САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

по заданию lab7

Дисциплина

«Языки описания аппаратных средств вычислительных систем»

выполнил:

Курякин Д. А

группа: 3540901/12001

преподаватель:

Антонов А. П.

Санкт-Петербург

2021

Оглавление

[1 Задание lab7\_1 3](#_Toc89552645)

[1.1 Задание 3](#_Toc89552646)

[1.2 Описание на языке Verilog 3](#_Toc89552647)

[1.3 Результат синтеза 4](#_Toc89552648)

[1.4 Моделирование 4](#_Toc89552649)

[1.5 Назначение выводов СБИС 5](#_Toc89552650)

[1.6 Тестирование на плате Nexys4 DDR 6](#_Toc89552651)

[1.7 Выводы 6](#_Toc89552652)

[2 Задание lab7\_2 7](#_Toc89552653)

[2.1 Задание 7](#_Toc89552654)

[2.2 Описание на языке Verilog 8](#_Toc89552655)

[2.3 Результат синтеза 11](#_Toc89552656)

[2.4 Моделирование 11](#_Toc89552657)

[2.5 Назначение выводов СБИС 15](#_Toc89552658)

[2.6 Тестирование на плате Nexys4 DDR 16](#_Toc89552659)

[2.7 Выводы 16](#_Toc89552660)

[3 Задание lab6\_3 17](#_Toc89552661)

[3.1 Задание 17](#_Toc89552662)

[3.2 Описание на языке Verilog 17](#_Toc89552663)

[3.3 Результат синтеза 18](#_Toc89552664)

[3.4 Моделирование 18](#_Toc89552665)

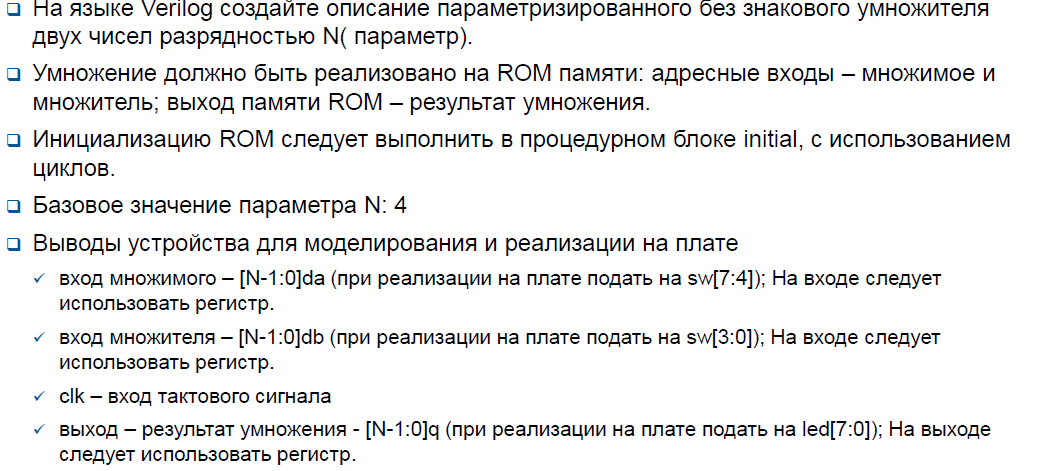
[3.5 Назначение выводов СБИС 19](#_Toc89552666)

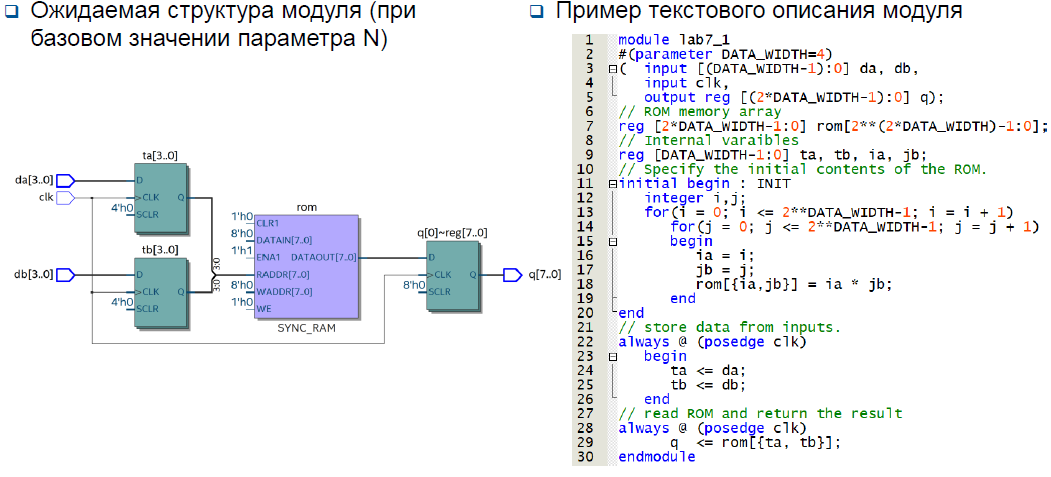
[3.6 Тестирование на плате Nexys4 DDR 19](#_Toc89552667)

