**Работа 3. Преобразователи кодов.**

**§ 3.1. Дешифратор – демультиплексор**

Дешифратор (обозначают DC - decoder) преобразует одну разновидность кода в другую. Самым распространенным является дешифратор, на входе которого n-разрядный двоичный код (d), а на выходе - единичный код (q) содержащий 2n разрядов. Единичный код называют также «1из N», он содержит только один разряд, равный 1.  
Дешифраторы используют для формирования сигналов включения в работу различных устройств или потоков данных.

Рис.3.1. Дешифратор – демультиплексор 2 в 4, символ,   
логические функции, таблица истинности, схема

е

q0

q1

q2

q3

d1 d0

|  |  |  |  |
| --- | --- | --- | --- |
| е | d1d0 | q3q2q1q0 | hex |
| 0 | x x | 0 0 0 0 | 0 |
| 1 | 0 0 | 0 0 0 1 | 1 |
| 1 | 0 1 | 0 0 1 0 | 2 |
| 1 | 1 0 | 0 1 0 0 | 4 |
| 1 | 1 1 | 1 0 0 0 | 8 |

q0

q1

q2

q3

DC

d0

d1

e

.

e

;

;

;

0

1

3

0

1

2

0

1

1

0

1

0

*d*

*d*

*q*

*d*

*d*

*e*

*q*

*d*

*d*

*e*

*q*

*d*

*d*

*e*

*q*

























d0

d1

d0

d1

&

&

q0

&

&

q1

q2

q3

е

Дешифратор, имеющий вход разрешения е (от слова enable) (рис.3.1), называется дешифратор – демультиплексор, его работу поясняет приведенная схема переключения сигналов. Дешифратор – демультиплексор позволяет переключать входной сигнал (е) на один из выходов (q3..q0) в соответствии с адресом (d1..d0), как показано на схеме переключения сигналов.

При е = 1 = const получим режим дешифратора. Схема формирует сигнал сигнал 1 на одном из выходов (q3..q0) в соответствии с кодом данных (d1 d0).

В режиме демультиплексора сигнал разрешения (е) используется в качестве входного сигнала. Его изменения передаются на один из выходов q в соответствии с кодом на входах d1 d0, который выполняет функцию адреса.

//Дешифратор

module v41\_dc (d, q); //2

input [1:0] d; //3

output [3:0] q; //4

assign q[0] = ~d[1] & ~d[0]; //5

assign q[1] = ~d[1] & d[0]; //6

assign q[2] = d[1] & ~d[0];//7

assign q[3] = d[1] & d[0];//8

endmodule //9

Таблица истинности дешифратора – демультиплексора составляется с учетом того, что что при е = 0 все выходные сигналы равны 0. При е = 1 сигнал, равный 1 появляется только на том выходе дешифратора, адрес которого подается на входы d. В столбце е «hex» приведены значения выходного кода в 16-ричной системе, для использования.

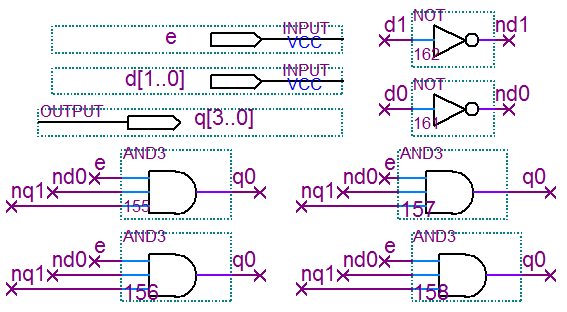


рис. 3.2. Схеиа дешифратора-  
демультиплексора s31\_dcdm

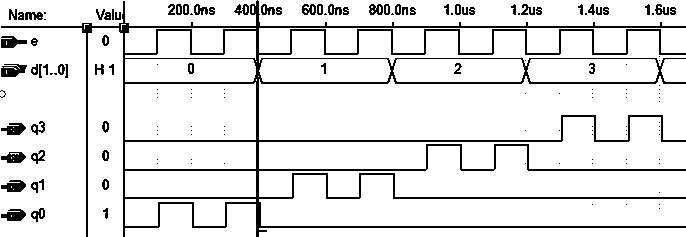


Рис.3.3. Режим демультиплексора

По таблице истинности записаны логические функции в СДНФ. Для каждого выходного сигнала функция содержит всего один минтерм. По логическим функциям составлены схема и описание на Verilog.

