

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

Е.Д. Тегай

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).

Требуемая последовательность выходных кодов устройства продемонстрирована в таблице 1, где для удобства восприятия номер варианта и другие необходимые данные выделены жёлтым цветом.

Таблица 1

Таблица 1 (начало) – Последовательность выходных кодов устройства																			
№ 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

Следует отметить, что значение M, упомянутое в таблице 1, взято из предыдущей лабораторной работы №3. Ссылаясь на отчёт, **M = 35**.

Ход работы

В качестве языка описания аппаратуры был выбран Verilog.

Перед началом выполнения самой работы необходимо корректно настроить среду разработки, чтобы получить временную диаграмму в среде

ModelSim. Для этого пришлось создавать новый проект, где на последнем этапе настройки нужно было указать, собственно, ModelSim. Это показано на рисунке 1. Следует также отметить, что дальнейшая настройка проводилась согласно алгоритму, рассмотренному в лекции №4 от 11 марта 2024 года.

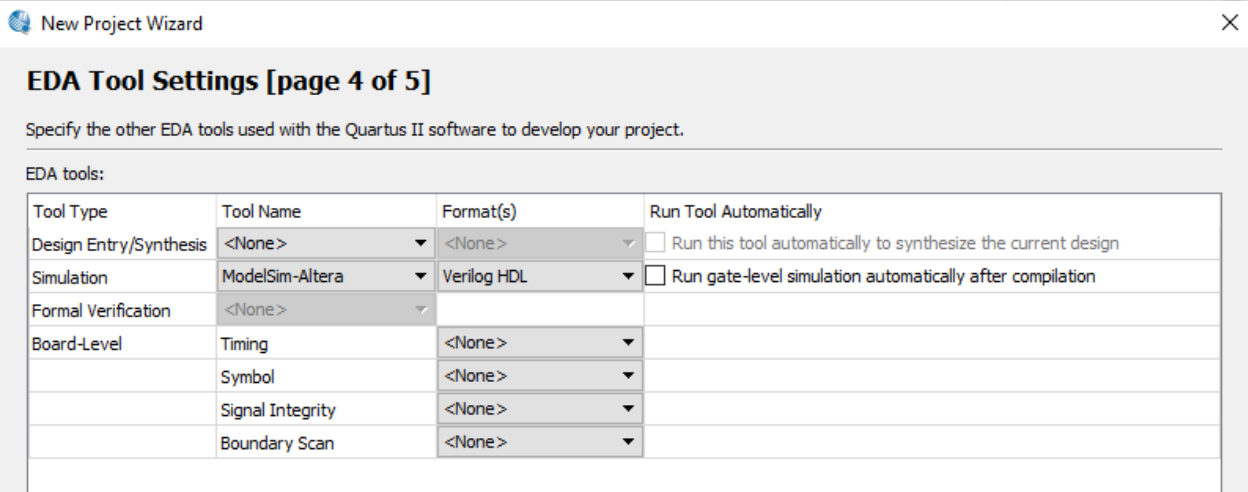


Рисунок 1 – Настройка проекта

Далее идёт настройка симуляции. Результат показан на рисунке 2.

