## Задание

Реализовать одноразрядный демультиплексор 1-в-8 со входом разрешения EN (активный уровень сигнала EN – логический ноль; при подаче на вход EN сигнала с уровнем логической единицы, выходные сигналы демультиплексора должны устанавливаться в высокоимпедансное состояние). Реализовать дешифратор для семисегментного индикатора (рис. 1), который преобразует входной четырехразрядный двоичный код в отображаемые на индикаторе шестнадцатеричные цифры от 0 до F. Выполнить прототипирование двух разработанных устройств.



Рисунок 1. Расположение сегментов индикатора.

## Выполнение

1. Демультиплексором называется комбинационное логическое устройство, предназначенное для управления передачей данных от одного входного канала на несколько выходных. В случае представленного выше задания, необходим один входной информационный сигнал, три входных адресных сигнала и восемь выходных сигналов. Составим таблицу истинности (табл. 1), где EN – вход разрешения, D – входной информационный сигнал, Z – высокоимпедансное состояние.

Таблица 1.

**Таблица истинности для одноразрядного демультиплексора 1-в-8.**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EN** | **A2** | **A1** | **A0** | **F7** | **F6** | **F5** | **F4** | **F3** | **F2** | **F1** | **F0** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **D** |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | **D** | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | **D** | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | **D** | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | **D** | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | **D** | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | **D** | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | **D** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | Z | Z | Z | Z | Z | Z | Z | Z |
| 1 | 0 | 0 | 1 | Z | Z | Z | Z | Z | Z | Z | Z |
| 1 | 0 | 1 | 0 | Z | Z | Z | Z | Z | Z | Z | Z |
| 1 | 0 | 1 | 1 | Z | Z | Z | Z | Z | Z | Z | Z |
| 1 | 1 | 0 | 0 | Z | Z | Z | Z | Z | Z | Z | Z |
| 1 | 1 | 0 | 1 | Z | Z | Z | Z | Z | Z | Z | Z |
| 1 | 1 | 1 | 0 | Z | Z | Z | Z | Z | Z | Z | Z |
| 1 | 1 | 1 | 1 | Z | Z | Z | Z | Z | Z | Z | Z |

1. Спроектируем устройство согласно таблице истинности.
2. Выполним моделирование с целью проверки корректности работы устройства. Результаты представлены на рисунке 2.