**Задание 3.1. Проект дешифратора - демультиплексора 2 в 4 по схеме.**

Разработайте проект дешифратоа-демультиплексора с именем s31\_dc на основе схемы (рис. 3.1), выполните этапы 1-10.

По заданной схеме (рис. 3.1), в которой использовано физическое соединение элементов (проводниками), введите в графическом редакторе схему (рис. 3.2), в которой используются логические соединения (использованием одинаковых имен), и добавляются терминалы для входных и выходных сигналов.

При моделировании получите три временные диаграммы отдельно для режима дешифратора, для режима демультиплексора (подобно рис. 3.3) и для режима измерения временных задержек в более крупном масштабе.

При работе с отладочной платой составьте таблицу подключения выводов. На входы дешифратора – демультиплексора подайте сигналы от кнопок. К выходам подключите светодтоды. Выберите элементы отладочной платы, которые необходимо подключить к проекту в ПЛИС, например, кнопки k4, k3, clr, а также светодиоды d34- d37, определите по схеме платы номера их контактов. Выполните назначение выводов (рис. 3.5), запишите результат назначения File > Saveе , выполните компиляцию и программирование ПЛИС. Опишите работу устройства от кнопок в режиме дешифратора и в режиме мультиплексора.

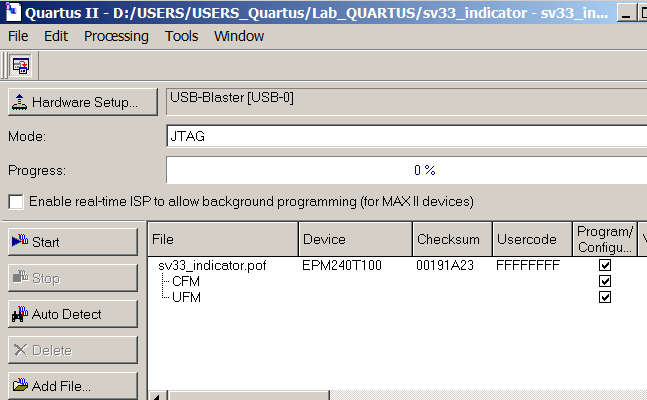
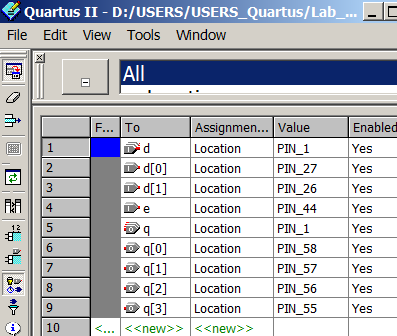


Рис. 3.4. Окна редактора назначения выводов и программатора



**§ 3.2. Дешифратор для вывода на семисегментный индикатор.**

На отладочной плате содержится блок 7-сегментных индикаторов, позволяющий выводить изображения цифр в виде комбинаций светящихся сегментов (рис. 3.5). Комбинацией сегментов могут быть представлены цифры 0—9, а также ряд букв. Схема и таблица подключения элементов платы приведены в Справочном материале.

