

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА 44

КУРСОВОЙ ПРОЕКТ
ЗАЩИЩЕНА С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

старший преподаватель
должность, уч. степень, звание

подпись, дата

А.Н. Долидзе
инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ

по дисциплине: СХЕМОТЕХНИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4143

подпись, дата

Д.В. Пономарев
инициалы, фамилия

Санкт-Петербург 2024

Проектное задание

Разработать электронное устройство в соответствии с предложенной схемой и исходными данными, которое обеспечило бы заданную точность и качество работы. Работоспособность устройства обязательно подтверждается моделированием с применением САПР Quartus.

Разработка устройства включает в себя следующие этапы:

- составление и обоснование выбора функциональной схемы;
- разработку проектируемой схемы в САПР Quartus с использованием стандартных блоков, предоставляемых САПР (И, НЕ, дешифратор, триггер и т.д.) и описание ее работы с использованием временных диаграмм;
- разработку проектируемой схемы в САПР Quartus с использованием языка описания аппаратуры SystemVerilog и описание ее работы с использованием временных диаграмм;
- анализ двух подходов к разработке проектируемой схемы;
- обоснование и выбор FPGA на базе которого будет синтезироваться разрабатываемая схема;
- моделированием с применением САПР Quartus.
- назначение входов и выходов проектируемой схемы на выбранном FPGA.

Согласно варианту (схема 1, вариант 14), устройство является преобразователем кодов и должно переводить код 5211 в 7421. Предложенная схема устройства представлена на рисунке 1.

Схема 1

Схема преобразователя кода

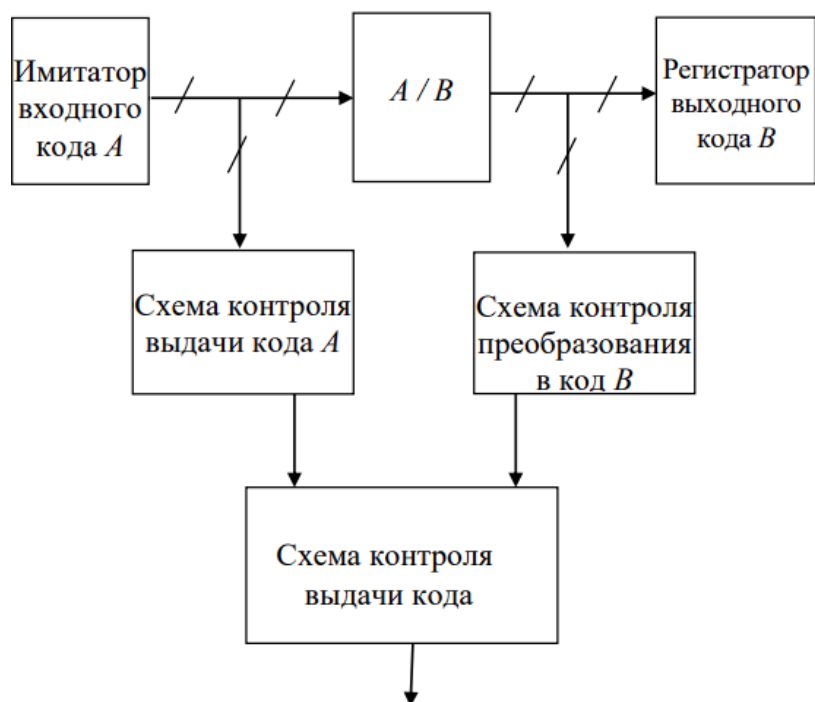


Рисунок 1 – Схема устройства

В таблице 1 представлено соответствие входных и выходных кодов, с которыми работает устройство.

Таблица 1

5211	7421
0000	0000
0001	0001
0011	0010
0101	0011
0111	0100
1000	0101
1010	0110
1101	1000
1110	1001
1111	1010

ОГЛАВЛЕНИЕ

Проектное задание.....	2
ОГЛАВЛЕНИЕ	4
ВВЕДЕНИЕ	5
1. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ	6
1.1. ВЫБОР ФУНКЦИОНАЛЬНОЙ СХЕМЫ	6
1.2. СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА СИСПОЛЬЗОВАНИЕМ СТАНДАРТНЫХ БЛОКОВ	7
1.3. СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА СИСПОЛЬЗОВАНИЕМ ЯЗЫКА ОПИСАНИЯ АППАРАТУРЫ SYSTEM VERILOG	10
2. ВЫБОР FPGA	13
ЗАКЛЮЧЕНИЕ.....	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

ВВЕДЕНИЕ

Целью курсового проектирования является освоение методов расчета, схемотехнического проектирования и конструирования элементов и блоков ЦВМ на базе программируемых логических интегральных схем (ПЛИС, далее FPGA).

