САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

по заданию labs1-6

Дисциплина

«Языки описания аппаратных средств вычислительных систем»

выполнил:

Курякин Д. А

группа: 3540901/12001

преподаватель:

Антонов А. П.

Санкт-Петербург

2021

Оглавление

[1 Задание labs\_1 4](#_Toc90344436)

[1.1 Задание 4](#_Toc90344437)

[1.2 Описание на языке Verilog 4](#_Toc90344438)

[1.3 Результат синтеза 4](#_Toc90344439)

[1.4 Моделирование 5](#_Toc90344440)

[1.5 Назначение выводов СБИС 8](#_Toc90344441)

[1.6 Тестирование на плате Nexys4 DDR 8](#_Toc90344442)

[1.7 Выводы 8](#_Toc90344443)

[2 Задание labs\_2 9](#_Toc90344444)

[2.1 Задание 9](#_Toc90344445)

[2.2 Описание на языке Verilog 9](#_Toc90344446)

[2.3 Результат синтеза 9](#_Toc90344447)

[2.4 Моделирование 10](#_Toc90344448)

[2.5 Назначение выводов СБИС 11](#_Toc90344449)

[2.6 Тестирование на плате Nexys4 DDR 12](#_Toc90344450)

[2.7 Выводы 12](#_Toc90344451)

[3 Задание labs\_3 13](#_Toc90344452)

[3.1 Задание 13](#_Toc90344453)

[3.2 Описание на языке Verilog 13](#_Toc90344454)

[3.3 Результат синтеза 13](#_Toc90344455)

[3.4 Моделирование 14](#_Toc90344456)

[3.5 Назначение выводов СБИС 16](#_Toc90344457)

[3.6 Тестирование на плате Nexys4 DDR 17](#_Toc90344458)

[3.7 Выводы 17](#_Toc90344459)

[4 Задание labs\_4 18](#_Toc90344460)

[4.1 Задание 18](#_Toc90344461)

[4.2 Описание на языке Verilog 18](#_Toc90344462)

[4.3 Результат синтеза 18](#_Toc90344463)

[4.4 Моделирование 18](#_Toc90344464)

[4.5 Назначение выводов СБИС 20](#_Toc90344465)

[4.6 Тестирование на плате Nexys4 DDR 21](#_Toc90344466)

[4.7 Выводы 21](#_Toc90344467)

[5 Задание labs\_5 22](#_Toc90344468)

[5.1 Задание 22](#_Toc90344469)

[5.2 Описание на языке Verilog 22](#_Toc90344470)

[5.3 Результат синтеза 23](#_Toc90344471)

[5.4 Моделирование 23](#_Toc90344472)

[5.5 Назначение выводов СБИС 25](#_Toc90344473)

[5.6 Тестирование на плате Nexys4 DDR 26](#_Toc90344474)

[5.7 Выводы 26](#_Toc90344475)

[6 Задание labs\_6 27](#_Toc90344476)

[6.1 Задание 27](#_Toc90344477)

[6.2 Описание на языке Verilog 27](#_Toc90344478)

[6.3 Результат синтеза 28](#_Toc90344479)

[6.4 Моделирование 28](#_Toc90344480)

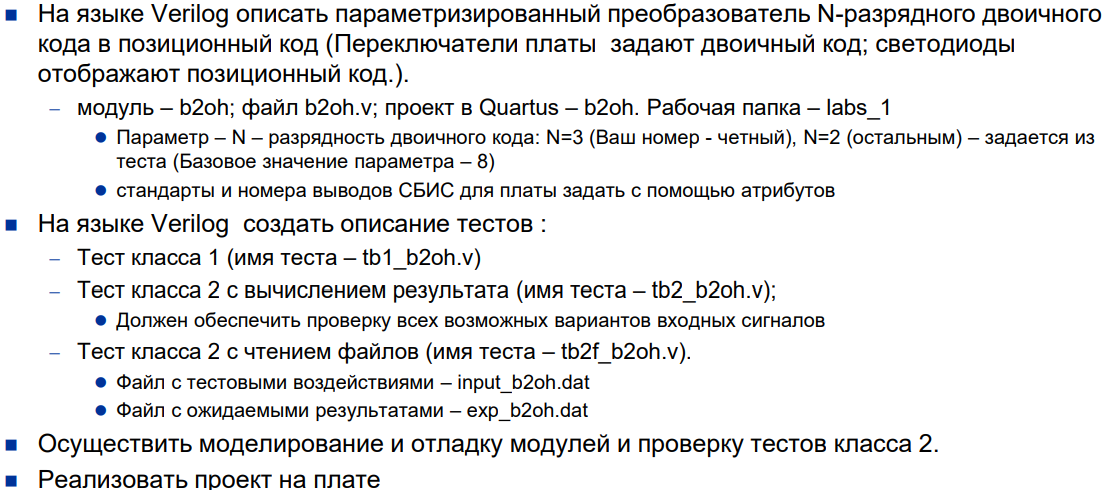
[6.5 Назначение выводов СБИС 32](#_Toc90344481)

[6.6 Тестирование на плате Nexys4 DDR 33](#_Toc90344482)

[6.7 Выводы 33](#_Toc90344483)

# Задание labs\_1

## Задание



## Описание на языке Verilog

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

Листинг 1-1. Labs\_1.v

|  |
| --- |
| module b2oh #(  parameter N = 3)(  input [N-1:0] sw,  output [(2\*\*N)-1:0] led  );  assign led = 1 << sw;  endmodule |

## Результат синтеза

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

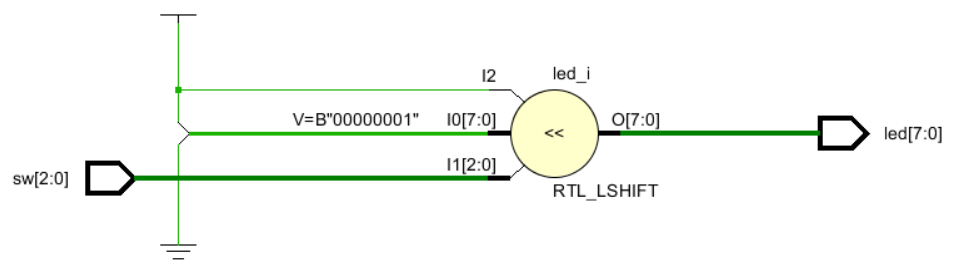


Рис. 1‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты:

