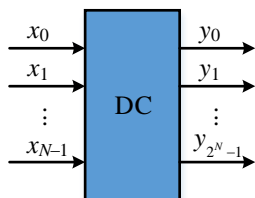


Рис. 1. Дешифратор

Таблица 1. Таблица истинности для двухвходового дешифратора

x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Часто дешифраторы дополняются входом разрешения работы **ce** (chip enable). Если на этот вход поступает активный логический сигнал, то один из выходов дешифратора переходит в активное состояние, иначе все выходы неактивны вне зависимости от состояния входов.

Шифратор (encoder)

Шифратор – это комбинационное цифровое устройство, которое выполняет преобразование позиционного n -разрядного кода в N -ичный двоичный, троичный или m -ичный код.

Двоичный шифратор выполняет логическую функцию преобразования унитарного n -ичного однозначного кода в двоичный (см. рис. 2). При подаче сигнала на один из n входов (только на один) на выходе появляется двоичный (в общем случае m -ичный) код номера активного входа.

В таблице 2 приведен пример таблицы истинности для четырехвходового шифратора. Важно отметить, что всего возможно $2^4 = 16$ различных комбинаций входных сигналов такого шифратора, но разрешены только четыре из них.

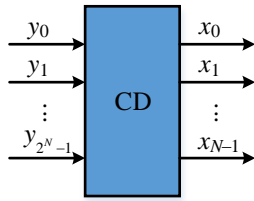


Рис. 2. Шифратор

Таблица 2. Таблица истинности для четырехвходового шифратора

x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
Другие комбинации запрещены				–	–

Если количество входов настолько велико, что в шифраторе используются все возможные комбинации сигналов на выходе, то такой шифратор называется полным, если не все, то неполным.

Приоритетный шифратор

Приоритетный шифратор – это комбинационное цифровое устройство, которое выполняет преобразование номера наиболее приоритетного входа в N -ичный двоичный, троичный или m -ичный код.

Для корректной работы шифратора только на один его вход можно подавать сигнал (единицу), при этом на всех остальных входах сигнала быть не должно (нули). Приоритетный шифратор применяется в случае, когда более, чем на одном входе может присутствовать активный сигнал. В таких случаях приоритетный шифратор обработает (шифрует) первый вход, на котором будет присутствовать единица.

Ниже в таблице 3 приведена таблица истинности для четырехвходового приоритетного шифратора. Приоритет входа описанного шифратора увеличивается с увеличением его номера. Символом n помечены входы и выходы шифратора, сигналы на которых не важны. Добавлен дополнительный выход v – valid, который сигнализирует о корректности выходных сигналов шифратора.

Таблица 3. Таблица истинности для четырехвходового приоритетного шифратора

x_3	x_2	x_1	x_0	y_1	y_0	v
0	0	0	0	n	n	0
0	0	0	1	0	0	1
0	0	1	n	0	1	1
0	1	n	n	1	0	1
1	n	n	n	1	1	1

У рассмотренного приоритетного шифратора отсутствуют запрещенные комбинации входных сигналов.

Пример 1. Логарифм числа

Интересно, что с помощью приоритетного шифратора можно очень просто приблизительно вычислять логарифм по основанию 2 целого числа. Логарифм по основанию 2 числа X можно представить в следующем виде (используется экспоненциальная запись с основанием 2):

$$X = 2^p M,$$

$$\log_2 X = \log_2(2^p M) = p + \log_2 M \approx p,$$

где p – порядок числа X , а M – мантисса числа x , $1 \leq M < 2$. Формулу (1) можно реализовать с помощью приоритетного шифратора. В таблице 4 для примера представлена реализация

логарифма по основанию 2 чисел от 0 до 15. Красным цветом выделены единицы, по которым срабатывает приоритетный шифратор. На выходе такого шифратора генерируется результат: необходимое значение порядка числа.