В ходе выполнения курсового проекта должны быть получены схема устройства и программный код на языке описания аппаратуры System Verilog, соответствующие заданному в проектном задании функционалу. Устройство представляет собой преобразователь кодов, включающее в себя схемы контроля выдачи входного и выходного кодов.

1. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ

1.1. ВЫБОР ФУНКЦИОНАЛЬНОЙ СХЕМЫ

В ходе работы было решено использовать готовые модули среды Quartus для построения схемы. Изначально рассматривалось применение пары шифратор-дешифратор для реализации схемы. Однако в стандартных модулях Quartus был доступен только 4-битный дешифратор с 16 выходами. Поэтому для второй части схемы решено использовать логические выражения, связанные с выходами дешифратора.

Для начала надо понять каких сочетаний нету в нашей таблице и составить таблицу истинности. Схемы контроля выдачи входного и выходного кода были получены с помощью построения нормальной-конъюнктивной формы, при условии, что функция схем активируется в случае распознавания недопустимого кода. В таблице 2 представлены таблицы истинности схем контроля выдачи кодов, где f_x – функция контроля выдачи входного кода, f_y – функция контроля выдачи выходного кода.

Таблица 2

x_3	x_2	x_1	x_0	f_x	f_y
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	1	0

0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

Теперь после того как мы получили значения, надо составить выражения в которых будут отображены выражения для битов выхода. Выражения составляются по такому принципу, что все возможные значения из таблицы 2 нумеруются от 0 до 15, после чего смотрятся все значения соответствующего выходного бита и в месте где бит равен 1 ставиться q_n , где n это номер из таблицы 2.

Составленные выражения приведены ниже:

$$y_0 = q_1 \vee q_5 \vee q_8 \vee q_{14} \quad (1)$$

$$y_1 = q_3 \vee q_5 \vee q_{10} \vee q_{15} \quad (2)$$

$$y_2 = q_7 \vee q_8 \vee q_{10} \quad (3)$$

$$y_3 = q_{13} \vee q_{14} \vee q_{15} \quad (4)$$

1.2. СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА С ИСПОЛЬЗОВАНИЕМ СТАНДАРТНЫХ БЛОКОВ

В ходе разработки схемы устройства были использован стандартный модуль *16dmux*, а также стандартные модули, такие как *or* и *and*. На рисунках 2-5 представлен результат моделирования схемы в графическом редакторе среды

Quartus.

