ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук, доц.		О.О. Жаринов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

РАЗРАБОТКА ФОРМИРОВАТЕЛЯ ИМПУЛЬСОВ, УПРАВЛЯЕМОГО ЦИФРОВЫМ КОДОМ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКОВ ОПИСАНИЯ АППАРАТУРЫ

по курсу: СХЕМОТЕХНИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4143		Д.В. Пономарев
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Вариант 24

Разработать проект формирователя импульсов, параметры которых задаются внешним двоичным параллельным кодом в среде программирования Quartus, используя языки описания аппаратуры.

Индивидуальное задание

Задание заключается в формировании импульсов, параметры которых однозначно определяются цифровым управляющим 6-разрядным двоичным кодом. Проект устройства, который нужно реализовать на языке Verilog, должен иметь 1 выход и 8 входов:

1) один вход для подачи тактовых импульсов,

2) один вход для подачи импульса загрузки управляющего кода,

3) 6 входов для подачи внешнего управляющего 6-разрядного двоичного кода.

Номер варианта, а также его содержимое продемонстрировано в таблице 1. Для удобства, оно выделено жёлтым цветом. Следует пояснить, что напротив ячейки «Вар.» находится номер варианта, напротив «$К_1$» - значение, равное тактовой длительности вывода единиц, а «$К_0$» - значение, равное тактовой длительности вывода нулей.

Значение управляющего кода может изменяться в произвольное время. Во время активного состояния входа загрузки состояние выхода не регламентируется. По окончании загрузочного импульса должна автоматически инициироваться новая фаза работы устройства, под управлением полученного значения кода (число N). Начало новой фазы работы соответствует первому переднему фронту тактового импульса после снятия импульса загрузки.

Таблица 1

Таблица вариантов заданий

Вар.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
K_1	N	1	N	N	2	N	3	N	4	N	5	N	6	N	7
K_0	N	N	1	2	N	3	N	4	N	5	N	6	N	7	N
Вар.	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
K_1	8	N	N	9	N	10	N	11	N	12	N	13	N	14	N
K_0	N	8	9	N	10	N	11	N	12	N	13	N	14	N	15

Ход работы

Я реализовал эту работу используя модульную иерархию на языке описания аппаратуры Verilog.

Модуль lab1 использует модуль PulseGenerator и подключает его к входным и выходным портам.

Входные порты:

- clk: тактовые импульсы.
- load_pulse: импульс загрузки управляющего кода.
- control: внешний управляющий 6-разрядный двоичный код.

Выходной порт:

- out_pulse: выходной импульс.

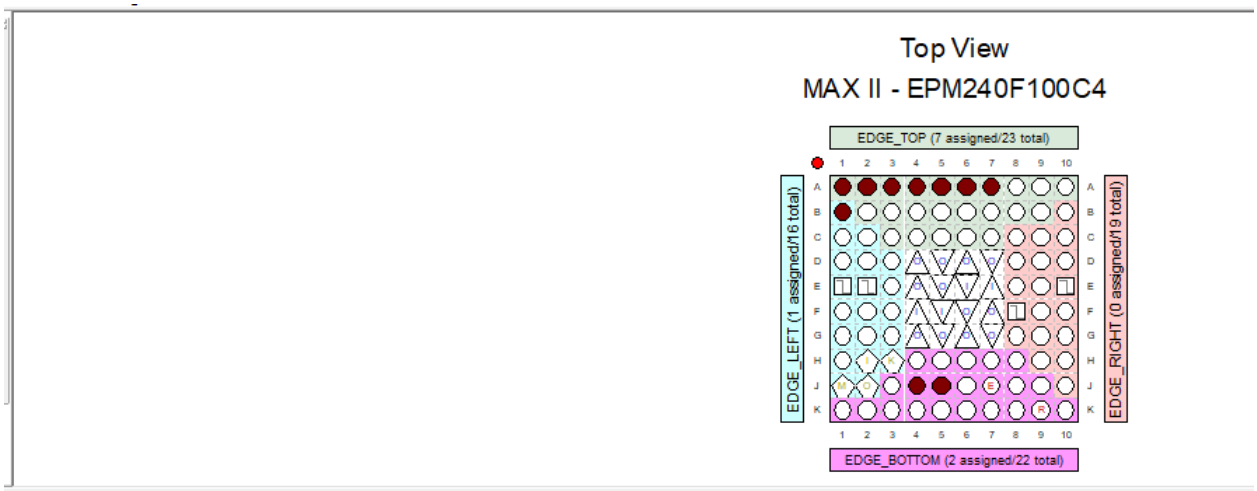
Модуль PulseGenerator генерирует выходной импульс (out_pulse) в зависимости от входного тактового сигнала (clk), сигнала загрузки управляющего кода (load_pulse) и внешнего управляющего кода (control). При активации сигнала load_pulse значение control загружается в регистр k1. Счётчик counter увеличивается на каждом положительном фронте тактового сигнала clk. Когда счётчик меньше значения в k1, выходной сигнал out_pulse устанавливается в 1, иначе он устанавливается в 0. После того, как out_pulse установлен в 0, он остаётся в этом состоянии в течение k0 тактов.