Рис. 3.7.   
Отображение  
букв

|  |  |  |
| --- | --- | --- |
| Сим- вол | Обр. коды k4 k3 k2 k1 | Выходы a b c d e f g |
| 0  1  2  3  4  5  6  7  8  9  а | 1 1 1 1  1 1 1 0  1 1 0 1  1 1 0 0  1 0 1 1  1 0 1 0  1 0 0 1  1 0 0 0  0 1 1 1  0 1 1 0  … | 0 0 0 0 0 0 1  1 0 0 1 1 1 1  0 0 1 0 0 1 0  0 0 0 0 1 1 0  1 0 0 1 1 0 0  0 1 0 0 1 0 0  0 1 0 0 0 0 0  0 0 0 1 1 1 1  0 0 0 0 0 0 0  0 0 0 0 1 0 0  … |

Рис. 3.5.  
Обозначения

сегментов

a

b

c

d

e

f

g

h

При составлении таблицу истинности дешифратора, необходимо учитывать, что входные сигналы будут поступать от кнопок, которые принимают значение 0 при нажатии кнопки. Для удобства ввода данных примем для символа 0 состояния кнопок, когда не нажата ни одна из них. При этом код от кнопок равен 1111. Для символа 1 примем состояния, когда нажата кнопка младшего разряда, при этом код равен 1110. В результате, с учетом включения кнопок, входные сигналы будут представлены обратными кодами символов.

Светодиоды в индикаторах включены с общим анодом, а управляющие сигналы подаются на катоды, соответствующие отдельным сегментам. Они должны быть равны 0, чтобы сегмент светился.

module v33\_indicator (k4,k3,k2,k1, a,b,c,d,e,f,g,h,z);

input k4,k3,k2,k1; output a,b,c,d,e,f,g, h,z;

assign a= k4&k3&k2&~k1| k4&~k3&k2&k1;

assign b= k4&~k3&k2&~k1|k4&~k3&~k2&k1;

assign c= k4&k3&~k2&k1;

assign d= k4&k3&k2&~k1|k4&~k3&k2&k1 | k4&~k3&~k2&~k1 ;

assign e= k4&k3&k2&~k1|k4&k3&~k2&~k1 | k4&~k3&k2&k1 | k4&~k3&k2&~k1

| k4&~k3&~k2&~k1 | ~k4&k3&k2&~k1;

assign f= k4&k3&k2&~k1|k4&k3&~k2&k1 | k4&k3&~k2&~k1 | k4&~k3&~k2&~k1;

assign g= k4&k3&k2&k1|k4&k3&k2&~k1 | k4&~k3&~k2&~k1 ;

assign h=0; assign z=0; endmodule

Первая строка таблицы соответствует выводу символа 0. Кнопки не нажаты, код от кнопок 1111. Должны светиться все сегменты, кроме g (см. рис. 3.5), поэтому в данной строке на сегмент g подается сигнал 1, а на остальные сегменты подается 0.

Вторая строка таблицы - ввод символа 1, код от кнопок 1110, должны светиться сегменты *b* и *с*, поэтому в данной строке *b* = *с=0*, а на остальные сегменты подается 1.

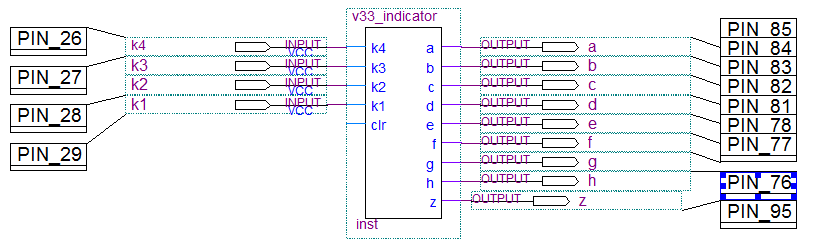


Рис. 3.6. Схема дешифратора после назначения выводов

Самостоятельно запишите в таблицу истинности строки, обеспечивающие отображение символов для вывода 16-ричных значений.

По таблице составлены логические функции и описание на Verilog. Несложно составить схему

**Задание 3.2. Проект дешифратора для вывода на семисегментный индикатор.**

Разработайте проект дешифратора для индикатора по описанию с именем v32\_indicator. Выполните этапы 1 – 5, создайте символ модуля.

2. Создайте новый файл схемы проекта с именем sv32\_indicator, на который поместите символ v32\_indicator ( рис. 3.8), выполните этапы 1 – 5 и 10. Подключите устройство к компьютеру, включите программатор.