[3.7 Выводы 20](#_Toc89552668)

# Задание lab7\_1

## Задание





## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже в листинге 1.

Листинг 1. Lab7\_1.v

|  |
| --- |
| module lab7\_1  #(parameter DATA\_WIDTH=4)  ( input [(DATA\_WIDTH-1):0] da, db,  input clk,  output reg [(2\*DATA\_WIDTH-1):0] q);  reg [2\*DATA\_WIDTH-1:0] rom[2\*\*(2\*DATA\_WIDTH)-1:0];  reg [DATA\_WIDTH-1:0] ta, tb, ia, jb;  initial begin : INIT  integer i, j;  for(i=0; i <= 2\*\*DATA\_WIDTH-1; i = i + 1)  for(j=0; j <= 2\*\*DATA\_WIDTH-1; j = j + 1)  begin  ia = i;  jb = j;  rom[{ia,jb}] = ia \* jb;  end  end  always @ (posedge clk)  begin  ta <= da;  tb <= db;  end  always @ (posedge clk)  q <= rom[{ta, tb}];    endmodule |

## Результат синтеза

Результат синтеза описания на языке Verilog в пакете Quartus приведен ниже, на Рис. 1‑1.

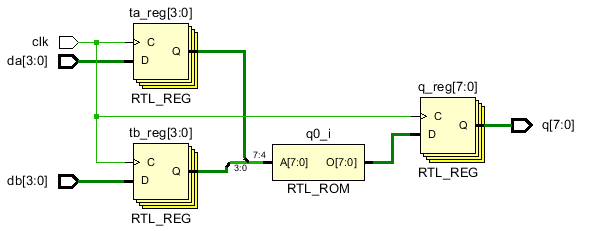


Рис. 1‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовался тест представленный в листинге 2:

Листинг 2. Tb7\_1.v

|  |
| --- |
| module tb7\_1();  parameter DATA\_WIDTH=4;  localparam period = 10;  integer i;  reg clk;  reg [(DATA\_WIDTH-1):0] da, db;  wire [(2\*DATA\_WIDTH-1):0] q;  lab7\_1 LAB(.clk(clk), .da(da), .db(db), .q(q));  initial begin  clk = 1'b0;  for(i = 1; i < 15; i = i + 1) begin  da = i;  db = i;  clk = ~clk; #period; clk = ~clk; #period;  end  end  endmodule |