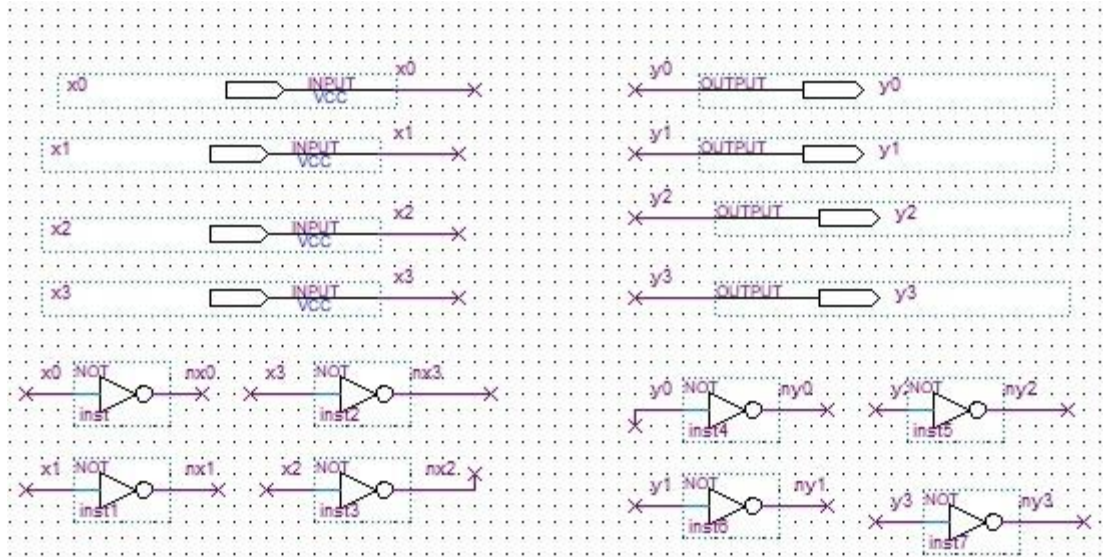


Рисунок 2 – Входы и выходы, а также блоки инверсии

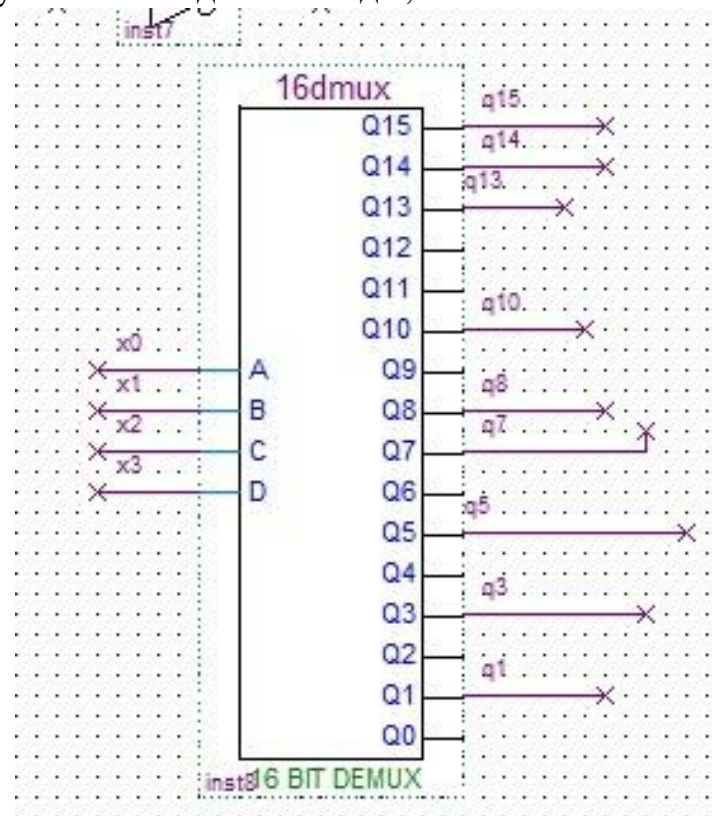


Рисунок 3 – Модуль с назначенными выходами q

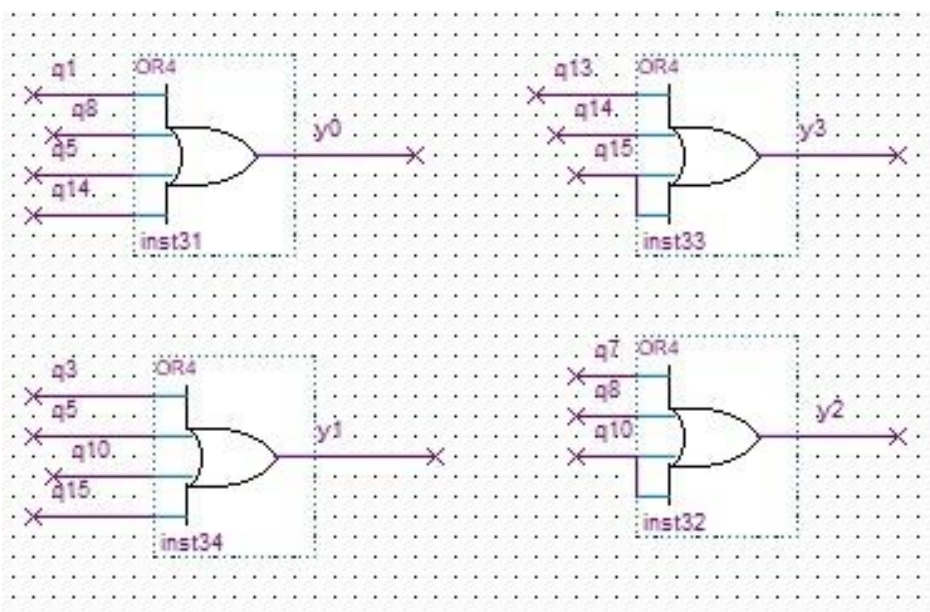


Рисунок 4— Полученные выражения на схеме

Далее будет часть схемы отвечающая за ошибки. На ней будут