

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., к.т.н., доц.

должность, уч. степень, звание

подпись, дата

Жаринов О.О.

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

Разработка модуля счетного устройства с заданным алгоритмом работы, с
использованием языков описания аппаратуры

по курсу: Схемотехника

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4241

подпись, дата

Булатов А.Ю.

инициалы, фамилия

Санкт-Петербург 2025

1. Цель работы.

Разработать проект модуля счетного устройства, работающего по заданному алгоритму, в среде программирования Quartus, используя языки описания аппаратуры.

2. Вариант 6.

Вариант задания наследуется с лабораторной работы №7 предыдущего семестра.

Задание заключается в разработке устройства формирования заданных последовательностей выходных кодов. По варианту основание счета $M = 21$ - сначала формируется последовательность из 21 тактов счётчика с шагом 1, затем – 7 подряд идущих нулей, на этом заканчивается один период формирования.

Требуемая последовательность выходных кодов представлена в таблице 1.

Таблица 1 – Вариант задания

№	0	1	2	...	M-2	M-1	M	M+1	M+2	M+3	M+4	M+5	M+6	M+7	M+8	M+9
6	0	1	2	...	M-2	M-1	0	0	0	0	0	0	0	1	2	...

3. Описание концепции разработки программного модуля

Для того, чтобы реализовать счет тактов, необходим непосредственно счетчик, который будет выполнять счет до значения $M-1$, то есть до 20, а после обнуляться и 7 тактов сохранять значение 0, после чего возобновлять счет.

Для достижения необходимого результата был использован модуль `lpm_counter`, который при каждом положительном фронте тактового сигнала увеличивает значение на единицу. При достижении значения 20 активируются логические элементы, формирующие управляющий сигнал.

Выходы счетчика анализируются с помощью логических элементов И и НЕ. Когда активны определённые биты ($q[4]$, $q[2]$, $q[1]$), на тактовый и сбросной вход D-триггера подаются сигналы, определяющие момент переключения и сброса. D-триггер управляет выбором значения на выходе мультиплексора `lpm_mux`.

Мультиплексор по сигналу *sel* выбирает между текущим значением счетчика и постоянной 20. Таким образом, до достижения значения 20 на выход поступает текущее значение счетчика, а при достижении — фиксированное значение 20. После этого счетчик сбрасывается и начинается новый цикл.

На рисунке 1 представлена схема устройства, которая была разработана в ходе выполнения лабораторной работы №7 первого семестра обучения дисциплине.