Листинг программы

```
module lab1 (

 input clk,                // тактовые импульсы
   input load_pulse,         // импульс загрузки управляющего кода
```

```
    input [5:0] control,       // внешний управляющий 6-разрядный двоичный код
    output  out_pulse       // выходной импульс
);

PulseGenerator PulseGen(
    .clk(clk),
    .load_pulse(load_pulse),
    .control(control),
    .out_pulse(out_pulse)
);

endmodule

module PulseGenerator(
    input clk,              // тактовые импульсы
    input load_pulse,      // импульс загрузки управляющего кода
    input [5:0] control,   // внешний управляющий 6-разрядный двоичный код
    output reg out_pulse   // выходной импульс
);

reg [6:0] k1;             // значение для длительности высокого уровня
parameter k0 = 11;        // значение для длительности низкого уровня

reg [6:0] counter;        // счётчик

always @(posedge clk) begin
    if (load_pulse) begin
        k1 <= control;  // Обновляем значение k1 при активации load_pulse
        counter <= 0;
    end else begin
        if (counter < k1) begin
            out_pulse <= 1;
            counter <= counter + 1;
        end else if (counter < k1 + k0) begin
            out_pulse <= 0;
            counter <= counter + 1;
        end else begin
            out_pulse <= 0;
            counter <= 0;
        end
    end
end

endmodule
```

ПЛИС



Edit: Output

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved
clk	Input	PIN_A1	2		3.3-V LVTTL (default)	
control[5]	Input	PIN_A2	2		3.3-V LVTTL (default)	
control[4]	Input	PIN_A3	2		3.3-V LVTTL (default)	
control[3]	Input	PIN_A4	2		3.3-V LVTTL (default)	
control[2]	Input	PIN_A5	2		3.3-V LVTTL (default)	
control[1]	Input	PIN_A6	2		3.3-V LVTTL (default)	
control[0]	Input	PIN_A7	2		3.3-V LVTTL (default)	
load_pulse	Input	PIN_B1	1		3.3-V LVTTL (default)	
out_pulse	Output	PIN_J5	1		3.3-V LVTTL (default)	
out	Unknown	PIN_J4	1		3.3-V LVTTL (default)	

Рисунок 1 - ПЛИС

Временная диаграмма

Результаты функциональной и временной симуляций при малых значениях N продемонстрированы на рисунках 2 – 4.

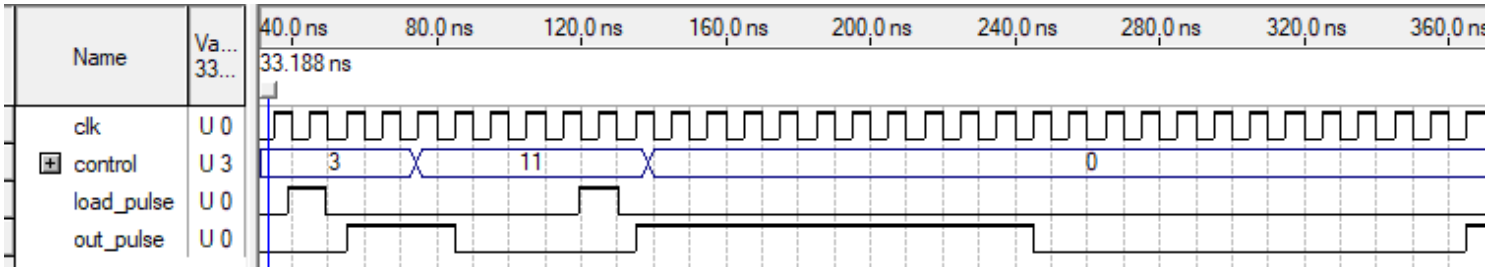


Рисунок 2 – Функциональная симуляция при малых значениях (11)