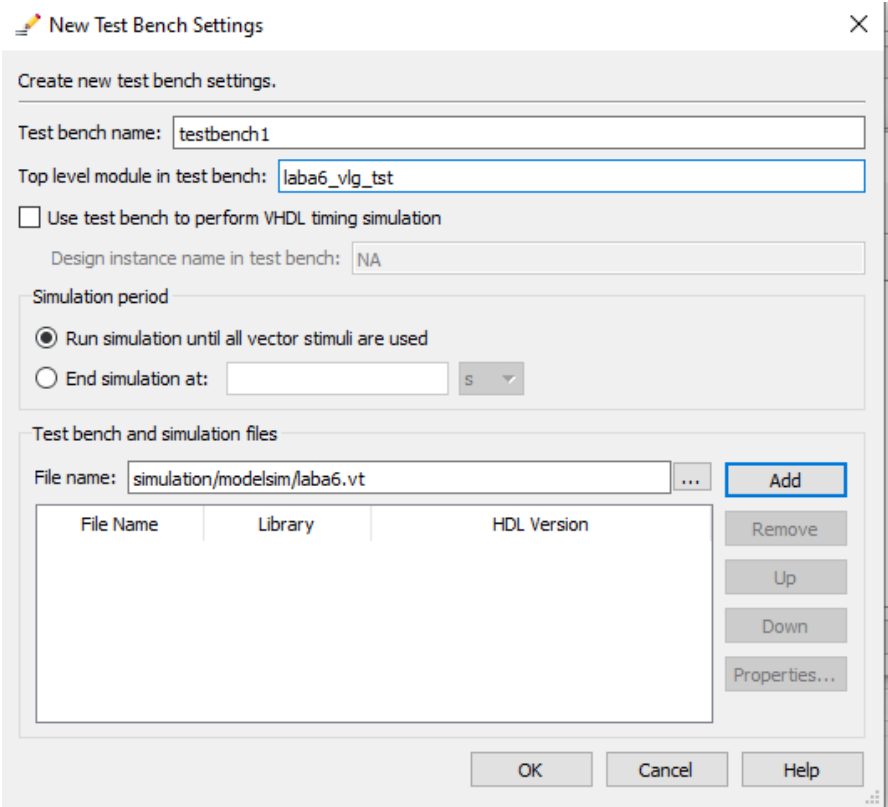


Рисунок 2 – Настройка симуляции

Настройка на этом завершается. Теперь приступим к решению поставленной задачи. Как видно из таблицы 1, один цикл счётчика можно разбить как-бы на три части: счёт от 0 до 34 с последующим сбросом в ноль, снова счёт от 0 до 34 с таким же сбросом в ноль, а затем последовательность нулей, длящаяся в течение тех же 35 тактов. Итоговый примерный вид продемонстрирован в таблице 2.

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
30	31	32	33	34	0	1	2	3	4
40	41	42	43	44	45	46	47	48	49
5	6	7	8	9	10	11	12	13	14
50	51	52	53	54	55	56	57	58	59
15	16	17	18	19	20	21	22	23	24
60	61	62	63	64	65	66	67	68	69
25	26	27	28	29	30	31	32	33	34
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	0	0
90	91	92	93	94	95	96	97	98	99
0	0	0	0	0	0	0	0	0	0
100	101	102	103	104	105	106	107	108	109
0	0	0	0	0	0	1	2	3	4

Следует отметить, что в таблице 2 жёлтым выделены номера тактов, красным – «ключевые» моменты, характеризующие выходную последовательность счётчика в рамках цикла, а в ячейках, не содержащих выделение, находятся выходные значения счётчика.

Теперь приступим к разработке программы. Хочется указать, что со времён выполнения лабораторной работы №4 для реализации поставленной цели первым делом думается на использование автомата. Эта лабораторная работа так же не стала исключением.

Рассмотрим автомат подробнее. В этот раз автомат будет состоять всего

лишь из двух состояний. Пусть первое состояние будет называться STATE_1, а его суть будет заключаться в выводе в течение 35 тактов нулей, а нулевое (начальное) состояние, пусть оно будет STATE_0, будет отвечать за вывод последовательности от 0 до 34 включительно.

Если с первым состоянием всё довольно просто, то нулевое будет содержать в себе довольно некомпактную логику. Необходимо следить, чтобы вывод от 0 до 34 проводился именно 2 раза после вывода 35 нулей. Тогда в программе было решено завести помимо основного счётчика, который будет выводить определённую последовательность в течение 35 тактов, ещё и циклический счётчик, с помощью которого и можно реализовать именно 2 вывода подряд последовательности 0 – 34.

Теперь рассмотрим более подробно разработанный код программы. Начнём со структуры разрабатываемого модуля. На вход будут подаваться обычные тактовые сигналы, сигнал Reset, а также асинхронный сброс. Включение этих входных сигналов, помимо тактовых, было решено на основе просмотренной лекции №5 от 25 марта 2024 года, чтобы привести к более правдоподобному виду. В качестве выходного сигнала идёт обычный шестибитный регистр out.

Далее необходимо как-то обозначить состояния автомата. Для этого используется конструкция parameter, где нулевому состоянию соответствует комбинация 00, а первому – 01.

Затем идёт объявление используемых регистровых переменных, таких как: двубитная переменная state, которой и будут присваиваться названия состояний в дальнейшем, основной шестибитный счётчик, который реализует вывод в течение 35 тактов, а также циклический шестибитный счётчик, с помощью которого идёт вывод в течение 2 итераций выходной последовательности 0 – 34.