приведены все значения для входных(erA) и выходных(erB) кодов, которые отсутствуют в моем варианте, то есть все строки у которых в Fa и Fb проставлены 1 из таблицы 2.

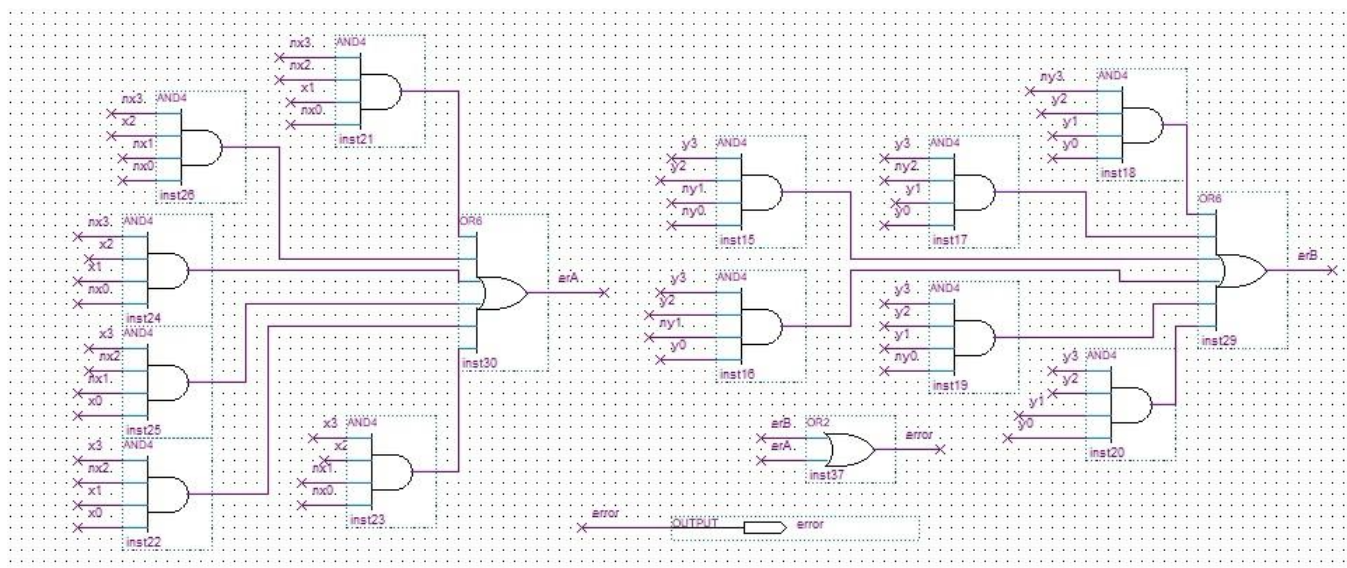


Рисунок 5 – Обработчик ошибок

На рисунке 6 представлен результат симулирования схемы в виде временной диаграммы.