Результаты моделирования приведены на Рис. 1‑2

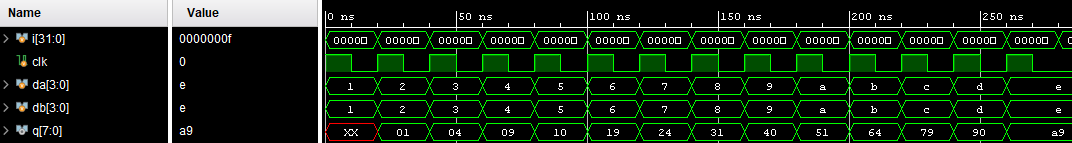


Рис. 1‑2 Результат моделирования

## Назначение выводов СБИС

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

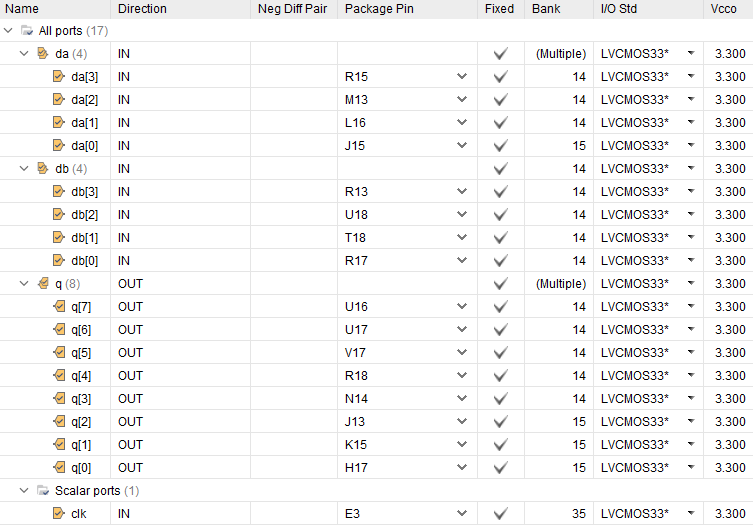


Рис. 1‑3 Назначение выводов

## Тестирование на плате Nexys4 DDR

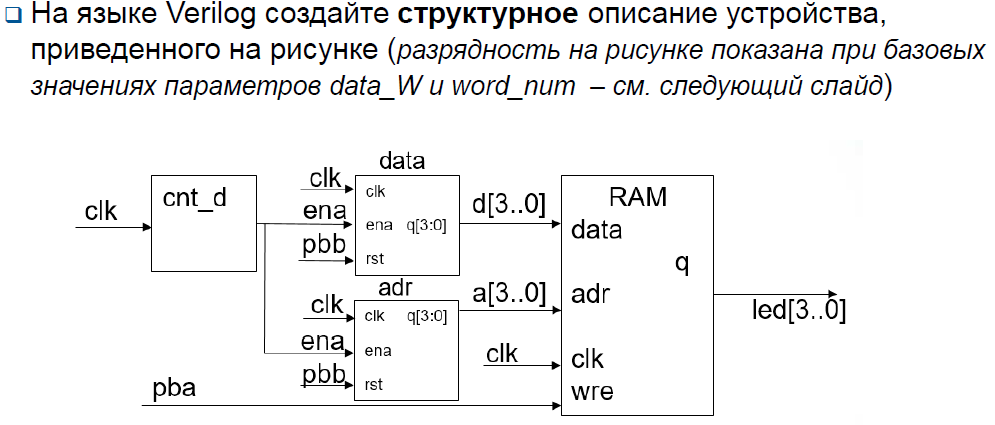
Для тестирования проекта на плате использовались тесты, описанные в разделе 1.4. Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

