ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А.Н.Тихонова

**Домашнее задание**

**часть 1**

по направлению 09.03.01 Информатика и вычислительная техника студента образовательной программы бакалавриата

«Информатика и вычислительная техника»

Дисциплина:

«[Проектирование систем на кристалле](https://edu.hse.ru/course/view.php?id=188925)»

Преподаватель:

Американов Александр Александрович

Выполнил:

Студенты группы: БИВ215

Борисова Софья Алексеевна

Розанцева Валерия Викторовна

Дата сдачи: 11.06.2024

Москва 2024

**Оглавление**

[1 Постановка задачи 3](#_gjdgxs)

[2 Выполнение работы 4](#_30j0zll)

[2.1 ЛР1. Основы комбинационный логики. Маршрут разработки цифровых схем 4](#_1pvbxgugudj0)

[2.1.1 Настройка проекта 4](#_1fob9te)

[2.1.2 Файл сценария 6](#_2et92p0)

[2.1.3 Результаты 8](#_gve9515ovwj9)

[2.2 ЛР2. Основы последовательностной логики. Управление энергопотреблением цифровой схемы 9](#_odgc70z5t1vz)

[2.2.1 RS-триггер 9](#_w7mcja3596gq)

[2.2.2 D-защелка 17](#_d700fv8jrn1d)

[2.2.3 JK-триггер 19](#_b707r0pa8x7x)

[2.2.4 T-триггер 23](#_klm9w0yybwn2)

[2.2.5 D-защелка по нисходящему фронту 24](#_50i8q59ghczp)

[2.2.6 D-триггер 26](#_37j96iabs23w)

[2.3 ЛР3. Шифраторы и дешифраторы. Скорость работы комбинационных блок 29](#_qjgj13laz4g3)

[2.3.1 Неприоритетный шифратор 29](#_i3rhl71u84vn)

[2.3.2 Приоритетный шифратор 31](#_83nvm4ob0a5u)

[2.3.3 Дешифратор 36](#_dh4w3h9wmil2)

[2.3.4 Самостоятельная работа 37](#_sdej6iowypkx)

[3 Выводы 39](#_nmf14n)

[4 Список литературы 40](#_37m2jsg)

# 1 Постановка задачи

Необходимо выполнить 3 первые лабораторные работы на удалённых стендах. Работы делаются через Телеграм бота.

# 2 Выполнение работы

## 2.1 ЛР1. Основы комбинационный логики. Маршрут разработки цифровых схем

### 2.1.1 Настройка проекта

Сначала необходимо выбрать плату DE10-Lite с чипом MAX10 10M50DAF484, так как на данный момент в удаленной лаборатории задействованы исключительно эти платы.

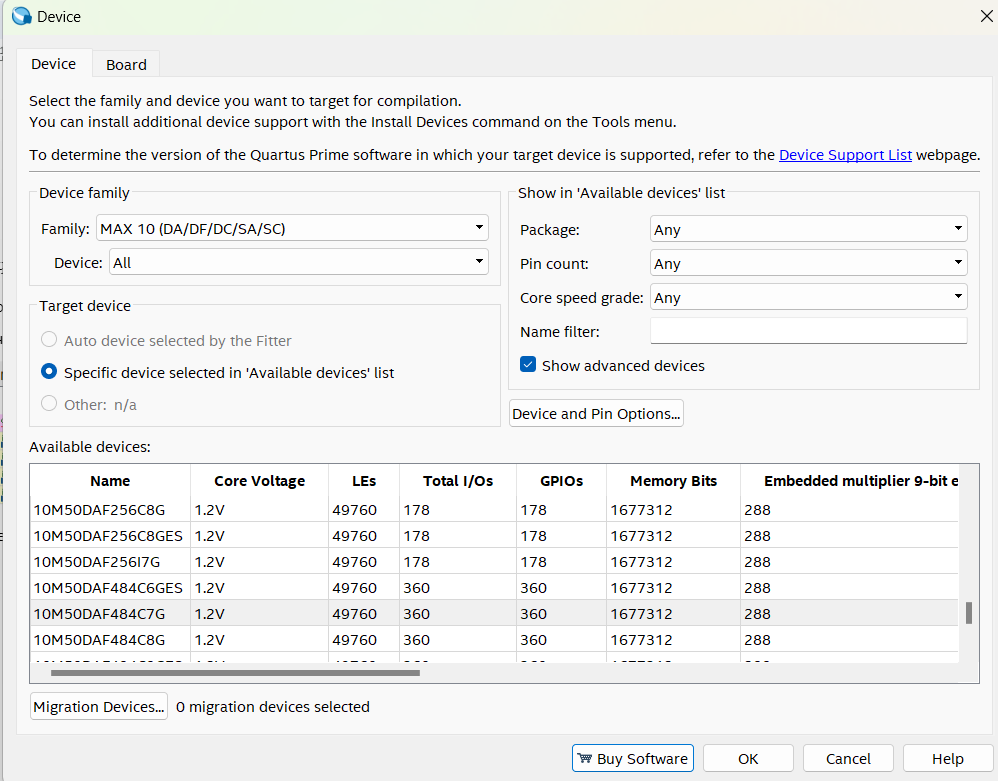


Рисунок 1. Выбор платы

При работе с платой ПЛИС в удалённой лаборатории нет возможности вручную с взаимодействовать с входными сигналы с кнопок и DIP переключателей. Поэтому необходимо переназначить входные порты на GPIO порты.

Ниже представлен пример прямого назначения портов GPIO на светодиоды. Важно: используются именно эти порты GPIO, а не все 40 имеющихся на плате.

assign LEDR[0] = GPIO[16];

assign LEDR[1] = GPIO[14];

assign LEDR[2] = GPIO[12];

assign LEDR[3] = GPIO[10];

assign LEDR[4] = GPIO[ 6];

assign LEDR[5] = GPIO[ 4];

assign LEDR[6] = GPIO[ 2];

assign LEDR[7] = GPIO[ 0];

Также необходимо поменять назначение портов в Pin-Planner. Согласно документации на плату DE10-Lite.

Таблица 1

| GPIO\_[0] | PIN\_V10 |
| --- | --- |
| GPIO\_[2] | PIN\_V9 |
| GPIO\_[4] | PIN\_V8 |
| GPIO\_[6] | PIN\_V7 |
| GPIO\_[10] | PIN\_W5 |
| GPIO\_[12] | PIN\_AA14 |
| GPIO\_[14] | PIN\_W12 |
| GPIO\_[16] | PIN\_AB12 |

Назначение пинов (Рисунок 2).

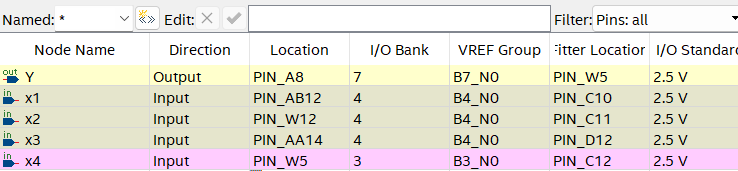


Рисунок 2. Назначение пинов

Далее следует сохранить и скомпилировать проект (Рисунок 3).

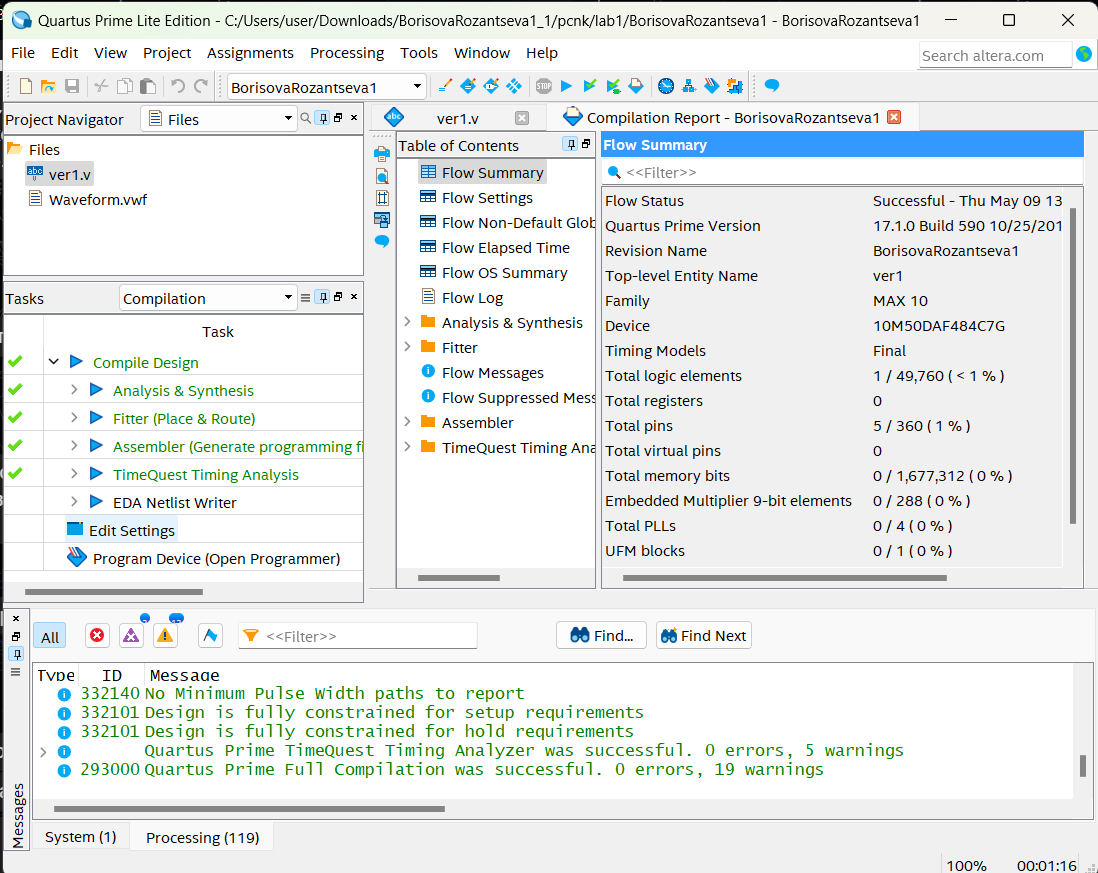


Рисунок 3. Компиляция проекта

На следующем этапе необходимо написать файл сценария. Файл сценария является файлом в формате .txt с указанием того, на какие пины необходимо подавать тестовые воздействия.