Листинг 1-2. tb1\_b2oh.v

|  |
| --- |
| `timescale 1ns / 1ps  module tb1\_b2oh();  localparam delay = 20;  parameter N = 3;  reg [N-1:0] sw;  wire [(2\*\*N)-1:0] led;  b2oh #(N) DUT(sw, led);  initial begin  for(sw = 0; sw < 2\*\*N; sw = sw + 1'b1)  #delay;  $display("Test is successful");  end  initial begin  $display("\t\t TIME binary position");  $monitor($time,,,,,,sw,,,,,,led);  end  endmodule |

Результаты моделирования приведены на Рис. 1‑2



Рис. ‑2 Результат моделирования

Листинг 1-3. Tb2\_b2oh.v

|  |
| --- |
| `timescale 1ns / 1ps  module tb2\_b2oh();  localparam delay = 20;  parameter N = 3;  reg [N-1:0] sw;  wire [(2\*\*N)-1:0] led;  b2oh #(N) DUT(sw, led);  initial begin  test(0, 1);  test(1, 2);  test(2, 4);  test(3, 8);  test(4, 16);  test(5, 32);  test(6, 64);  test(7, 128);  $display("Test is successful");  end  task test;  input [N - 1 : 0] binary;  input [2\*\*N - 1 : 0] expected;  begin  sw = binary;  #delay if (led != expected)  begin  $display("Test is failed, expected result is %d, actual is %d", expected, led);  $stop;  end  end  endtask  initial begin  $display("\t\t TIME binary position");  $monitor($time,,,,,,sw,,,,,,led);  end  endmodule |

Результаты моделирования приведены на Рис. 1‑3

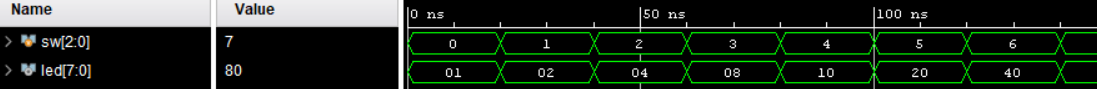


Рис. ‑3 Результат моделирования

Листинг 1-4. Tb2f\_b2oh.v

|  |
| --- |
| `timescale 1ns / 1ps  module tb2f\_b2oh();  localparam delay = 20;  parameter N = 3;  integer i;  reg [N-1:0] sw;  wire [(2\*\*N)-1:0] led;  reg [N-1:0] in [0:2\*\*N - 1];  reg [(2\*\*N)-1:0] exp [0:2\*\*N - 1];  b2oh #(N) DUT(sw, led);  initial begin  for(i = 0; i < 2\*\*N; i = i + 1)  test(in[i], exp[i]);  $display("Test is successful");  end  task test;  input [N - 1 : 0] binary;  input [2\*\*N - 1 : 0] expected;  begin  sw = binary;  #delay if (led != expected)  begin  $display("Test is failed, expected result is %d, actual is %d", expected, led);  $stop;  end  end  endtask  initial begin  $display("\t\t TIME binary position");  $monitor($time,,,,,,sw,,,,,,led);  end  initial begin  $readmemb("input\_b2oh.mem", in);  $readmemb("exp\_b2oh.mem", exp);  end  endmodule |

Результаты моделирования приведены на Рис. 1‑4

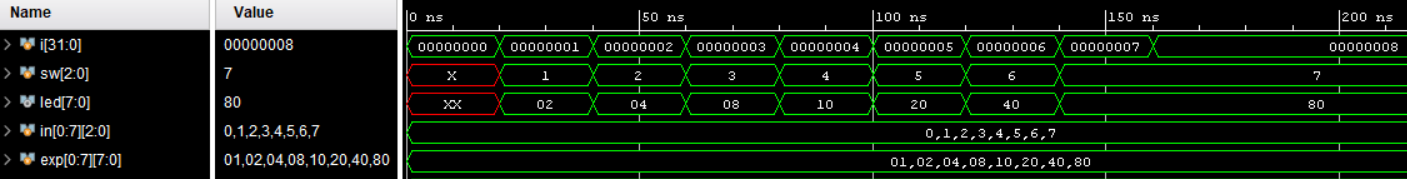


Рис. ‑4 Результат моделирования

## Назначение выводов СБИС

Назначение выводов СБИС и стандартов ввода вывода приведено на Рис. 1‑5

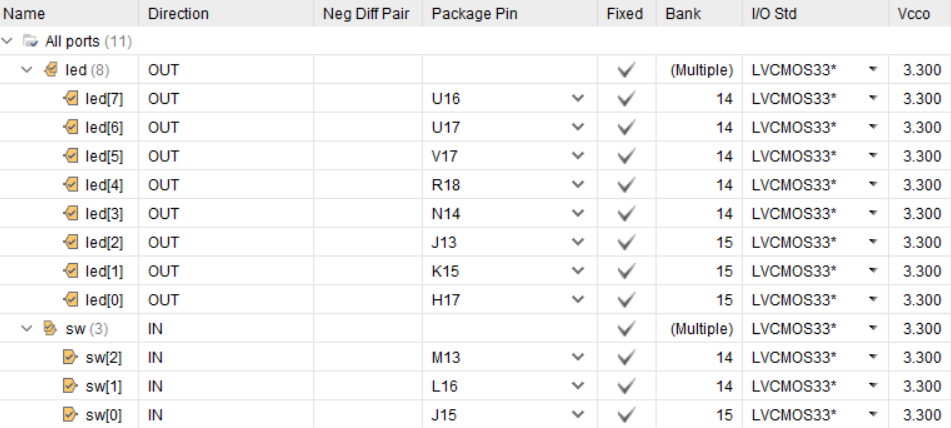


Рис. ‑5 Назначение выводов

## Тестирование на плате Nexys4 DDR

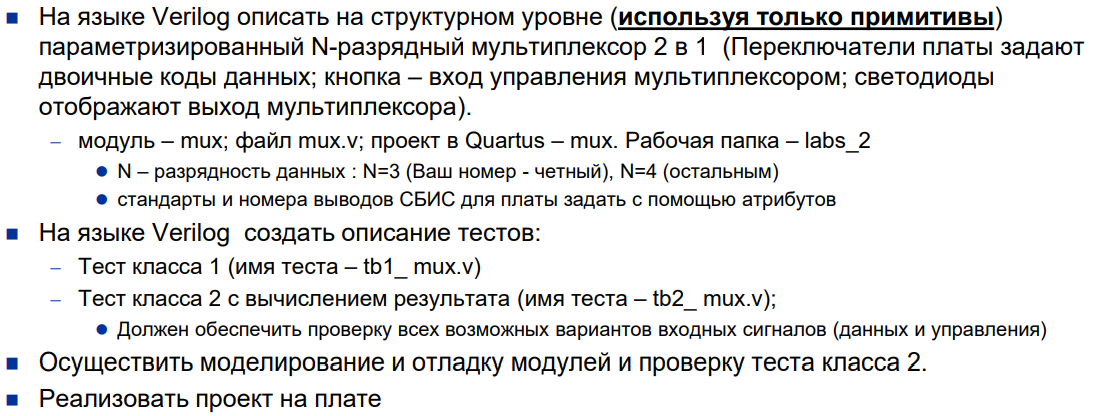
Для тестирования проекта на плате использовались тесты, описанные в разделе 1.4. Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

## Выводы

В результате было реализовано устройство, описанное в задании, и были написаны тесты, требуемые в задании.

