САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

по заданию lab7

Дисциплина

«Технологии проектирования аппаратных средств компьютерных систем»

выполнил:

Курякин Д. А

группа:

преподаватель:

Антонов А. П.

Санкт-Петербург

2022

Оглавление

[1 Задание lab1\_1 4](#_Toc98192533)

[1.1 Задание 4](#_Toc98192534)

[1.2 Описание на языке Verilog 4](#_Toc98192535)

[1.3 Результат синтеза (RTL) 4](#_Toc98192536)

[1.4 Моделирование 4](#_Toc98192537)

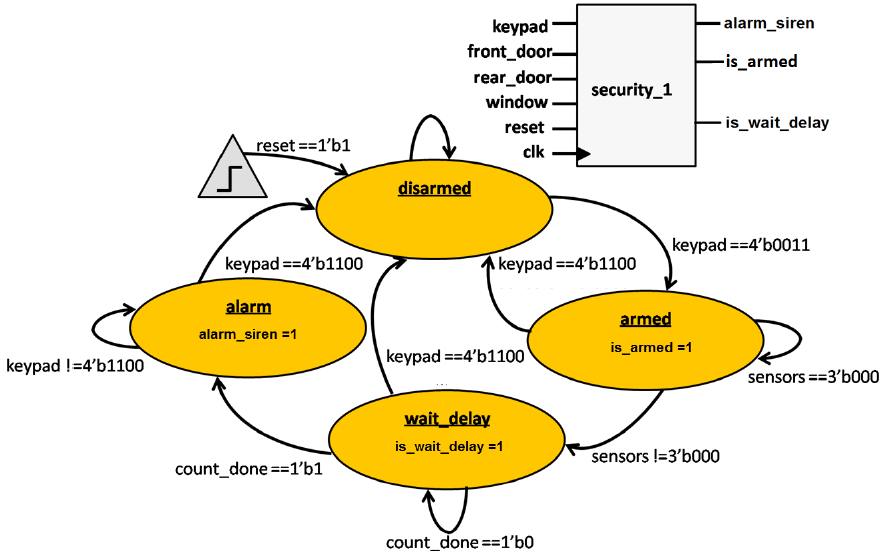
[1.5 Назначение выводов СБИС 5](#_Toc98192538)

[1.6 Тестирование на плате Nexys4 DDR 6](#_Toc98192539)

[1.7 Выводы 6](#_Toc98192540)

# Задание lab7

## Задание



Изображение выглядит как текст

Автоматически созданное описание

## Описание на языке System Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже листингах 1–2.

Листинг 1. lab7\_1.sv

|  |
| --- |
| `timescale 1ns / 1ps  module lab7\_1(input logic clk);  bit front\_door;  bit rear\_door;  bit window;  bit clk;  bit reset;  bit ena;  bit [3:0] keypad;  bit alarm\_siren;  bit is\_armed;  bit is\_wait\_delay;  security\_verilog SV(.clk(clk),  .alarm\_siren(alarm\_siren),  .is\_armed(is\_armed),  .is\_wait\_delay(is\_wait\_delay),  .front\_door(front\_door),  .rear\_door(rear\_door),  .window(window),  .reset(reset),  .ena(ena),  .keypad(keypad));  vio\_0 VIO(.clk(clk),  .probe\_in0(alarm\_siren),  .probe\_in1(is\_armed),  .probe\_in2(is\_wait\_delay),  .probe\_out0(front\_door),  .probe\_out1(rear\_door),  .probe\_out2(window),  .probe\_out3(reset),  .probe\_out4(ena),  .probe\_out5(keypad));  ila\_0 ILA(.clk(clk),  .probe0(alarm\_siren),  .probe1(is\_armed),  .probe2(is\_wait\_delay));  endmodule |

Листинг 2. security\_verilog.sv

|  |
| --- |
| `timescale 1ns / 1ps  module security\_verilog(  input logic front\_door,  input logic rear\_door,  input logic window,  input logic clk,  input logic reset,  input logic ena,  input logic [3:0] keypad,  output logic alarm\_siren,  output logic is\_armed,  output logic is\_wait\_delay  );  //integer MAX = 10000000;  integer MAX = 4;  logic [26:0] c = '0;  logic clk\_10gz = '0;  parameter delay\_val = 100;  logic start\_count;  logic count\_done;  logic [6:0] delay\_cntr = 0;  localparam disarmed = 2'd0,  armed = 2'd1,  wait\_delay = 2'd2,  alarm = 2'd3;  logic [1:0] curr\_state, next\_state;  logic [2:0] sensors;  assign sensors = {front\_door, rear\_door, window};  always\_ff @ (posedge clk\_10gz)  if (ena)  if (reset)  curr\_state <= disarmed;  else  curr\_state <= next\_state;    always\_ff @ (curr\_state, sensors, keypad, count\_done, ena) begin  if (ena)  case (curr\_state)  disarmed: begin  if (keypad == 4'b0011)  next\_state <= armed;  else  next\_state <= curr\_state;  end    armed: begin  if (sensors != 3'b000)  next\_state <= wait\_delay;  else if (keypad == 4'b1100)  next\_state <= disarmed;  else  next\_state <= curr\_state;  end    wait\_delay: begin  if (count\_done == 1'b1)  next\_state <= alarm;  else if (keypad == 4'b1100)  next\_state <= disarmed;  else  next\_state <= curr\_state;  end    alarm: begin  if (keypad == 4'b1100)  next\_state <= disarmed;  else  next\_state <= curr\_state;  end  endcase  end    always\_ff @ (posedge clk\_10gz) begin  if (ena)  if (reset) begin  is\_armed <= 1'b0;  is\_wait\_delay <= 1'b0;  alarm\_siren <= 1'b0;  end else begin  is\_armed <= (next\_state == armed);  is\_wait\_delay <= (next\_state == wait\_delay);  alarm\_siren <= (next\_state == alarm);  end  end  assign start\_count = ((curr\_state == armed) && (sensors != 3'b000));  always\_ff @ (posedge clk\_10gz) begin  if (ena)  if (reset)  delay\_cntr <= 0;  else if (start\_count)  delay\_cntr <= delay\_val - 1'b1;  else if (curr\_state != wait\_delay)  delay\_cntr <= 0;  else if (delay\_cntr != 0)  delay\_cntr <= delay\_cntr - 1'b1;  end    assign count\_done = (delay\_cntr == 0);  always\_ff @(posedge clk)  begin  if(c == MAX) begin  c = 0;  clk\_10gz = 1'b1;  end else begin  c = c + 1'b1;  clk\_10gz = 1'b0;  end  end  endmodule |

