

ГУАП

КАФЕДРА № 44

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук, доц.  
\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

О.О. Жаринов  
\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

### РАЗРАБОТКА МОДУЛЯ СЧЁТНОГО УСТРОЙСТВА С ЗАДАНЫМ АЛГОРИТМОМ РАБОТЫ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКОВ ОПИСАНИЯ АППАРАТУРЫ

по курсу: СХЕМОТЕХНИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4143

\_\_\_\_\_  
подпись, дата

Д.В. Пономарев  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2024

## Цель работы

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

## Индивидуальное задание

Задание заключается в разработке устройства формирования заданных последовательностей выходных кодов. Здесь особенно важно подчеркнуть недопустимость даже кратковременных “врезок” двоичных кодов, нарушающих заданную последовательность. Наличие таких элементов на временной диаграмме допускается только по согласованию с преподавателем, при предъявлении результатов работы в часы занятий.

Моделирование проекта следует осуществить в среде ModelSim Altera, с использованием файла, описывающего тестовые воздействия (testbench).

Последовательность выходных кодов устройства для варианта №5 показана в таблице 1. Подметим, что значение М взято из предыдущей лабораторной работы №3. Согласно отчёту, **М = 29**.

Таблица 1

№ варианта	порядковый номер входного импульса счетного модуля																		
	0	1	2	...	M-2	M-1	M	M+1	M+2	...	2M-1	2M	2M+1	2M+2	...	3M-1	3M	3M+1	3M+2
1	0	1	2	...	M-2	M-1	M	M-1	M-2	...	1	0	1	2	...	M-1	M	M-1	M-2
2	0	1	2	...	M-2	M-1	M	M	M	...	M	0	1	2	...	M-1	M	M	M
3	0	1	2	...	M-2	M-1	M	M	M	...	M	M	M-1	M-2	...	1	0	1	2
4	0	1	2	...	M-2	M-1	0	0	0	...	0	0	1	2	...	M-1	0	0	0
5	0	1	2	...	M-2	M-1	M	M-1	M-2	...	1	0	0	0	...	0	0	1	2
6	0	1	2	...	M-2	M-1	0	0	0	...	0	0	0	0	...	0	0	1	2
7	0	0	0	...	0	0	0	1	2	...	M-1	M	M-1	M-2	...	1	0	0	0
8	0	1	2	...	M-2	M-1	M	M	M	...	M	0	0	0	...	0	0	1	2
9	0	0	0	...	0	0	0	1	2	...	M-1	M	0	0	...	0	0	0	0
10	0	1	2	...	M-2	M-1	0	1	2	...	M-1	0	0	0	...	0	0	1	2

## Ход работы

В качестве языка описания аппаратуры был выбран Verilog.

Начнём с настройки ModelSim, чтобы правильно построить временную диаграмму. Дальнейшая настройка проводилась согласно лекции №4 от 11 марта 2024 года. Во время создания проекта на этапе настройки EDA Tool Settings нужно указать ModelSim Altera и Verilog HDL. Это показано на

рисунке 1.

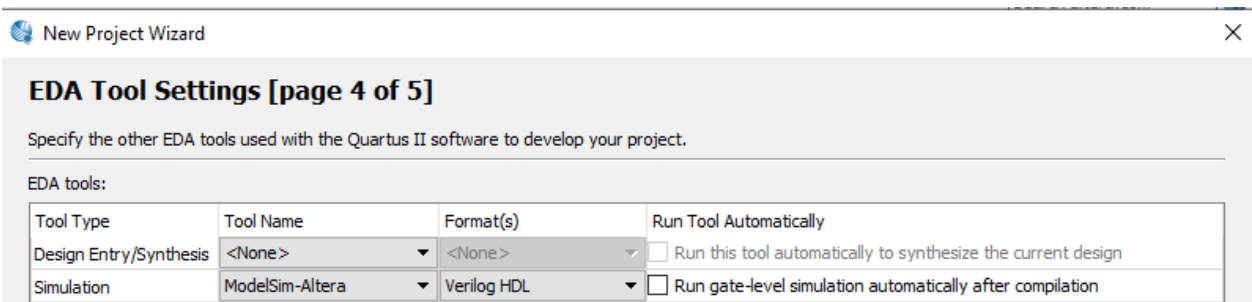


Рисунок 1 – Настройка EDA Tool Settings

Затем в настройках симуляции EDA Tool укажем имена созданного ранее тестбенча и добавим файл с расширением vt. Результат показан на рисунке 2.