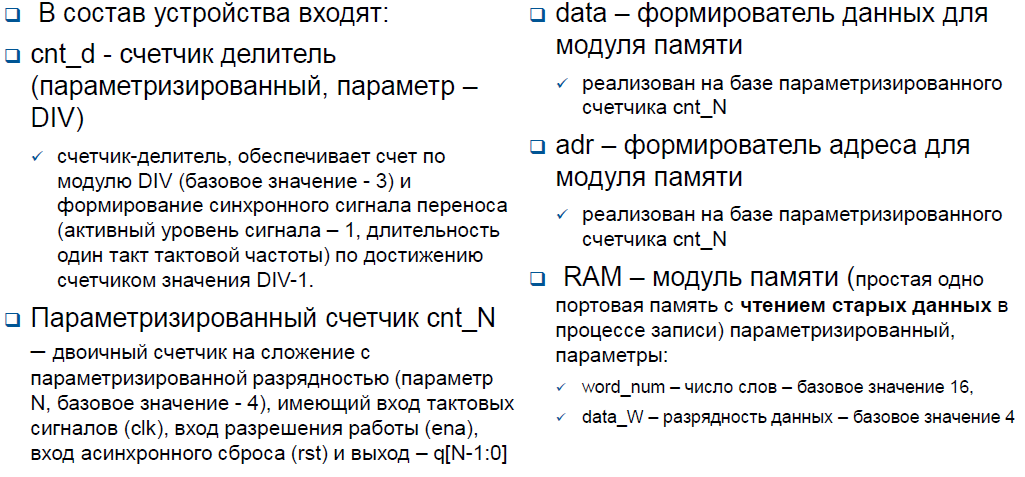
## Выводы

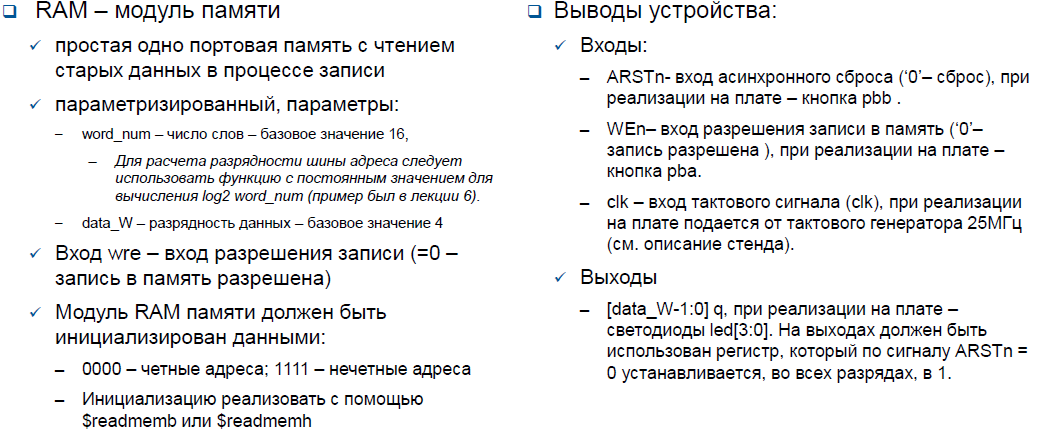
В результате было реализовано параметризированный без знаковый умножитель двух чисел разрядностью N.

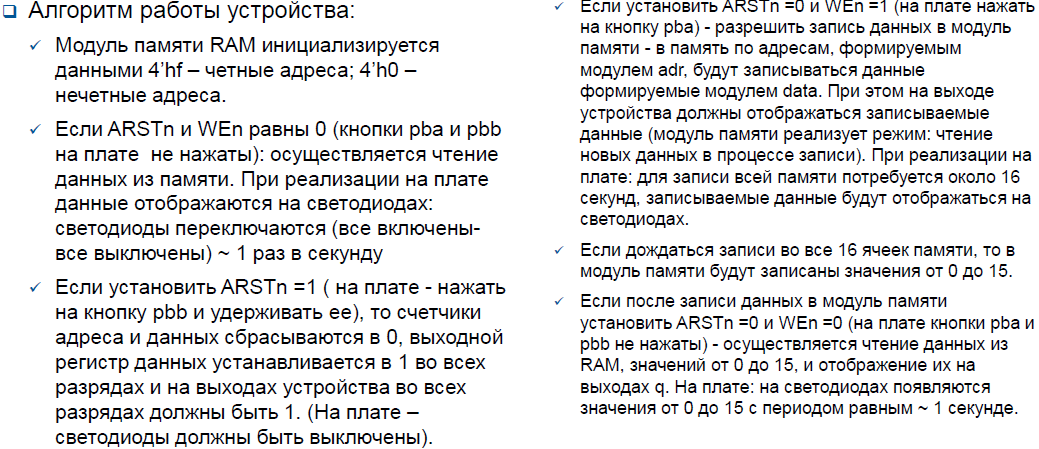
# Задание lab7\_2

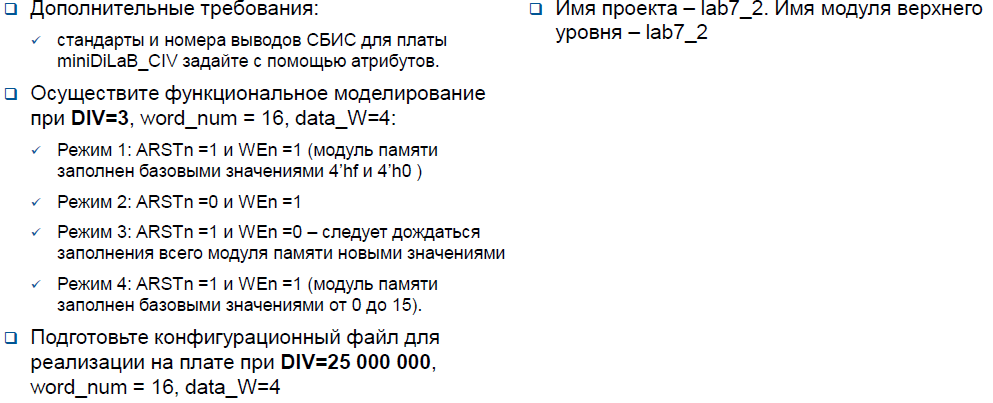
## Задание











## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже в листингах 3-7.