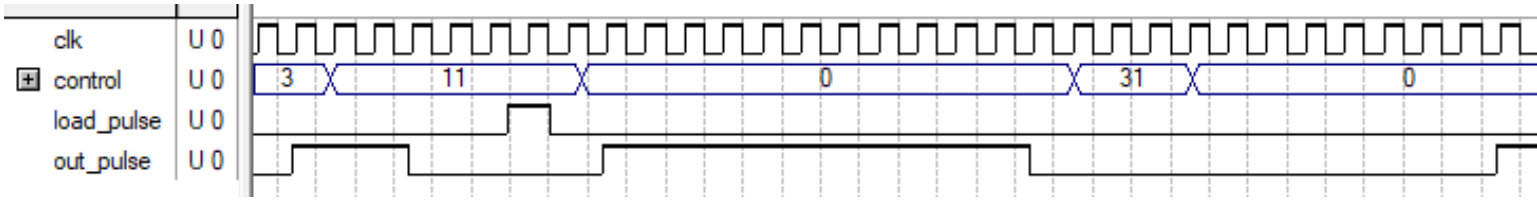


Рисунок 3 – Временная симуляция при малых значениях

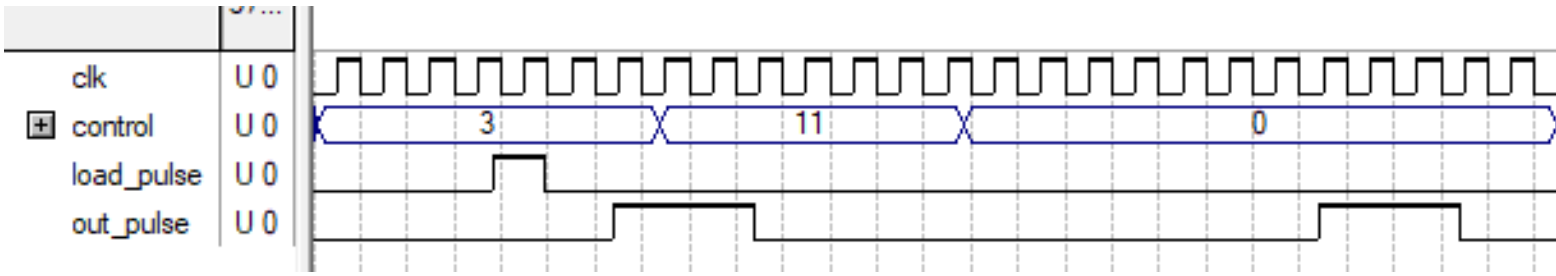


Рисунок 4 – Временная симуляция при малых значениях

Результаты функциональной и временной симуляций при средних значениях N продемонстрированы на рисунках 5 – 8.