# Задание labs\_2

## Задание



## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже в листинге 2-1.

Листинг 2-1. mux.v

|  |
| --- |
| module mux #(  parameter N = 4  )(  input [N-1:0] sw1,  input [N-1:0] sw2,  input bt,  output [N-1:0] led  );    wire [N-1:0] a, b;  wire not\_bt;    not not1 (not\_bt, bt);  and and1 [N-1:0] (a, bt, sw1);  and and2 [N-1:0] (b, not\_bt, sw2);  or or1 [N-1:0] (led, a, b);  endmodule |

## Результат синтеза

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

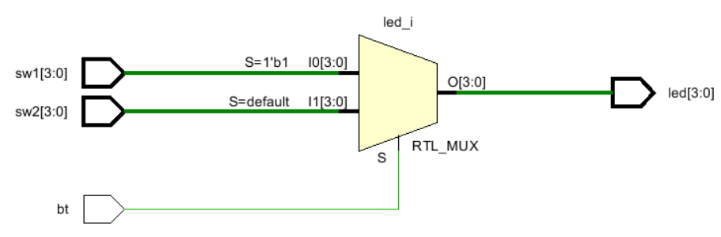


Рис. ‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты:

Листинг 2-2. tb1\_mux.v

|  |
| --- |
| module tb1\_mux();  localparam delay = 20;  parameter N = 4;  reg [N-1:0] sw1, sw2;  reg bt;  wire [N-1:0] led;  integer i;  mux #(N) LAB(sw1, sw2, bt, led);  initial begin  for(i = 16; i < 64; i = i + 1) begin  bt = 0; {sw2, sw1} = i; #delay;  bt = 1; #delay;  end  $display("Test is successful");  end  initial begin  $display("\t\t TIME binary position");  $monitor($time,,,,,,sw2,,,,,,sw1,,,,,,bt,,,,,,led);  end  endmodule |

Результаты моделирования приведены на Рис. 2-2.

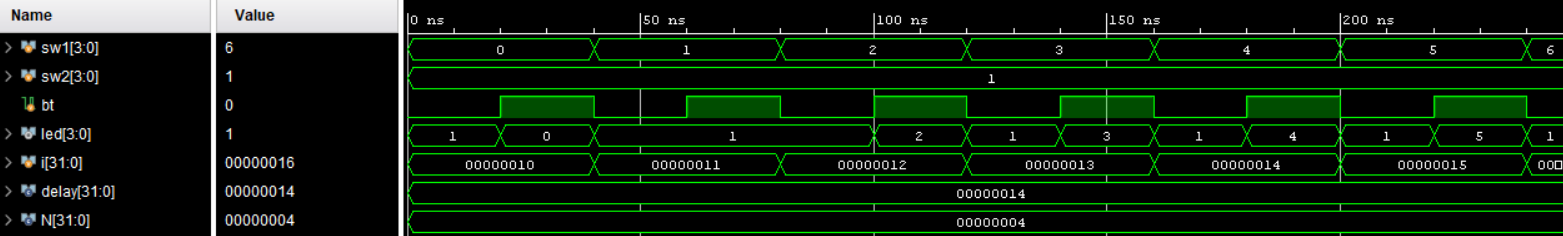


Рис. ‑2 Результат моделирования

Листинг 2-3. tb1\_mux.v

|  |
| --- |
| module tb1\_mux();  localparam delay = 20;  parameter N = 4;  reg [N-1:0] sw1, sw2;  reg bt;  wire [N-1:0] led;  mux #(N) LAB(sw1, sw2, bt, led);  initial begin  test(4, 8, 0, 8);  test(5, 0, 1, 5);  test(0, 2, 0, 2);  test(0, 7, 1, 0);  test(1, 1, 0, 1);  test(3, 1, 1, 3);  $display("Test is successful");  end  initial begin  $display("\t\t TIME binary position");  $monitor($time,,,,,,sw2,,,,,,sw1,,,,,,bt,,,,,,led);  end |

Результаты моделирования приведены на Рис. 2-3.