Листинг 3. Lab7\_2.v

|  |
| --- |
| module lab7\_2(  input clk,  input bt1,  input bt2,  output [data\_W - 1:0] led  );  parameter DIV = 25000000;  parameter N = 4;  parameter word\_num = 16;  parameter data\_W = 8;    wire ena;  wire [N-1 : 0] d, a;  cnt\_d #(DIV) cnt\_rg (clk, ena);  data #(N) cnt\_data (clk, ena, bt2, d);  adr #(N) cnt\_adr (clk, ena, pb2, a);  store #(word\_num, data\_W) st (clk, bt1, d, a, led);  endmodule |

Листинг 4. cnt\_d.v

|  |
| --- |
| module cnt\_d(  input clk,  output reg ena  );  parameter DIV = 3;    reg [27:0] tmp;  initial begin  tmp = 28'h0;  end    always @ (posedge clk)  begin  tmp <= tmp + 1'b1;  if (tmp == DIV - 1)  begin  tmp <= 1'b0;  ena <= 1'b1;  end else  ena <= 1'b0;  end  endmodule |

Листинг 5. data.v

|  |
| --- |
| module data (  input clk, ena, bt,  output reg [N-1:0] d  );  parameter N = 16;    initial begin  d = 4'h0;  end    always @ (posedge clk or posedge bt)  if (bt)  d <= 4'h0;  else if (ena) begin  d <= d + 1'b1;  end  endmodule |

Листинг 6. adr.v

|  |
| --- |
| module adr (  input clk, ena, bt,  output reg [N-1:0] q  );  parameter N = 16;    initial begin  q = 4'h0;  end    always @ (posedge clk or posedge bt)  if (bt)  q <= 4'h0;  else if (ena) begin  q <= q + 4'h1;  end  endmodule |

Листинг 7. store.v

|  |
| --- |
| module store #(  parameter WORD\_NUM = 16,  parameter DATA\_W = 8)(  input clk, wre,  input [WORD\_NUM-1 : 0] data, adr,  output [DATA\_W-1 : 0] led  );  reg [DATA\_W-1:0] list [WORD\_NUM:0];  integer i;  initial  $readmemh("mem.mem", list);  always @ (posedge clk)  begin  if (wre) begin  list[adr] <= data;  end  end  assign led = list[adr];  endmodule |

## Результат синтеза

Результат синтеза описания на языке Verilog приведен ниже на Рис. 2‑1.

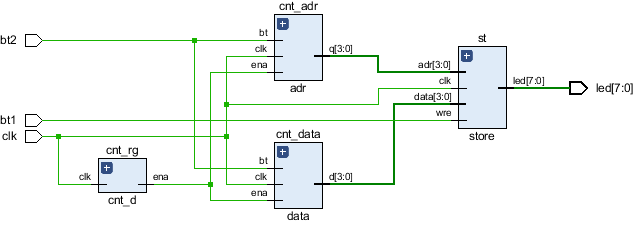


Рис. 2‑1 Синтезированная схема

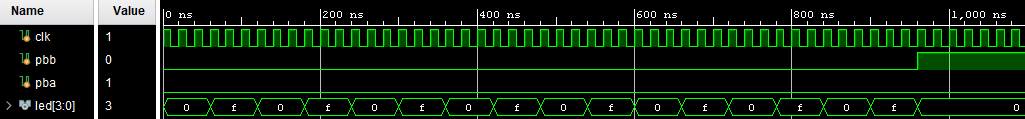
## Моделирование

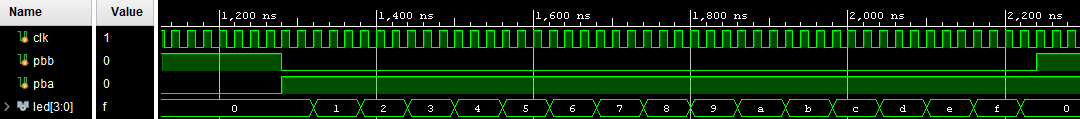
Для проверки правильности работы созданного Verilog описания использовались тесты, представленный в листингах 8-11:

Листинг 8. tb7\_2.v

|  |
| --- |
| `timescale 1ns / 1ps  module tb7\_2();  parameter DIV = 3;  parameter N = 16;  parameter WORD\_NUM = 16;  parameter DATA\_W = 4;  localparam period = 10;  reg clk;  reg pbb;  reg pba;  wire [DATA\_W - 1:0] led;  integer i;  lab7\_2 #(DIV, N, WORD\_NUM, DATA\_W) lb(clk, pba, pbb, led);  initial begin  clk = 1'b0;  pba = 1'b0;  pbb = 1'b0;    for(i = 0; i < 16\*DIV; i = i + 1) begin  clk = ~clk;#period; clk = ~clk;#period;  end  pbb = 1'b1;  for(i = 0; i < 16; i = i + 1) begin  clk = ~clk;#period; clk = ~clk;#period;  end  pba = 1'b1;  pbb = 1'b0;  for(i = 0; i < 16\*DIV; i = i + 1) begin  clk = ~clk;#period; clk = ~clk;#period;  end  pba = 1'b1;  pbb = 1'b1;  for(i = 0; i < 16; i = i + 1) begin  clk = ~clk;#period; clk = ~clk;#period;  end  pba = 1'b0;  pbb = 1'b0;  for(i = 0; i < 16; i = i + 1) begin  clk = ~clk;#period; clk = ~clk;#period;  end  end  endmodule |

Результаты моделирования приведены на Рис. 2-2.





