



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт (филиал) КБиСП направление 12.03.01

Кафедра КБ6 “Приборы и информационно-измерительные системы”

Дисциплина “РУНБПЛИС”

ОТЧЕТ

по лабораторной работе на тему:

Проектирование мультиплексора в Vivado

Студент А.Д. Глухов

подпись, дата

инициалы и фамилия

Группа БПБО-02-20 шифр 20Б0924

МОСКВА 2022 г.

Содержание

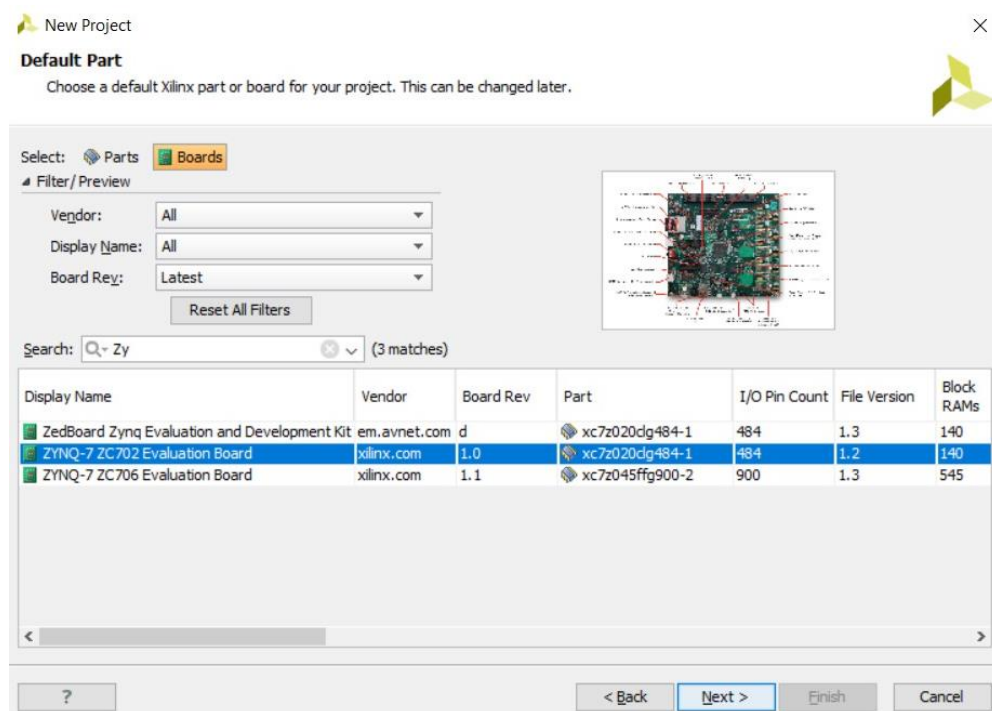
Задание на лабораторную работу	3
Проектирование мультиплексора в среде Vivado.....	4
Заключение	11
Список источников	12
Приложение А.	13
Приложение Б.....	16

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Необходимо осуществить проектирование мультиплексор 4x1 в среде разработки Xilinx Vivado, написать код модуля на языке Verilog, выполнить тестирование, получить электрическую схему и провести моделирование, доказывающее работоспособность устройства.

ПРОЕКТИРОВАНИЕ МУЛЬТИПЛЕКСОРА В СРЕДЕ VIVADO

При создании нового проекта в среде разработки Vivado необходимо выбрать модель ПЛИС или отладочную плату. Для отработки проектирования дешифратора на базе ПЛИС плата «ZYNQ-7 ZC702 Evaluation Board». На рисунке 2 представлен скриншот среды Vivado и выбор ПЛИС.



Рисунке 2 - Скриншот среды Vivado и выбор ПЛИС

После создания проекта необходимо создать файл с исходным кодом для дешифратора, который будет написан на языке Verilog. Для этого нужно выбрать инструмент «Add Sources» и создать файл «Design Source» в директории проекта, выбрав расширение Verilog. На рисунке 3 представлен скриншот окна управления исходными файлами.