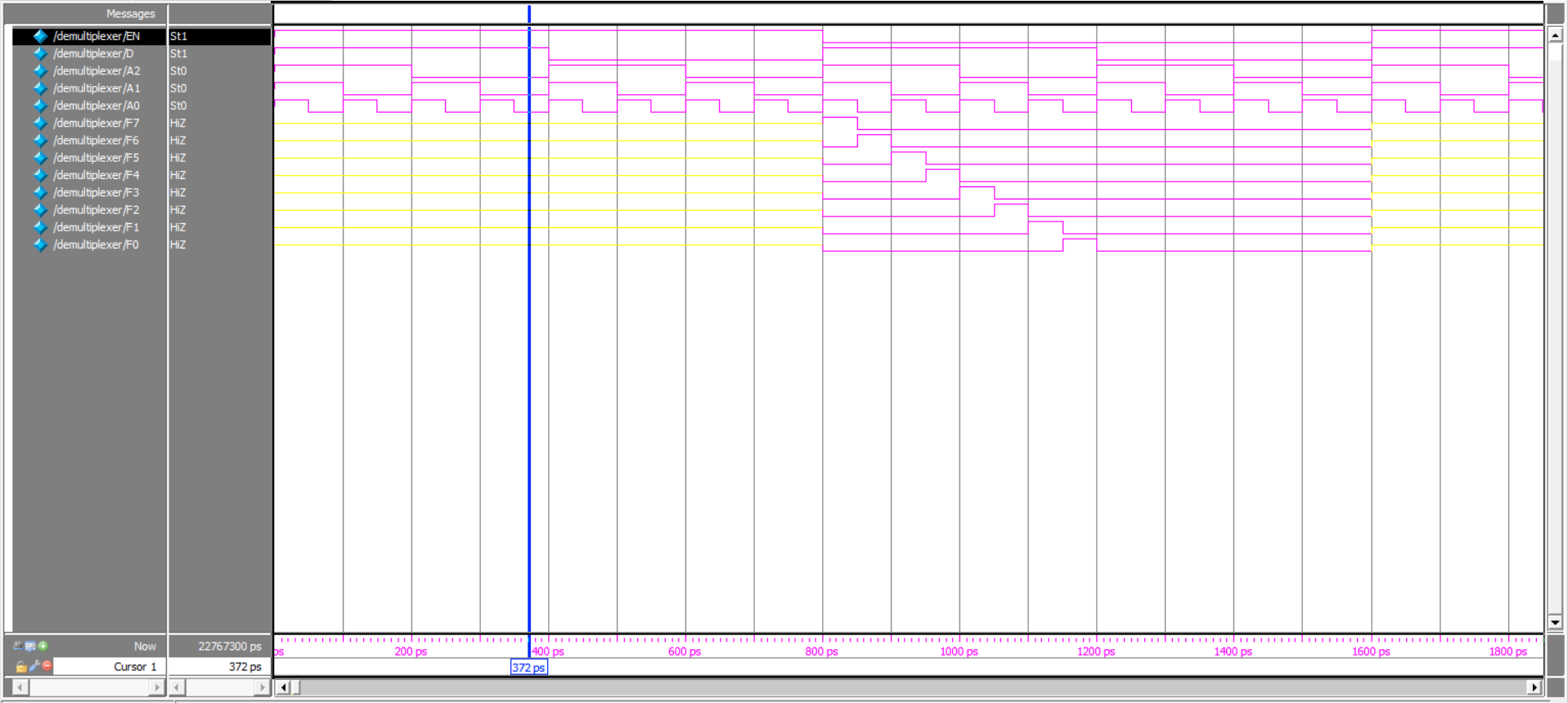


Рисунок 2. Результаты моделирования демультиплексора 1-в-8.

1. Выполним прототипирование с помощью онлайн лаборатории. Для этого подключим файл Verilog к проекту Quartus, назначим пины и скомпилируем проект. После чего прошьем отладочную плату используя rbf файл прошивки. Результаты приведены на рисунках 3-5.

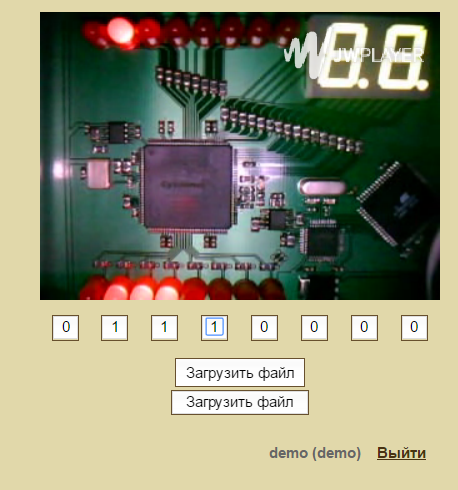


Рисунок 3. EN = 0, D = 1, A2 = 1, A1 = 1, A0 = 0. Результат: F6

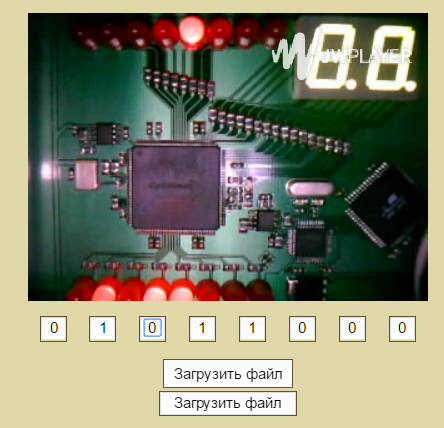


Рисунок 4. EN = 0, D = 1, A2 = 0, A1 = 1, A0 = 1. Результат: F3

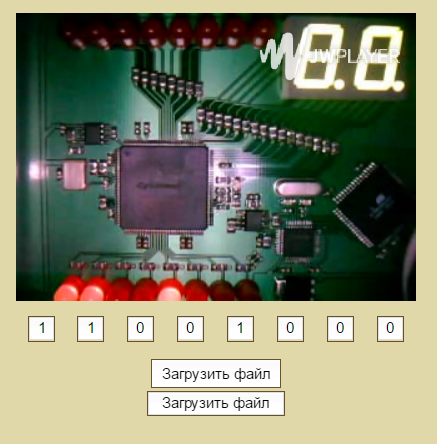


Рисунок 5. EN = 1, D = 1, A2 = 0, A1 = 0, A0 = 1. Результат: высокоимпедансный сигнал на всех выходах

1. Дешифратор – комбинационное устройство, преобразующие двоичный код в код другой системы счисления, например, двоичной или восьмеричной. Составим таблицу истинности (табл. 2) для дешифратора, отображающего четырехразрядные двоичные числа 0000-1111 на семисегментном индикаторе в шестнадцатеричной системе счисления.

Таблица 2.