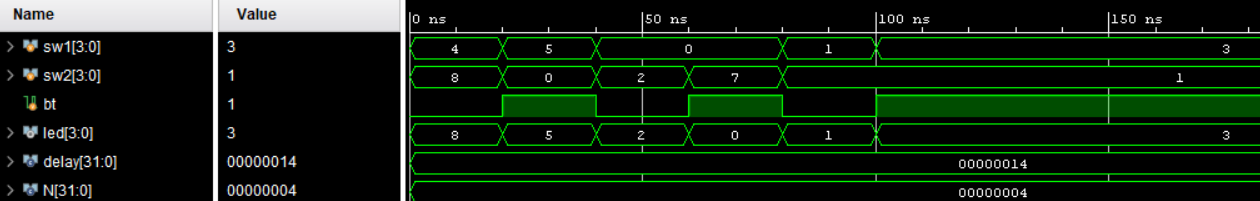


Рис. ‑3 Результат моделирования

## Назначение выводов СБИС

Назначение выводов СБИС и стандартов ввода вывода приведено на Рис. 2‑4

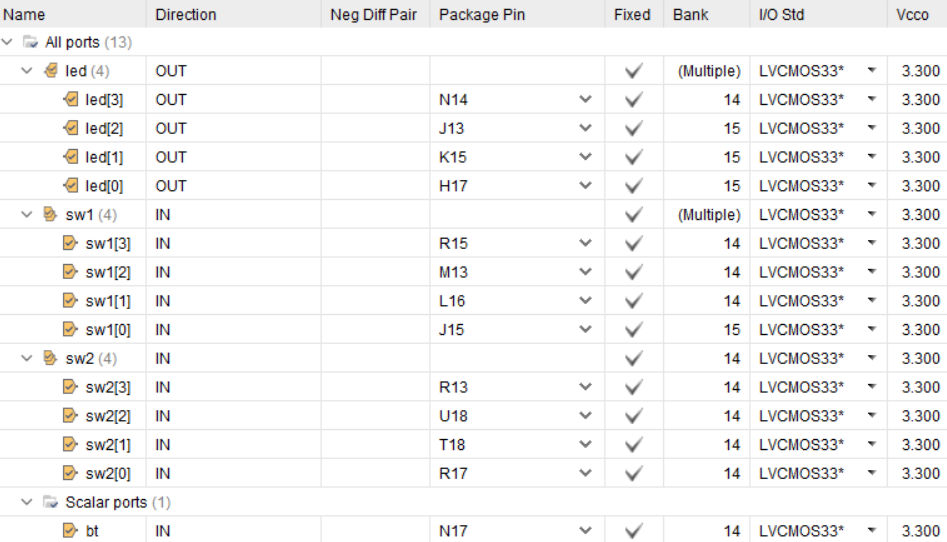


Рис. ‑4 Назначение выводов

## Тестирование на плате Nexys4 DDR

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

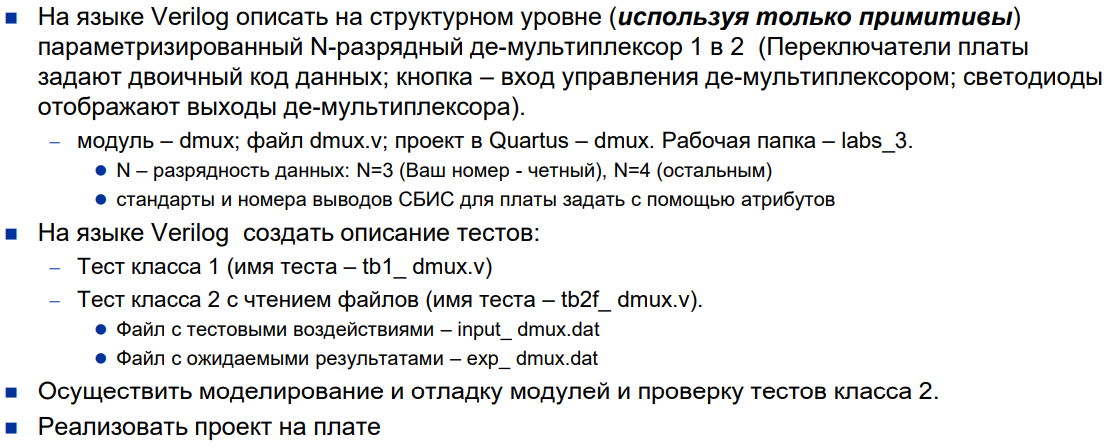
Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

## Выводы

В результате было реализовано устройство, описанное в задании, и были написаны тесты, требуемые в задании.

# Задание labs\_3

## Задание



## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже листинге 3-1.

Листинг 3-1. dmux.v

|  |
| --- |
| `timescale 1ns / 1ps  module dmux  #(parameter N = 4)  (input sel,  input [N-1:0] data,  output [N-1:0] a,  output [N-1:0] b);  wire not\_sel;  not not1 (not\_sel, sel);  and and1 [N-1:0] (a, not\_sel, data);  and and2 [N-1:0] (b, sel, data);  endmodule |

## Результат синтеза

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

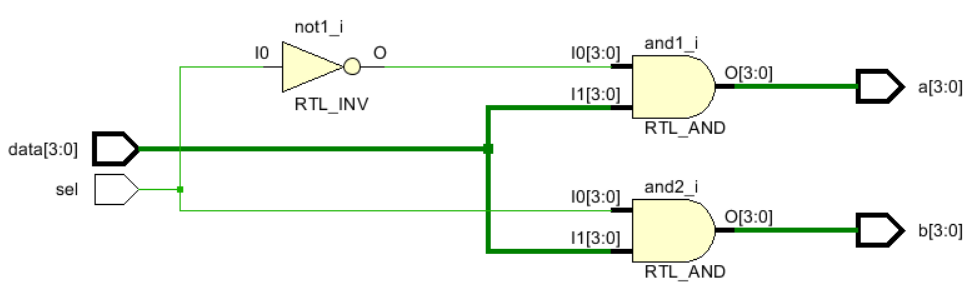


Рис. 3‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты:

Листинг 3-2. tb1\_dmux.v

|  |
| --- |
| module tb1\_dmux();  localparam N = 4;  reg [N-1:0] data;  wire [N-1:0] a;  wire [N-1:0] b;  reg sel;  dmux #(N) DUT (sel, data, a, b);  initial begin  sel = 0; data = 7;  #10 sel = 1; data = 7;  #10 sel = 0; data = 1;  #10 sel = 1; data = 1;  #10 $display("Test is over");  end  initial begin  $display("\t\t TIME SEL DATA A B");  $monitor($time,,,,,,sel,,,,,,data,,,,,,a,,,,,,b);  end  endmodule |

Результаты моделирования приведены на Рис. 3-2.

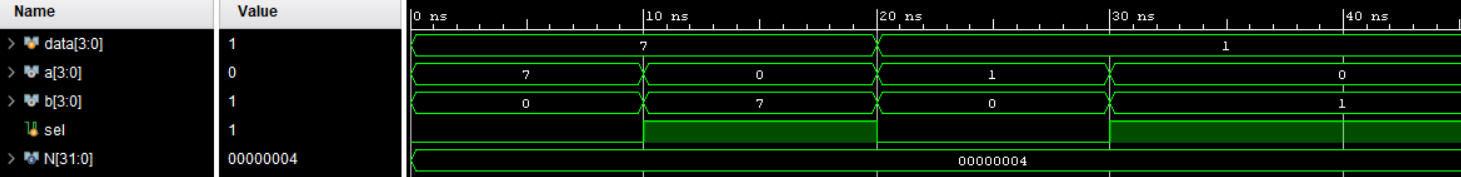


Рис. 3‑2 Результат моделирования

Листинг 3-3. Tb2f\_dmux.v

|  |
| --- |
| `timescale 1ns / 1ps  module tb2f\_dmux();  localparam N = 4;  reg [N-1:0] data;  reg sel;  wire [N-1:0] a;  wire [N-1:0] b;  reg [N-1:0] in [0:7];  reg [N-1:0] exp [0:7];  dmux #(N) fun (sel, data, a, b);  initial begin  $display("Test start");  $readmemh("input\_ dmux.mem", in);  $readmemh("exp\_ dmux.mem", exp);  test(in[0], in[1], exp[0], exp[1]);  test(in[2], in[3], exp[2], exp[3]);  test(in[4], in[5], exp[4], exp[5]);  test(in[6], in[6], exp[6], exp[7]);  $display("Test is successful");  end  task test;  input in\_sel;  input [N-1:0] in\_data;  input [N-1:0] expected\_a;  input [N-1:0] expected\_b;  begin  data = in\_data;  sel = in\_sel;  #1 if (expected\_a != a || expected\_b != b)  begin  $display("Test is failed");  if (expected\_a != a)  begin  $display("expected a is %d, actual a is %d", expected\_a, a);  end  if (expected\_b != b)  begin  $display("expected b is %d, actual b is %d", expected\_b, b);  end  $stop;  end  end  endtask  initial begin  $display("\t\t TIME SEL DATA A B");  $monitor($time,,,,,,sel,,,,,,data,,,,,,a,,,,,,b);  end  endmodule |

Результаты моделирования приведены на Рис. 3-3.

