МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

**Отчет по лабораторной работе № 4**

**по курсу «Схемотехника ЭВМ»**

**Проектирование устройств с Z-состоянием выхода.**

**Вариант 31.**

Выполнил: студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Лаптев

Проверил: ст. преп. каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Белозерских

Лабораторная работа защищена

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Барнаул 2022

**Цель работы:** Получение навыков синтеза различных комбинационных устройств с помощью языка описания аппаратуры AHDL в САПР Altera MAX+PLUS II.

**Задачи:** С помощью текстового редактора (Text Editor) синтезировать комбинационные схемы преобразователя кода, сдвигающего регистра с параллельной загрузкой и дешифратора адреса с заданными параметрами. Подключить их к двунаправленной шине данных в режиме графического редактора (Graphic Editor) и произвести исследование особенностей работы элементов с выходами, имеющими Z-состояние. Произвести измерение временных параметров полученной схемы (Simulator).

**Задание:** Вариант 31 – **61966A**. Необходимо синтезировать схему, состоящую из 2-х устройств, подсоединенных к двунаправленной шине. Выбор какое из устройств в данный момент подключено к шине осуществляется с помощью дешифратора адреса. Управление устройствами осуществляется с помощью внешних сигналов.

**Выполнение работы:**

Получаем данные, необходимые для синтеза комбинационных устройств. Вариант расшифровывается следующим образом:

0x61966A = 01100001’10010110’01101010

Первые 2 байта задают адреса, на которые реагирует дешифратор адреса:

* 01100001 – адресами устройства Регистр являются 0, 5, 6 (при этом дешифратор выдает **1** на линию CS). Т.е. номер бита в байте и является тем адресом, в котором дешифратор должен выдавать **1**.
* 10010110 – адресами устройства Преобразователь Кода являются 1, 2, 4, 7 (при этом дешифратор выдает **1** на линию CS1)

Третий байт отвечает за виды устройств в схеме и способы их синтеза:

10010110 = 1, 0, 0, 1, 0, 1, 1, 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 6-5 | 4 | 3-1 | 0 |
| TRI\_Р | Регистр | TRI\_ПК | Преобр.Кода | ДшА |

ДшА (1бит) – Задает вид синтеза дешифратора адреса:

* **0** – булево выражение (используется для синтеза CS, CS1);

Преобр.Кода (3 бита) – Задает тип преобразователя кода:

* **011** – BIN => код для 7-ми сегментного индикатора + точка;

TRI\_ПК (1 бит) – Задает вид синтеза буферов выхода с Z-состоянием, которые будут использоваться в преобразователе кода.

* **1** – так как указано в Altera MAX+PLUS Help в разделе Implementing Bidirectional Pins, см. **SUBDESIGN bus\_reg3** (будет использоваться при синтезе устройства **Registr**).

Регистр (2 бита) – Задает тип синтезируемого регистра:

* **00** – 8-ми разрядный циклический сдвиговый регистр с параллельной загрузкой (сдвиг осуществляется влево);

TRI\_Р (1 бит) – Задает вид синтеза буферов выхода с Z-состоянием, котороые будут использоваться в устройстве Регистр.

* **1** – так как указано в Altera MAX+PLUS Help в разделе Implementing Bidirectional Pins, см. **SUBDESIGN bus\_reg3** (будет использоваться при синтезе устройства **Registr**).

Следовательно, нам необходимо синтезировать схему с 8-ми разрядным преобразователем кода из двоичных чисел в коды 7-ми сегментного индикатора с точкой и буферами выхода по методу **bus\_reg3**, с 8-ми разрядным циклическим сдвигающим влево регистром с параллельной загрузкой и буферами выхода по методу **bus\_reg3** и дешифратором адреса на основе булева выражения.