Выполните исследование работы дешифратора, , используя отладочную плату, подавая входные сигналы от кнопок. Проверьте правильность отображения всех символов.

**§ 3.3. Приоритетный шифратор.**

Шифратор (обозначают CD - coder) (Рис. 3.8) выполняет функцию, противоположную дешифратору, он преобразует сигнал, равный единице только на одном из входов (единичный код), в двоичный код отображающий номер входной цепи. Шифраторы выполняют функции ввода кода нажатой клавиши, а также определения номера устройства, подавшего запрос на прерывание. Полный двоичный шифратор имеет *2N* входов и *N* выходов. Таблица истинности для двоичного шифратора получается из таблицы для дешифратора, в которой меняются местами столбцы входов и выходов.

В реальных условиях, при воздействии помех, или одновременном нажатии нескольких кнопок, могут появиться единичные сигналы на нескольких входах.

В этом случае схема шифратора должна учитывать приоритеты, назначенные каждому входу, и выдать на выход код запроса с самым высоким приоритетом, не учитывая остальные запросы. Приоритет назначается в зависимости от номера канала. В этом случае приоритетный шифратор при одновременном запросе с нескольких каналов формирует на выходе код старшего канала. Легко видеть, что при наличии всего одного возбужденного входа приоритетный шифратор работает так же, как и двоичный.

В приведенной таблице d3..d0 – входные сигналы, представленные единичным кодом, q1..q0 – выходной двоичный код, g – выходной сигнал, отмечающий наличие запросов. При составлен таблицы учитывается , ято приоритетны шифратор выдает требуемый код qi, если di=1, а на входах с более высокими номерами dj = 0 (j > i). Cостояния входов с более низкими номерами (j<i) являются безразличными.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| d3 | d2 | d1 | d0 | q1 | q0 | g |
| 1 | х | х | х | 1 | 1 | 1 |
| 0 | 1 | х | х | 1 | 0 | 1 |
| 0 | 0 | 1 | х | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

0  
1  
2  
3

0  
1

g

CD

Рис. 3.7. Приоритетный шифратор, таблица истинности, обозначение,   
схема, временные диаграммы

g

1

d3  
d2  
d1

d0

d1

&

q0

d2

1

d3

1

q1

d3

d2

d

t

t

1

2

3

0

0

1

2

3

4

5

6

7

8

9

A

B

C

D

E

F

q

По таблице построены временные диаграммы и записаны логические функции для выходных сигналов, для упрощения которых, использовано тождество: .



По данным функциям построена приведенная схема (рис. 3.7). Описание приоритетного шифратора на языке Verilog может быть составлено по логическим функциям.

Другой вариант описания основан на использовании последовательного оператора, который применительно к комбинационной схеме содержит только ключевое слово *always*, а символ *@* и список чувствительности отсутствуют. В этом случае последовательный оператор будет срабатывать после изменения любого из входных сигналов подобно параллельному оператору, что позволяет описывать комбинационные схемы и использовать в этих описаниях операторы *case* и *if*, что расширяет функциональные возможности. Выходной сигнал, формируемый данным оператором, необходимо описать как *reg* (Строка 5). Выход «g» - наличие запросов формируется параллельным оператором.

Селектором оператора «case» является 4-разрядный двоичный сигнал «d». Оператор «case» формирует выходной сигнал q в соответствии с таблицей истинности, в которой входной сигнал может содержать значения «z» и «x».

// Шифратор 1

module v33\_cd (d, q,g); /2

input [3:0] d; //3

output [1:0] q; //4

reg [1:0] q; //5

output g; //6

always case (d) //7

4'b0001:q=2’b00; //8

4'b001x:q=2’b01; //9

4'b01xx:q=2’b10; //10

4'b1xxx:q=2’b11; //11

endcase //12

assign g = | q; //13

endmodule //14

**Задание 3.3.** Иерархический **проект приоритетного шифратора 4 в 2.**

Разработайте иерархический проект, содержащий модули шифратора введенный по схеме *s33\_cd*  и введенный по писанию *v33\_cd*. Выполните этапы 1-5, создайте символы модулей.