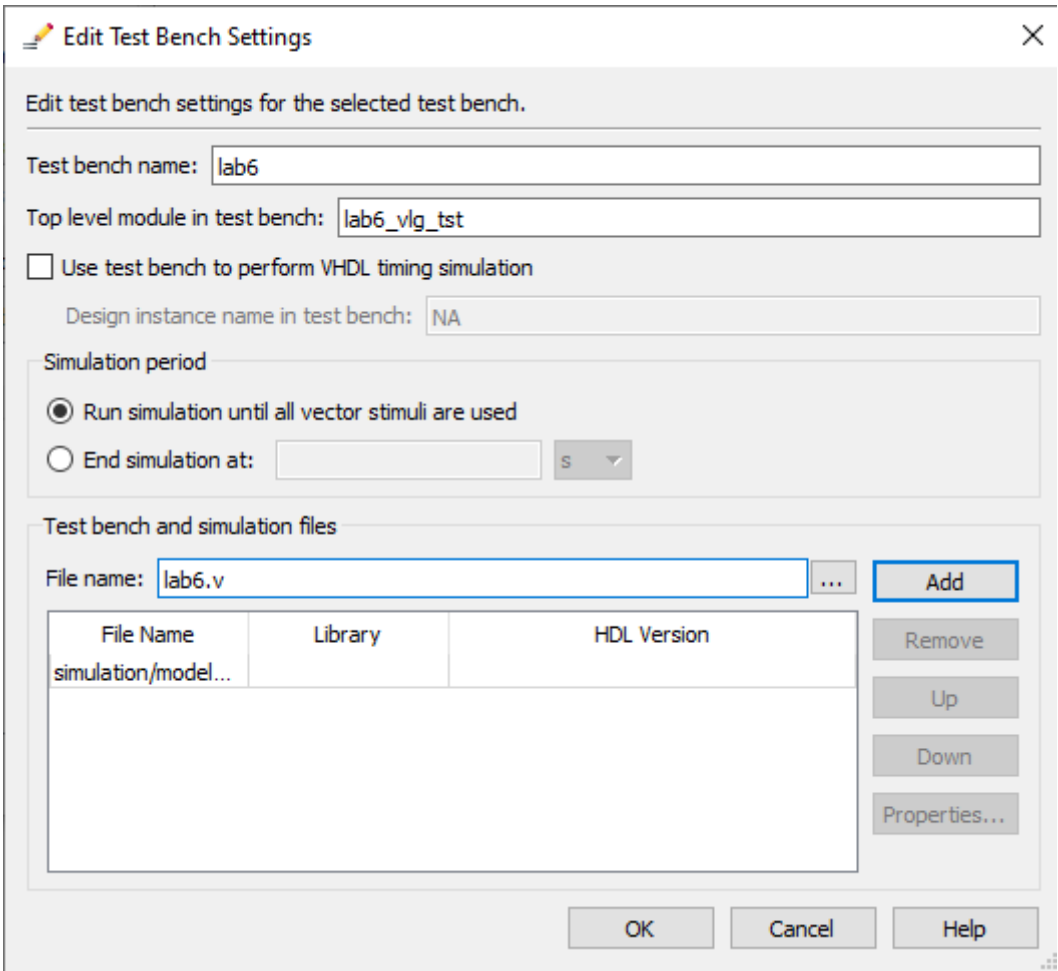


Рисунок 2 – Настройка симуляции

Опираясь на таблицу 1, можно сделать вывод о том, то один цикл счётчика состоит из нескольких этапов. Рассмотрим каждый из них:

Этап 0: счёт от 0 до 29 (такт 0 – 29).

Этап 1: счёт от 28 до 0 (такт 30 – 58).

Этап 2: счёт 0 (такт 59 – 86).