С помощью текстового редактора (Text Editor) синтезируем заданный тип преобразователя кода. Код необходимый для синтеза преобразователем кода из двоичных чисел в коды семисегментного индикатора с точкой и буферами выхода по методу **bus\_reg3** и получившийся символ устройства приведены на Рис. 4.1:

SUBDESIGN 7segm

(

i[8..1]: bidir;

en\_din, clk, cs: input;

)

VARIABLE

tt[5..1]:dffe;

t[8..1]:TRI;

BEGIN

tt[].d = i[5..1];

tt[].clk = clk;

tt[].ena = cs AND !en\_din;

TABLE

tt[] => t8, t7, t6, t5, t4, t3, t2, t1;

H"0" => 0, 1, 1, 1, 1, 1, 1, 0;

H"1" => 0, 0, 1, 1, 0, 0, 0, 0;

H"2" => 0, 1, 1, 0, 1, 1, 0, 1;

H"3" => 0, 1, 1, 1, 1, 0, 0, 1;

H"4" => 0, 0, 1, 1, 0, 0, 1, 1;

H"5" => 0, 1, 0, 1, 1, 0, 1, 1;

H"6" => 0, 1, 0, 1, 1, 1, 1, 1;

H"7" => 0, 1, 1, 1, 0, 0, 0, 0;

H"8" => 0, 1, 1, 1, 1, 1, 1, 1;

H"9" => 0, 1, 1, 1, 1, 0, 1, 1;

H"A" => 0, 1, 1, 1, 0, 1, 1, 1;

H"B" => 0, 0, 0, 1, 1, 1, 1, 1;

H"C" => 0, 1, 0, 0, 1, 1, 1, 0;

H"D" => 0, 0, 1, 1, 1, 1, 0, 1;

H"E" => 0, 1, 0, 0, 1, 1, 1, 1;

H"F" => 0, 1, 0, 0, 0, 1, 1, 1;

H"10" => 1, 1, 1, 1, 1, 1, 1, 0;

H"11" => 1, 0, 1, 1, 0, 0, 0, 0;

H"12" => 1, 1, 1, 0, 1, 1, 0, 1;

H"13" => 1, 1, 1, 1, 1, 0, 0, 1;

H"14" => 1, 0, 1, 1, 0, 0, 1, 1;

H"15" => 1, 1, 0, 1, 1, 0, 1, 1;

H"16" => 1, 1, 0, 1, 1, 1, 1, 1;

H"17" => 1, 1, 1, 1, 0, 0, 0, 0;

H"18" => 1, 1, 1, 1, 1, 1, 1, 1;

H"19" => 1, 1, 1, 1, 1, 0, 1, 1;

H"1A" => 1, 1, 1, 1, 0, 1, 1, 1;

H"1B" => 1, 0, 0, 1, 1, 1, 1, 1;

H"1C" => 1, 1, 0, 0, 1, 1, 1, 0;

H"1D" => 1, 0, 1, 1, 1, 1, 0, 1;

H"1E" => 1, 1, 0, 0, 1, 1, 1, 1;

H"1F" => 1, 1, 0, 0, 0, 1, 1, 1;

END TABLE;

t[].oe = cs & en\_din;

FOR j IN 1 TO 8 GENERATE

i[j] = t[j].out;

END GENERATE;

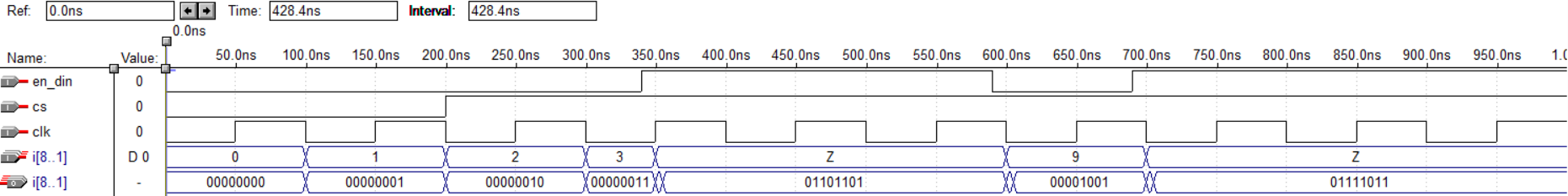
END;

