

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

подпись, дата

А.Н. Долидзе

инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ

по курсу: СХЕМОТЕХНИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2025

Проектное задание

Задание на курсовое проектирование выдается индивидуально каждому студенту. В задании указывается тема и исходные данные для проектирования.

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 3), устройство является преобразователем кодов и должно переводить код Грея в 8421. Функция: необходимо определить какому десятичному числу соответствует сформированное слово В, чётному или нечётному (смотрите таблицу кодов). Добавьте в схему счётчик, увеличивающий своё значение при чётном числе и уменьшающий при нечётном. Предложенная схема устройства представлена на рисунке 1.

Схема преобразователя кода

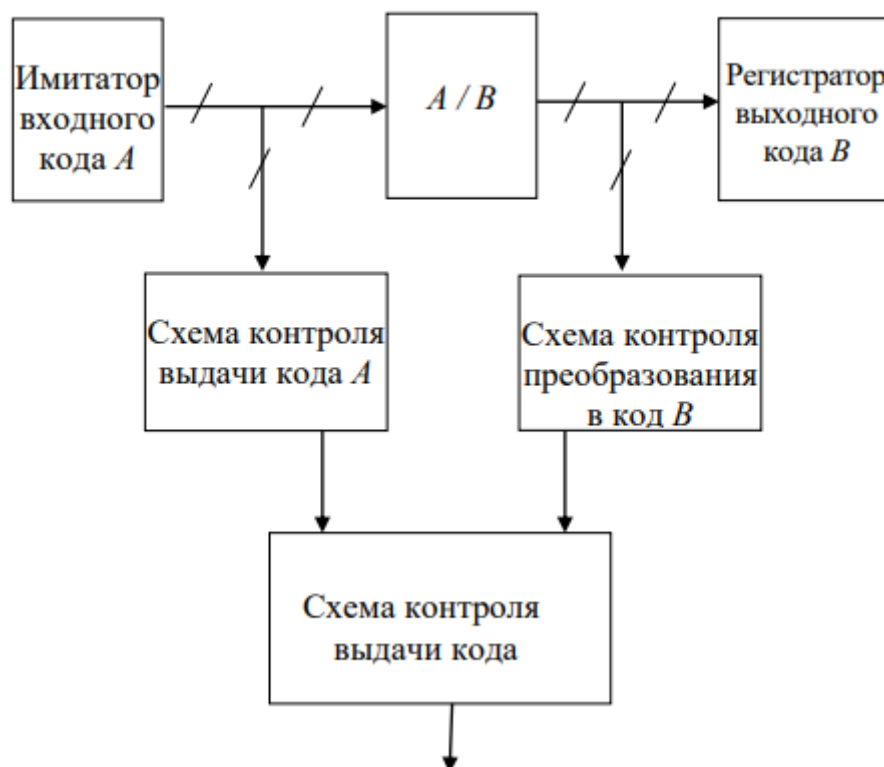


Рисунок 1 – Схема преобразователя кодов (Схема №1)

В таблице №1 представлено соответствие двоично-десятичное представление входных и выходных кодов, с которыми работает устройство.

Таблица №1

Входной код “Грея”	Выходной код “8421”
0000	0000
0001	0001
0011	0010
0010	0011
0110	0100
0111	0101
0101	0110
0100	0111
1100	1000
1101	1001

Также есть дополнительное условие: необходимо определить какому десятичному числу соответствует сформированное слово В, чётному или нечётному (смотрите таблицу

кодов). Добавьте в схему счётчик, увеличивающий своё значение при чётном числе и уменьшающий при нечётном.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	6
1. РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ.....	7
1.1 ВЫБОР ФУНКЦИОНАЛЬНОЙ СХЕМЫ	7
1.2 СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА С ИСПОЛЬЗОВАНИЕМ СТАНДАРТНЫХ БЛОКОВ САПР QUARTUS И ПОСТРОЕНИЕ ВРЕМЕННЫХ ДИАГРАММ В КОНТРОЛЬНЫХ ТОЧКАХ	9
1.3 СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ОПИСАНИЯ АППАРАТУРЫ SYSTEM VERILOG	11
2. ВЫБОР FPGA	14
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).