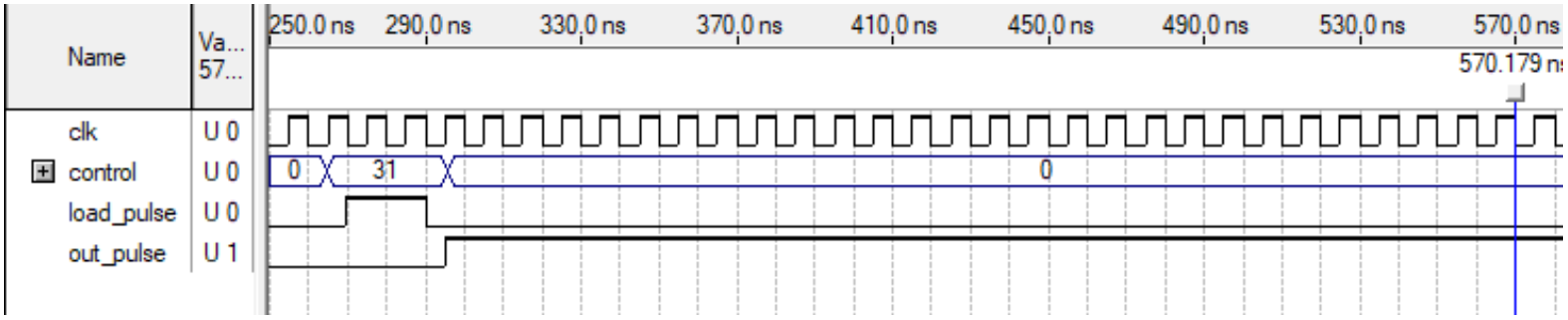


Рисунок 5 – Функциональная симуляция при средних значениях (начало)

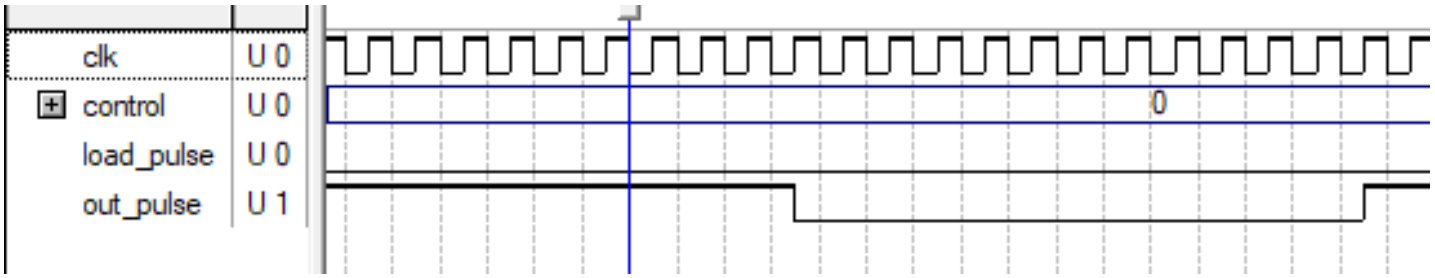


Рисунок 6 – Функциональная симуляция при средних значениях (конец)

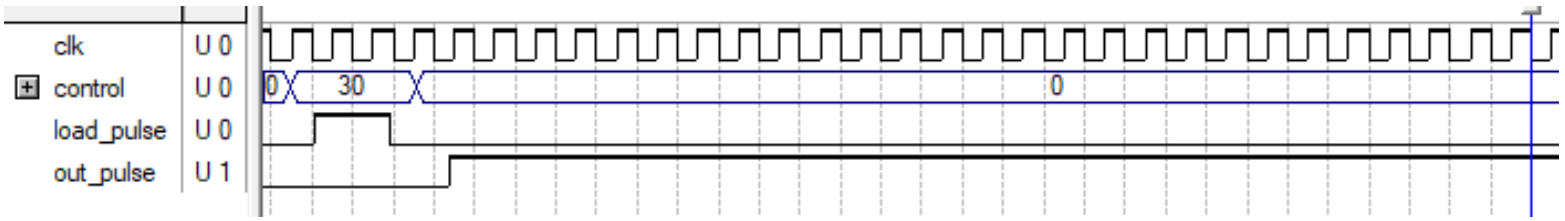


Рисунок 7 – Временная симуляция при средних значениях (начало)

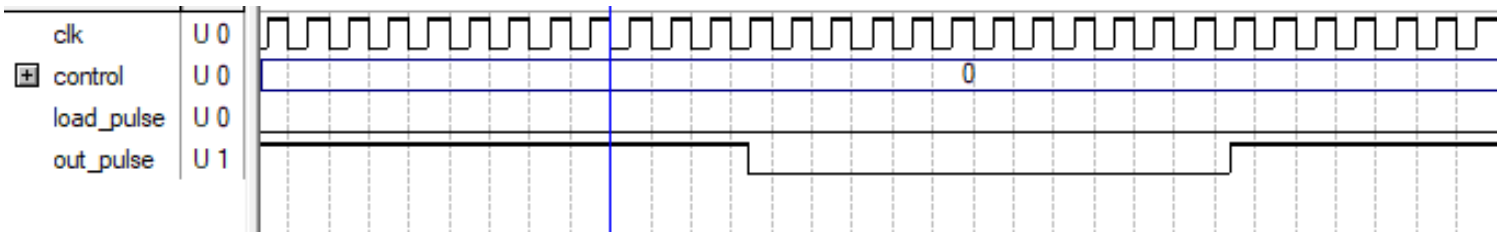


Рисунок 8 – Временная симуляция при средних значениях (конец)

Наконец, результаты функциональной и временной симуляций с максимальным значением N продемонстрированы на рисунках 16 – 21.