### 2.1.2 Файл сценария

Пользовательские команды управления платой ПЛИС задаются в специализированном текстовом файле сценария. Он необходим, чтобы в асинхронном формате доступа к удаленному стенду пользователь имел возможность задать необходимую последовательность команд для последующего их выполнения.

Правила оформления файла сценария:

1. Каждая новая команда обязательно начинается с новой строки.
2. Номер пина от 1 до 8.
3. В конце файла должна присутствовать команда «end», которая означает конец распознаваемого сценария.

Список команд, которые может использовать пользователь:

1. «Кнопка + номер пина». Правила записи: button 1, button1, but 1, but1. Данная команда будет распознана, как нажатие кнопки с номером 1 на 0.5 с.
2. «Переключатель + номер пина». Правила записи: switch 1, switch1, sw 1, sw1. Данная команда будет распознана, как перевод переключателя в противоположное текущему положение. Изначально свитч находится в положении 0.
3. «Задержка + время». Данная команда записывается как delay, и через пробел указано время в секундах для задержки. Она будет применяться к той команде, которая находится на строчку выше.

Пример корректного оформления файла сценария с соблюдением всех правил его написания представлен на рисунке 4.

Изображение выглядит как текст, Шрифт, типография

Автоматически созданное описание

Рисунок 4. Пример составления файла сценария

Написанный сценарий (Рисунок 5).

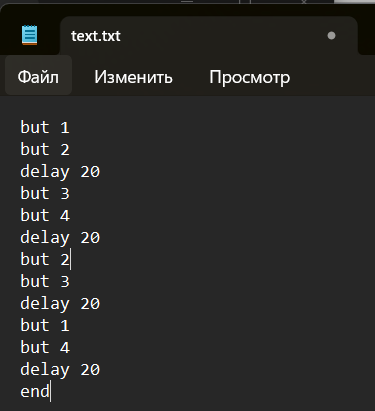


Рисунок 5. Сценарий для первой лабораторной работы

### 2.1.3 Результаты

На указанную почту чат-бот присылает файлы после загрузки прошивки и сценария (Рисунок 6).

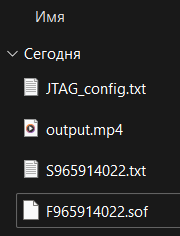


Рисунок 6. Результат

Для просмотра результата необходимо открыть файл “.mp4”.

В данной лабораторной работе для функции при любой комбинации x0, x1, x2, x3 будет гореть LED (Рисунок 7).

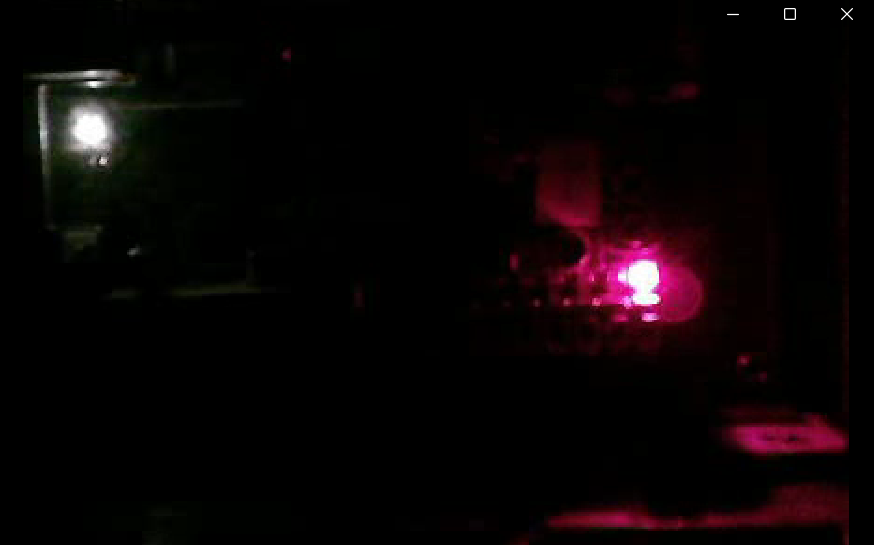


Рисунок 7. Результат

## 2.2 ЛР2. Основы последовательностной логики. Управление энергопотреблением цифровой схемы

### 2.2.1 RS-триггер

Папка **00\_sr\_latch.**

Изменение файла верхнего уровня (Рисунок 8).

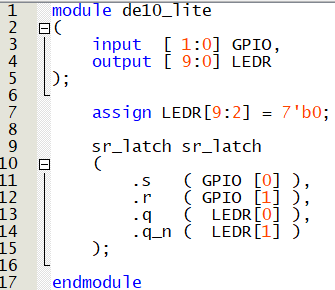


Рисунок 8. Файл верхнего уровня для RS-триггера

Назначение пинов (Рисунок 9).

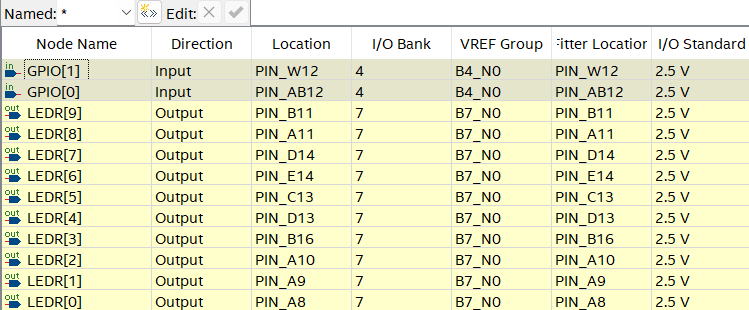


Рисунок 9. Назначение пинов

Написанный сценарий (Рисунок 10).

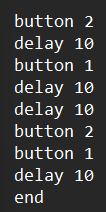


Рисунок 10. Сценарий для RS-триггера

Далее приведены результаты с удаленного стенда (Рисунок 11–15).