Разработайте ведущий модуль sv3*3\_cd.* Выполните этапы 1-10. При работе с отладочной платой подайте входные сигналы от кнопок, а к выходам подключите светодиоды. Объясните работу приоритетного шифратора при одновременном нажатии нескольких кнопок.

**§ 3.4. Преобразователя двоичного кода в код Грея.**

Преобразование перемещений (или угла поворота) в двоичный код производится с использованием специальных кодирующих масок, имеющих прозрачные и непрозрачные участки. Кодирующая маска перемещается в пространстве между светодиодами (СД) и фотодиодами (ФД), при этом фотодиоды формируют выходной код (Рис. 3.8).

d3

d2

d1

d0

0 1 2 3 4 5 6 7 8 9 10 1 12 13 14 15

Рис. 3.8. Структура преобразователя перемещения в двоичный код

СД

ФД

Кодирующая маска

При использовании двоичного кодирования элементов маски, переход от одного дискретного значения к другому, когда изменяются значения нескольких разрядов, сопровождается ошибками неоднозначности. Изменение состояний отдельных разрядов не может быть строго одновременным из-за погрешностей изготовления маски.

Для устранения ошибок неоднозначности в преобразователях используют код Грея, в котором при переходе от любого значения к следующему изменяется только один разряд. Получение последовательности чисел в коде Грея иллюстрируется геометрическими представлениями двоичных чисел, в которых отрезки имеют единичную длину, а координаты вершин образуют двоичные коды. (Рис. 3.9). Двухразрядные числа кода Грея получаются при обходе контура на плоскости (двумерного куба): 00 - 01 - 11 – 10). Действительно, каждый переход от одного значения к другому сопровождается изменением только одного разряда. Последовательность кодов Грея обозначает клетки карты Карно, для соседних клеток изменяется только один разряд. Последовательность 3-разрядных чисел в коде Грея получается при обходе вершин трехмерного куба: 000 - 001 - 011 - 010 - 110 - 111 - 101 - 100.

11

01

00

10

011

001

000

010

111

101

100

110

Рис. 3.9 Двумерный и  
 3-мерный кубы

Преобразование двоичного кода в код Грея выполняется по правилу: старшие разряды совпадают, а каждый следующий разряд кода Грея равен сумме по модулю два данного и предыдущего разрядов двоичного кода. В формулах разряды двоичного кода обозначены буквой b, а кода Грея – g.

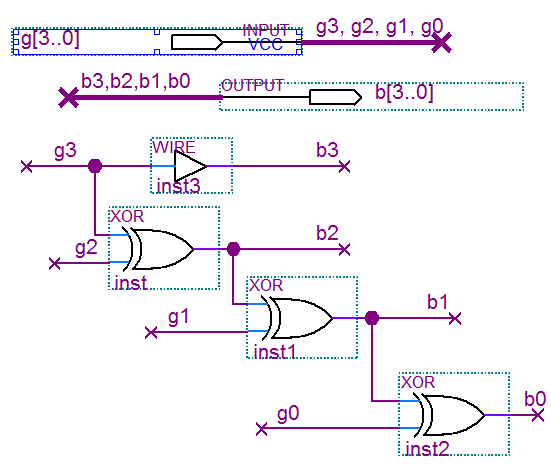
Для обратного преобразования кода Грея в двоичный код старшие разряды также совпадают, но каждый следующий двоичный разряд двоичного кода получается суммированием по модулю два всех предыдущих разрядов кода Грея.

|  |  |
| --- | --- |
| Преобразование двоичного кода  в код Грея | Преобразование кода Грея  в двоичный код |
|  |  |

Приведенные логические функции позволяют составить схемы и описания преобразователей.