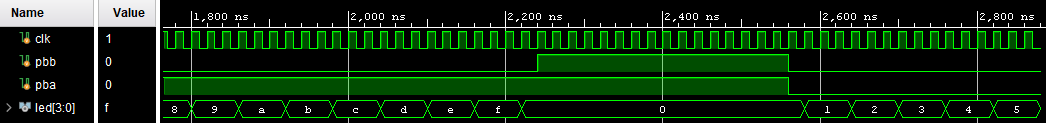


Рис. 2‑2 Результат моделирования

Листинг 9. cnt\_d\_tb.v

|  |
| --- |
| module cnt\_d\_tb();  parameter DIV = 6;  localparam period = 10;  reg clk;  wire ena;  integer i;  cnt\_d #(DIV) cnt(.clk(clk), .ena(ena));  initial begin  clk = 1'b0;  for(i = 0; i < 100; i = i + 1) begin  clk = ~clk;#period;clk = ~clk;#period;  end  end  endmodule |

Результаты моделирования приведены на Рис. 2-3.

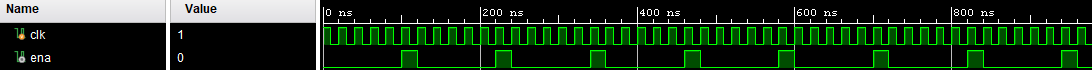


Рис. 2‑3 Результат моделирования

Листинг 10. adr\_tb.v

|  |
| --- |
| module adr\_tb();  parameter N = 4;  localparam period = 10;  reg clk, ena, bt;  wire [N-1:0] q, d;  integer i;  adr #(N) ADR(.clk(clk), .ena(ena), .bt(bt), .q(q));  data #(N) DATA(.clk(clk), .ena(ena), .bt(bt), .d(d));  initial begin  clk = 1'b0;  ena = 1'b1;  bt = 1'b0;    for(i = 0; i < 5; i = i + 1) begin  clk = ~clk;#period;clk = ~clk;#period;  end  bt = 1'b1;  clk = ~clk;#period;clk = ~clk;#period;  bt = 1'b0;    for(i = 0; i < 100; i = i + 1) begin  clk = ~clk;#period;clk = ~clk;#period;  end  end  endmodule |

Результаты моделирования приведены на Рис. 2-4.

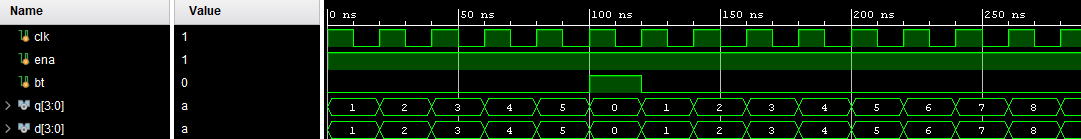


Рис. 2‑4 Результат моделирования

Листинг 11 tb\_store.v

|  |
| --- |
| module tb\_store();  parameter word\_num = 16;  parameter data\_W = 8;  localparam period = 10;  reg clk, wre;  reg [word\_num-1 : 0] data, adr;  wire [data\_W-1 : 0] led;  integer i;  store #(word\_num, data\_W) st(clk, wre, data, adr, led);  initial begin  clk = 1'b0;  wre = 1'b1;  adr = 3'h5;  data = 8'h55;  clk = ~clk;#period;clk = ~clk;#period;    wre = 1'h1;  adr = 3'h6;  data = 8'haa;  clk = ~clk;#period;clk = ~clk;#period;  wre = 1'h0;  data = 8'h0;    for(i = 0; i < 100; i = i + 1) begin  adr = i;  clk = ~clk;#period;clk = ~clk;#period;  end  end  endmodule |

Результаты моделирования приведены на Рис. 2-5.