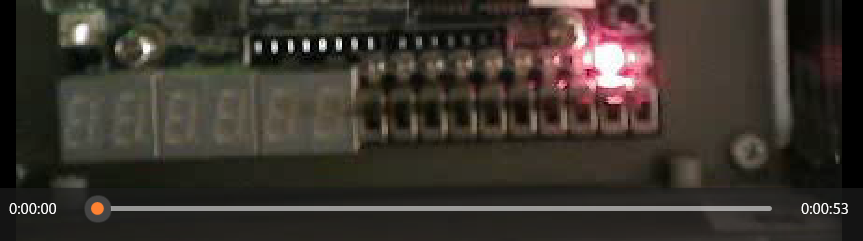


Рисунок 11. Режим сброса: S=0, R=1, Q=0, ~Q=1

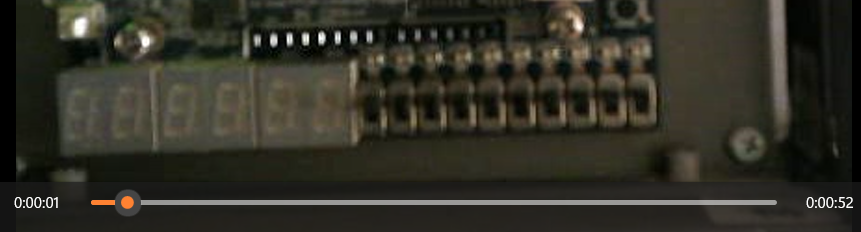


Рисунок 12. Delay 10

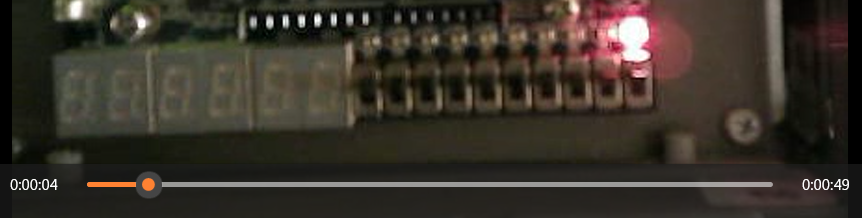


Рисунок 13. Режим установки: S=1, R=0, Q=1, ~Q=0

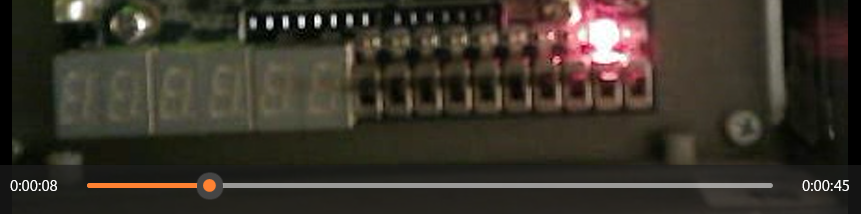


Рисунок 14. Запрещенная комбинация: S=R=1, фото 1

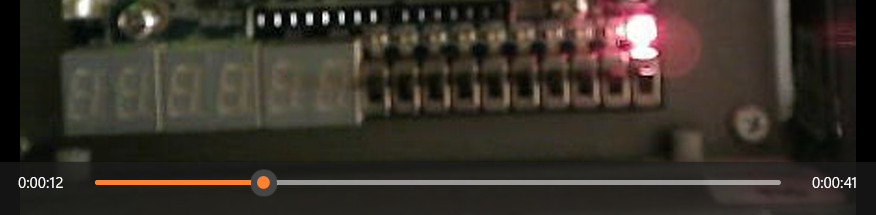


Рисунок 15. Запрещенная комбинация: S=R=1, фото 2

Реализовать запрещенную комбинацию нет возможностей, так как нажатие кнопок осуществляется не одновременно.

Далее приведен проект, где в сценарии используются свитчи.

Изменение файла верхнего уровня (Рисунок 16). Входные сигналы – массив GPIO, выходные – светодиоды.

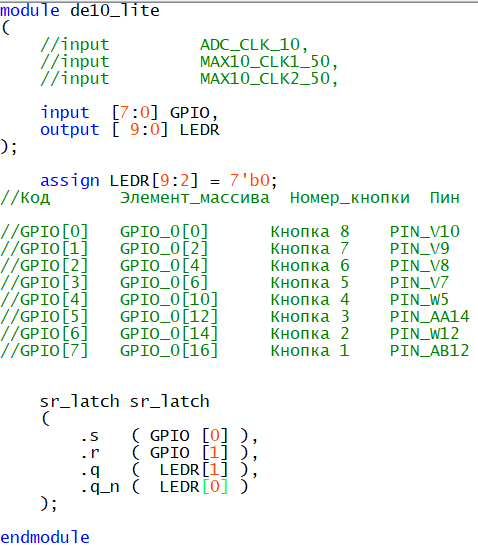


Рисунок 16. Файл верхнего уровня 00\_sr\_latch

Реализация RS-триггера (Рисунок 17).

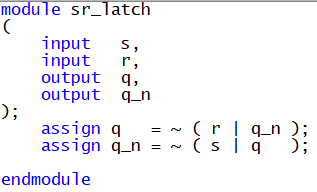


Рисунок 17. Реализация RS-триггера

Назначение пинов (Рисунок 18).

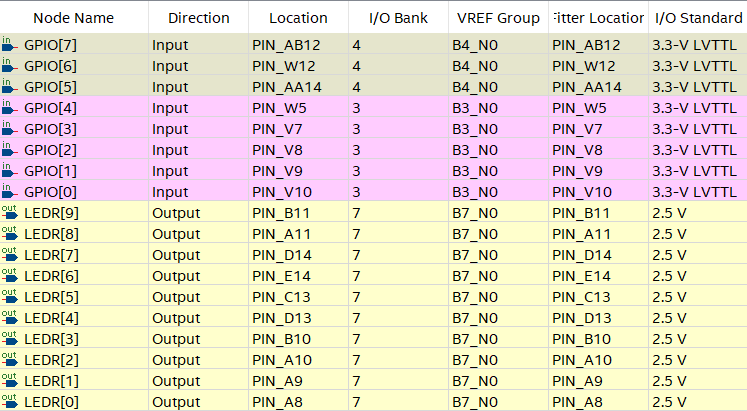


Рисунок 18. Назначение пинов

Или файл qsf изменить:

# LEDR

set\_location\_assignment PIN\_A8 -to LEDR[0]

set\_location\_assignment PIN\_A9 -to LEDR[1]

set\_location\_assignment PIN\_A10 -to LEDR[2]

set\_location\_assignment PIN\_B10 -to LEDR[3]

set\_location\_assignment PIN\_D13 -to LEDR[4]

set\_location\_assignment PIN\_C13 -to LEDR[5]

set\_location\_assignment PIN\_E14 -to LEDR[6]

set\_location\_assignment PIN\_D14 -to LEDR[7]

set\_location\_assignment PIN\_A11 -to LEDR[8]

set\_location\_assignment PIN\_B11 -to LEDR[9]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to LEDR\*

# GPIO

set\_location\_assignment PIN\_V10 -to GPIO[0]

set\_location\_assignment PIN\_V9 -to GPIO[1]

set\_location\_assignment PIN\_V8 -to GPIO[2]

set\_location\_assignment PIN\_V7 -to GPIO[3]

set\_location\_assignment PIN\_W5 -to GPIO[4]

set\_location\_assignment PIN\_AA14 -to GPIO[5]

set\_location\_assignment PIN\_W12 -to GPIO[6]

set\_location\_assignment PIN\_AB12 -to GPIO[7]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[0]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[1]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[2]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[3]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[4]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[5]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[6]

set\_instance\_assignment -name IO\_STANDARD "3.3-V LVTTL" -to GPIO[7]

Назначены:

s ( GPIO [0] ) – кнопка (свитч) 8, r ( GPIO [1]) – кнопка (свитч) 7, q – LEDR [1], ~q – LEDR [0].

Сценарий создается в текстовом файле, на одной строке – одна команда. Команда sw 8 – Написанный сценарий (Рисунок 19).

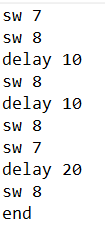


Рисунок 19. Сценарий для 00\_sr\_latch

Далее приведены результаты с удалённого стенда (Рисунок 20–23).