Заглядывая вперёд, данные этапы и будут будущими состояниями автомата.

Примерный вид последовательности в течение 1 итерации показан в таблице 2, где голубым выделены номера тактов, а под ними – соответствующие выходные значения счётчика.

Таблица 2

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
28	27	26	25	24	23	22	21	20	19
40	41	42	43	44	45	46	47	48	49
18	17	16	15	14	13	12	11	10	9
50	51	52	53	54	55	56	57	58	59
8	7	6	5	4	3	2	1	0	0
60	61	62	63	64	65	66	67	68	69
0	0	0	0	0	0	0	0	0	0
70	71	72	73	74	75	76	77	78	79
0	0	0	0	0	0	0	0	0	0
80	81	82	83	84	85	86	87	88	89
0	0	0	0	0	0	0	0	1	2

Идея заключается в использовании автомата с тремя состояниями. Рассмотрим автомат подробнее. Первое состояние имеет название S0. Оно является начальным, предназначен же для вывода последовательности от 0 до 29 включительно. Второе состояние имеет название S1. Оно является «обратным» по отношению к начальному в каком-то смысле, потому что оно предназначено для вывода обратного счёта от 28 до 0 включительно. Наконец, третье состояние имеет название S2 и предназначено для вывода нулей.

Перейдём к разработанной программе. На вход модуля будут подаваться обычные тактовые сигналы, а на выходе будет обычный регистр out размером 6 бит. Затем идёт инициализация состояний автомата. Итого получаем:

$$S0 = 00$$
$$S1 = 01$$
$$S2 = 10$$

После этого идёт объявление регистров. Всего их будет два: регистр s размером 2 бита, которому будет присваиваться значение состояния, а также счётчик counter размером 5 битов. С помощью инициализации зададим начальное состояние автомата, а именно: автомат пребывает в состоянии S0 и счётчик обнулён.

Затем идёт обширный блок always @, который реагирует на каждый положительный фронт тактового сигнала. Внутри него, с помощью конструкции case, расписано поведение каждого из состояний. Логика в целом соответствует расписанной ранее, но есть момент. В состоянии S0, когда счётчик дошёл до значения 30, идёт переход в блок else, где внутри него сначала идёт вывод значения 28, потом присваивание счётчику значения 27 и только потом переход в состояние S1. Это сделано для того, чтобы вывелась последовательность вида

**...28 29 28...**

### **Листинг программы**

```
module lab6 (  
    input clk,  
    output reg [5:0] out  
);  
  
parameter S0 = 2'b00;  
parameter S1 = 2'b01;
```

```
parameter S2 = 2'b10;
```

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

```
reg [4:0] counter;
```

```
initial begin
```

```
    s = S0;
```

```
    counter = 0;
```

```
end
```

```
always @(posedge clk) begin
```

```
    case(s)
```

```
        S0: begin
```

```
            if (counter < 30) begin
```

```
                out <= counter;
```

```
                counter = counter + 1;
```

```
            end
```

```
            else begin
```

```
                out <= counter - 2;
```

```
                counter <= 27;
```

```
                s <= S1;
```

```
            end
```

```
        end
```

```
        S1: begin
```

```
            if (counter > 0) begin
```

```
                out <= counter;
```

```
                counter = counter - 1;
```

```
            end
```

```
            else begin
```

```
                out <= 0;
```

```
                counter <= 0;
```

```
                s <= S2;
```

```
            end
```

```
        end
```

```
        S2: begin
```

```
            if (counter < 28) begin
```

```
                out <= 0;
```

```

        counter = counter + 1;
    end
    else begin
        counter <= 0;
        s <= S0;
    end

end

endcase
end
endmodule

```

## ПЛИС

Результат назначения выводов ПЛИС показан на рисунке 3, а на рисунке 4 показана сама ПЛИС.