При выполнении проекта автор должен 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.

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

По ходу выполнения проекта в качестве реализации для разработки заданного функционала в виде функциональной схемы были выбраны стандартные модули из пакета Quartus, а именно компоненты шифратор и дешифратор с выходом в виде логических выражений дешифратора.

Составим таблицу, в которой каждая входная последовательность будет соответствовать определенному выходу блока 16dmux.

Таблица №2

A_3	A_2	A_1	A_0	№Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	3
0	0	1	1	2
0	1	0	0	7
0	1	0	1	6
0	1	1	0	4
0	1	1	1	5
1	0	0	0	15
1	0	0	1	14
1	0	1	0	12
1	0	1	1	13
1	1	0	0	8
1	1	0	1	9
1	1	1	0	11
1	1	1	1	10

Из таблицы видно, что начиная с Q_{10} , на вход начинают подаваться последовательности, которые отсутствуют по заданию. Соответственно их мы не будем выводить. При этом их мы будем использовать для контроля выдачи входного кода, аналогичным образом поступим и с выходным кодом.

Сопоставив входную и выходную последовательность, можно определить при каких выходных Q , на каждый бит выходной последовательности приходится единица. Исходя

из этого составим логическое выражение для каждого бита выходной последовательности:

$$Q_{14} \vee Q_{15} = B_3$$

$$Q_4 \vee Q_5 \vee Q_6 \vee Q_7 = B_2$$

$$Q_2 \vee Q_3 \vee Q_4 \vee Q_5 = B_1$$

$$Q_1 \vee Q_2 \vee Q_5 \vee Q_6 \vee Q_{14} = B_0$$

1.2 СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА С ИСПОЛЬЗОВАНИЕМ СТАНДАРТНЫХ БЛОКОВ САПР QUARTUS И ПОСТРОЕНИЕ ВРЕМЕННЫХ ДИАГРАММ В КОНТРОЛЬНЫХ ТОЧКАХ

В ходе разработки схемы устройства были использован стандартный модуль 16dmux, а также стандартные модули, такие как or и and. Для создания счётчика был добавлен тактирующий сигнал clk, который подаёт 1 при изменении смене подающего числа. Используется сумматор с разрешением счёта при подаче значений, которые присутствуют по заданию и проверке на ноль. Чётность определяется по нулевому разряду (соотв-но 1-нечётный, 0-чётный). На рисунках 2- 3 представлен результат моделирования схемы в графическом редакторе среды Quartus.