Изображение выглядит как текст

Автоматически созданное описание

*Рис. 4.1. Преобразователь кода из двоичных чисел в коды 7-ми сегментного индикатора.*

В редакторе Waveform Editor строим диаграмму входных воздействий:



*Рис. 4.2. Отклик преобразователя кода на входные сигналы.*

Из диаграммы на Рис.4.2 видно, что пока устройство не выбрано (CS=0) никакие сигналы на выходы двунаправленной шины I[8..1] с него не поступают. Таким образом выходы двунаправленной шины I[8..1] повторяют значения входов шины I[8..1]. При CS=1 устройство начинает работать (происходит защелкивание данных с входов I[8..1] и преобразование кода), но данные на выход двунаправленной шины I[8..1] поступают только в моменты, когда она переключается на вывод с помощью сигнала en\_din=1. При этом входы двунаправленной шины I[8..1] переходят в Z-состояние, а на выходах появляются результаты работы устройства. Далее, с помощью текстового редактора (Text Editor) синтезируем заданный тип регистра. Код необходимый для синтеза циклического сдвигающего влево регистра с параллельной загрузкой и буферами выхода по методу bus\_reg3 и получившийся символ устройства приведены на Рис. 4.3:

SUBDESIGN registr

(

Data[8..1]: bidir;

clk, load, en\_din, cs: input;

)

VARIABLE

FF[8..1]: DFFE;

Tri\_[8..1]:TRI;

BEGIN

FF[].clk = clk;

FF[].ena = cs;

if load==1 then

FF[].D = Data[];

else

FF[].D = (FF[7..1].Q, FF[8].q);

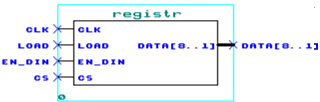
end if;

Tri\_[].in = FF[].Q;

Tri\_[].oe = cs & en\_din;

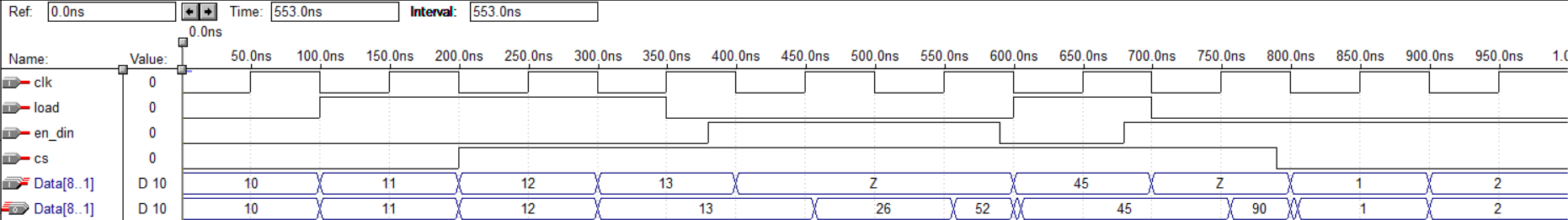
Data[] = Tri\_[].out;

END;



*Рис. 4.3. 8-ми разрядный циклический сдвиговый влево регистр с параллельной загрузкой.*

В редакторе Waveform Editor строим диаграмму входных воздействий:



*Рис. 4.4. Отклик циклического сдвигового регистра на входные сигналы.*

Из диаграммы на Рис.4.4 видно, что пока устройство не выбрано (CS=0) никакие сигналы на выходы двунаправленной шины Data[8..1] с него не поступают. Таким образом выходы двунаправленной шины Data[8..1] повторяют значения входов шины Data[8..1]. При CS=1 устройство начинает работать (по сигналу load=1 происходит защелкивание данных с входов Data[8..1] и, при load=0, осуществляется циклический сдвиг влево), но данные на выход двунаправленной шины Data[8..1] поступают только в моменты, когда она переключается на вывод с помощью сигнала en\_din=1. При этом входы двунаправленной шины Data[8..1] переходят в Z-состояние, а на выходах появляются результаты работы устройства.

