Практическая работа № 1

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ПРЕОБРАЗОВАТЕЛЯ КОДА

Целью работы является разработка и исследование электронного устройства преобразующего дво-

ично-десятичный код в семисегментный код с целью дальнейшей визуальной индикации.

1.1. Задание

Для заданного двоично-десятичного кода разработать принципиальную схему преобразователя ко-

да. Схему реализовать в базисе И-НЕ на логических элементах 555 серии.

1.1.1. Указания к разработке принципиальной схемы

Для получения необходимого вида выходного кода устройства рассмотрим стандартный однораз-

рядный семисегментный индикатор (рис. 1.1). На рисунке каждый сегмент индикатора обозначен соответствующей буквой (a, b, c, d, e, f, g). Состояние каждого сегмента (по- гашен – светится) определяется состоянием соответствующего входного сигнала (0 или

1).

Исходными данными для разработки схемы является заданный двоично-

десятичный код.

Рассмотрим процедуру разработки принципиальной схемы преобразователя ко-

да на примере двоично-десятичного кода 1-2 4 8.

*a*

*f g b*

(4)

Преобразователь кода

(7)

Семисегментный индикатор

*e d с*

Рис. 1.1

Рис. 1.2

Функциональная схема преобразователя кода.

Проектирование преобразователя осуществляется на основании таблицы 1.1, определяющей связи выходных сигналов с состояниями входного кода.

Таблица 1.1

|  |  |  |
| --- | --- | --- |
| №сост. | Входной код | Выходной код |
| Q4 | Q3 | Q2 | Q1 | a | b | c | d | e | f | g |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 1 0 01 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Из таблицы 1.1 необходимо определить 7 логических функций:

*a* = *Fa* (*Q*4 , *Q*3 , *Q*2 , *Q*1 );

*b* = *Fb* (*Q*4 , *Q*3 , *Q*2 , *Q*1 );

*c* = *Fc* (*Q*4 , *Q*3 , *Q*2 , *Q*1 );

*d* = *Fd* (*Q*4 , *Q*3 , *Q*2 , *Q*1 );

*e* = *Fe* (*Q*4 , *Q*3 , *Q*2 , *Q*1 );

*f* = *F f* (*Q*4 , *Q*3 , *Q*2 , *Q*1);*g* = *Fg* (*Q*4 , *Q*3 , *Q*2 , *Q*1 ).

Минимизация данных функций в этом случае может быть проведена при помощи карт Карно.

Поскольку разметка карт Карно не будет изменяться, то имеет смысл составить вспомогательную карту Карно. В клетках данной карты проставлены № состояний таблицы 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Всп. |  |  |  |  |
|  | 9 | 7 | 6 | 8 |
|  | 1 | - | - | 0 |
|  | - | - | - | - |
|  | 5 | 3 | 2 | 4 |

В таблице “-” означает запрещенное состояние т.е. состояние, в котором преобразователь не может находиться в нормальном режиме работы.

Далее, исходя из данных табл. 1, при помощи вспомогательной карты заполняются карты Карно для искомых логических функций. При этом в клетки карт проставляются значения логических функций для соответствующих состояний. Затем производится объединение единичных клеток в контура в соответствии с правилами:

- в контур может входить 1, 2, 4, 8,16 клеток;

- контур имеет прямоугольную форму;

- одна и та же единица может входить в несколько контуров;

- контур может включать в себя запрещенные состояния;

- чем больше единиц объединяет контур, тем проще схемная реализация логической функции.

Карты Карно для искомых логических функций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a |  |  |  |  |
|  | 1 | 1 | 1 | 1 |
|  | 0 | - | - | 1 |
|  | - | - | - | - |
|  | 1 | 1 | 1 | 0 |

 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| b |  |  |  |  |
|  | 1 | 1 | 0 | 1 |
|  | 1 | - | - | 1 |
|  | - | - | - | - |
|  | 0 | 1 | 1 | 1 |

 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| c |  |  |  |  |
|  | 1 | 1 | 1 | 1 |
|  | 1 | - | - | 1 |
|  | - | - | - | - |
|  | 1 | 1 | 0 | 1 |

 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| d |  |  |  |  |
|  | 1 | 0 | 1 | 1 |
|  | 0 | - | - | 1 |
|  | - | - | - | - |
|  | 1 | 1 | 1 | 0 |

 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| e |  |  |  |  |
|  | 0 | 0 | 1 | 1 |
|  | 0 | - | - | 1 |
|  | - | - | - | - |
|  | 0 | 0 | 1 | 0 |

 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| f |  |  |  |  |
|  | 1 | 0 | 1 | 1 |
|  | 0 | - | - | 1 |
|  | - | - | - | - |
|  | 1 | 0 | 0 | 1 |

 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| g |  |  |  |  |
|  | 1 | 0 | 1 | 1 |
|  | 0 | - | - | 0 |
|  | - | - | - | - |
|  | 1 | 1 | 1 | 1 |

 

Далее при помощи закона де Моргана уравнения преобразуются к виду необходимому для реализации схемы в базисе И-НЕ.

 

 

 



Реализация схемы производится на ТТЛ-микросхемах малой степени интеграции К155:

- К555ЛА3 – четыре логических элемента 2И-НЕ;

- К555ЛА4 – три логических элемента 3И-НЕ;

- К555ЛА1 – два логических элемента 4И-НЕ;

Напряжение питания микросхем – 5В. Свободные входы логических элементов необходимо подключать к источнику питания +5В±10% через резистор 1кОм. К каждому резистору допускается подключение 20 свободных входов.

Схема электрическая принципиальная

DD4.1

DD1.1

&

a

&

&

DD5.1

DD1.2

b

&

&

DD1.3

DD5.2

DD1.4

&

c

&

DD2.1

&

DD4.2

&

DD2.2

d

&

DD3.1

&

DD2.4

&

DD2.3

&

e

&

DD6.1

DD3.2

f

&

&

DD6.2

DD3.3

g

&

# Проверка схемы в Proteus 7.7

Для того, чтоб получить 0, по заданию нужно на вход подать 0110, который соответствует нулевому состоянию.



На вход Q1 подаем вместо нуля единицу. Получается код 1110, который соответствует первому состоянию.



На вход Q3 подаем 0 вместо 1. Получается код 0101, который соответствует состоянию 4.



Выводы:

В данной практической работе мы исследовали электронное устройство, которое преобразовывает двоично-десятичный код в семисегментный для дальнейшей визуальной индикации. Для этого мы используя заданный двоично-десятичный код разработали принципиальную схему преобразователя кода, смоделировали в программе Proteus 7.7, и убедились, что она работает.