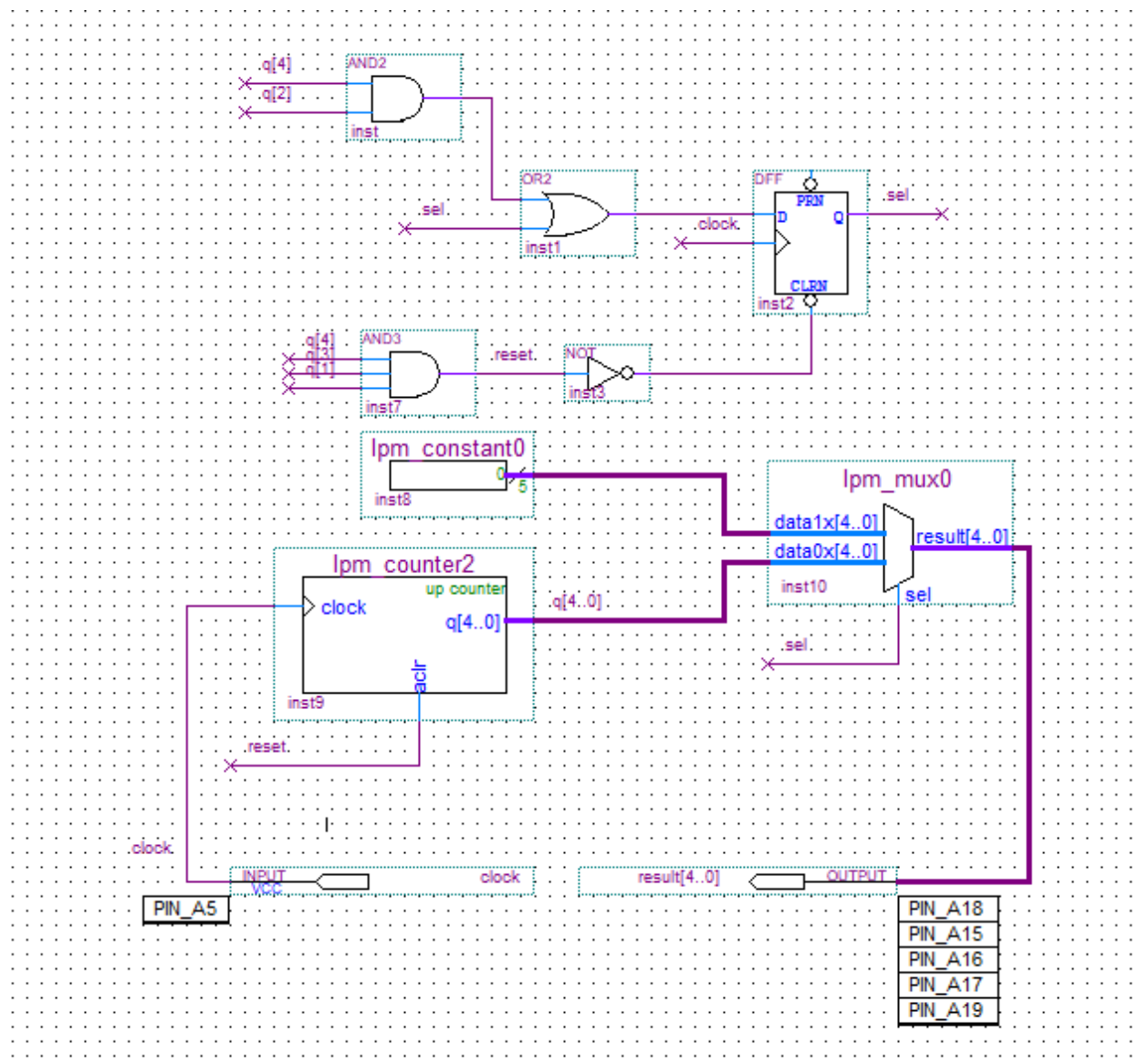


Рисунок 1 – Схема устройства в среде Quartus II

4. Листинги программ, которые были необходимы для выполнения работы.

Листинг 1 – Код D-триггера d_flipflop.sv

```
module d_flipflop (  
    input logic clk,  
    input logic reset,  
    input logic d,  
    output logic q  
);  
    // Асинхронный сброс, запись данных по фронту clk  
always_ff @(posedge clk or posedge reset) begin  
    if (reset)  
        q <= 1'b0;    //При сбросе устанавливаем в 0  
    else  
        q <= d;        //а иначе запоминаем значение триггера  
    end  
endmodule
```

Листинг 2 – Код счетчика lpm_counter.sv

```
module lpm_counter #(  
    parameter WIDTH = 5  
)(  
    input logic clk,  
    input logic reset,  
    output logic [WIDTH-1:0] q  
);  
    //Счетчик с инкрементом на каждом такте  
always_ff @(posedge clk or posedge reset) begin  
    if (reset)  
        q <= 0;        //Если reset == 1, то сбрасываем  
    else  
        q <= q + 1;    //А иначе инкрементируем значение  
    end  
endmodule
```

Листинг 3 – Код мультиплексора lpm_mux.sv

```
module lpm_mux #(
    parameter WIDTH = 5
) (
    input logic sel,
    input logic [WIDTH-1:0] data0,
    input logic [WIDTH-1:0] data1,
    output logic [WIDTH-1:0] result
);
    //логика мультиплексора
    always_comb begin
        case (sel)
            1'b0: result = data0; //0 - выбираем data0
            1'b1: result = data1; //1 - выбираем data1
        endcase
    end
endmodule
```