Теперь, с помощью текстового редактора (Text Editor), используя булевы выражения, синтезируем дешифратор адреса Рис. 4.5:

SUBDESIGN addr

(

Addr[3..1]: input;

CS: output; -- 0, 5, 6

CS1: output; -- 1, 2, 4, 7

)

BEGIN

CS = (Addr[] == b"000") # (Addr[] == b"101") # (Addr[] == b"110");

CS1 = (Addr[] == b"001") # (Addr[] == b"010") # (Addr[] == b"100") # (Addr[] == b"111");

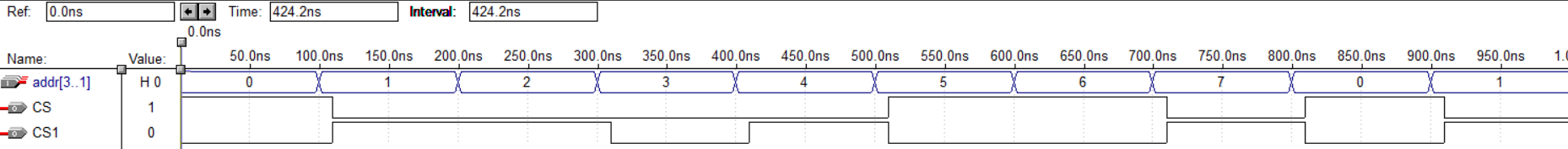
END;

Изображение выглядит как текст

Автоматически созданное описание

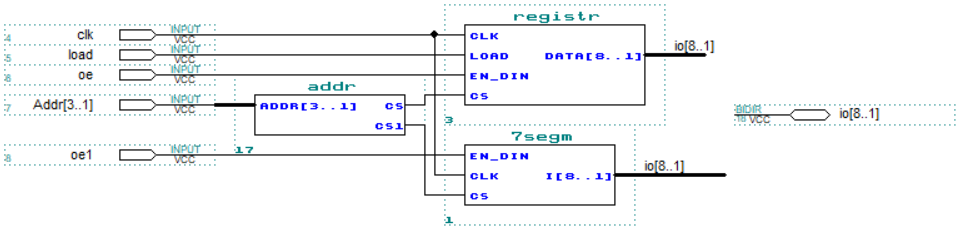
*Рис. 4.5. Дешифратор адреса.*

В редакторе Waveform Editor строим диаграмму входных воздействий и проверяем правильность выработки сигналов, которые будут отвечать за выбор того или иного устройства нашей схемы:



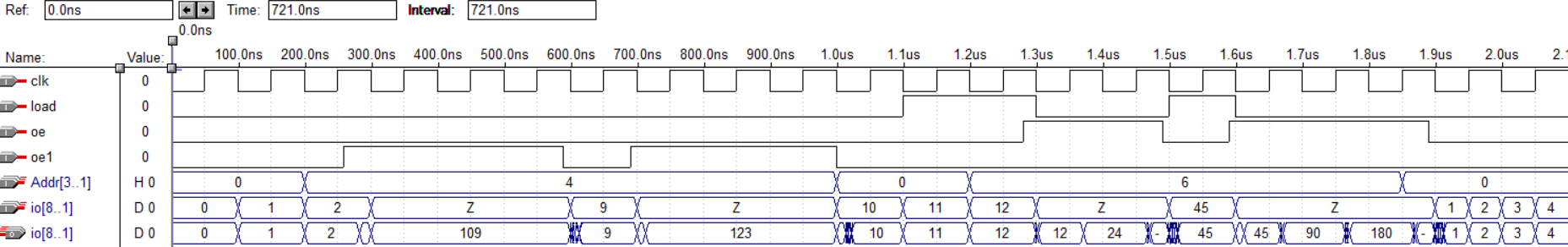
*Рис. 4.6. Отклик дешифратора адреса на входные сигналы.*

Используя графический редактор (Graphic Editor), строим итоговую схему, состоящую из символов синтезированных устройств, с присоединенными к ним элементами входов и выходов. См. Рис.4.7.