	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard
1	 clk	Input	PIN_A1	2		3.3-V LVTTL (default)
2	 out[5]	Output	PIN_A2	2		3.3-V LVTTL (default)
3	 out[4]	Output	PIN_A3	2		3.3-V LVTTL (default)
4	 out[3]	Output	PIN_A4	2		3.3-V LVTTL (default)
5	 out[2]	Output	PIN_A5	2		3.3-V LVTTL (default)
6	 out[1]	Output	PIN_A6	2		3.3-V LVTTL (default)
7	 out[0]	Output	PIN_A7	2		3.3-V LVTTL (default)
8	<new node>>					

Рисунок 3 – Назначение выводов ПЛИС

## Top View MAX II - EPM570ZM256I8

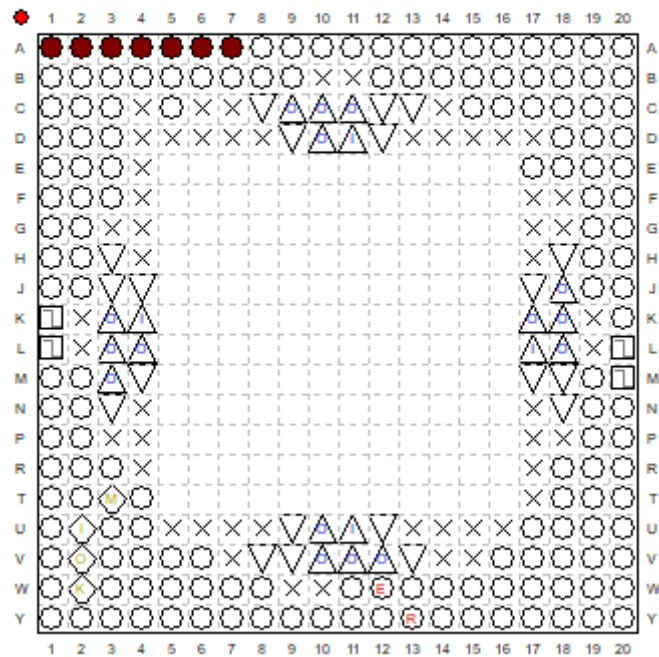


Рисунок 4 - ПЛИС

### Временная диаграмма

Временная диаграмма продемонстрирована на рисунках 5 – 9.

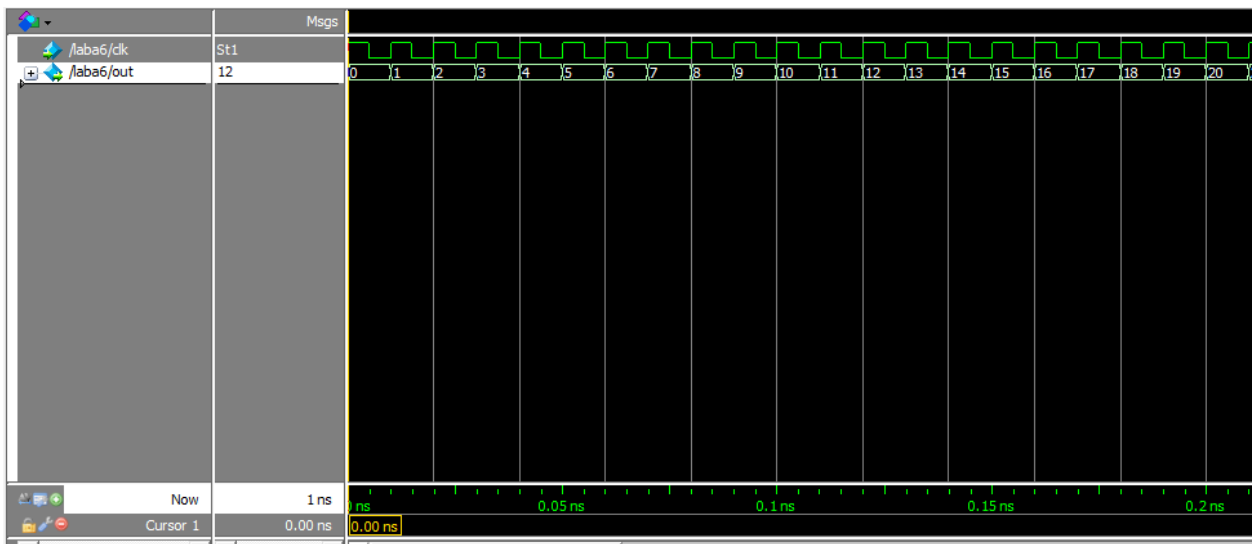


Рисунок 5 – Временная диаграмма (первая часть)