Таблица 4. Реализация логарифма по основанию 2 на основе приоритетного шифратора для чисел от 0 до 15

Десятичный код	Прямой двоичный код				p
	x_3	x_2	x_1	x_0	
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	2
5	0	1	0	1	2
6	0	1	1	0	2
7	0	1	1	1	2
8	1	0	0	0	3
9	1	0	0	1	3
10	1	0	1	0	3
11	1	0	1	1	3
12	1	1	0	0	3
13	1	1	0	1	3
14	1	1	1	0	3
15	1	1	1	1	3

Кодопреобразователь

Преобразователь кодов (кодопреобразователь) – это комбинационное цифровое устройство, которое изменяет код представления числа. Наиболее часто используемые кодопреобразователи: двоично-десятичный код в семисегментный; двоично-десятичный код в двоичный; двоичный код в двоично-десятичный; двоичный код в код Грея; двоичный код в код Хэмминга; двоичный код в код m из n .

Двоично-десятичный код

При использовании двоично-десятичного кода каждый разряд десятичного числа записывается в виде его четырехбитного прямого двоичного кода. Пример таблицы истинности для кодопреобразователя из прямого двоичного кода в двоично-десятичный для чисел от 0 до 15 представлен в таблице 5.

Таблица 5. Таблица истинности для кодопреобразователя из прямого двоичного кода в двоично-десятичный для чисел от 0 до 15

Десятичный й код	Прямой двоичный код				Двоично-десятичный код									
	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0		0	0	0	0
1	0	0	0	1	0	0	0	0	0		0	0	0	1
2	0	0	1	0	0	0	0	0	0		0	0	1	0
3	0	0	1	1	0	0	0	0	0		0	0	1	1
4	0	1	0	0	0	0	0	0	0		0	1	0	0
5	0	1	0	1	0	0	0	0	0		0	1	0	1
6	0	1	1	0	0	0	0	0	0		0	1	1	0
7	0	1	1	1	0	0	0	0	0		0	1	1	1
8	1	0	0	0	0	0	0	0	0		1	0	0	0
9	1	0	0	1	0	0	0	0	0		1	0	0	1
10	1	0	1	0	0	0	0	0	1		0	0	0	0
11	1	0	1	1	0	0	0	0	1		0	0	0	1
12	1	1	0	0	0	0	0	0	1		0	0	1	0
13	1	1	0	1	0	0	0	0	1		0	0	1	1
14	1	1	1	0	0	0	0	0	1		0	1	0	0
15	1	1	1	1	0	0	0	0	1		0	1	0	1

В двоично-десятичном коде для кодирования одного десятичного разряда из 16 возможных значений прямого двоичного кода применяются первые десять. Остальные шесть являются запрещенными.

Код Грея (одношаговый)

Одношаговый код Грея (часто слово “одношаговый” пропускают) – это такой двоичный код, в котором любые два смежных слова различаются только в одной позиции. Другими словами, расстояние Хэмминга¹ между любыми двумя смежными словами равно единице, включая первое и последнее слова (циклически замкнутый одношаговый код Грея).

В таблице 6 представлен пример таблицы истинности для кодопреобразователя из прямого двоичного кода в одношаговый код Грея.

Таблица 6. Таблица истинности для кодопреобразователя из прямого двоичного кода в одношаговый код Грея

Десятичный код	Прямой двоичный код			Код Грея		
	x_2	x_1	x_0	y_2	y_1	y_0
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

Перевод N -разрядного числа из прямого двоичного кода в двоичный код Грея выполняется по следующей формуле: $y_i = x_i \text{ XOR } x_{i+1}$, $i = 0, 1, \dots, N-1$. Т.е. для вычисления i -го разряда в коде Грея необходимо взять исключающее ИЛИ от i -го и $(i + 1)$ -го разрядов в прямом двоичном коде. При $i = N-1$ считать $x_{i+1} = 0$. Здесь x_i – разряды исходного числа в прямом двоичном коде, а y_i – разряды числа в коде Грея.

Обратный перевод числа из кода Грея в прямой двоичный код выполняется так: $x_i = y_i \text{ XOR } y_{i+1} \dots \text{ XOR } y_{N-1}$. Т.е. i -й разряд в прямом двоичном коде равен исключающему ИЛИ от всех разрядов в коде Грея с номерами не менее i .