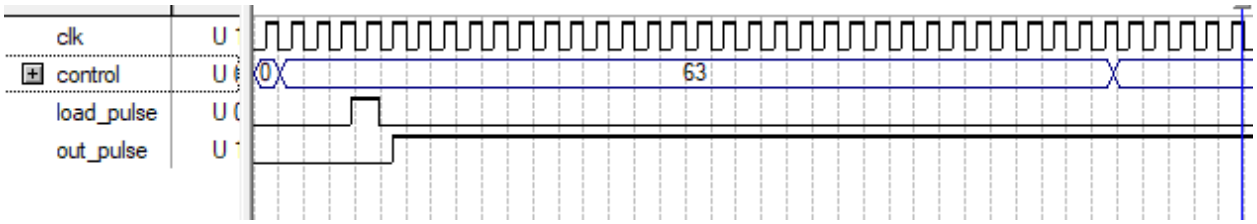


Рисунок 9 – Функциональная симуляция при максимальных значениях (начало)

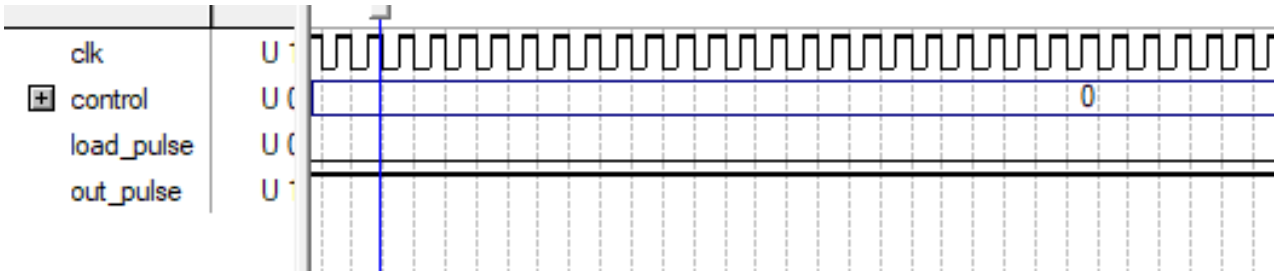


Рисунок 10 – Функциональная симуляция при максимальных значениях (середина)

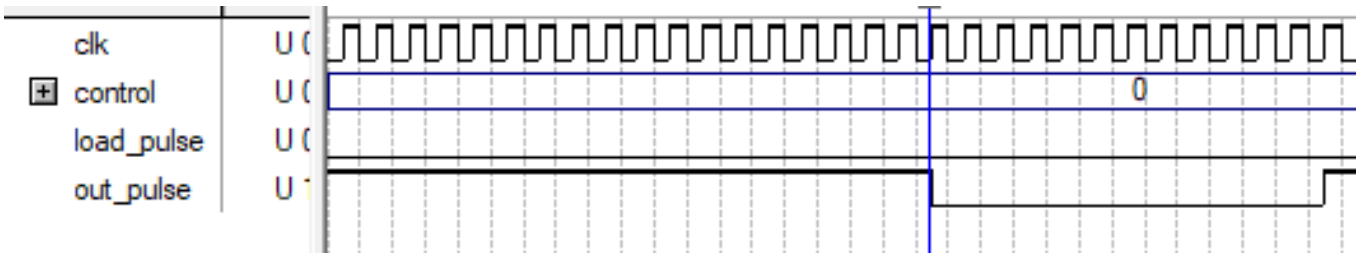


Рисунок 11 – Функциональная симуляция при максимальных значениях (конец)