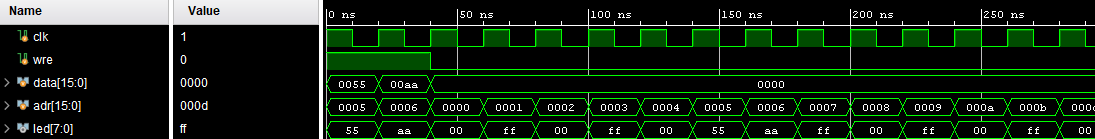


Рис. 2‑5 Результат моделирования

## Назначение выводов СБИС

Назначение выводов СБИС и стандартов ввода вывода приведено на Рис. 2‑6

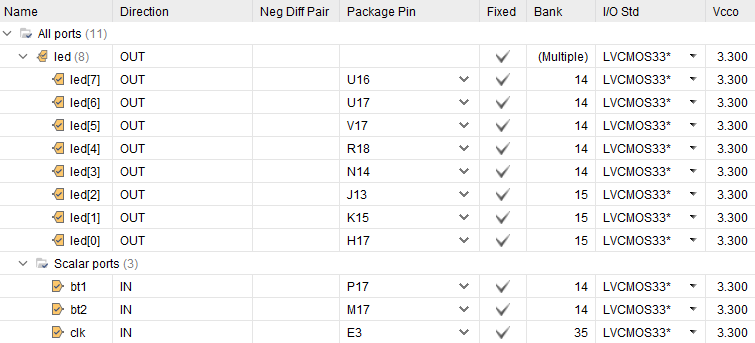


Рис. 2‑6 Назначение выводов

## Тестирование на плате Nexys4 DDR

Для тестирования проекта на плате использовались тесты, описанные в разделе 2.4

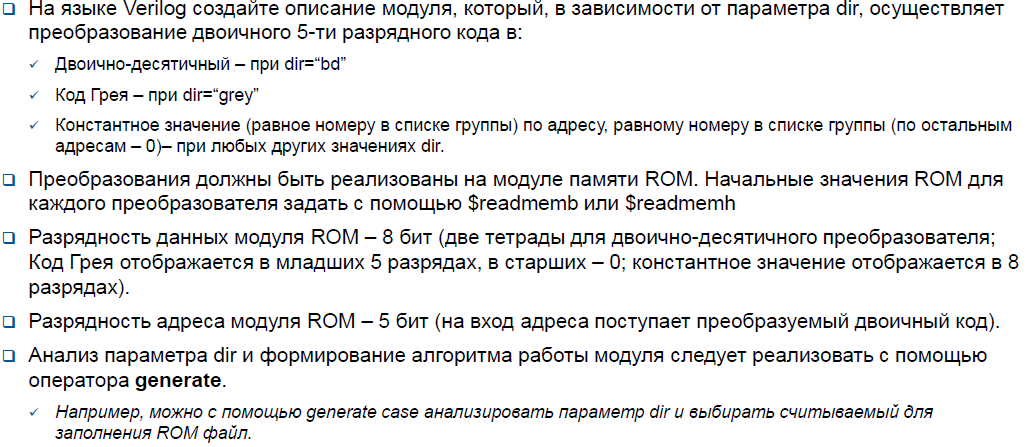
Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

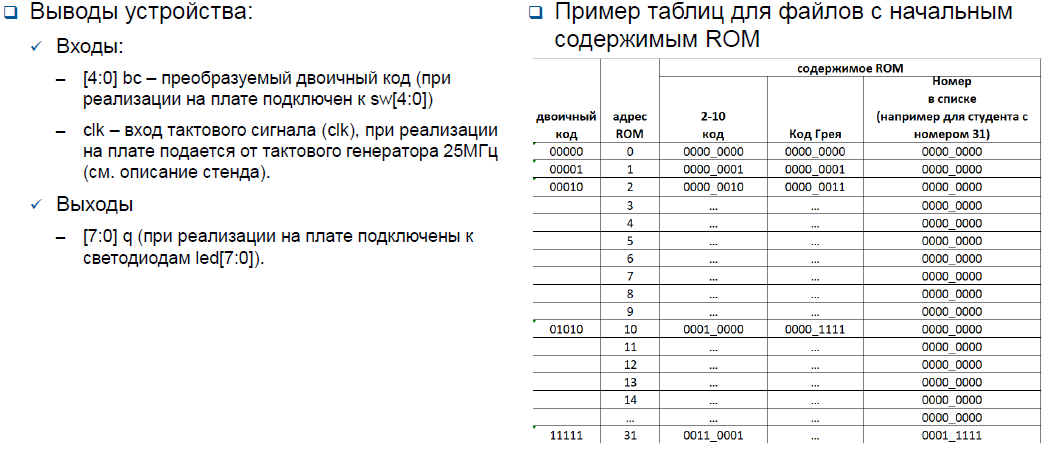
## Выводы

В результате было реализовано устройство, описанное в задание.