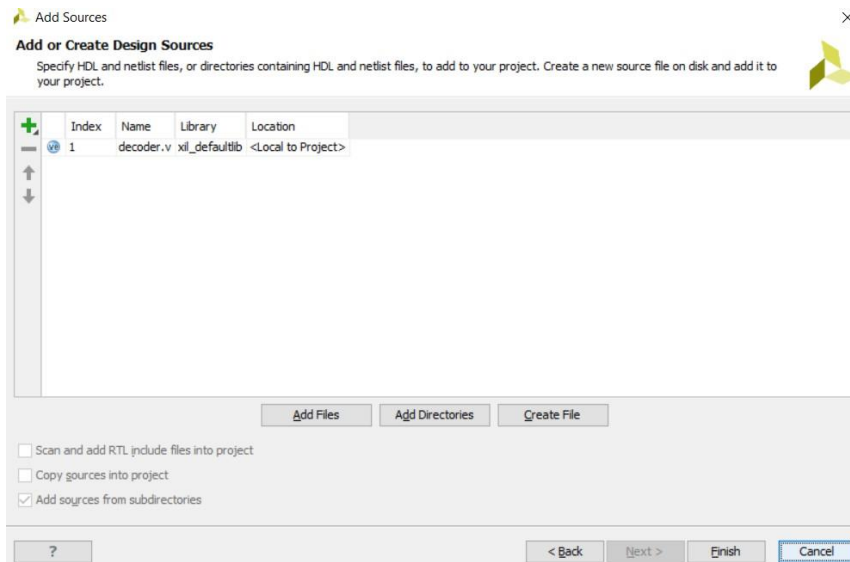


Рисунок 3 - Скриншот окна управления исходными файлами

Во время создания файла с исходным кодом модуля мультиплексора можно определить входные и выходные сигналы для предварительной генерации или же пропустить и написать их вручную. По знанию принципов функционирования мультиплексора был написан код программы двумя способами, который представлен на рисунках 4, 5.

```

23 module mult(
24     input [3:0] X,
25     input [1:0] S,
26     input EN,
27     output Y
28 );
29
30     reg Y;
31
32     always @*
33     if (EN)
34     begin
35         Y = 0;
36         Y = X[S];
37     end
38 endmodule

```

Рисунок 4 – Скриншот с исходным кодом мультиплексора с использованием оператора присвоения

```

24 module mult(
25     input [3:0] X,
26     input [1:0] S,
27     input EN,
28     output Y
29 );
30
31     reg Y;
32
33     always @*
34     if (EN)
35     begin
36         Y = 0;
37         case (S)
38             2'b00 : Y = X[0];
39             2'b01 : Y = X[1];
40             2'b10 : Y = X[2];
41             2'b11 : Y = X[3];
42         endcase
43     end
44 endmodule

```

Рисунок 5 – Скриншот с исходным кодом мультиплексора с использованием оператора case

Для проверки полученных результатов необходимо написать модуль тестирования (testbench) и осуществить моделирование мультиплексора. Скриншот с кодом модуля тестирования, в котором создаётся экземпляр основного модуля мультиплексора с переопределёнными значениями сигналов, представлен на рисунке 6.

```

42 module testbench;
43     reg [1:0] S;
44     reg EN;
45     reg [3:0] X;
46
47     mult m(X,S,EN,Y);
48
49     initial
50     begin
51         EN = 1;
52         S = 0;
53         X = 1;
54         #50;
55         S = 1;
56         X = 2;
57         #50;
58         S = 2;
59         X = 4;
60         #50;
61         S = 2;
62         X = 8;
63         #50;
64     end
65 endmodule

```

Рисунок 6 – Скриншот с исходным кодом модуля тестирования

После создания модуля тестирования можно запустить моделирование, предварительно выбрав в опциях симуляции нужный модуль. На рисунке 7 представлен скриншот моделирования мультиплексора 4x1.



Рисунок 7 - Скриншот моделирования мультиплексора 4x1

На симуляции видно временные диаграммы тех сигналов, которые описаны в модуле тестирования. Это важный и обязательный этап проверки описанной схемы, доказывающий её полную работоспособность.

Полный исходный код модуля мультиплексора 4x1 и модуля тестирования представлены в приложениях А, Б.

После создания кода и моделирования, среда разработки Vivado в автоматизированном режиме генерирует электрическую схему модуля мультиплексора 4x1. Схемы для обоих способов реализации представлены на рисунках 8, 9.