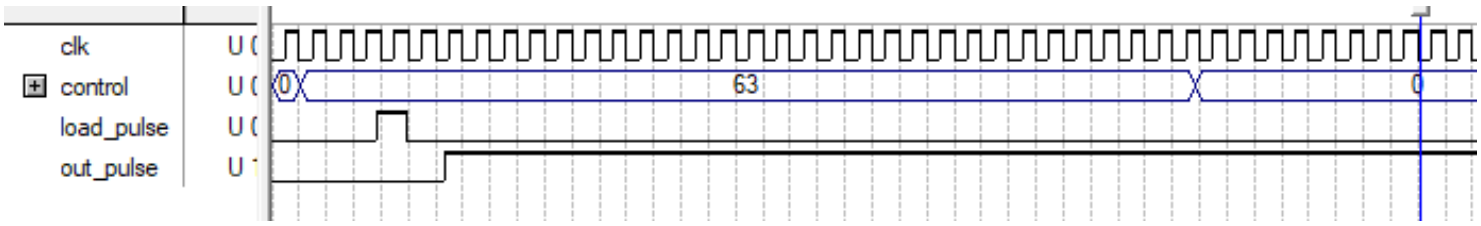


Рисунок 12 – Временная симуляция при максимальных значениях (начало)

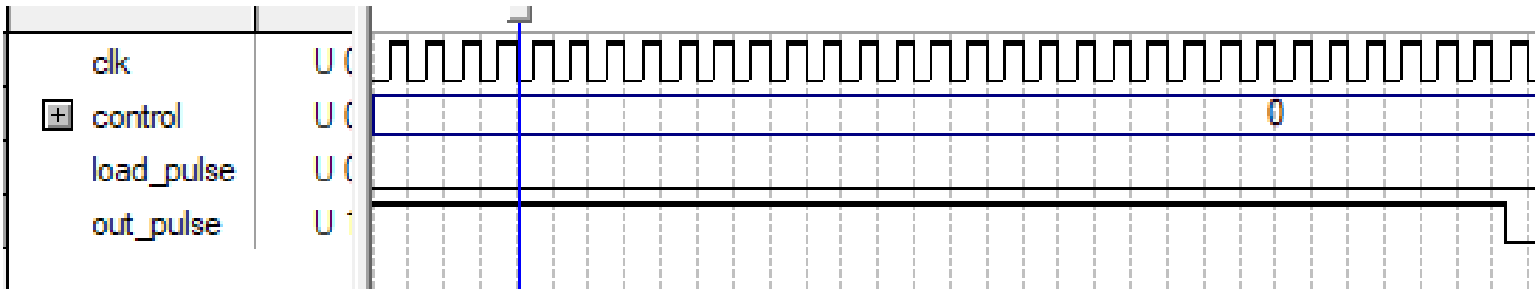


Рисунок 13 – Временная симуляция при максимальных значениях (середина)

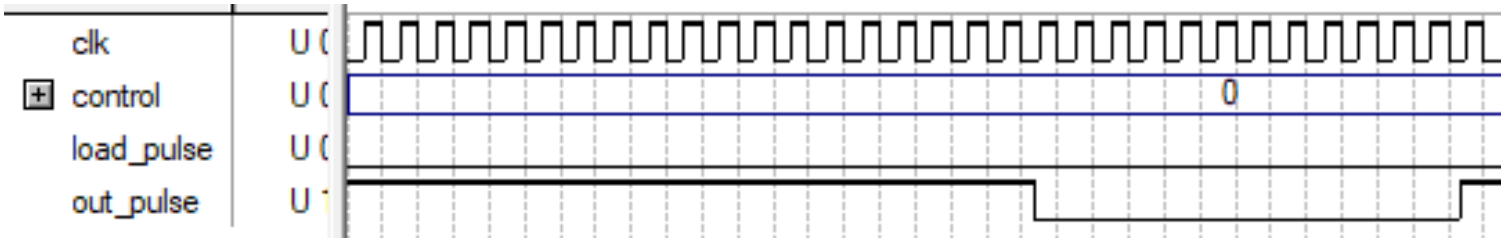


Рисунок 14 – Временная симуляция при максимальных значениях (конец)

Выводы

В данной лабораторной работе был разработан проект формирователя импульсов, параметры которых задаются внешним двоичным параллельным кодом 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. ImaginationTechnologies. – М.: ДМК Пресс, 2018. - 792 с.

4. Учебно-методические материалы к выполнению лабораторной работы №5 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/4f645754182c1092dee745c6a09dc3fe/download. (Дата обращения: 29.03.24).

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: 29.03.24).

6. 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: 29.03.24).