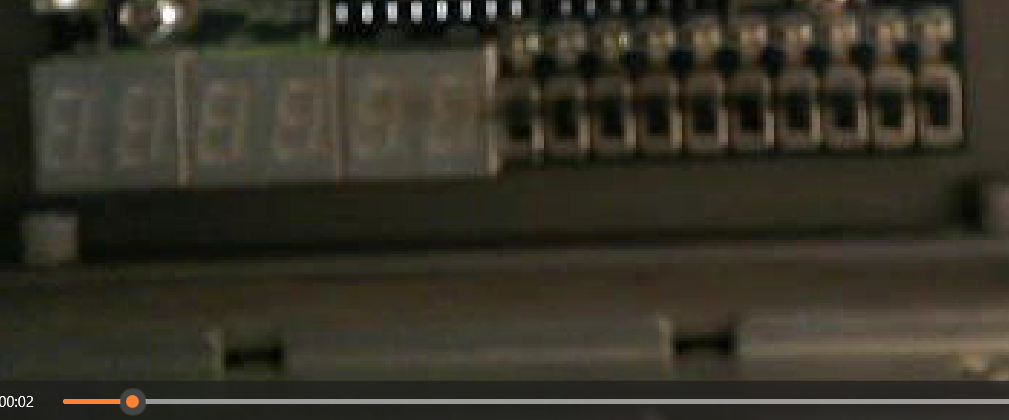


Рисунок 20. Запрещенная комбинация: S=1, R=1, Q=0, ~Q=0

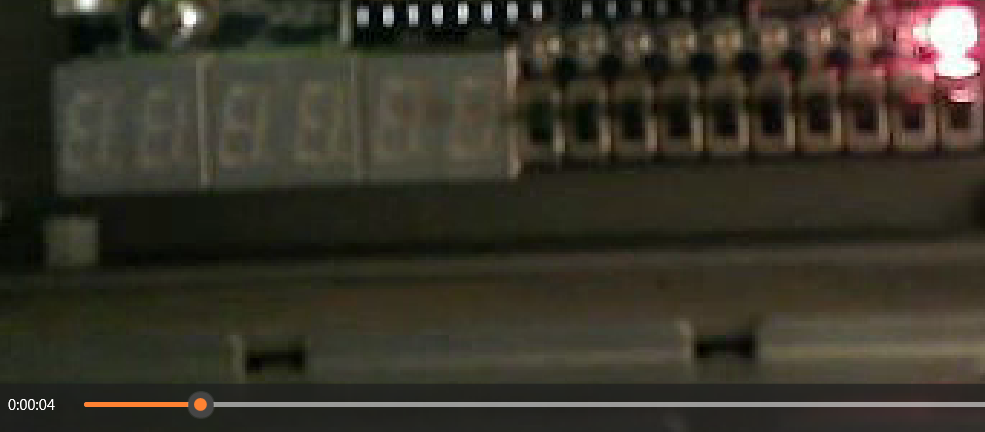


Рисунок 21. Установка в нулевое состояние: s=0, r=1, Q=0, ~Q=1

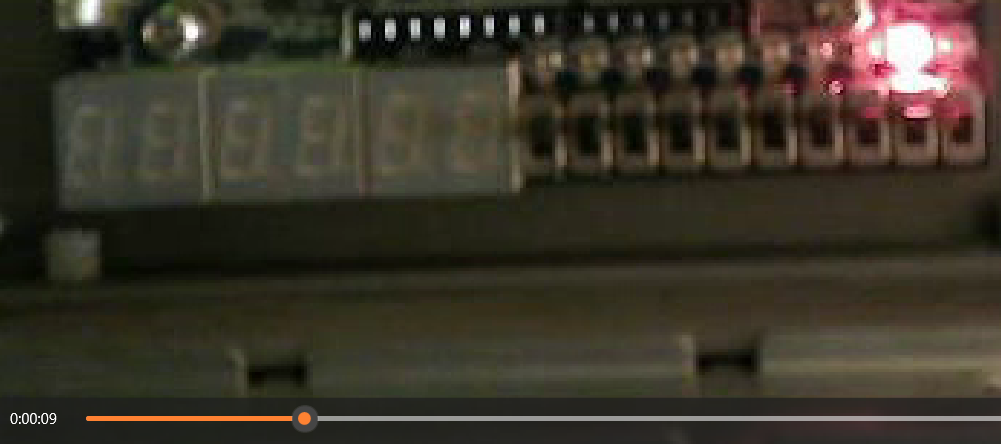


Рисунок 22. Установка в единичное состояние: s=1, r=0, Q=1, ~Q=0



Рисунок 23. Режим хранения предыдущего состояния

### 2.2.2 D-защелка

Папка **01\_d\_latch**.

Файл верхнего уровня (Рисунок 24).

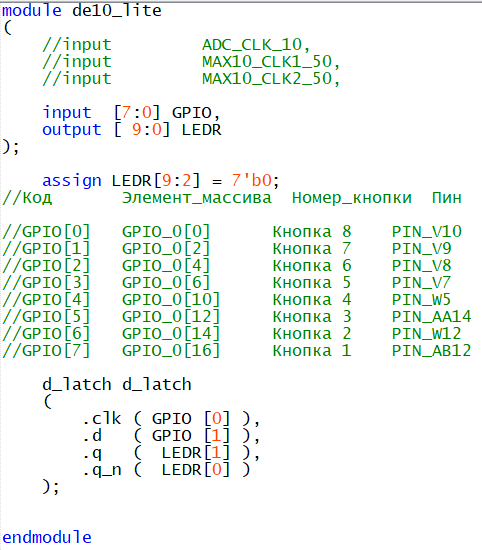


Рисунок 24. Файл верхнего уровня 01\_d\_latch

Реализация D-защелки (Рисунок 25).

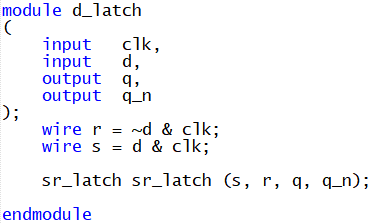


Рисунок 25. Реализация D-защелки

Назначения пинов сохраняются.

clk ( GPIO [0] ) – свитч 8, d ( GPIO [1]) – свитч 7, q – LEDR [1], ~q – LEDR [0].

Файл сценария (Рисунок 26).

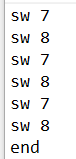


Рисунок 26. Сценарий для 01\_d\_latch

Работа удаленного стенда (Рисунок 27–29).

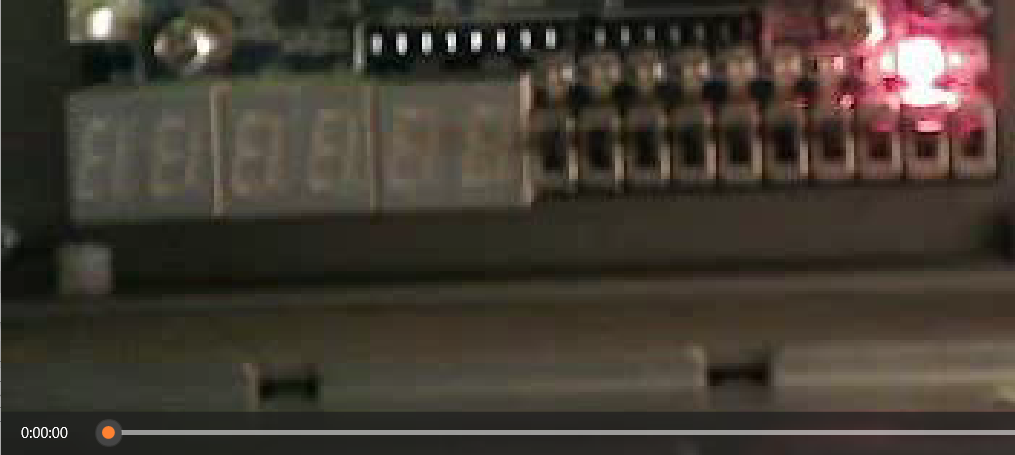


Рисунок 27. d=1, clk=1, выходы Q=1, ~Q=0

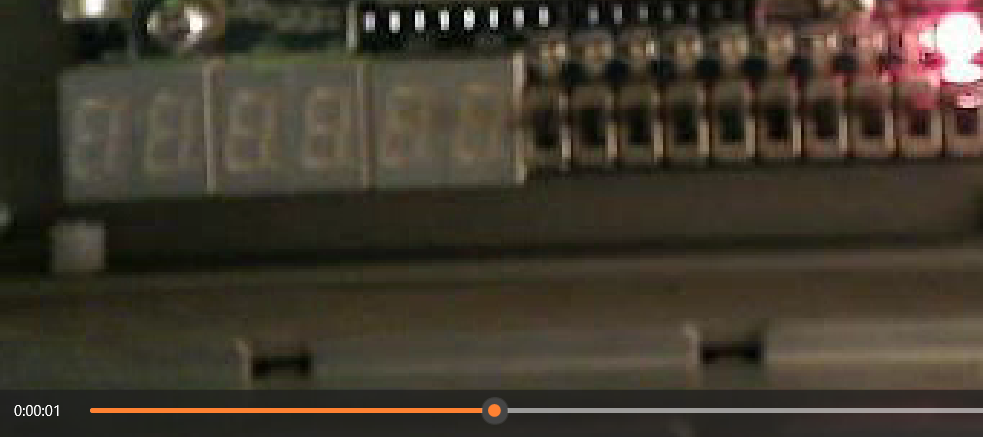


Рисунок 28. d=0, clk=0, выходы Q=0, ~Q=1

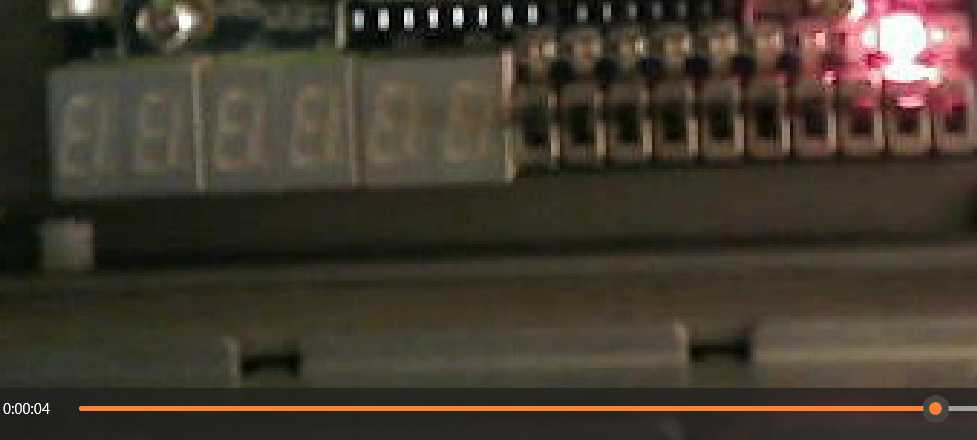


Рисунок 29. d=1, clk=1, выходы Q=1, ~Q=0

### 2.2.3 JK-триггер

Файл верхнего уровня (Рисунок 30).

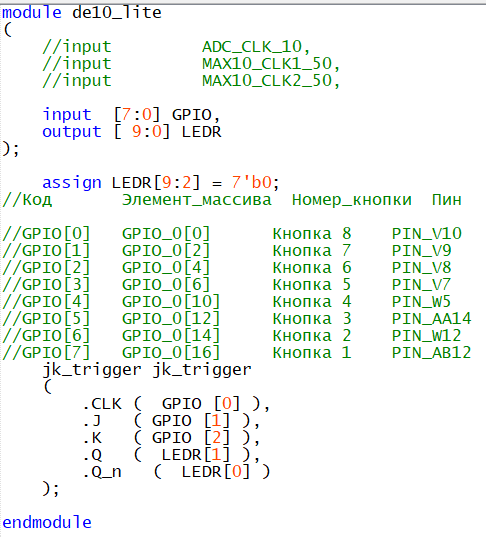


Рисунок 30. Файл верхнего уровня JK-триггера

Реализация JK-триггера (Рисунок 31).