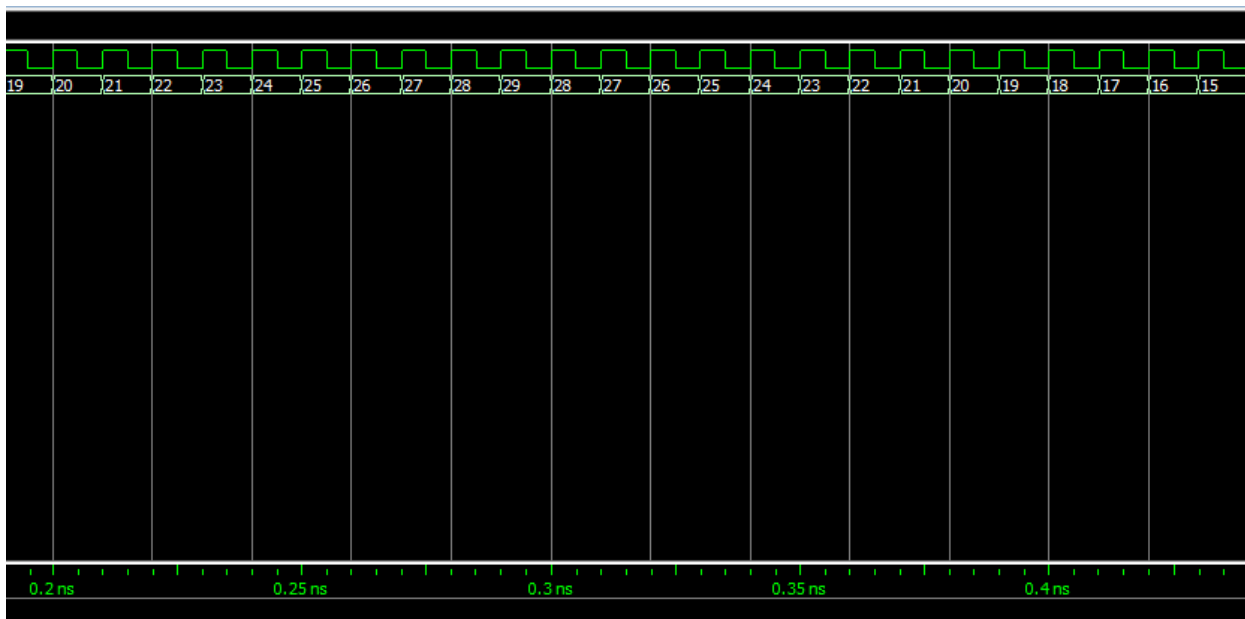


Рисунок 6 – Временная диаграмма (вторая часть)