**Задание 3.4. Преобразователь кода Грея в двоичный код.**

Разработайте иерархический проект преобразователя 4-разрядного кода Грея в двоичнsq код с именем sv34\_gray\_no\_bin, содержащий модуль, созданный по схеме s34\_gray\_no\_bin и модуль по описанию языке Verilog v34\_gray\_no\_bin(рис. 3.10 – 3.13).



//Код Грея - в двоичный 1

module v34\_gray\_to\_bin (g, b);//2

input [3:0]g; //3

output [3:0] b; //4

assign b[3]= g[3]; //5

assign b[2]= g[3]^g[2]; //6

assign b[1]= g[3]^g[2]^g[1]; //7

assign b[0]= g[3]^g[2]^g[1]^g[0]; //8

endmodule //9

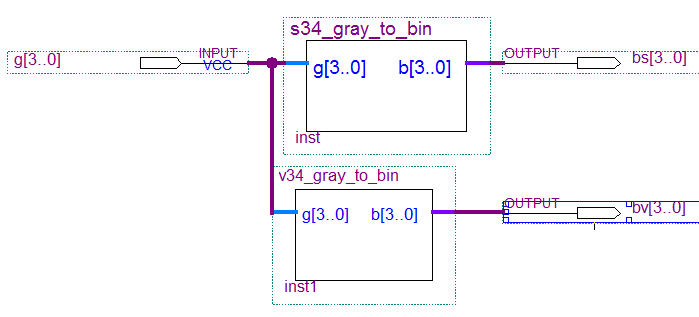


Рис. 3. 10 Проект s34\_gray\_no\_bin

Рис. 3. 11 Описание v34\_gray\_no\_bin

Рис. 3. 12. Проект sv34\_gray\_no\_bin

При моделировании и формировании тестовых сигналов от счетчика (кнопка с буквой «С») в окне Count Value в строке Radix установите Hexadecimal и выберите Gray Code - код Грея (рис. 3.13).



Рис. 3.13. Временные диаграммы преобразователя кода Грея в двоичный код

Изобразите кодирующую маску датчика перемещения для кода Грея, поясните работу преобразователей. Определите максимальное время задержки сигнала.

**Задание 3.5.** Разработайте иерархический проект преобразователя двоичного 4-разрядного кода в код Грея с именем sv35\_bin\_to\_graay, содержащий модули sv35\_bin\_to\_graay и sv35\_bin\_to\_graay. Выполните моделирование, определите максимальное время задержки сигнала. Изобразите кодирующую масоку датчика перемещения для двоичного кода, поясните работу преобразователя и возможные ошибки. Проведите сравнительную оценку быстродействия прямого и обратного преобразователей в код Грея.

**Задание 3.6.** Иерархический **проект двоичного шифратора 4 в 2.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| d3 | d2 | d1 | d0 | q1 | q0 | g |
| 1 | х | х | х | 1 | 1 | 1 |
| 0 | 1 | х | х | 1 | 0 | 1 |
| 0 | 0 | 1 | х | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

По методике задания 3.3 Разработайте схему и описание двоичного шифратора (обычного, не приоритетного) в соответствии с приведенной таблицей истинности. Разработайте иерархический проект s*v36\_cd*, содержащий два модуля шифратора: введенный по схеме *s36\_cd*  и введенный по писанию *v36\_cd*. Выполните моделирование.

Объясните работу двоичного шифратора, срсавните ее с работой приоритетного шифратора при одновременной подаче нескольких единичных сигналов.

**Контрольные вопросы**

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | Функциональное назначение,области применения дешифраторов и шифраторов  Поясните составление таблицы истинности приоритетного шифратора.  Поясните временные диаграммы для приоритетного шифратора.  Код Грея является непозиционным и рефлексивным. Поясните эти свойства.  Символы и знаки, которые отображаются на семисегментном индикаторе.  Приведите несколько способов описания дешифратора на Verilog.  Составьте таблицу истинности для компаратора с 2-разрядными входами.  Запишите логическую функцию для признака «не равно».  Кодировка сигналов от кнопок на отладочной плате.  Кодировка сигналов, подаваемых на индикаторы и светодиоды. |