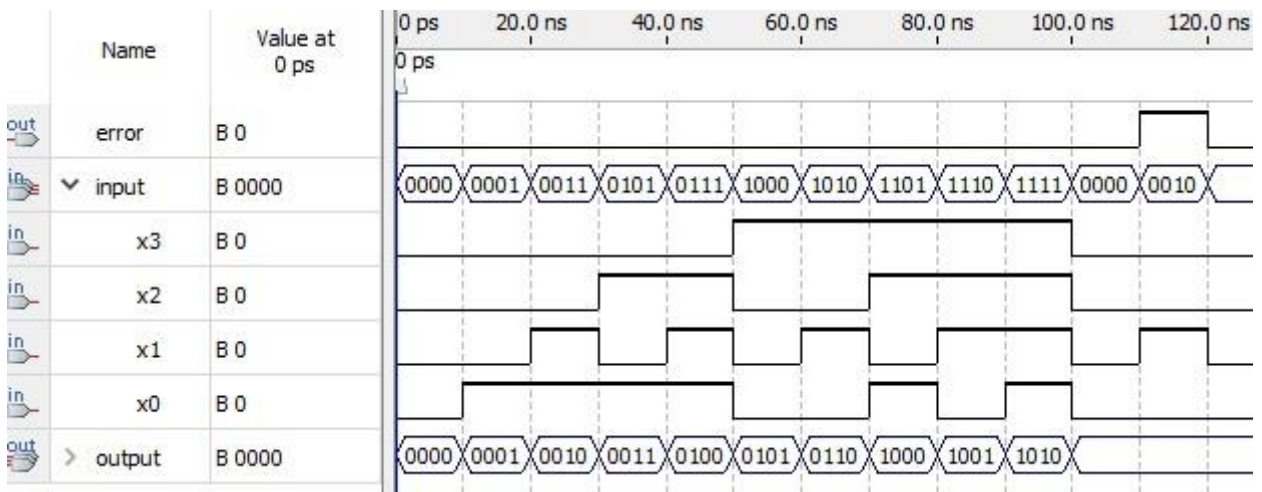


Рисунок 6 – Временная диаграмма

1.3. СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ОПИСАНИЯ АППАРАТУРЫ SYSTEM VERILOG

В ходе разработки программного кода для проекта было принято решение придерживаться модульной архитектуры проекта. Программа состоит из четырёх модулей: основной модуль проекта, модуль преобразователя кода, модуль контроля выдачи входного кода, модуль контроля выдачи выходного кода. Код программы представлен в листинге 1.

Листинг кода 1

```
module OutputReg (
input wire [3:0] outputCode,
output reg result
);
always @* begin
result <=
```

```

outputCode == 4'b0111 ||
outputCode == 4'b1011 ||
outputCode == 4'b1100 ||
outputCode == 4'b1101 ||
outputCode == 4'b1110 ||
outputCode == 4'b1111;
end
endmodule
module InputIm (
input wire [3:0] inputCode,
output reg result
);
always @* begin
result <=
inputCode == 4'b0010 ||
inputCode == 4'b0100 ||
inputCode == 4'b0110 ||
inputCode == 4'b1001 ||
inputCode == 4'b1100 ||
inputCode == 4'b1011;
end
endmodule
module KontrV (
input wire [3:0] inputCode,
output reg [3:0] outputCode
);
always @* begin
case (inputCode)
4'b0000: outputCode = 4'b0000;
4'b0001: outputCode = 4'b0001;
4'b0011: outputCode = 4'b0010;
4'b0101: outputCode = 4'b0011;
4'b0111: outputCode = 4'b0100;
4'b1000: outputCode = 4'b0101;
4'b1010: outputCode = 4'b0110;
4'b1101: outputCode = 4'b1000;
4'b1110: outputCode = 4'b1001;
4'b1111: outputCode = 4'b1010;
default: outputCode = 4'b0000;
endcase
end
endmodule
module lab7(

```

```

input wire [3:0] inputCode,
output wire [3:0] outputCode,

output wire error
);
wire error_x;
wire error_y;
KontrV m_KontrV(
.inputCode(inputCode),
.outputCode(outputCode)
);
InputIm m_inputIm(
.inputCode(inputCode),
.result(error_x)
);
OutputReg m_outputReg(
.outputCode(outputCode),
.result(error_y)
);
assign error = error_x || error_y;
endmodule

```

Описание кода:

Модуль OutputReg:

- Принимает на вход 4-битный входной код outputCode.
- Определяет result в зависимости от значений outputCode.
- Если outputCode равен одному из указанных значений, result устанавливается в 1, иначе в 0.

Модуль InputIm:

- Принимает на вход 4-битный входной код inputCode.
- Определяет result в зависимости от значений inputCode.
- Если inputCode равен одному из указанных значений, result устанавливается в 1, иначе в 0.