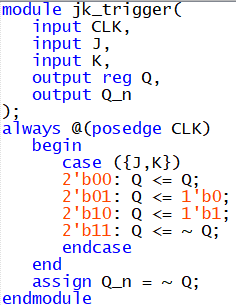


Рисунок 31. Реализация JK-триггера

Назначения пинов сохраняются.

clk ( GPIO [0] ) – кнопка 8, J ( GPIO [1]) – свитч 7, K – свитч 6, q – LEDR [1], ~q – LEDR [0].

Файл сценария (Рисунок 32).

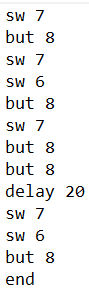


Рисунок 32. Сценарий для JK-триггера

Тестбенч, созданный по файлу сценария (Рисунок 33).

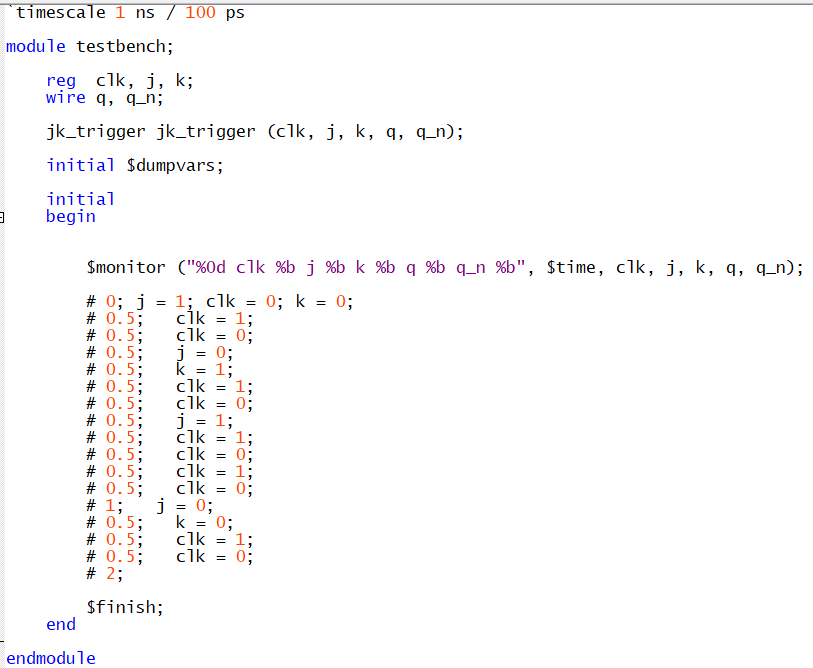


Рисунок 33. Тестбенч для JK-триггера

Результат моделирования (Рисунок 34).

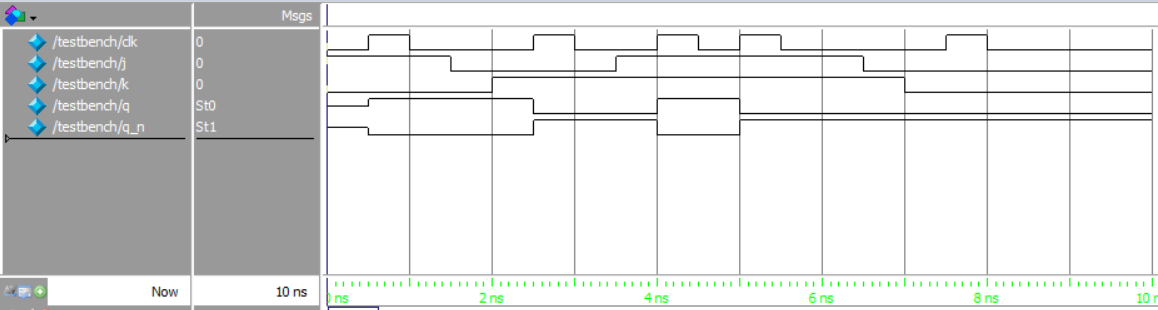


Рисунок 34. ModelSim

Результат тестбенча совпадает с работой платы (Рисунок 35–39).

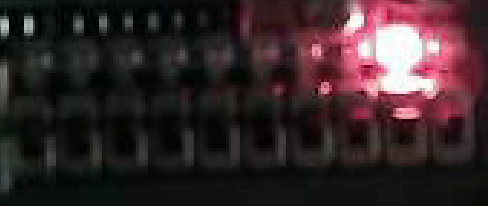


Рисунок 35. Установка в единичное состояние: CLK=1, J=1, K=0, Q=1, ~Q=0

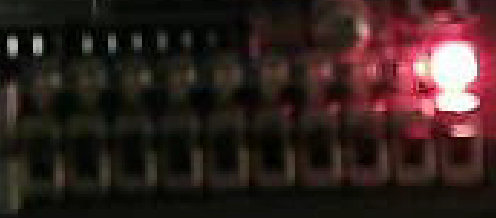


Рисунок 36. Установка в нулевое состояние: CLK=1, J=0, K=1, Q=0, ~Q=1



Рисунок 37. Режим счетчика: CLK=1, J=1, K=1, Q=1, ~Q=0

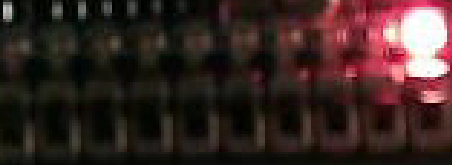


Рисунок 38. Режим счетчика: CLK=1, J=1, K=1, Q=0, ~Q=1

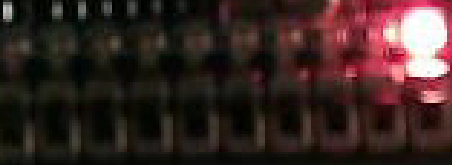


Рисунок 39. Режим хранения: CLK=1, J=0, K=0, Q=1, ~Q=0

### 2.2.4 T-триггер

Папка **02\_t\_latch**.

Файл верхнего уровня (Рисунок 40).

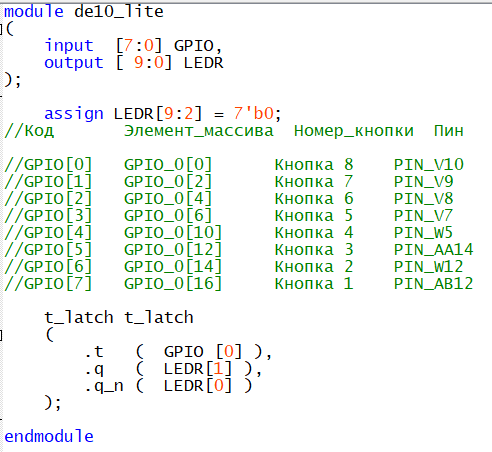


Рисунок 40. Файл верхнего уровня 02\_t\_latch

Реализация T-триггера на основе D-защелки (Рисунок 41).

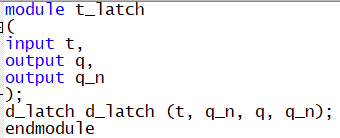


Рисунок 41. Реализация T-триггера

Назначения пинов сохраняются. Написанный сценарий (Рисунок 42).

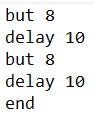


Рисунок 42. Сценарий для 02\_t\_latch

Далее приведены результаты с удаленного стенда (Рисунок 43–44).

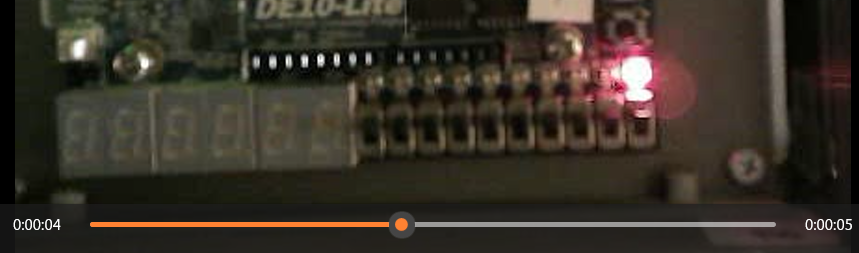


Рисунок 43. На входе: T=1, Q=0, ~Q=1

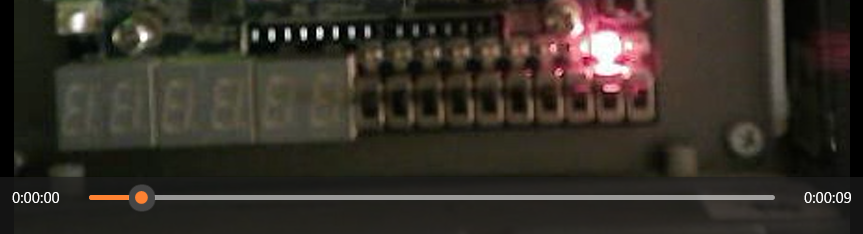


Рисунок 44. На входе: T=1, Q=1, ~Q=0

### 2.2.5 D-защелка по нисходящему фронту

Папка **d\_dop\_latch**.

Файл верхнего уровня (Рисунок 45).