Само поведение автомата расписано внутри блока always @, который

чувствителен к положительному фронту тактового сигнала, положительному фронту сигнала `aclr` или же положительному фронту сигнала сброса.

Рассмотрим более подробно внутреннюю структуру блока `always`. Первым делом идёт проверочное условие, а не обнаружен ли положительный сигнал на `R`. Если обнаружен, то идёт основной сброс автомата, соответственно состояние будет начальным, а счётчики – обнуляются. Если же, например, обнаружен положительный сигнал на `aclr`, то также сбрасывается основной счётчик.

Затем идёт описание состояний. Логика первого состояния проста: пока счётчик не дойдет до 35, идёт вывод нулей. На каждой итерации идёт инкрементация счётчика. Если же счётчик дошёл до 35, то идёт последующий сброс счётчика и дальнейший переход на начальное состояние.

Первое, что делается внутри нулевого состояния, это проверка циклического счётчика. Если счётчик не дошёл до значения 2, то идёт заход далее в проверочное условие основного счётчика. Опять таки, если счётчик < 35, то идёт вывод значения счётчика и последующая его инкрементация. А иначе, если основной счётчик достиг граничного значения, идёт его последующий сброс, тогда как затем идёт инкрементация циклического счётчика. Если же в самом начале циклический счётчик достиг 2, то идёт обнуление всех счётчиков с последующим переходом в состояние `STATE_1`.

Листинг программы

Код разработанной программы продемонстрирован ниже.

```
module laba6 (  
    input clk,  
    input R,  
    input aclr,  
    output reg [5:0] out  
);  
  
parameter STATE_0 = 2'b00;
```

```

parameter STATE_1 = 2'b01;
// Регистровые переменные, обозначающие состояние, осн. Счётчик и
циклический счётчик
reg [1:0] state;
reg [5:0] counter;
reg [5:0] counter_cycle;
// Начальные параметры
initial begin
    state = STATE_0;
    counter = 0;
    counter_cycle = 0;
end
// Автомат
always @(posedge clk or posedge aclr or posedge R) begin
    if (R) begin
        state <= STATE_0;
        counter <= 0;
    end
    else if (aclr) begin
        counter <= 6'd0;
    end
    else begin
        case(state)
            STATE_0: begin
                if (counter_cycle < 2) begin
                    if (counter < 35) begin
                        out <= counter;
                        counter = counter + 1;
                    end
                end
                else begin
                    counter <= 0;
                    counter_cycle = counter_cycle + 1;
                    state <= STATE_0;
                end
            end
        end
    end
end
else begin