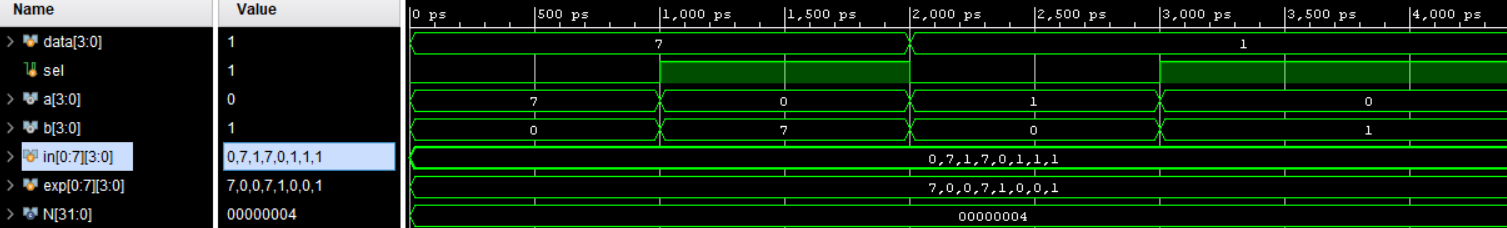


Рис. 3‑3 Результат моделирования

## Назначение выводов СБИС

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

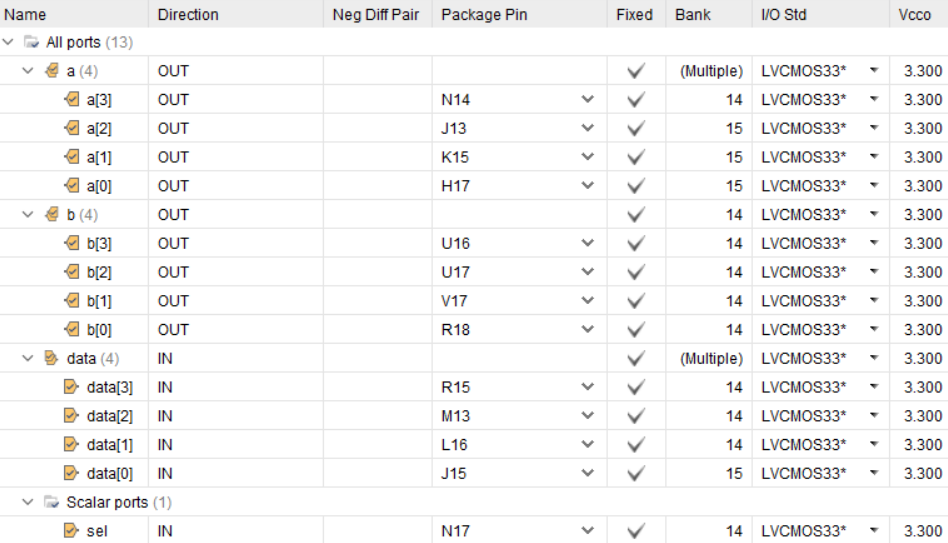


Рис. 3‑3 Назначение выводов

## Тестирование на плате Nexys4 DDR

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

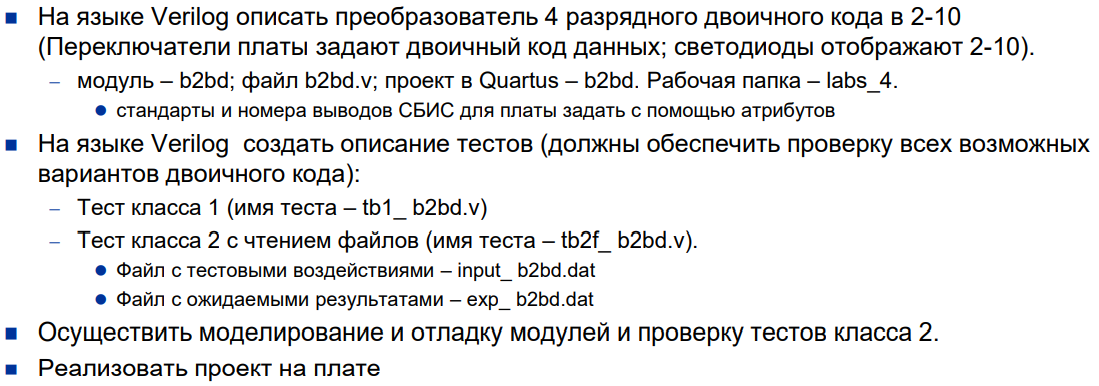
Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

## Выводы

В результате было реализовано устройство, описанное в задании, и были написаны тесты, требуемые в задании.

# Задание labs\_4

## Задание



## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже листинге 3-1.

Листинг 4-1. b2bd.v

|  |
| --- |
| module b2bd(  input [3:0] sw,  output [7:0] led  );  assign led = (sw < 10) ? sw : (sw+6);  endmodule |

## Результат синтеза

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

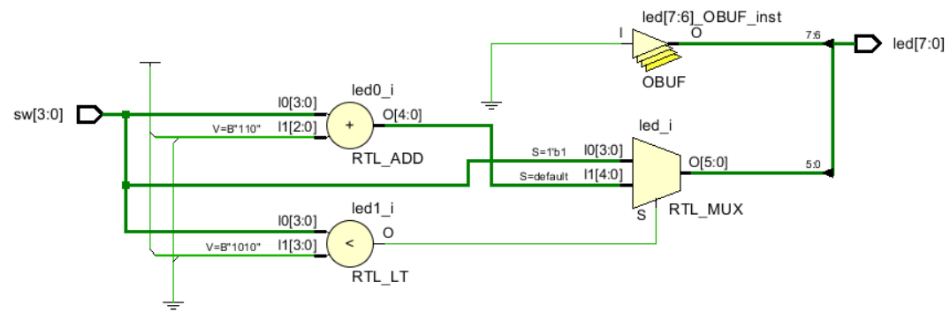


Рис. 4‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты:

Листинг 4-2. tb1\_b2bd.v

|  |
| --- |
| module tb1\_b2bd();  reg [3:0] data;  wire [7:0] result;  b2bd fun (data, result);  integer i;  initial begin  data = 0;  for (i=1; i<17; i=i+1)  #10 data = i;  end  initial begin  $display("\t\t TIME B BD");  $monitor($time,,,,,,data,,,,,,result);  end  endmodule |

Результаты моделирования приведены на Рис. 4-2.