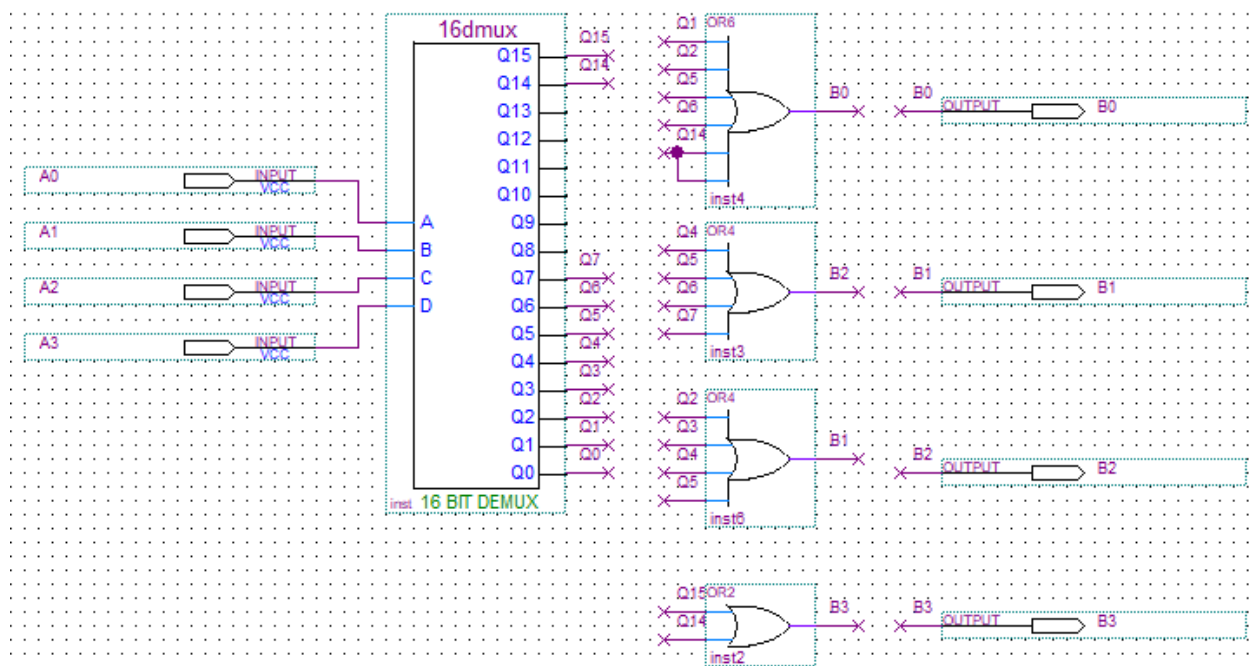


Рисунок 2 – Схема преобразователя кодов

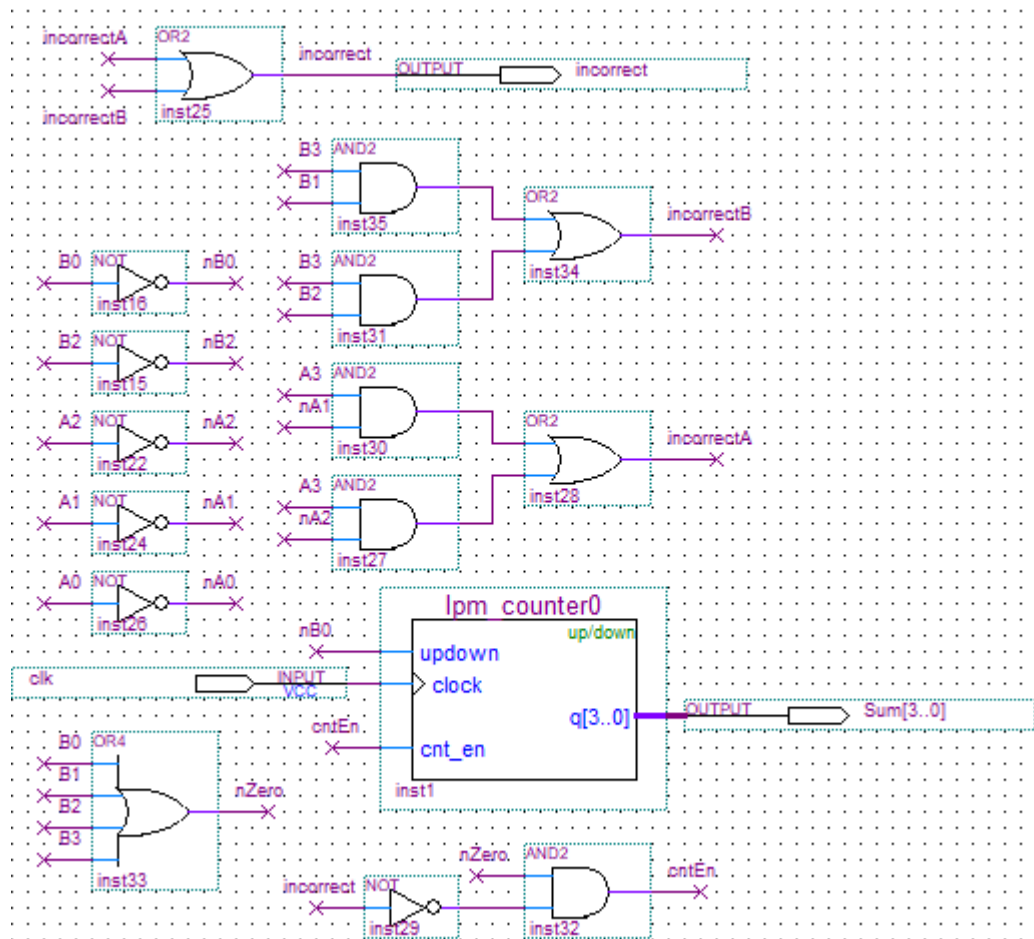
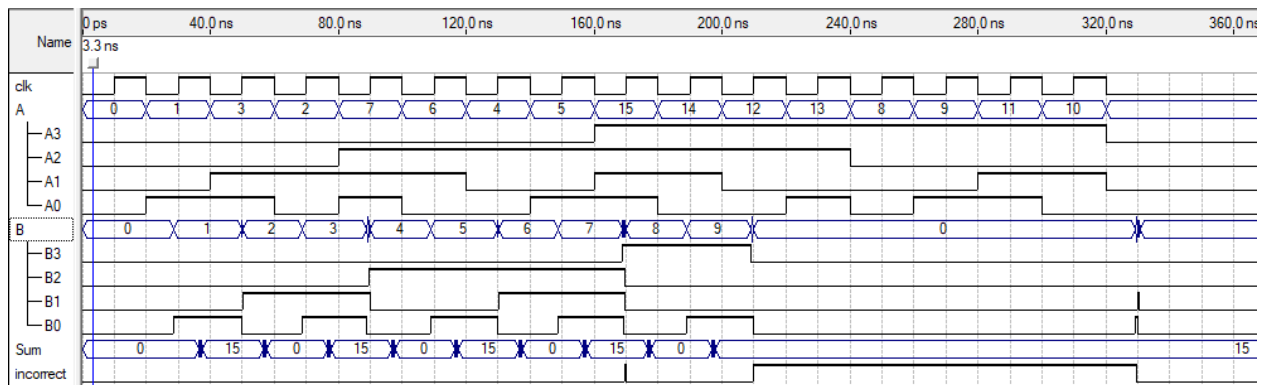


Рисунок 3 – Схема контроля выдачи кодов и счётчика

На рисунке 4 результат выполнения схемы в виде временной диаграммы.



1.3 СОСТАВЛЕНИЕ СХЕМЫ УСТРОЙСТВА С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ОПИСАНИЯ АППАРАТУРЫ SYSTEM VERILOG

Листинг программы:

```
module InputAnalyzer (  
    input wire [3:0] inputCode,  
    output reg result  
);  
    always @* begin  
        result <=  
            inputCode == 4'b1000 ||  
            inputCode == 4'b1001 ||  
            inputCode == 4'b1010 ||  
            inputCode == 4'b1011 ||  
            inputCode == 4'b1100 ||  
            inputCode == 4'b1101;  
    end  
endmodule  
  
module OutputAnalyzer (  
    input wire [3:0] outputCode,  
    output reg result  
);  
  
    always @* begin  
        result <=  
            outputCode == 4'b1010 ||  
            outputCode == 4'b1011 ||  
            outputCode == 4'b1100 ||  
            outputCode == 4'b1101 ||  
            outputCode == 4'b1110 ||  
            outputCode == 4'b1111;  
    end  
endmodule
```

```

module Logic (
    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'b0010: outputCode = 4'b0011;
            4'b0110: outputCode = 4'b0101;
            4'b0111: outputCode = 4'b0100;
            4'b0101: outputCode = 4'b0111;
            4'b0100: outputCode = 4'b0110;
            4'b1111: outputCode = 4'b1000;
            4'b1110: outputCode = 4'b1001;
            default: outputCode = 4'b0000;
        endcase
    end
endmodule