Листинг 4 – Основной код lab6.sv

```
module lab6 (
    input logic clk,
    input logic reset,
    output logic [4:0] result
);
    logic [4:0] counter_out;
    logic [2:0] pause_counter;
    logic in_pause, next_in_pause;

    //Основной счетчик до 20
    lpm_counter #(.WIDTH(5)) counter_inst (
        .clk(clk),
        .reset(reset | in_pause), //сбрасываем, если пауза
        .q(counter_out)
    );
    next_in_pause = pause_counter == 3;
```

```

//Регистр паузы (от 0 до 5)
always_ff @(posedge clk or posedge reset) begin
    if (reset) begin
        pause_counter <= 3'd0; //сброс счетчика паузы
    end else if (in_pause) begin
        pause_counter <= pause_counter + 3'd1; //инкремент в
состоянии паузы
    end else begin
        pause_counter <= 3'd0; //Обнуление счетчика, если в не
паузе
    end
end

//Логика перехода in_pause
always_comb begin
    if (reset)
        next_in_pause = 1'b0;
    else if (!in_pause && counter_out == 5'd20)
        next_in_pause = 1'b1; // Вход в паузу при достижении
20
    else if (in_pause && pause_counter == 3'd5)
        next_in_pause = 1'b0;
    else
        next_in_pause = in_pause;
end

//Хранение in_pause через D-триггер
d_flipflop pause_ff (
    .clk(clk),
    .reset(reset),
    .d(next_in_pause),
    .q(in_pause)
);

//Мультиплексор
lpm_mux #(.WIDTH(5)) result_mux (
    .sel(in_pause),
    .data0(counter_out),

```

```

        .data1(5'd0),
        .result(result)
    );
endmodule

```

Листинг 5 – lab6_tb.sv

```

`timescale 1ns/100ps
module lab6_tb;
    logic clk;
    logic reset;
    logic [4:0] result;

    //Основной модуль
    lab6 uut (
        .clk(clk),
        .reset(reset),
        .result(result)
    );

    always #50 clk = ~clk; //Один такт=100 нс

    initial begin
        $dumpfile("lab6.vcd"); //VCD-файл для просмотра в
GTKWave
        $dumpvars(0, lab6_tb);

        clk = 0;
        reset = 1;
        #20; // 20 нс держим сброс
        reset = 0;

        #6000; //Работа 6000 нс
        $finish;
    end
endmodule

```

5. Рисунок, на котором представлена информация о назначенных выводах ПЛИС для проекта.

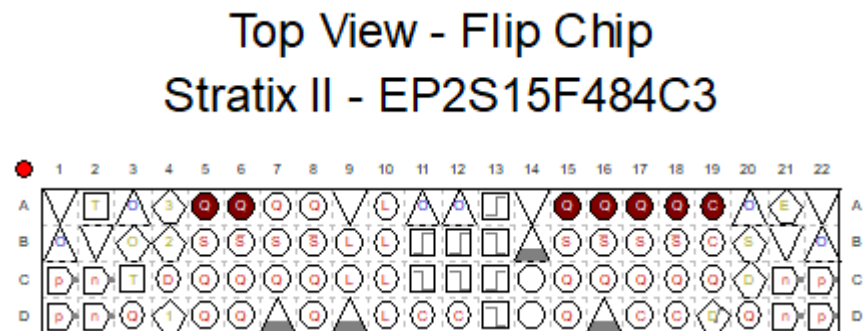


Рисунок 2 – Назначенные выводы ПЛИС

clk	Input	PIN_A5	4	B4_N1	3.3-V LVTTL (default)	
reset	Input	PIN_A6	4	B4_N1	3.3-V LVTTL (default)	
result[4]	Output	PIN_A19	3	B3_N1	3.3-V LVTTL (default)	
result[3]	Output	PIN_A18	3	B3_N0	3.3-V LVTTL (default)	
result[2]	Output	PIN_A17	3	B3_N0	3.3-V LVTTL (default)	
result[1]	Output	PIN_A16	3	B3_N0	3.3-V LVTTL (default)	
result[0]	Output	PIN_A15	3	B3_N0	3.3-V LVTTL (default)	