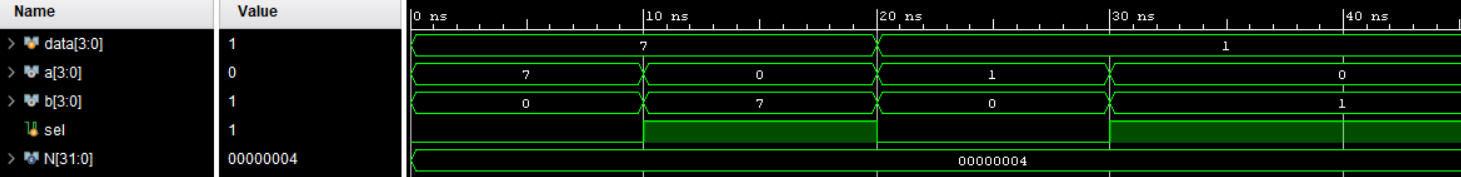


Рис. 4‑2 Результат моделирования

Листинг 4-3. Tb2f\_b2bd.v

|  |
| --- |
| `timescale 1 ns / 1 ns  module tb2f\_b2bd();  reg [3:0] data;  wire [7:0] result;  reg [3:0] in [0:15];  reg [7:0] exp [0:15];  b2bd fun (data, result);  integer i;  initial begin  $readmemh("input\_b2bd.mem", in);  $readmemh("exp\_b2bd.mem", exp);  for (i=1; i<17; i=i+1)  test(in[i], exp[i]);  $display("Test is successful");  end  task test;  input [3:0] in\_data;  input [7:0] expected;  begin  data = in\_data;  #10 if (result != expected)  begin  $display("Test is failed, expected is %b, actual is %b", expected, result);  $stop;  end  end  endtask  initial begin  $display("\t\t TIME B BD");  $monitor("%t %d %8b", $time, data, result);  end  endmodule |

Результаты моделирования приведены на Рис. 4-3.

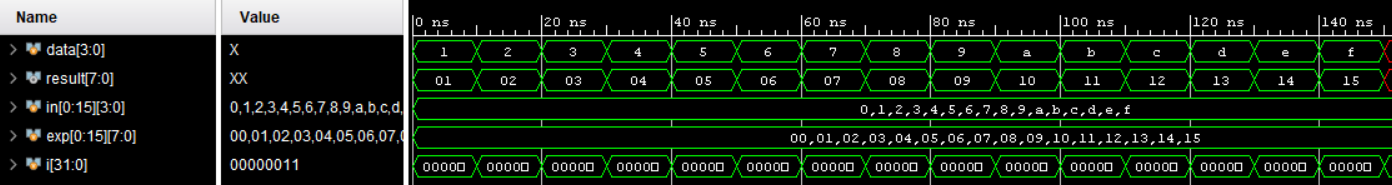


Рис. 4‑3 Результат моделирования

## Назначение выводов СБИС

Назначение выводов СБИС и стандартов ввода вывода приведено на Рис. 4‑3.



Рис. 4‑3 Назначение выводов

## Тестирование на плате Nexys4 DDR

Для тестирования проекта на плате использовались тесты, описанные в разделе 4.4

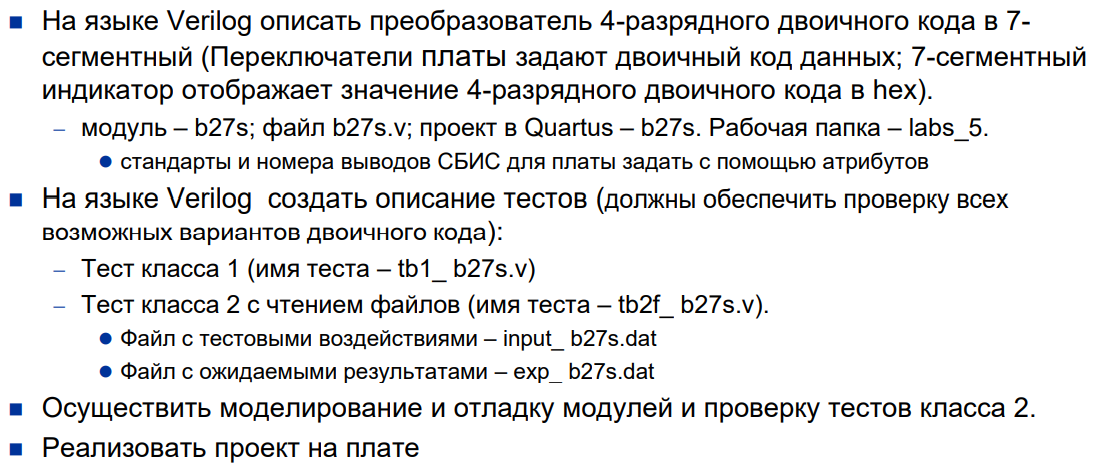
Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

## Выводы

В результате было реализовано устройство, описанное в задании, и были написаны тесты, требуемые в задании.

# Задание labs\_5

## Задание



## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже листинге 5-1.

Листинг 5-1. b27s.v

|  |
| --- |
| module b27s(  input [3:0] num,  output reg [6:0] ind  );  reg [6:0] arr[15:0];  initial begin  arr[0] = 7'h40;  arr[1] = 7'h79;  arr[2] = 7'h24;  arr[3] = 7'h30;  arr[4] = 7'h19;  arr[5] = 7'h12;  arr[6] = 7'h02;  arr[7] = 7'h78;  arr[8] = 7'h00;  arr[9] = 7'h10;  arr[10] = 7'h08;  arr[11] = 7'h03;  arr[12] = 7'h46;  arr[13] = 7'h21;  arr[14] = 7'h06;  arr[15] = 7'h0e;  end  always @ \*  begin  ind = ~arr[num];  end  endmodule |

## Результат синтеза

Результат синтеза описания на языке Verilog приведен ниже на Рис. 5‑1.