Двоичный код Грея широко применяется в качестве манипуляционного кода в различных способах цифровой модуляции (например, ФМ-4, КАМ-64, АФМ-32 и т.д.).

В общем случае удобство применения кода Грея в качестве манипуляционного кода, заключается в том, что при случайном (обычно, ошибочном) переходе между соседними состояниями в системе будет возникать ошибка только в одном разряде.

Семисегментный код

Одним из простейших устройств индикации (для отображения цифр и чисел) является семисегментный индикатор (обычно, светодиодный) (см. рис. 3 и 4). Такие индикаторы, как следует из названия, состоят из семи светодиодов (иногда добавляют восьмой светодиод – для отображения точки). Зажигая различные комбинации светодиодов (см. рис. 5) можно отображать цифры от 0 до 9. Сегменты обозначаются буквами от А до G; восьмой сегмент (десятичная точка, decimal point, DP), предназначен для отображения десятичных дробных чисел.

Таблица истинности для кодопреобразователя из прямого двоичного кода в семисегментный код представлена в таблице 7.

¹ Расстояние Хэмминга – число позиций, в которых соответствующие символы двух слов одинаковой длины различны.

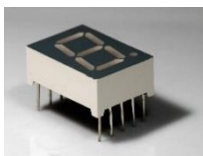


Рис. 3. Внешний вид семисегментного индикатора с точкой

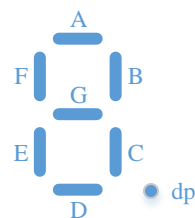


Рис. 4. Схематичное обозначение сегментов



Рис. 5. Пример изображения на семисегментном индикаторе цифр 2 и 9

Таблица 7. Таблица истинности для кодопреобразователя из прямого двоичного кода в семисегментный код

Двоичный код				Семисегментный код							Отображаемые цифры
x_3	x_2	x_1	x_0	g	f	e	d	c	b	a	
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0	1	1	0	1
0	0	1	0	1	0	1	1	0	1	1	2
0	0	1	1	1	0	0	1	1	1	1	3
0	1	0	0	1	1	0	0	1	1	0	4
0	1	0	1	1	1	0	1	1	0	1	5
0	1	1	0	1	1	1	1	1	0	1	6
0	1	1	1	0	0	0	0	1	1	1	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	0	1	1	1	1	9
1	0	1	0	1	1	1	0	1	1	1	A
1	0	1	1	1	1	1	1	1	0	0	B
1	1	0	0	0	1	1	1	0	0	1	C
1	1	0	1	1	0	1	1	1	1	0	D
1	1	1	0	1	1	1	1	0	0	1	E
1	1	1	1	1	1	1	0	0	0	1	F

Особенности FPGA

Основным ресурсом FPGA Xilinx 7-о поколения является конфигурируемый логический блок (Configurable Logical Block, CLB). В каждом логическом блоке содержится два модуля, которые называются slice, не связанных друг с другом. Логический блок подключен к матрице подключений (рис. 6²), которая в свою очередь подключена к глобальной матрице соединений.

² www.xilinx.com

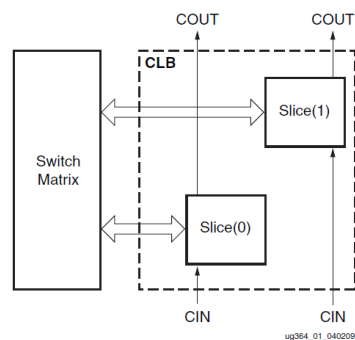


Рис. 6. Структура конфигурируемого логического блока

Логические блоки (и slice'ы) в ПЛИСе объединяются в столбцы, при этом slice, которые находятся в одном столбце соединены очень быстрыми линиями для реализации цепей переноса (необходимы, например, при выполнении арифметических операций).