Модуль KontrV:

- Принимает на вход 4-битный входной код inputCode.
- Производит преобразование inputCode в соответствующий outputCode согласно заданному в таблице преобразования.
- Если входной код inputCode соответствует одному из заданных значений в таблице, устанавливает соответствующий outputCode. В противном случае outputCode устанавливается в 4-битный нулевой код.

Модуль lab7:

- Комбинирует работу модулей InputIm, OutputReg и KontrV.

- Принимает на вход 4-битный входной код inputCode.
- Выводит 4-битный выходной код outputCode.
- Устанавливает флаг ошибки error, если входные коды не соответствуют ожидаемым значениям.

На рисунке 7 представлен результат симуляции схемы, реализованной с помощью языка описания аппаратуры System Verilog.

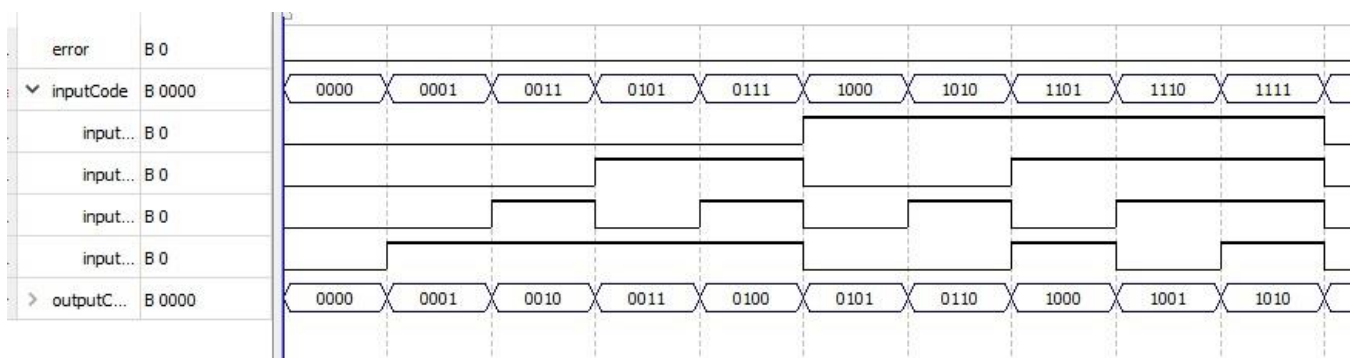


Рисунок 7 – Временная диаграмма

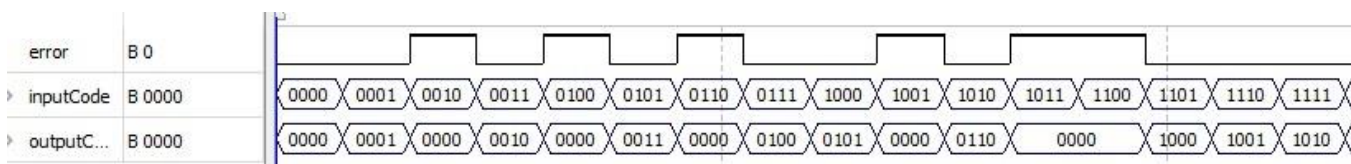


Рисунок 8– Временная диаграмма с с ошибками

2. ВЫБОР FPGA

В ходе выполнения проекта было принято решение использовать ПЛИС Cyclone IV E– EP4CE6E22C6 как было указано в задании. На рисунке 9-10 представлена схема подключения кПЛИС.