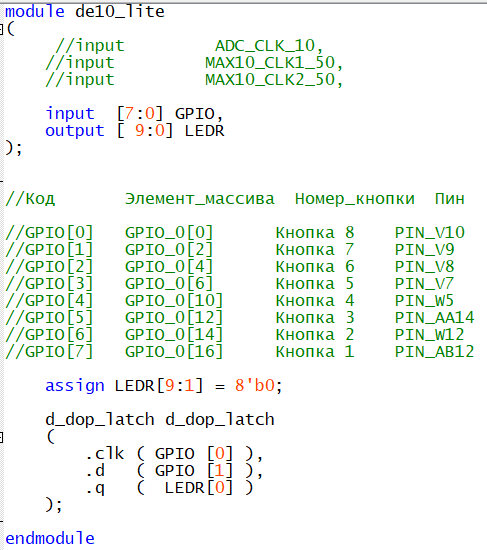


Рисунок 45. Файл верхнего уровня d\_dop\_latch

Реализация D-защелки, где изменения происходит при отрицательном фронте clk (Рисунок 46).

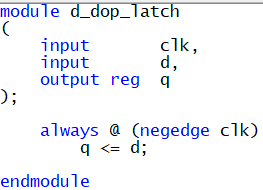


Рисунок 46. Реализация D-защелки по нисходящему фронту

Назначения пинов сохраняются. Написанный сценарий (Рисунок 47).

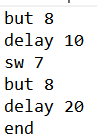


Рисунок 47. Сценарий для D-защелки по нисходящему фронту

clk ( GPIO [0] ) – кнопка 8, d – свитч 7, q – LEDR [0].

Далее приведены результаты с удаленного стенда (Рисунок 48–49).

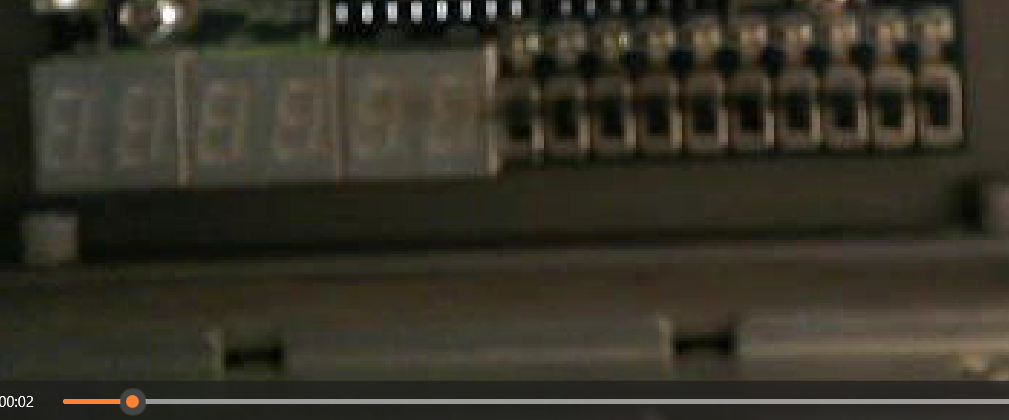


Рисунок 48. d=0, clk=1, Q=0

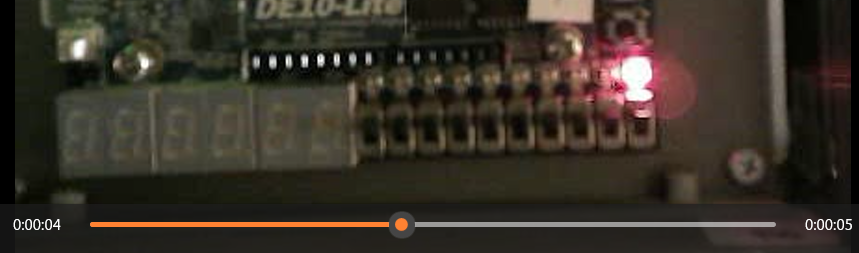


Рисунок 49. d=1, clk=1, Q=1

### 2.2.6 D-триггер

Папка **dff\_sync\_rst**.

Файл верхнего уровня (Рисунок 50).

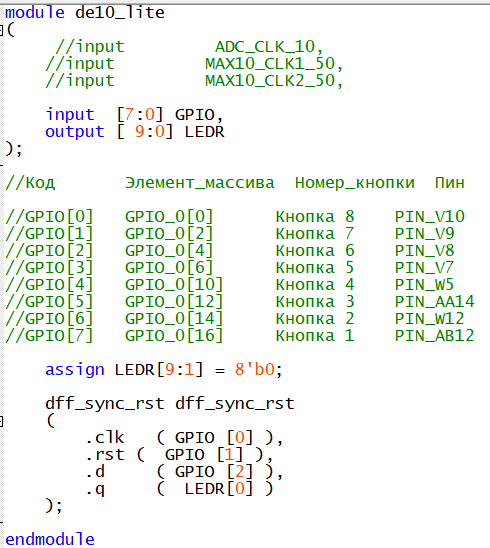


Рисунок 50. Файл верхнего уровня dff\_sync\_rst

Реализация D-триггера с синхронным сигналом очистки (Рисунок 51).

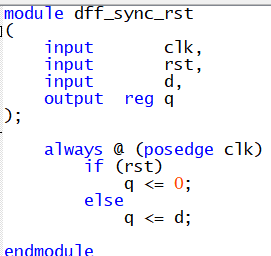


Рисунок 51. Реализация D-триггера с синхронным сигналом очистки

Назначения пинов сохраняются.

clk ( GPIO [0] ) – кнопка 8, rst ( GPIO [1]) – свитч 7, d – свитч 6, q – LEDR [0].

При rst=0 сигнал очистки (не происходит изменений).

Файл сценария (Рисунок 52).

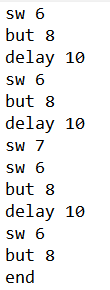


Рисунок 52. Сценарий для D-триггера с синхронным сигналом очистки

Работа удаленного стенда (Рисунок 53–56).