**Таблица истинности для дешифратора.**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Q3** | **Q2** | **Q1** | **Q0** | **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

1. Спроектируем устройство согласно приведенной таблице истинности. На рисунке 6 приведен пример работы.

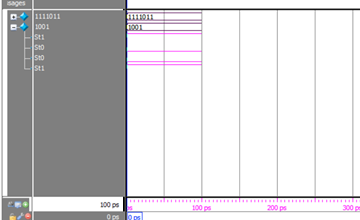


Рисунок 7. Пример работы дешифратора в режиме моделирования.

1. Выполним прототипирование с помощью онлайн лаборатории. Для этого подключим файл Verilog к проекту Quartus, назначим пины и скомпилируем проект (рис. 8). После чего прошьем отладочную плату используя rbf файл прошивки. Результаты приведены на рисунках 9 и 10. Кроме того, были проверены все значения от 0 до F.

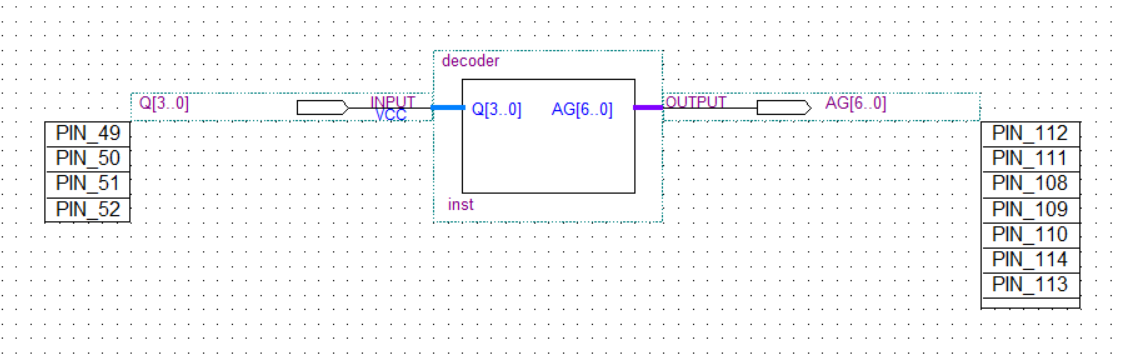


Рисунок 8. Файл bdf в Quartus для второго задания.

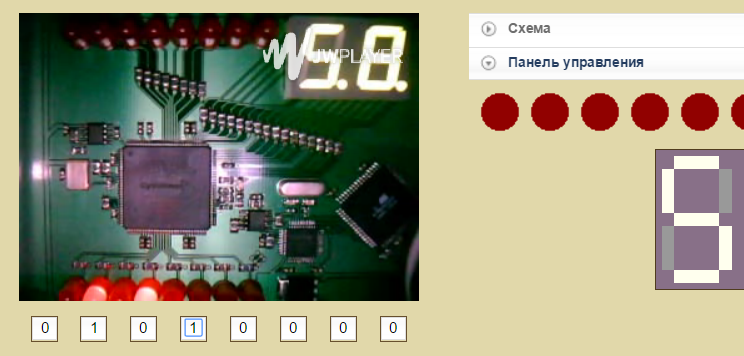


Рисунок 9. 0101 в двоичной – 5 в шестнадцатеричной.

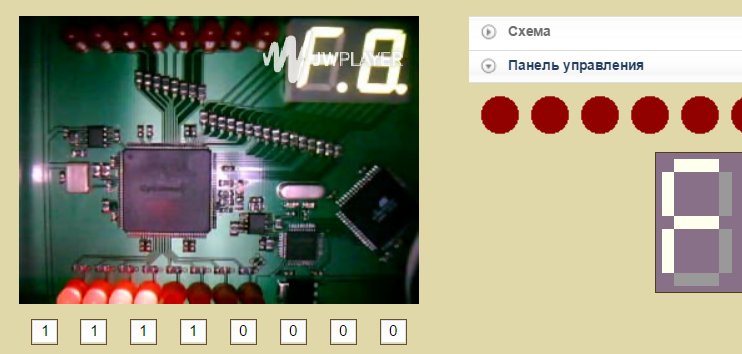


Рисунок 10. 1111 в двоичной – F в шестнадцатеричной.

## Выводы

В результате выполнения данной лабораторной работы были рассмотрены устройство и принцип действия таких комбинационных устройств как мультиплексор, демультиплексор, шифратор и дешифратор. Самостоятельно реализован демультиплексор со входом разрешения работы, а также дешифратор для семисегментного индикатора, благодаря чему были получены навыки работы с ним. Был более подробно изучен синтаксис языка Verilog. Рассмотрены принципы работы с неопределенным и высокоимпедансным состояниями.