```

```

        counter <= 0;
        counter_cycle <= 0;
        state <= STATE_1;
    end
end
STATE_1: begin
    if (counter < 35) begin
        out <= 0;
        counter = counter + 1;
    end
    else begin
        counter <= 0;
        state <= STATE_0;
    end
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	 R	Input	PIN_A1	2		3.3-V LVTTTL (default)
2	 adr	Input	PIN_A2	2		3.3-V LVTTTL (default)
3	 clk	Input	PIN_A3	2		3.3-V LVTTTL (default)
4	 out[5]	Output	PIN_A4	2		3.3-V LVTTTL (default)
5	 out[4]	Output	PIN_A5	2		3.3-V LVTTTL (default)
6	 out[3]	Output	PIN_A6	2		3.3-V LVTTTL (default)
7	 out[2]	Output	PIN_A7	2		3.3-V LVTTTL (default)
8	 out[1]	Output	PIN_A8	2		3.3-V LVTTTL (default)
9	 out[0]	Output	PIN_A9	2		3.3-V LVTTTL (default)

Рисунок 3 – Назначение выводов ПЛИС

Top View MAX II - EPM570ZM256I8

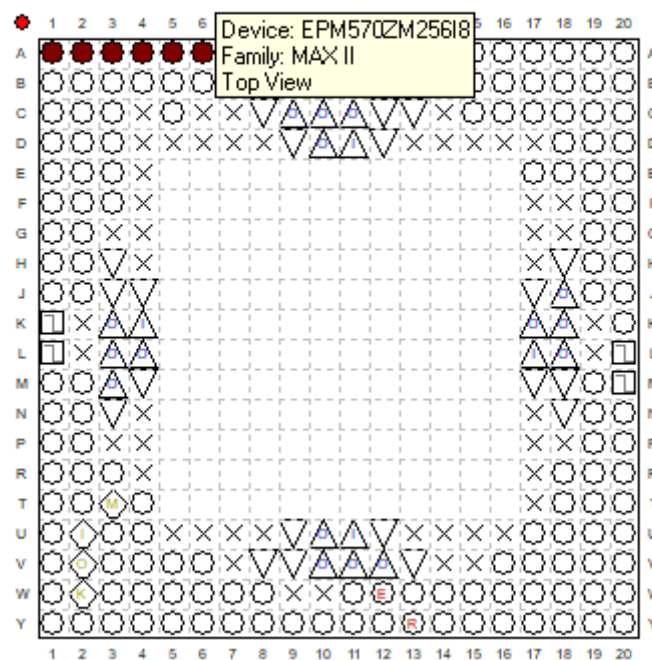


Рисунок 4 - ПЛИС

Временная диаграмма

Временная диаграмма продемонстрирована на рисунках 5 –

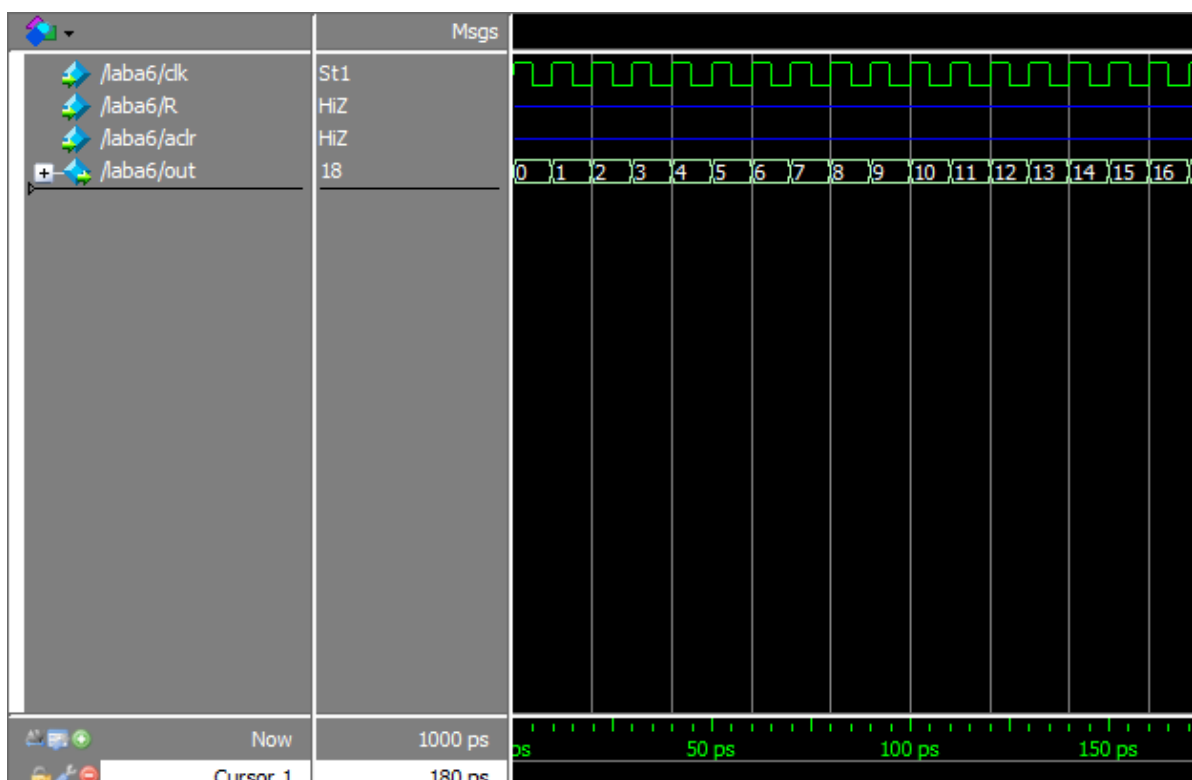


Рисунок 5 – Временная диаграмма (первая часть)

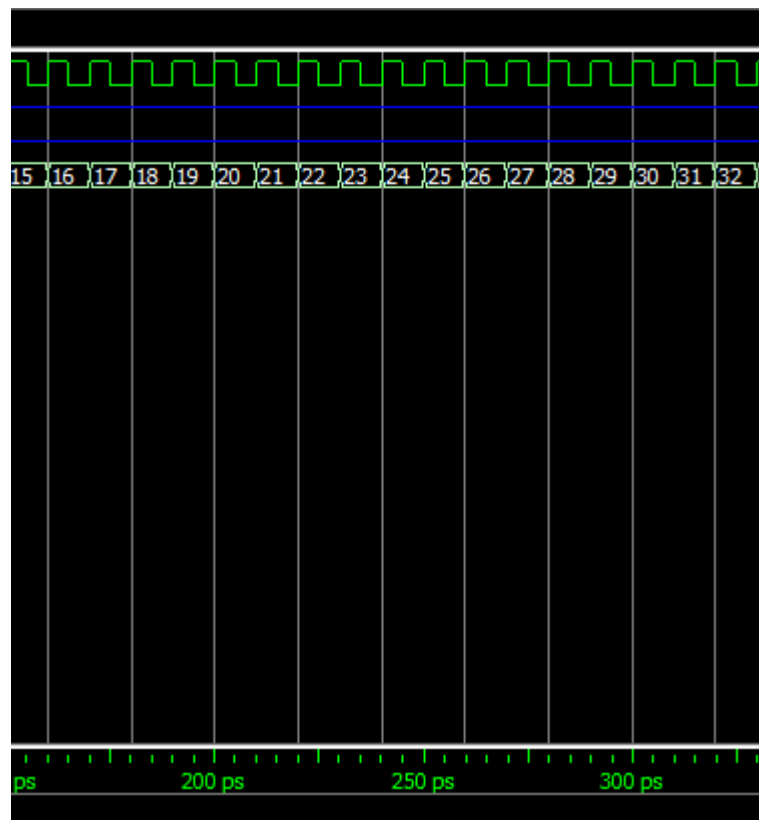


Рисунок 6 – Временная диаграмма (вторая часть)

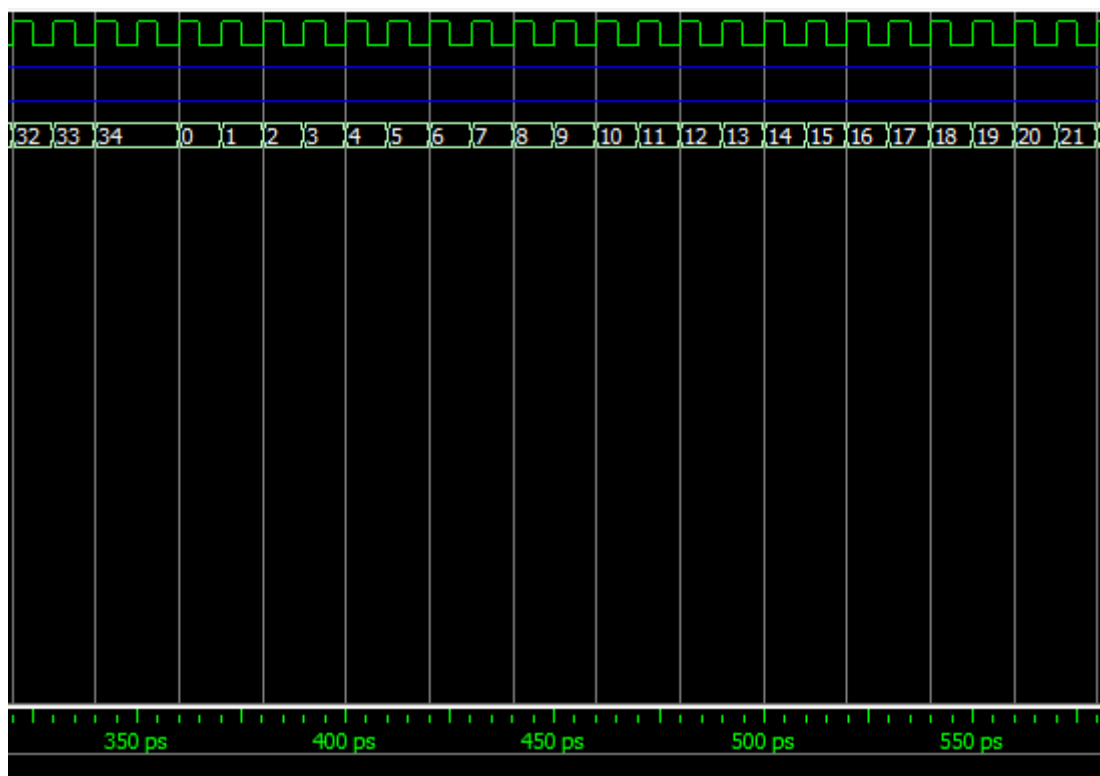


Рисунок 7 – Временная диаграмма (третья часть)

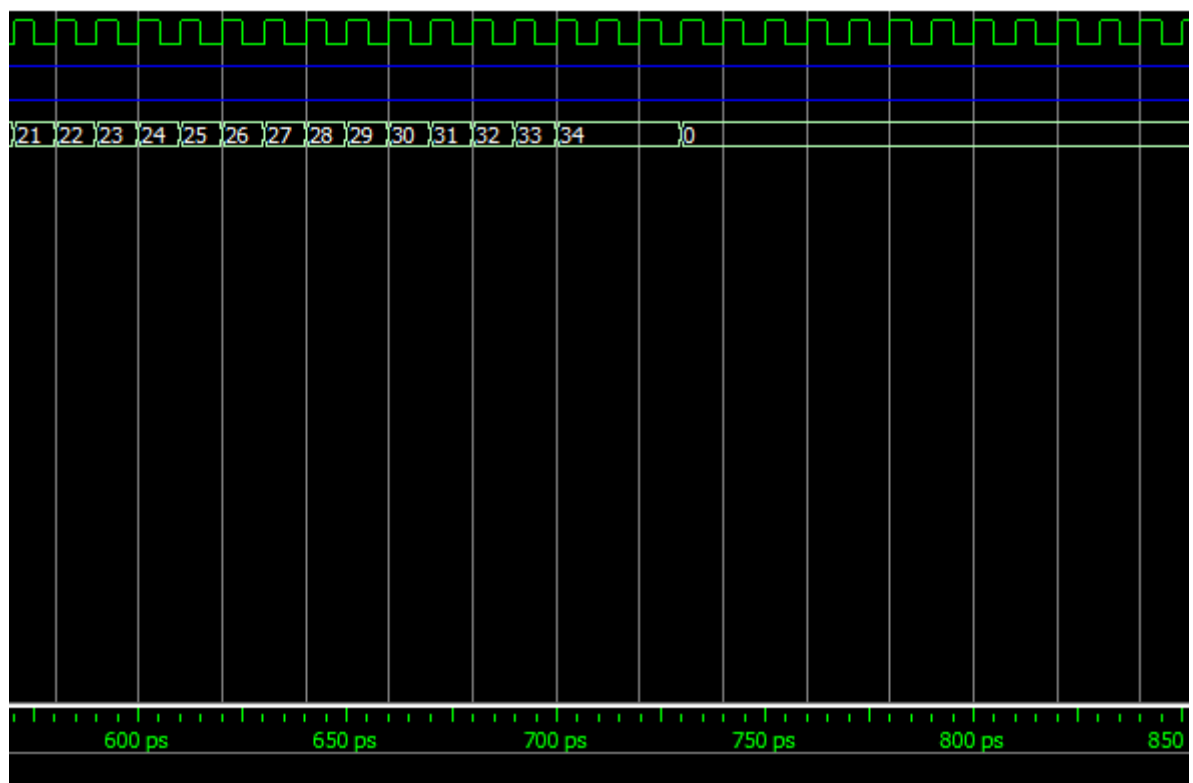


Рисунок 8 – Временная диаграмма (четвёртая часть)

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 №3 от 11 марта 2024 года по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Жаринов. О.О: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://bbb1.guap.ru/playback/presentation/2.3/4e99f54650dea30ef1dc263d08fafdd7f0c36944-1710157682528>. (Дата обращения: 05.04.24).
6. 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/2.3/8df832b35e59b7b17bb499a557e56942cdbfdf0b-1711367744513>. (Дата обращения: 05.04.24).

7. Отчёт о выполнении лабораторной работы №4 по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Тегай. Е.Д: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://pro.guap.ru/inside/student/reports/3910573/download>. (Дата обращения: 05.04.24).

8. Отчёт о выполнении лабораторной работы №3 по дисциплине «Схемотехника» (2-й семестр изучения дисциплины) // Тегай. Е.Д: [Электронный ресурс] // Санкт-Петербургский государственный университет аэрокосмического приборостроения. URL.: <https://pro.guap.ru/inside/student/reports/3907456/download>. (Дата обращения: 05.04.24).