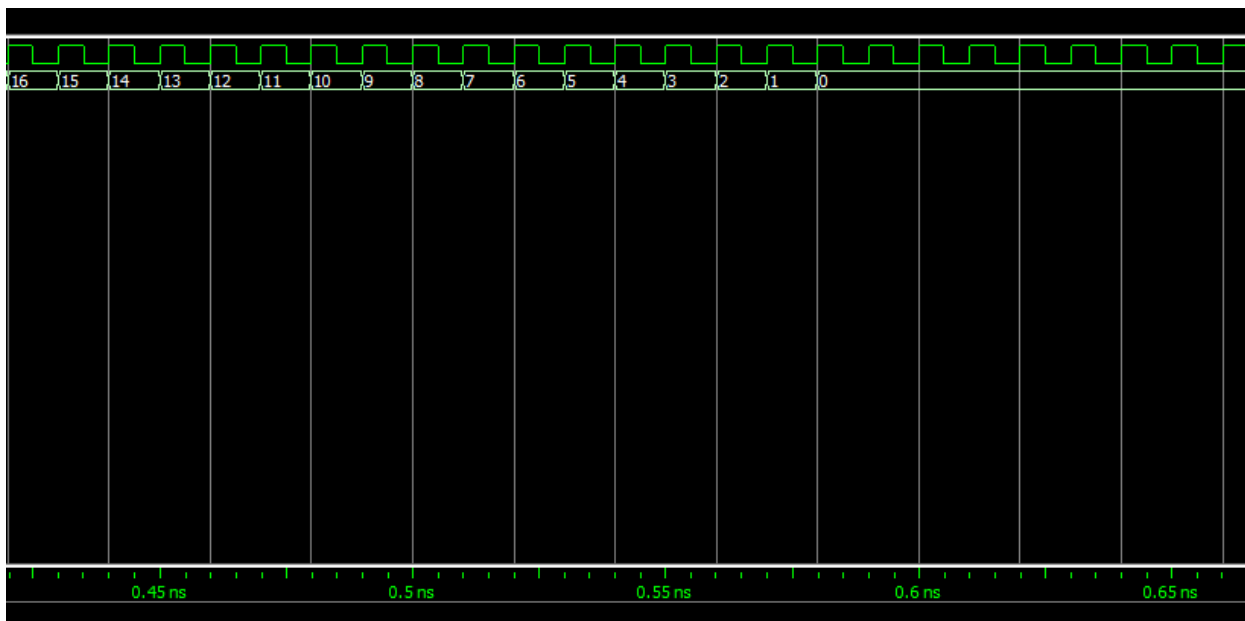


Рисунок 7 – Временная диаграмма (третья часть)

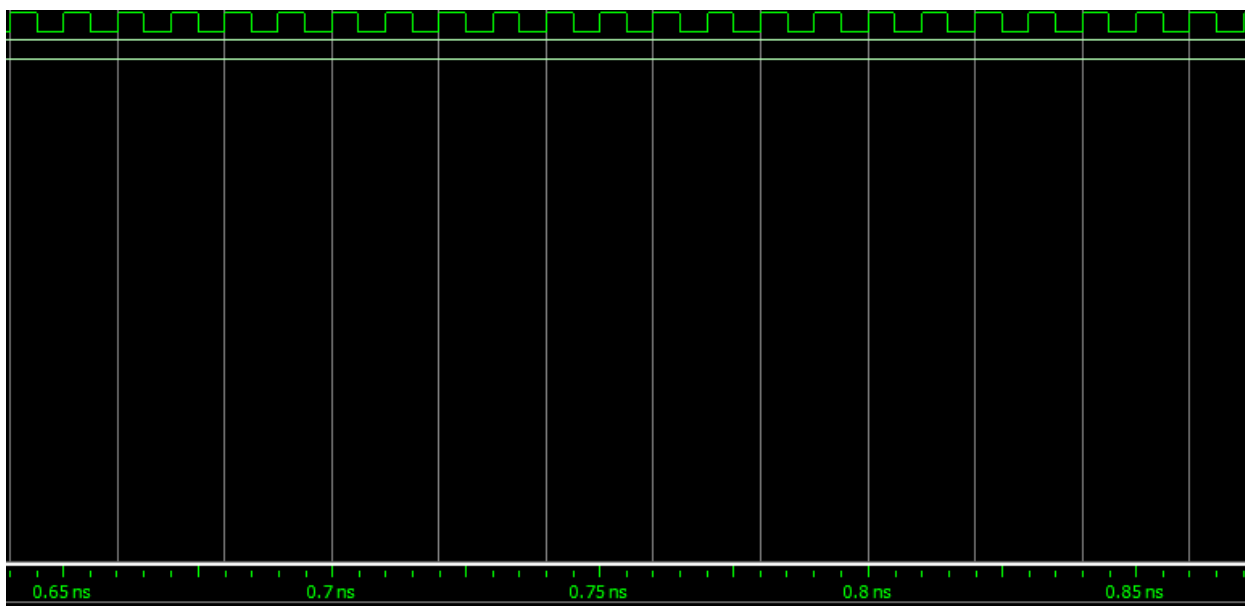


Рисунок 8 – Временная диаграмма (четвёртая часть)