## Результат синтеза (RTL)

Результат синтеза описания на языке Verilog в пакете Quartus приведен ниже, на рис 1-1.

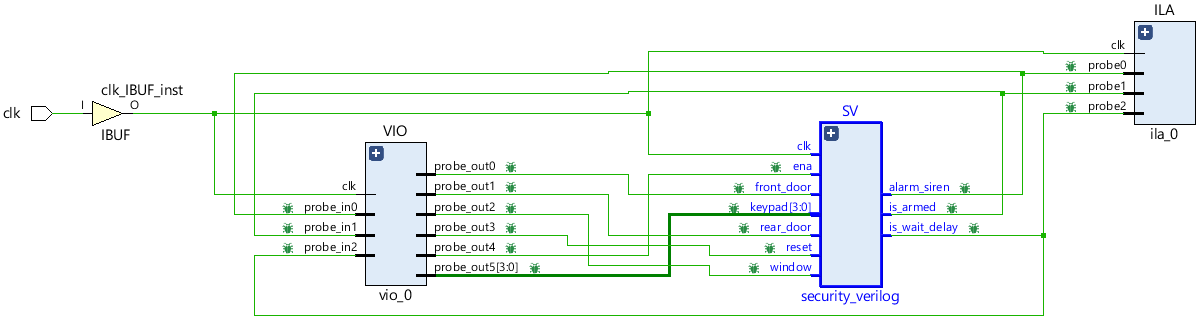


Рис. ‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты представленные в листингt 3:

Листинг 3. tb\_security\_verilog.sv

|  |
| --- |
| `timescale 1ns / 1ps  module tb\_security\_verilog();  bit front\_door = 0;  bit rear\_door = 0;  bit window = 0;  bit clk = 0;  bit reset = 0;  bit ena = 1;  bit [3:0] keypad = 0;  bit alarm\_siren = 0;  bit is\_armed = 0;  bit is\_wait\_delay = 0;  security\_verilog SV(.\*);  always #10 clk=~clk;  initial begin  #20;  reset = 1;  #100;  reset = 0;  keypad = 4'b0011;  #200;  keypad = 4'b1100;  #100;  keypad = 4'b0011;  #200;  front\_door = 1;  #200;  keypad = 4'b1100;  front\_door = 0;  #100;  keypad = 4'b0011;  #100;  rear\_door = 1;  #10250;  keypad = 4'b1100;  #200;  $stop;  end  endmodule |

Результаты моделирования приведены на Рис. 1‑2

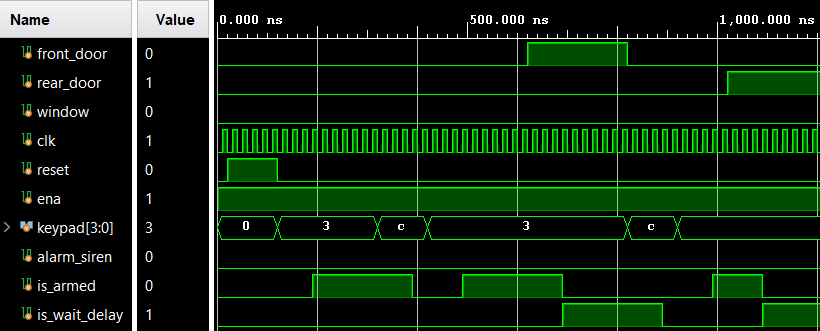


Рис. ‑2 Результат моделирования tb\_lfsr

## Назначение выводов СБИС

Назначение выводов СБИС и стандартов ввода вывода приведено на Рис. 1‑3

Изображение выглядит как стол

Автоматически созданное описание

Рис. ‑3 Назначение выводов

## Тестирование на плате Nexys4 DDR

Для тестирования проекта на плате использовались тесты, описанные в разделе 1.4. Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

## Выводы

В результате было реализовано устройство, описанное в лабораторной работе.