```

```

module KursCode (
    input clk,
    input wire [3:0] A,
    output wire [3:0] B,
    output wire incorrect,
    output wire [3:0] Sum
);
    wire incorrectA;
    wire incorrectB;
    Logic m_logic(
        .inputCode(A),
        .outputCode(B)
    );

```

```

);
InputAnalyzer m_inputAnalyzer(
    .inputCode(A),
    .result(incorrectA)
);

OutputAnalyzer m_outputAnalyzer(
    .outputCode(B),
    .result(incorrectB)
);

assign incorrect = incorrectA || incorrectB;
always @(posedge clk) begin
    if (incorrect==0) begin
        if (B[0]==1'b1) Sum=Sum-1;
        else Sum=Sum+1;
    end
end
endmodule

```

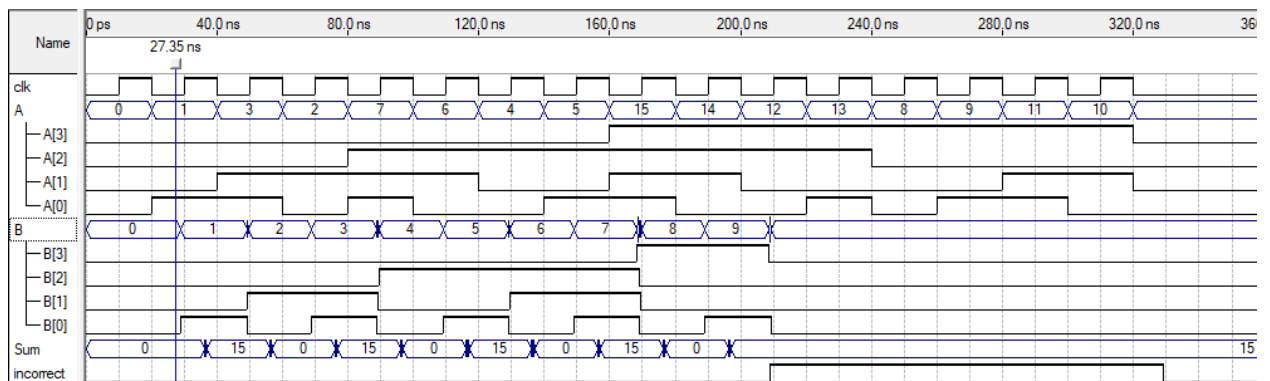


Рисунок 5 – Временная диаграмма языка описания аппаратуры

2. ВЫБОР FPGA

В ходе выполнения проекта было принято решение использовать ПЛИС Stratix II – EP2S15F484C3 ввиду того, что память в ПЛИС требует большое количество ресурсов и пинов. На рисунке 6 представлена схема подключения к ПЛИС. Схема подключения представлена на рисунке 6.