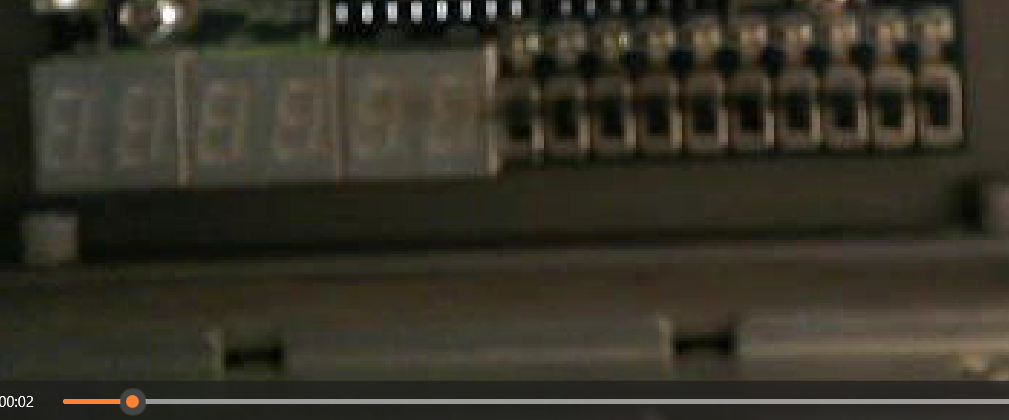


Рисунок 53. rst=0, d=1, clk=1, Q=0

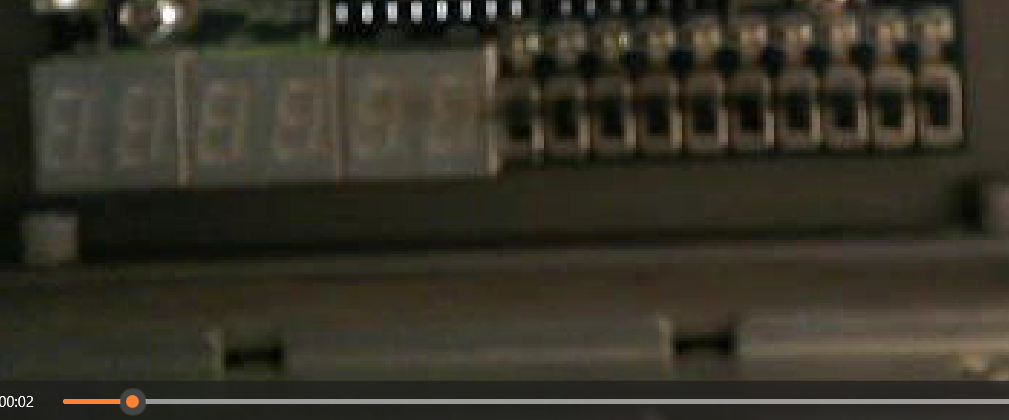


Рисунок 54. rst=0, d=0, clk=1, Q=0

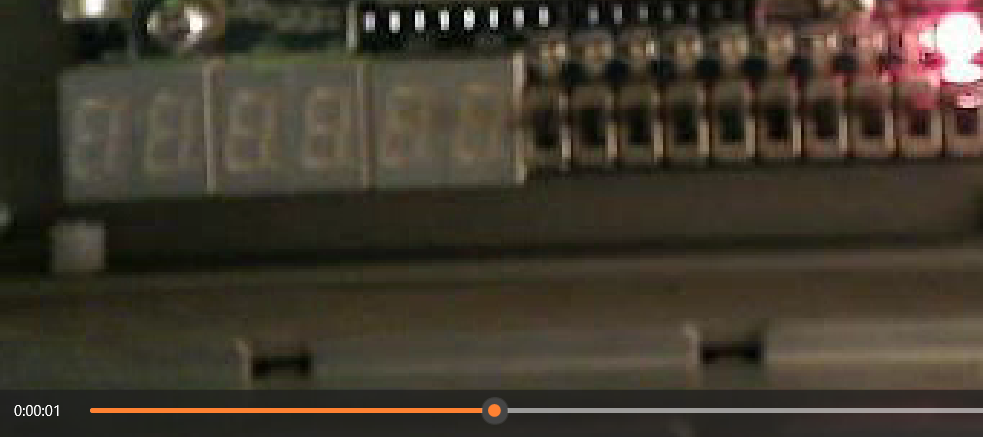


Рисунок 55. rst=1, d=1, clk=1, Q=1

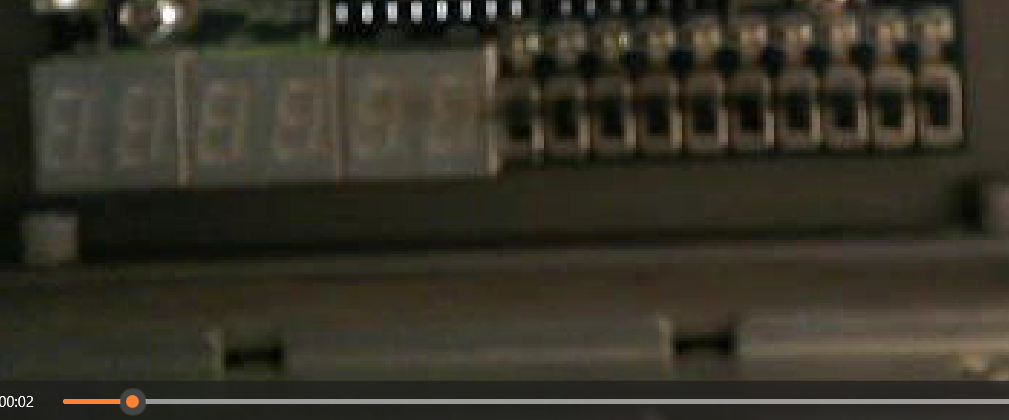


Рисунок 56. rst=1, d=0, clk=1, Q=0

## 2.3 ЛР3. Шифраторы и дешифраторы. Скорость работы комбинационных блок

### 2.3.1 Неприоритетный шифратор

Неприоритетный шифратор 16x4 имеет следующее преобразование кода (Рисунок 57).



Рисунок 57. Таблица истинности для доп. задания 1

При программировании платы удаленного стенда возможно использовать только 8 gpio, то входам a8..a15 были назначены 0.

Реализация неприоритетного шифратора (Рисунок 58).

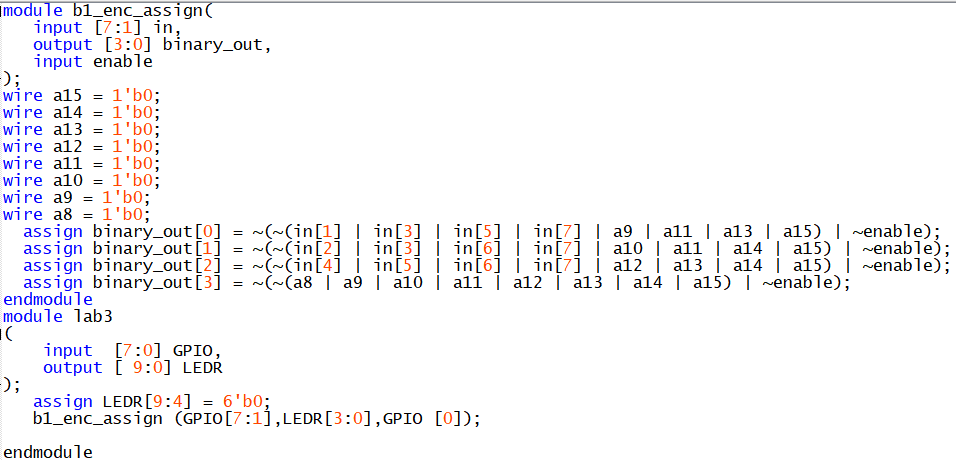


Рисунок 58. Реализация доп. задания 1

Файл сценария (Рисунок 59).

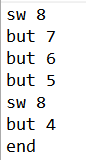


Рисунок 59. Сценарий для доп. задания 1

Работа удаленного стенда (Рисунок 60–63).

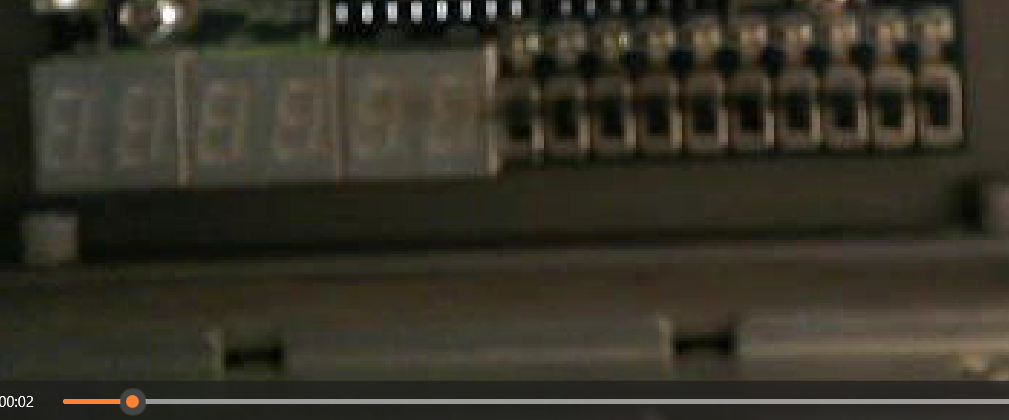


Рисунок 60. enable=1, На входе 0000000000000001, на выходе 0000

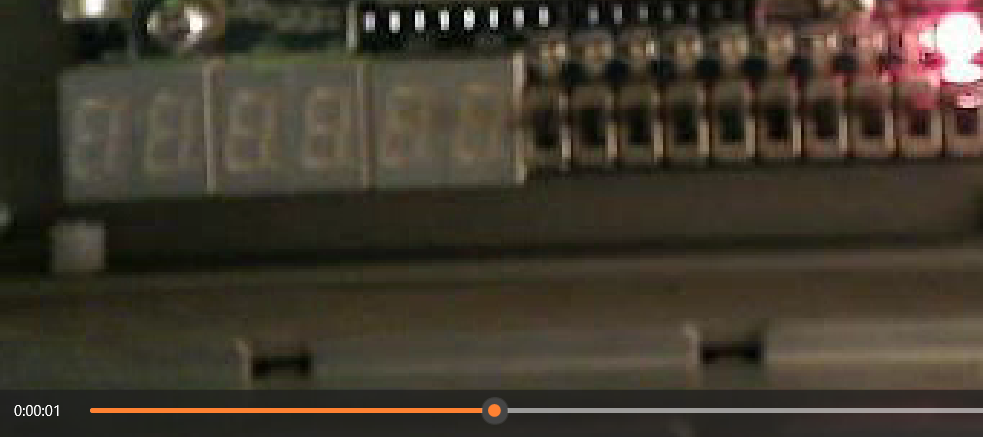


Рисунок 61. enable=1, На входе 0000000000000010, на выходе 0001

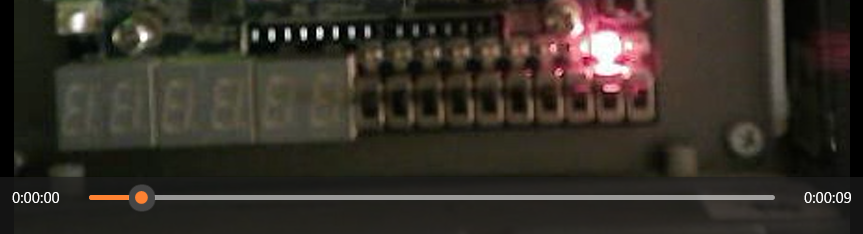


Рисунок 62. enable=1, На входе 0000000000000100, на выходе 0010

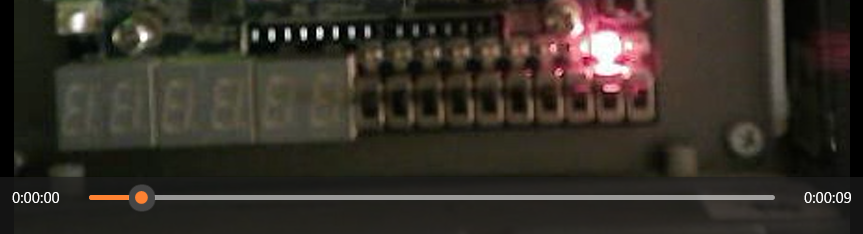


Рисунок 63. enable=0, На входе 0000000000001000, на выходе 0010

### 2.3.2 Приоритетный шифратор

Файл верхнего уровня для приоритетного шифратора, реализованный с помощью операторов if (рисунок 64):