Рисунок 3 – Назначенные выводы ПЛИС

6. Схема RTL Viewer

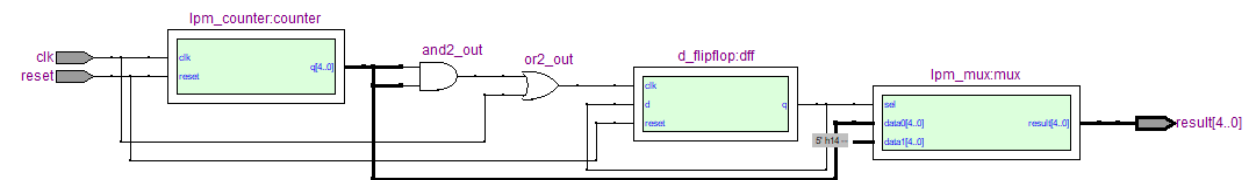


Рисунок 4 - Эквивалентная схема разработанного проекта, полученная с использованием RTL Viewer

7. Скриншот окна с результатами компиляции проекта

На рисунке 5 представлены результаты компиляции проекта

Flow Status	Successful - Tue Apr 08 15:54:44 2025
Quartus II Version	9.1 Build 222 10/21/2009 SJ Web Edition
Revision Name	lab6
Top-level Entity Name	lab6
Family	Stratix II
Device	EP2S15F484C3
Timing Models	Final
Met timing requirements	Yes
Logic utilization	< 1 %
Combinational ALUTs	5 / 12,480 (< 1 %)
Dedicated logic registers	5 / 12,480 (< 1 %)
Total registers	5
Total pins	7 / 343 (2 %)
Total virtual pins	0
Total block memory bits	0 / 419,328 (0 %)
DSP block 9-bit elements	0 / 96 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)

Рисунок 5 – Результаты компиляции

8. Рисунки временных диаграмм работы схемы в среде Quartus:
функциональная (functional simulation), временная (timing simulation) и
моделирование в Icarus Verilog.

На рисунках 6 и 7 представлены функциональная и временная
симуляции работы схемы.