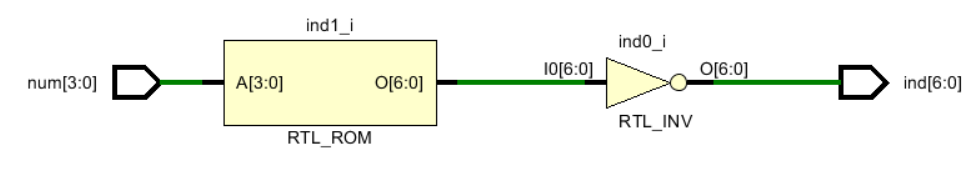


Рис. 5‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты:

Листинг 5-2. tb1\_b27s.v

|  |
| --- |
| module tb1\_b27s();  reg [3:0] data;  wire [6:0] result;  integer i;  b27s fun (data, result);  initial begin  data = 0;  for (i=1; i<17; i=i+1)  #10 data = i;  end  initial begin  $display("\t\t TIME B BD");  $monitor($time,,,,,,data,,,,,,result);  end  endmodule |

Результаты моделирования приведены на Рис. 5-2.

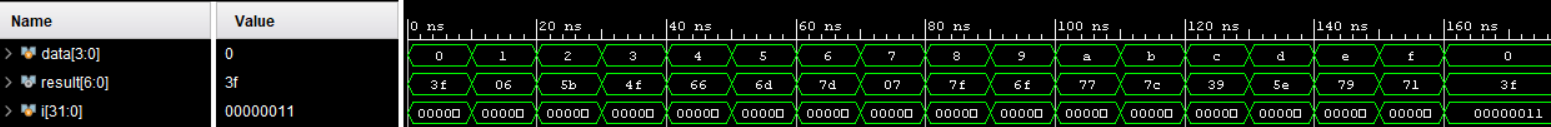


Рис. 5‑2 Результат моделирования

Листинг 5-3. tb2\_b27s.v

|  |
| --- |
| `timescale 1ns / 1ps  module tb2\_b27s();  reg [3:0] data;  reg [6:0] exp;  wire [6:0] result;  b27s fun1 (data, result);  initial begin  data = 4'b0000;  forever #10 data = data + 1;  end  initial begin  forever begin  @ (data) case(data)  4'h0: exp = ~7'h40; //0  4'h1: exp = ~7'h79; //1  4'h2: exp = ~7'h24; //2  4'h3: exp = ~7'h30; //3  4'h4: exp = ~7'h19; //4  4'h5: exp = ~7'h12; //5  4'h6: exp = ~7'h02; //6  4'h7: exp = ~7'h78; //7  4'h8: exp = ~7'h00; //8  4'h9: exp = ~7'h10; //9  4'ha: exp = ~7'h08; //A  4'hb: exp = ~7'h03; //B  4'hc: exp = ~7'h46; //C  4'hd: exp = ~7'h21; //D  4'he: exp = ~7'h06; //E  4'hf: exp = ~7'h0e; //F  endcase  end end  initial begin: assert\_scope  forever begin  @ (data); #0.6;  if (result != exp)  err\_task;  end end  task err\_task;  begin  $display("TIME: %t ERROR: expected: %b actual: %b", $realtime, exp, result);  $stop;  end  endtask  initial begin  $display("\t\t TIME Binary 7th segment");  $monitor("%t %d %8b", $time, data, result);  end  initial begin  #160  $display(" \*\*\*SUCCESS\*\*\* ");  $stop;  end  endmodule |

Результаты моделирования приведены на Рис. 5-3.

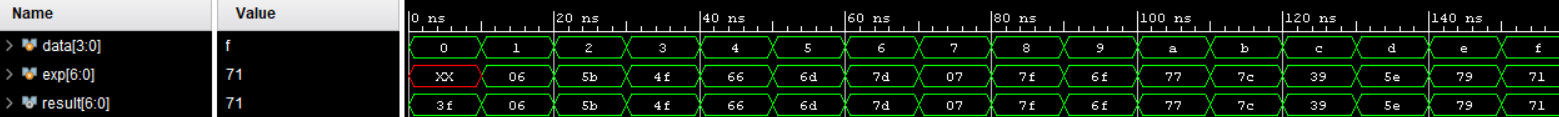


Рис. 5‑3 Результат моделирования

## Назначение выводов СБИС

Назначение выводов СБИС и стандартов ввода вывода приведено на Рис. 5‑3.

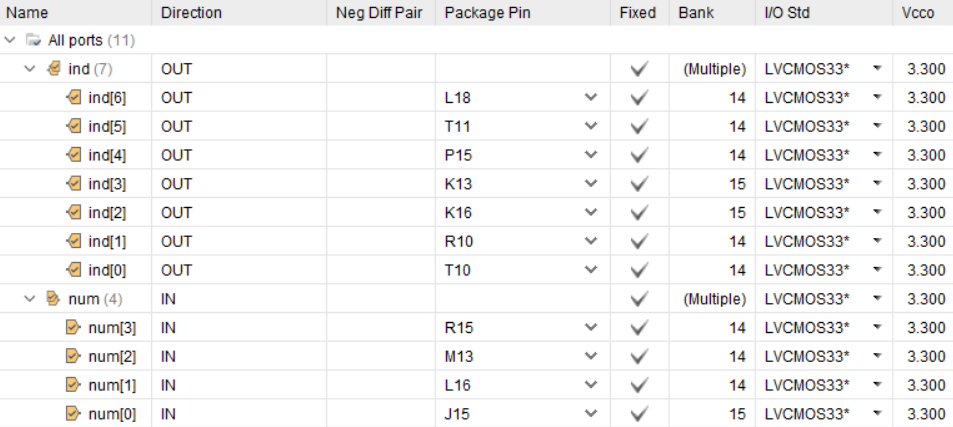


Рис. 5‑3 Назначение выводов

## Тестирование на плате Nexys4 DDR

Для тестирования проекта на плате использовались тесты, описанные в разделе 5.4

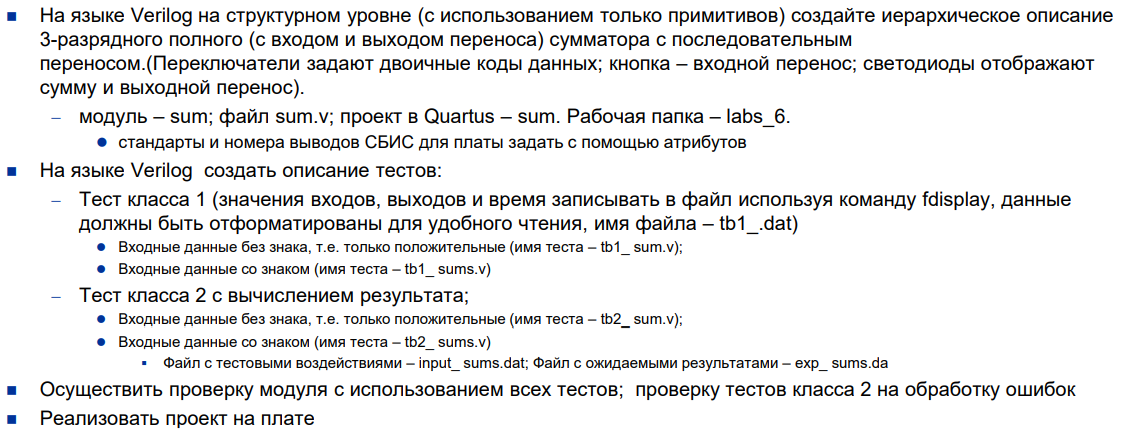
Проведенное на плате Nexys4 DDR тестирование разработанного устройства показало: результаты совпадают с ожидаемыми, устройство работает в соответствии с заданием.