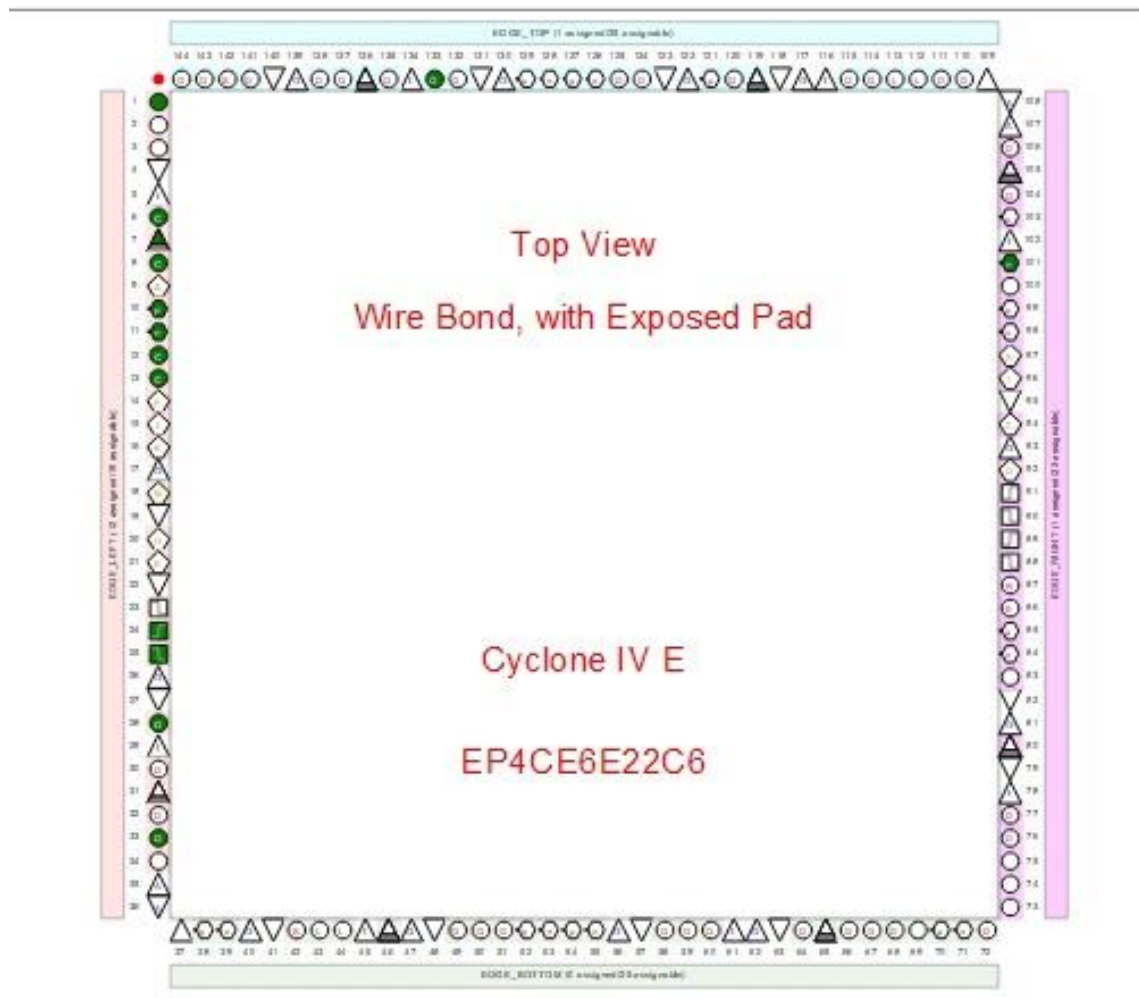


Рисунок 9 – Схема подключения выходов

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
out error	Output				PIN_11	2.5 V (default)
in inputCode[3]	Input				PIN_28	2.5 V (default)
in inputCode[2]	Input				PIN_33	2.5 V (default)
in inputCode[1]	Input				PIN_25	2.5 V (default)
in inputCode[0]	Input				PIN_24	2.5 V (default)
out outputCode[3]	Output				PIN_133	2.5 V (default)
out outputCode[2]	Output				PIN_7	2.5 V (default)
out outputCode[1]	Output				PIN_10	2.5 V (default)
out outputCode[0]	Output				PIN_1	2.5 V (default)

Рисунок 10 – Таблица назначения входов и выходов схемы

ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта было получено устройство преобразователя кодов. Моделирование устройства выполнено с помощью графической среды САПР Quartus и с помощью языка описания аппаратуры System Verilog. Обе реализации дают одинаковый и подобный друг другу результат.

Устройство работает конкретно и уверенно и соответствует поставленной задаче и заданному функционалу, исправно преобразовывает входной код в выходной, сообщает об ошибках, в случае некорректных кодов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Чернышев С. А. Синтез и компьютерный анализ элементов и узлов ЦВМ на FPGA с использованием САПР Quartus: метод. указания по курсовому проектированию / С.А. Чернышев. – СПб., 2018. – 31 с.