*Рис. 4.7. Итоговая схема устройства в редакторе Graphic Editor.*

В редакторе Waveform Editor строим диаграмму входных воздействий и проверяем правильность работы схемы в целом. При этом можно использовать заготовки, полученные при проверке устройств, входящих в состав схемы. Результаты симуляции приведены на Рис.4.8 и Рис.4.9.

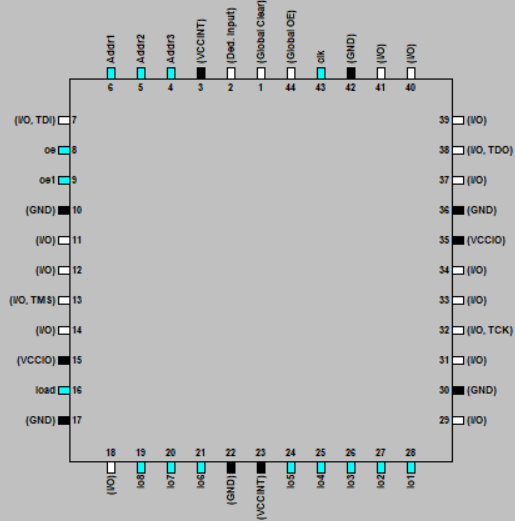


*Рис. 4.8. Отклик итоговой схемы устройства на входные сигналы.*

*Рис. 4.9. Отклик итоговой схемы устройства на входные сигналы.*

Из диаграммы на Рис.4.8 видно, что пока ни одно из устройств, входящих в состав схемы, не выбрано (Addr[3..1] не равен ни одному из адресов этих устройств – в данном случае это при Addr[3..1]=0), они никак не вмешиваются в сигналы на выходах двунаправленной шины IO[8..1] с него не поступают. Таким образом выходы двунаправленной шины IO[8..1] повторяют значения входов шины IO[8..1]. При Addr[3..1]=5 начинает работать преобразователь кода (происходит защелкивание данных с входов IO[8..1] и преобразование кода), но данные на выход двунаправленной шины IO[8..1] поступают только в моменты, когда она переключается на вывод с помощью сигнала оe1=1. При этом входы двунаправленной шины IO[8..1] переходят в Z-состояние, а на выходах появляются результаты работы преобразователя кода. При Addr[3..1]=6 начинает работать циклический сдвигающий регистр (по сигналу load=1 происходит защелкивание данных с входов IO[8..1] и, при load=0, осуществляется циклический сдвиг влево), но данные на выход двунаправленной шины IO[8..1] поступают только в моменты, когда она переключается на вывод с помощью сигнала оe=1. При этом входы двунаправленной шины IO[8..1] переходят в Z-состояние, а на выходах появляются результаты работы устройства.

Диаграмма на Рис.4.9, доказывает, что работа устройств, входящих в состав схемы, с двунаправленной шиной IO[8..1] и их реакция на селектирующий сигнал Addr[3..1] реализованы правильно и полноценно, т.к. видно, что никакие сигналы не только на шине IO[8..1], но и на управляющих линиях не влияют на устройства до тех пор, пока одно из них не будет выбрано.



*Рис. 4.10. Floorplan Editor.*

Изображение выглядит как стол

Автоматически созданное описание

*Рис. 4.11. Измерение временных задержек в Timing Analyzer.*

**Вывод:** Получены навыки синтеза различных комбинационных устройств с помощью языка описания аппаратуры AHDL в САПР Altera MAX+PLUS II. С помощью текстового редактора (Text Editor) синтезированы комбинационные схемы преобразователя кода, сдвигающего регистра с параллельной загрузкой и дешифратора адреса с заданными параметрами, подключены к двунаправленной шине данных в режиме графического редактора (Graphic Editor) и произведено исследование особенностей работы элементов с выходами, имеющими Z-состояние. Произведено измерение временных параметров полученной схемы (Simulator).