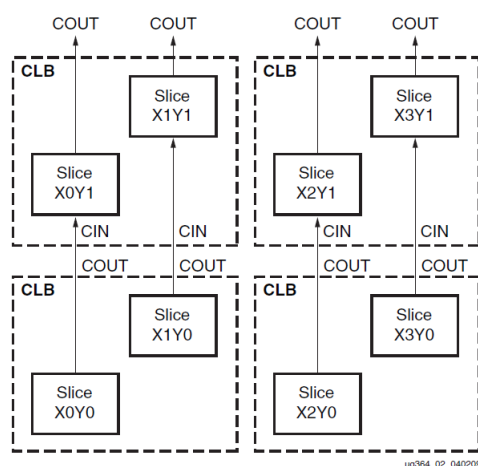


Рис. 7. Объединение логических блоков в столбцы

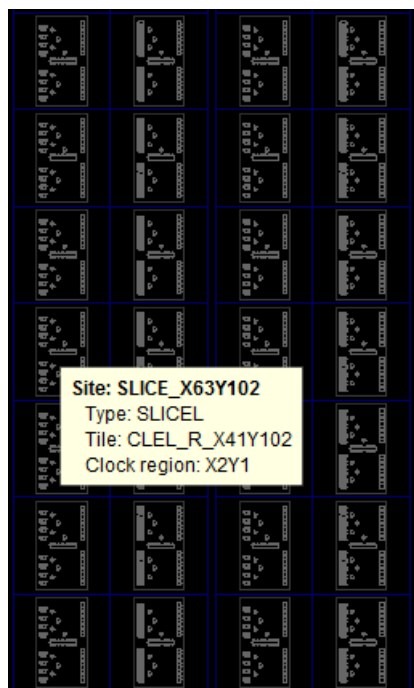
Для возможности реализации логических и арифметических функций, а также памяти в состав slice входят:

- 4 таблицы истинности (генераторы функций);
- 8 элементов хранения (триггеры);
- мультиплексоры;
- линии переноса.

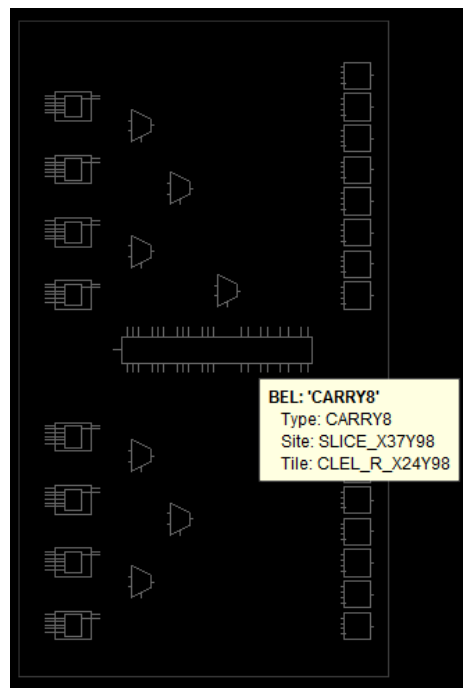
Существует два типа slice: SLICEM и SLICEL. SLICEM включает в себя поддержку двух дополнительных функций: хранение данных в распределенной памяти и сдвиг данных в 32-х разрядном сдвигающем регистре. В каждом конфигурируемом логическом блоке может быть либо два slice типа SLICEL, либо один slice типа SLICEL и один slice типа SLICEM. Каждый второй столбец CLB содержит SLICEM. Всего в ПЛИС slice типа SLICEM меньше, чем slice типа SLICEL.

Генераторы функций реализованы в виде 6-входовых таблиц истинности. Каждая таблица истинности имеет 6 независимых входов и 2 независимых выхода. Каждая таблица истинности может реализовывать 2 логических функции, если входы для них общие. Вывод данных осуществляется непосредственно с выходов slice либо через соответствующие мультиплексоры. В slice типа SLICEM в таблицах истинности могут быть также реализованы 256 бит распределенной RAM или 128-битный сдвигающий регистр.

На рис. 8 представлен внешний вид slice Xilinx FPGA серии Ultrascale. Фактически это вдвоенные slice предыдущей серии. При увеличенном масштабе в левой части slice видны 8 таблиц истинности, в правой – 16 триггеров, а посередине – мультиплексоры выходов таблиц истинности и цепь переноса.



а) Slice внутри Xilinx FPGA серии Ultrascale



б) Slice. Увеличенный масштаб

Рис. 8. Slice внутри Xilinx FPGA серии Ultrascale