

## Лабораторная работа № 2. Исследование комбинационного цифрового устройства.

**Цель работы:** получение основных навыков проектирования схем в редакторе пакета **Quartus15**. Изучение функционирования КЦУ, состоящего из кодопреобразователя и демультимплексора.

### Задание на работу в лаборатории.

1. Создать проект под названием **coder1**. Открыть VHDL файл и записать **программу 2.1**, отражающую функционирование преобразователя кода четырехразрядного двоичного числа в соответствующий ему символ на семисегментном индикаторе. Сохранить файл с названием **coder**, установить его старшим в иерархии и откомпилировать.
2. Пользуясь «Приложением 2», получить диаграммы при **интервалах импульса на входной шине – 20нс. Поразрядно шину не разворачивать!**

#### Программа 2.1.

```
library ieee;
use ieee.std_logic_1164.all;

entity coder is
port(symb: in bit_vector(3 downto 0);
ind: out bit_vector(6 downto 0)
);
end coder;

architecture BBB of coder is
begin
p0: process (symb)
begin
case symb is
when "0000"=>ind<="1000000";
when "0001"=>ind<="1111001";
when "0010"=>ind<="0100100";
when "0011"=>ind<="0110000";
when "0100"=>ind<="0011001";
when "0101"=>ind<="0010010";
when "0110"=>ind<="0000010";
when "0111"=>ind<="1111000";
when "1000"=>ind<="0000000";
when "1001"=>ind<="0010000";
when "1010"=>ind<="0001000";
when "1011"=>ind<="0000011";
```

```

when "1100"=>ind<="1000110";
when "1101"=>ind<="0100001";
when "1110"=>ind<="0000110";
when "1111"=>ind<="0001110";
end case;
end process;
end;

```

3. Открыть новый VHDL файл и записать **программу 2.2**, отражающую функционирование кодопреобразователя с управляемым демультиплексором на выходе. Сохранить файл под новым именем - **coder1**, установить его старшим в иерархии и откомпилировать.
4. Получить временные диаграммы для устройства, задав параметры для **символов 20нс, адреса – 80нс, разрешение вывода – 1.**

## Программа 2.2

```

library ieee;
use ieee.std_logic_1164.all;

entity coder1 is
port(symb: in bit_vector(3 downto 0);
adr:in bit_vector(1 downto 0);
x:in bit;
i_0: out bit_vector(6 downto 0);
i_1: out bit_vector(6 downto 0);
i_2: out bit_vector(6 downto 0);
i_3: out bit_vector(6 downto 0)
);
end coder1;

architecture BBB of coder1 is
signal ind: bit_vector(6 downto 0);
begin
p0: process (symb)
begin
case symb is
when "0000"=>ind<="1000000";
when "0001"=>ind<="1111001";
when "0010"=>ind<="0100100";
when "0011"=>ind<="0110000";
when "0100"=>ind<="0011001";
when "0101"=>ind<="0010010";
when "0110"=>ind<="0000010";
when "0111"=>ind<="1111000";

```

```

when "1000"=>ind<="0000000";
when "1001"=>ind<="0010000";
when "1010"=>ind<="0001000";
when "1011"=>ind<="0000011";
when "1100"=>ind<="1000110";
when "1101"=>ind<="0100001";
when "1110"=>ind<="0000110";
when "1111"=>ind<="0001110";
end case;
end process;
p1: process(adr,x)
begin
if (adr='00') then
    if(x='1') then i_0<=ind;
    elsif(x='0') then i_0<="1111111";
    end if;
elsif(adr='01') then
    if(x='1') then i_1<=ind;
    elsif(x='0') then i_1<="1111111";
    end if;
elsif(adr='10') then
    if(x='1') then i_2<=ind;
    elsif(x='0') then i_2<="1111111";
    end if;
elsif(adr='11') then
    if(x='1') then i_3<=ind;
    elsif(x='0') then i_3<="1111111";
    end if;
end if;
end process;
end;

```

5. Пользуясь «Приложением 3» произвести разводку выводов схемы для работы в макете таким образом, чтобы ввод **числа** осуществлялся с тумблеров **SW9,SW8,SW7,SW6**(SW9 – старший разряд), ввод **адреса** – с тумблеров **SW1,SW0**(SW1 – старший разряд), ввод разрешения «X» - с тумблера **SW2**. Вывод производить на **сегментные индикаторы с 0-го по 3-ий**. **После компиляции файла планировщика еще раз откомпилируйте файл верхнего уровня!**
6. Пользуясь «Приложением 4» произвести программирование кристалла FPGA макета. **Изначально установить на макете тумблер SW2, соответствующий значению «X», в «0».**
7. Проверить работу устройства. **Порядок следующий: устанавливаем адрес, устанавливаем X=1, устанавливаем любое число.**

**Убедившись, что вывод производится, устанавливаем  $X=0$ , затем устанавливаем новый адрес и повторяем вывод любого числа. Продемонстрировать работу преподавателю.**

**Отчет должен содержать программы функционирования устройств и диаграммы их работы.**