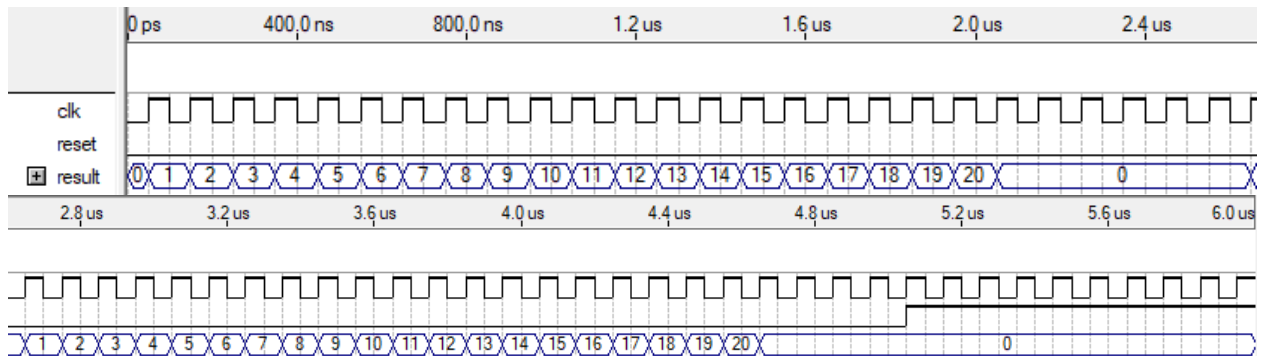


Рисунок 6 – Функциональная симуляция работы устройства

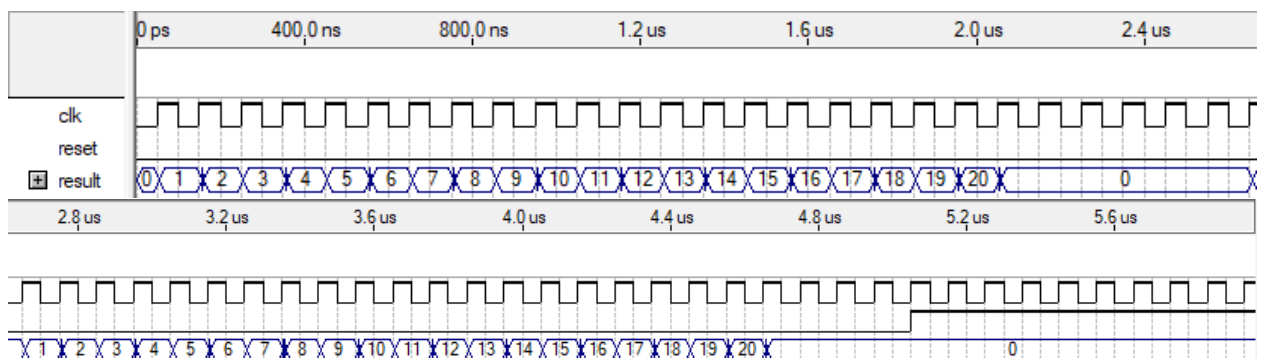


Рисунок 7 – Временная симуляция работы устройства

Далее приступим к моделированию при помощи Icarus Verilog,
результаты симуляции представлены на рисунке 8.

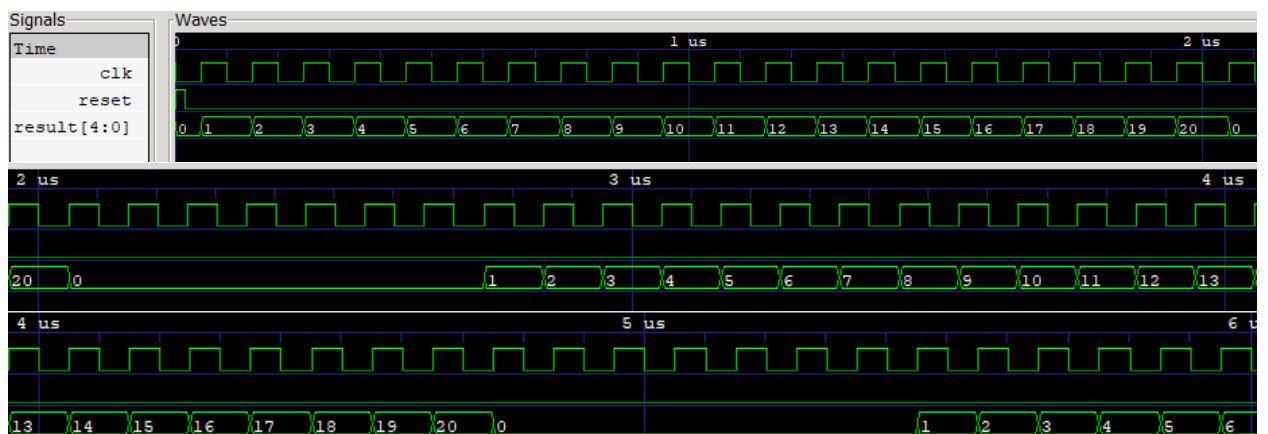


Рисунок 8 – Моделирование при помощи Icarus Verilog

9. Вывод.

В ходе выполнения лабораторной работы был разработан проект модуля счетного устройства, работающего по заданному алгоритму, в среде программирования Quartus, используя языки описания аппаратуры.

Оценка задержки.

Задержка в программном методе описания аппаратуры составила порядка 2.53 нс.

Задержка в графическом методе описания аппаратуры составила около 2.56 нс.

Основываясь на этих данных, могу сделать вывод о том, что задержки в обоих методах реализации устройства незначительны.

10. Список использованных источников.

1. Жаринов О.О. Учебно-методические материалы к выполнению лабораторной работы №7 по дисциплине “Схемотехника” (1-й семестр изучения дисциплины). ГУАП, 2024. – 3 с. (Интернет-ресурс) //URL: <https://pro.guap.ru/inside/student/tasks/4df74d6b7d0896ca3fd17787ac068686/download>
2. Жаринов О.О. Учебно-методические материалы к выполнению лабораторной работы №6 по дисциплине “Схемотехника” (2-й семестр изучения дисциплины). ГУАП, 2024. – 2 с. (Интернет-ресурс) //URL: <https://pro.guap.ru/inside/student/tasks/063e3f23174353340a351bd858f4dab9/download>
3. Булатов А.Ю. Отчет о лабораторной работе №7 «Разработка модуля счетного устройства с заданным алгоритмом работы, в среде Quartus» 1-й семестр //URL: <https://pro.guap.ru/inside/student/reports/4327133/download>
4. Интернет-ресурс установки Icarus Verilog
//URL: <https://bleyer.org/icarus/>