# Задание lab6\_3

## Задание





## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже листинге 5.

Листинг 5. lab7\_3.v

|  |
| --- |
| `timescale 1ns / 1ps  module lab7\_3 #(  parameter DIR = "const",  parameter DATA\_W = 8,  parameter N = 5)(  input [N-1:0] sw,  input clk,  output [DATA\_W-1:0] led  );    reg [DATA\_W-1:0] lists[(2\*\*N)-1:0];    generate  always @(posedge clk)  begin  if(DIR == "grey") begin : g\_grey  $readmemb("grey.mem", lists); end  if(DIR == "bd") begin : g\_bd  $readmemh("bd.mem", lists); end  if(DIR == "const") begin : g\_const  $readmemh("consts.mem", lists); end  end  endgenerate  assign led = lists[sw];  endmodule |

## Результат синтеза

Результат синтеза описания на языке Verilog приведен ниже на Рис. 3‑1.

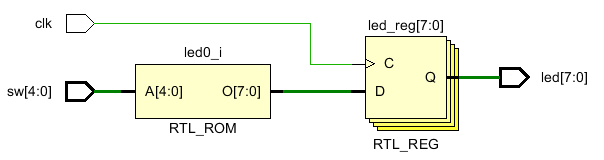


Рис. 3‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовался тест, представленный в листинге 6:

Листинг 6. tb7\_3.v

|  |
| --- |
| module tb7\_3();  parameter DIR = "bd";  parameter DATA\_W = 8;  parameter N = 5;  localparam period = 10;  reg clk;  reg [7:0] sw;  wire [7:0] led;  integer i;  lab7\_3 #(DIR, DATA\_W, N) LAB(.clk(clk), .sw(sw), .led(led));  initial begin  clk = 1'h0;  for(i = 0; i < 2\*\*N; i = i + 1) begin  sw = i;  #period;clk = ~clk; #period;clk = ~clk;  end  end  endmodule |

Результаты моделирования приведены на Рис. 3-2.

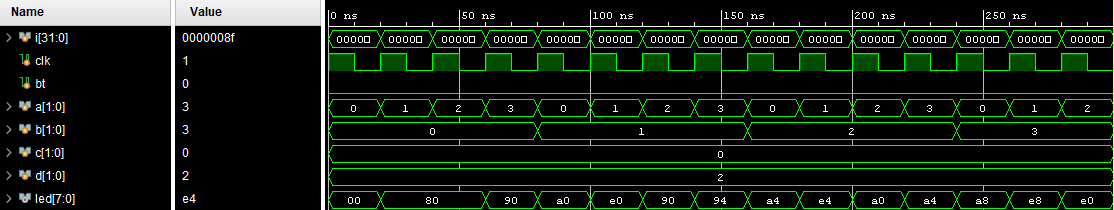


Рис. 3‑2 Результат моделирования

## Назначение выводов СБИС

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

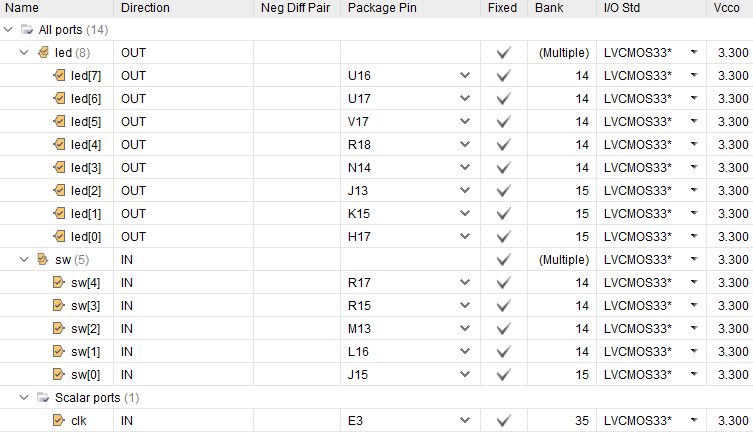


Рис. 3‑3 Назначение выводов

## Тестирование на плате Nexys4 DDR

Для тестирования проекта на плате использовались тесты, описанные в разделе 3.4

Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

## Выводы

В результате реализовано устройство которое, в зависимости от параметра dir, осуществляет преобразование двоичного 5-ти разрядного кода в код Грея, 2-10 код или константу.