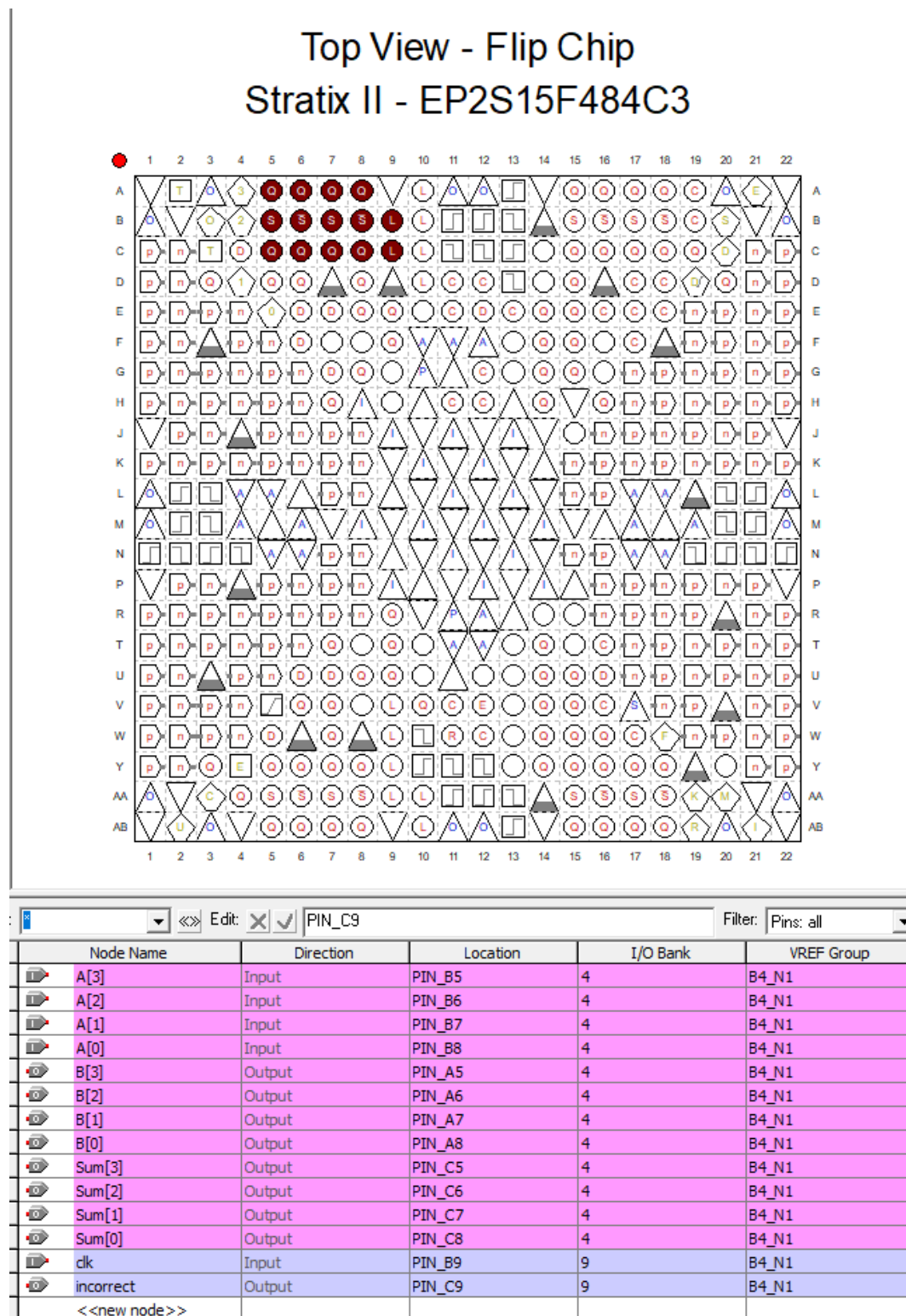


Рисунок 6 – Схема подключения

ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта было разработано электронное устройство в соответствии с предложенной схемой и исходными данными. Также была подтверждена работоспособность устройства при помощи моделирования с применения САПР Quartus. В ходе выполнения работы были выполнены следующие этапы: разработка проектируемой схемы в САПР Quartus с использованием стандартных блоков и описание ее работы с использованием временных диаграмм, разработка проектируемой схемы в САПР Quartus с использованием языка описания аппаратуры SystemVerilog и описание ее работы с использованием временных диаграмм. Обе реализации дали одинаковый результат. Устройство работает конкретно и уверенно и соответствует поставленной задачи и заданному функционалу, исправно преобразовывает входной код в выходной, сообщает об ошибках, в случае некорректных кодов, счётчик выполняет работы соответственно заданию.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