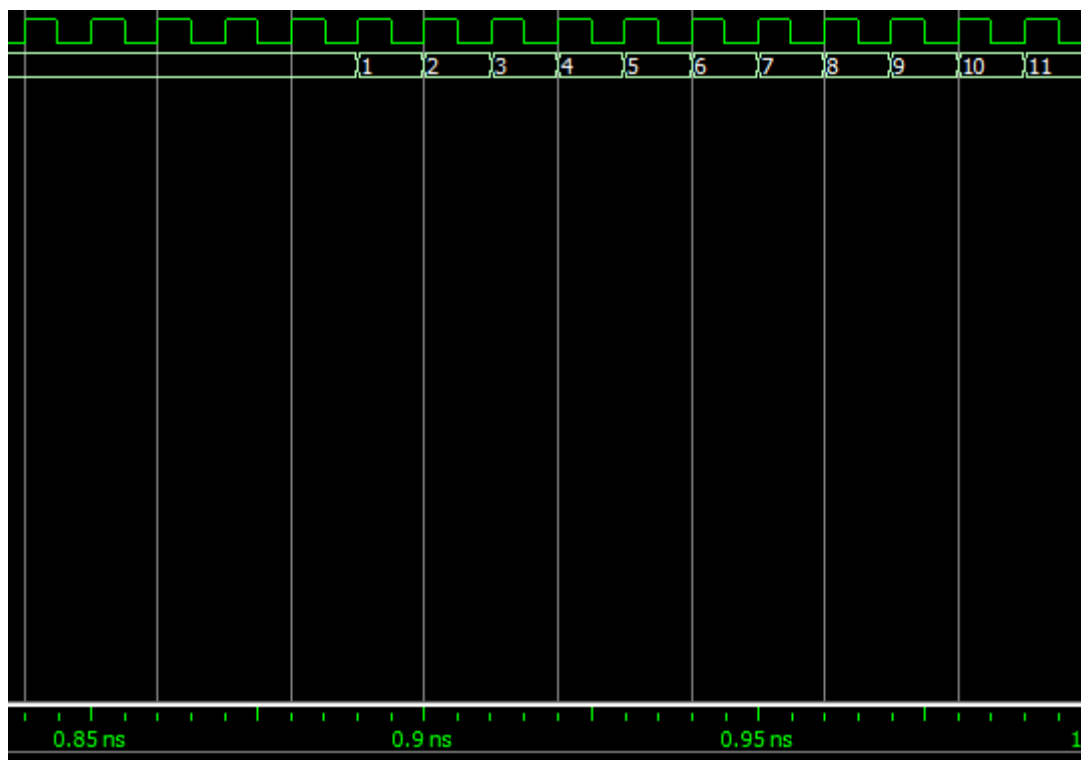


Рисунок 9 – Временная диаграмма (пятая часть)

## Выводы

В данной лабораторной работе был разработан проект модуля счётного устройства, работающего по заданному алгоритму в среде программирования Quartus, используя языки описания аппаратуры.

## Список используемых источников

1. Проектирование встраиваемых систем на ПЛИС. / З.Наваби; перев. с англ. В.В. Соловьева. – М.: ДМК Пресс, 2016. - 464 с.
2. Проектирование цифровых устройств на ПЛИС: учеб. пособие / И.В. Ушенина. - СПб: Лань, 2022. - 408 с.
3. Цифровая схемотехника и архитектура компьютера / Д.М. Харрис, С.Л. Харрис; пер. с англ. Imagination Technologies. – М.: ДМК Пресс, 2018. - 792 с.
4. Учебно-методические материалы к выполнению лабораторной работы №6 по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.:

<https://pro.guap.ru/inside/student/tasks/43730981ca7ca6713e1a6eadb8e83b51/download>. (Дата обращения: 05.04.24).

5. Лекция №4 от 25 марта 2024 года по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.:

<https://bbb2.guap.ru/playback/presentation/2.3/8df832b35e59b7b17bb499a557e56942cdbfd0b-1711367744513>. (Дата обращения: 05.04.24).

6. Отчёт о выполнении лабораторной работы №4 по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Пономарев Д.В: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.:

<https://pro.guap.ru/inside/student/reports/3920740/download>. (Дата обращения: 05.04.24).