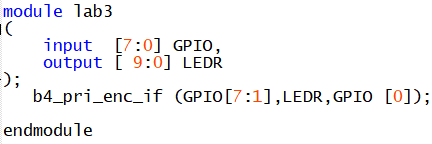


Рисунок 64. Файл верхнего уровня b4\_pri\_enc\_if

Код приоритетного шифратора, реализованный с помощью операторов if :

module b4\_pri\_enc\_if(

input [15:0] encoder\_in,

output reg [3:0] binary\_out,

input enable

);

always @ (\*)

begin

binary\_out = 0;

if (enable) begin

if (encoder\_in[0] == 1'b1)

binary\_out = 0;

else if (encoder\_in[1] == 1'b1)

binary\_out = 1;

else if (encoder\_in[2] == 1'b1)

binary\_out = 2;

else if (encoder\_in[3] == 1'b1)

binary\_out = 3;

else if (encoder\_in[4] == 1'b1)

binary\_out = 4;

else if (encoder\_in[5] == 1'b1)

binary\_out = 5;

else if (encoder\_in[6] == 1'b1)

binary\_out = 6;

else if (encoder\_in[7] == 1'b1)

binary\_out = 7;

else if (encoder\_in[8] == 1'b1)

binary\_out = 8;

else if (encoder\_in[9] == 1'b1)

binary\_out = 9;

else if (encoder\_in[10] == 1'b1)

binary\_out = 10;

else if (encoder\_in[11] == 1'b1)

binary\_out = 11;

else if (encoder\_in[12] == 1'b1)

binary\_out = 12;

else if (encoder\_in[13] == 1'b1)

binary\_out = 13;

else if (encoder\_in[14] == 1'b1)

binary\_out = 14;

else if (encoder\_in[15] == 1'b1)

binary\_out = 15;

end

end

endmodule

Файл сценария (Рисунок 65).

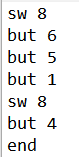


Рисунок 65. Сценарий для b4\_pri\_enc\_if

Приоритетный шифратор с использованием оператора assign, его код:

module b5\_pri\_enc\_assign(

input [15:0] encoder\_in,

output [3:0] binary\_out,

input enable

);

assign binary\_out = (enable) ? (

(encoder\_in[15] == 1'b1) ? 15 :

(encoder\_in[14] == 1'b1) ? 14 :

(encoder\_in[13] == 1'b1) ? 13 :

(encoder\_in[12] == 1'b1) ? 12 :

(encoder\_in[11] == 1'b1) ? 11 :

(encoder\_in[10] == 1'b1) ? 10 :

(encoder\_in[9] == 1'b1) ? 9 :

(encoder\_in[8] == 1'b1) ? 8 :

(encoder\_in[7] == 1'b1) ? 7 :

(encoder\_in[6] == 1'b1) ? 6 :

(encoder\_in[5] == 1'b1) ? 5 :

(encoder\_in[4] == 1'b1) ? 4 :

(encoder\_in[3] == 1'b1) ? 3 :

(encoder\_in[2] == 1'b1) ? 2 :

(encoder\_in[1] == 1'b1) ? 1 :

(encoder\_in[0] == 1'b1) ? 0 : 4'bxxxx):0;

endmodule

В файле верхнего уровня вызывается модуль b5\_pri\_enc\_assign.

Файл сценария (Рисунок 65).

Код шифратора, который указывает на младшую единицу и выполнен с использованием оператора assign:

module b5\_pri\_enc\_assign(

input [15:0] encoder\_in,

output [3:0] binary\_out,

input enable

);

assign binary\_out = (enable) ? (

(encoder\_in[1] == 1'b1) ? 1 :

(encoder\_in[2] == 1'b1) ? 2 :

(encoder\_in[3] == 1'b1) ? 3 :

(encoder\_in[4] == 1'b1) ? 4 :

(encoder\_in[5] == 1'b1) ? 5 :

(encoder\_in[6] == 1'b1) ? 6 :

(encoder\_in[7] == 1'b1) ? 7 :

(encoder\_in[8] == 1'b1) ? 8 :

(encoder\_in[9] == 1'b1) ? 9 :

(encoder\_in[10] == 1'b1) ? 10 :

(encoder\_in[11] == 1'b1) ? 11 :

(encoder\_in[12] == 1'b1) ? 12 :

(encoder\_in[13] == 1'b1) ? 13 :

(encoder\_in[14] == 1'b1) ? 14 :

(encoder\_in[15] == 1'b1) ? 15 :

(encoder\_in[0] == 1'b1) ? 0 : 4'bxxxx):0 ;

endmodule

В файле верхнего уровня вызывается модуль b5\_pri\_enc\_assign.

Файл сценария (Рисунок 65).

Код приоритетного параметрического шифратора из папки XII\_lab3\_hdl\_anybit\_enc:

module b12\_anybit\_enc (in,enc\_out,enable);

parameter OUT\_SIZE = 4; // configure this parameter only to get amount of output bits

parameter IN\_SIZE = 1<<OUT\_SIZE; // input bits are calculated like 2^(output bits)

input wire [IN\_SIZE-1:0] in;

output wire [OUT\_SIZE-1:0] enc\_out;

reg [OUT\_SIZE-1:0] out;

assign enc\_out = out;

input enable;

integer i;

always @(in) begin

if(enable) begin

i=0;

while (i<IN\_SIZE-1 && !in[i]) i=i+1;

out <= i;

end else out<=0;

end

endmodule

В файле верхнего уровня вызывается модуль b12\_anybit\_enc.

Файл сценария (Рисунок 65).

### 2.3.3 Дешифратор

Код для создания параметрического дешифратора со сдвигом находится в папке 7\_lab3\_hdl\_4bit\_dec\_shift (Рисунок 66).

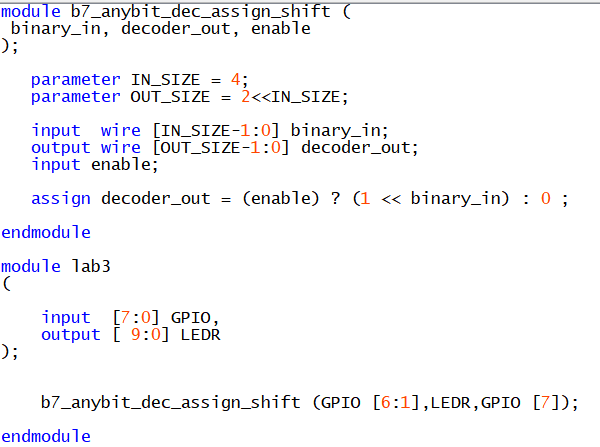


Рисунок 66. Код дешифратора

Файл сценария (Рисунок 67).

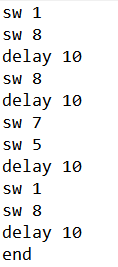


Рисунок 67. Сценарий для дешифратора

### 2.3.4 Самостоятельная работа

*Задание: разработать конвертор кода Грея в унитарный код*

Файл конвертора (Рисунок 68).

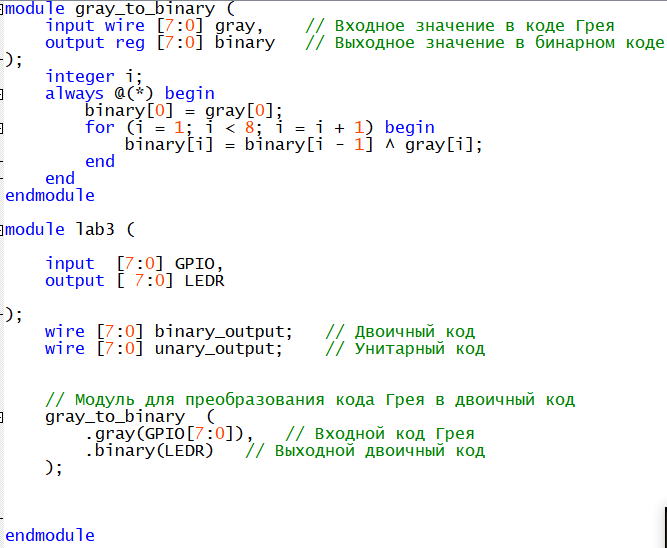


Рисунок 68. Файл конвертора

Файл сценария (Рисунок 69).



Рисунок 68. Файл сценария

# 3 Выводы

В ходе выполнения данной домашней работы было осуществлено знакомство с удаленной лабораторией САПР. Работа была выполнена через Телеграм бота.

При работе с платой ПЛИС в удаленной лаборатории необходимо переназначить входные порты на GPIO порты. Практически для всех входных данных лучше использовать переключатели (switch), а не кнопки (button) для эффективной работы программы. Однако для CLK предпочтительней использовать именно кнопки. Свитчи сохраняют свое положение, пока не будет применима команда еще раз.

В боте хотелось бы сохранения адреса почты, тогда для изменения нужно возвращаться в меню и выбирать соответствующую команду.

# 

# 4 Список литературы

1. User Manual DE10-Lite.
2. Инструкция по работе с удаленным стендом. URL - <https://docs.google.com/document/d/133p6ukIBe1n7TiZjt3gZZhJh8sG4tNq5/edit>.
3. GitHub// PSofya [Электронный ресурс]. Режим доступа:

<https://github.com/PSofya/dz_chip/tree/main> (дата обращения 10.06.2024).