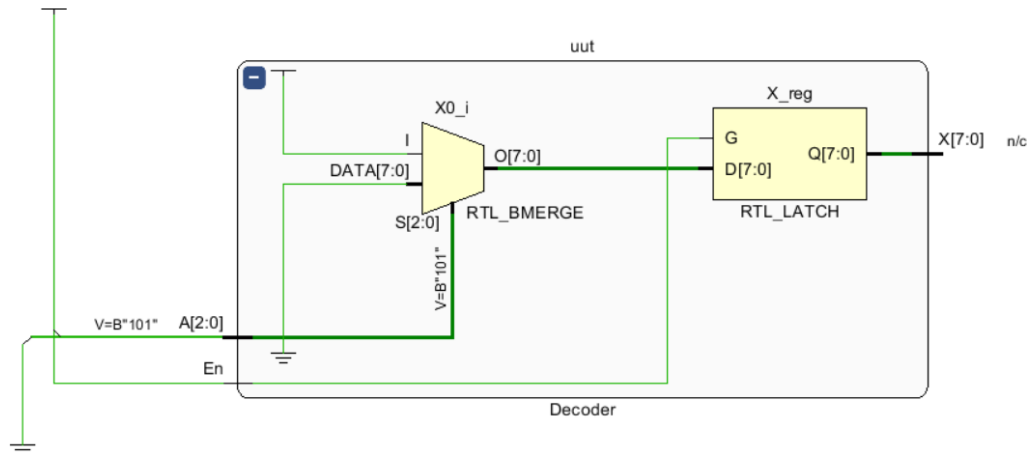


Рисунок 8 – Электрическая схема мультиплексора, реализованного 1 способом

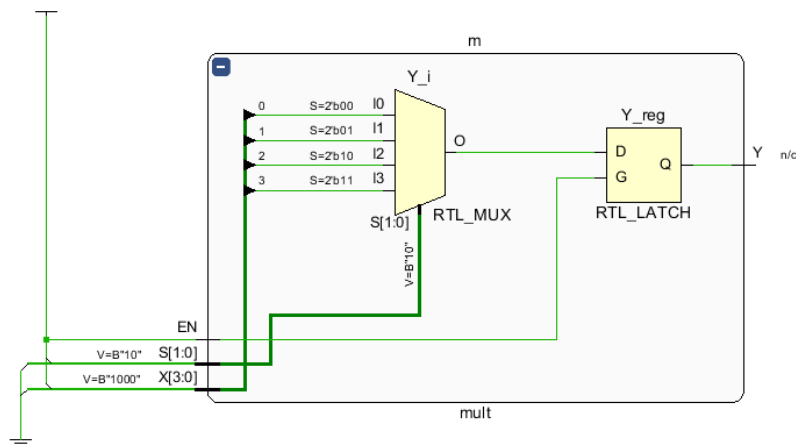


Рисунок 8 – Электрическая схема мультиплексора, реализованного 1 способом

ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы было осуществлено проектирование мультиплексора 4x1 в среде разработки Xilinx Vivado, написан код модуля на языке Verilog, выполнено тестирование (написан testbench), получена электрическая схема модуля и проведено моделирование, доказывающее работоспособность устройства.

СПИСОК ИСТОЧНИКОВ

1. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. СПб.: БХВ-Петербург, 2002. – 608 с.
2. Максфилд К. Проектирование ПЛИС. Курс молодого бойца. М.: Издательский дом «Додэка-XXI», 2007. – 408 с. (перевод с английского).
3. Тарасов И. Маршрут проектирования ПЛИС Xilinx в САПР Vivado // Компоненты и технологии. – 2012. – № 12.

ПРИЛОЖЕНИЕ А

Исходный код модуля мультиплексора и модуля тестирования

```
`timescale 1ns / 1ps

/////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 19.09.2022 14:48:54
// Design Name:
// Module Name: m
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module mult(
    input [3:0] X,
    input [1:0] S,
    input EN,
    output Y
```

```
);
```

```
reg Y;
```

```
always @*
```

```
if(EN)
```

```
begin
```

```
    Y = 0;
```

```
    Y = X[S];
```

```
end
```

```
endmodule
```

```
module testbench;
```

```
    reg [1:0] S;
```

```
    reg EN;
```

```
    reg [3:0] X;
```

```
    mult m(X,S,EN,Y);
```

```
    initial
```

```
    begin
```

```
        EN = 1;
```

```
        S = 0;
```

```
        X = 1;
```

```
        #50;
```

```
        S = 1;
```

```
        X = 2;
```

```
        #50;
```

```
S = 2;  
X = 4;  
#50;  
S = 2;  
X = 8;  
#50;  
end  
endmodule
```

ПРИЛОЖЕНИЕ Б

Исходный код модуля мультиплексора и модуля тестирования

```
`timescale 1ns / 1ps

/////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 19.09.2022 15:44:29
// Design Name:
// Module Name: LR_12
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module mult(
    input [3:0] X,
    input [1:0] S,
```

```

input EN,
output Y
);

reg Y;

always @*
if(EN)
begin
    Y = 0;
    case(S)
        2'b00 : Y = X[0];
        2'b01 : Y = X[1];
        2'b10 : Y = X[2];
        2'b11 : Y = X[3];
    endcase
end
endmodule

```

```

module testbench;
    reg [1:0] S;
    reg EN;
    reg [3:0] X;
    //wire Y;

    mult m(X,S,EN,Y);

    initial

```

```
begin
    EN = 1;
    S = 0;
    X = 1;
    #50;

    S = 1;
    X = 2;
    #50;

    S = 2;
    X = 4;
    #50;

    S = 2;
    X = 8;
    // EN = 1;
    #50;
    //$stop;
end
endmodule
```