## Выводы

В результате было реализовано устройство, описанное в задании, и были написаны тесты, требуемые в задании.

# Задание labs\_6

## Задание



## Описание на языке Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже листинге 6-1.

Листинг 6-1. sum.v

|  |
| --- |
| module sum(  input [2:0] a, b,  input cin,  output reg [2:0] sum,  output reg cout);  reg t0, t1, t2;  reg c0, c1, c2;  always @\* begin  sum1(a[0], b[0], cin, t0, c0);  sum1(a[1], b[1], c0, t1, c1);  sum1(a[2], b[2], c1, t2, c2);  sum <= {t2, t1, t0};  cout <= c2;  end  task sum1;  input a1, b1, per;  output sum1, out;  begin  sum1 = per & (~b1 & ~a1 | a1 & b1) | ~per & (~b1 & a1 | ~a1 & b1);  out = b1 & a1 | b1 & per | a1 & per;  end  endtask  endmodule |

## Результат синтеза

Результат синтеза описания на языке Verilog приведен ниже на Рис. 6‑1.

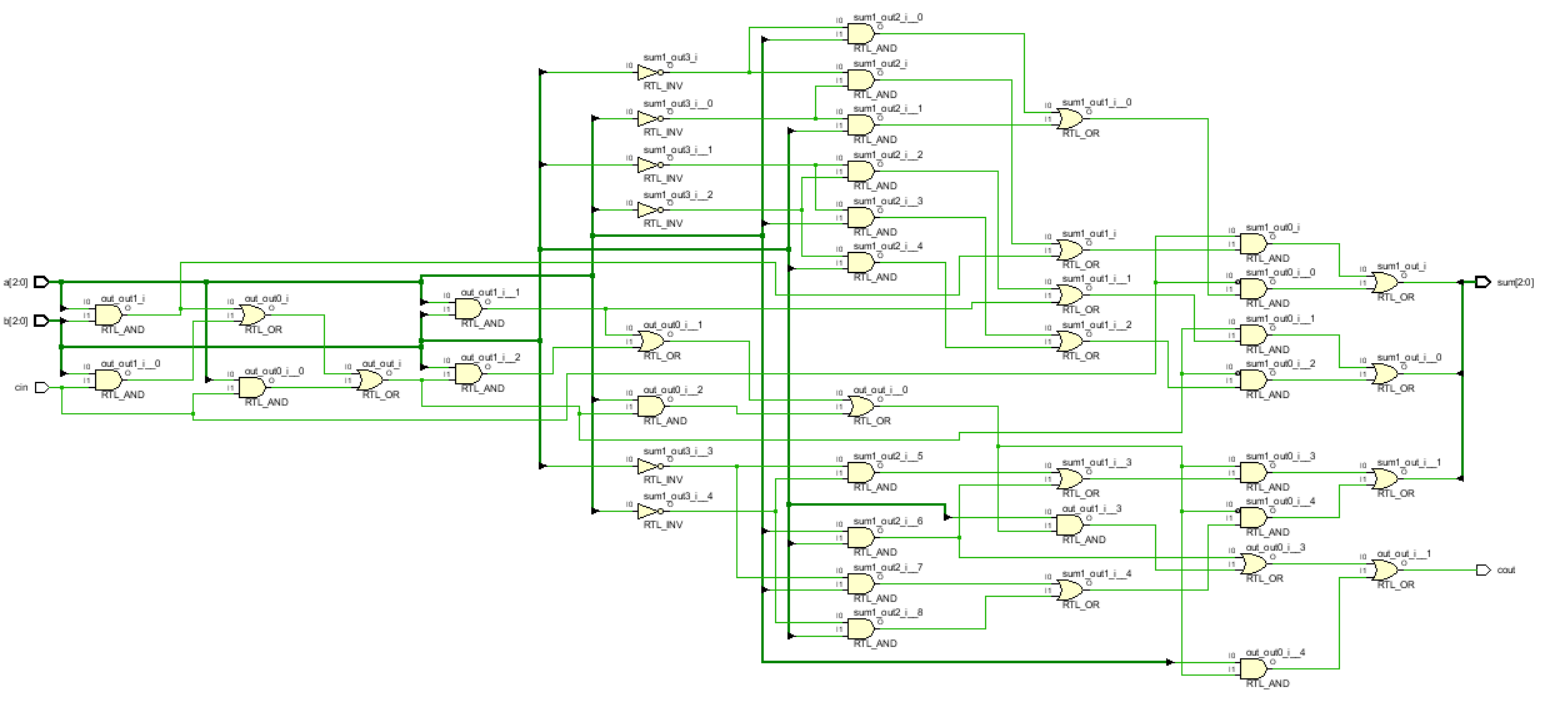


Рис. 6‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты:

Листинг 6-2. tb1\_sum.v

|  |
| --- |
| module tb1\_sum();  localparam N = 3;  reg [N-1:0] a, b;  reg cin;  wire cout;  wire [N-1:0] res;  sum LAB(a, b, cin, res, cout);  integer fd;  initial begin  cin = 0; a = 1; b = 2;  #100 cin = 0; a = 3; b = 3;  #100 cin = 1; a = 1; b = 2;  #100 cin = 1; a = -1; b = -2;  #100 cin = 1; a = 1; b = -3;  #100 cin = 1; a = -1; b = 3;  #100  $fdisplay(fd, " \*\*\*SUCCESS\*\*\* ");  $display(" \*\*\*SUCCESS\*\*\* ");  $fclose (fd);  end  initial begin  fd = $fopen("tb1\_.da", "w");  $monitor("time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,res,cout);  $fmonitor(fd, "time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,res,cout);  end  endmodule |

Результаты моделирования приведены на Рис. 6-2.

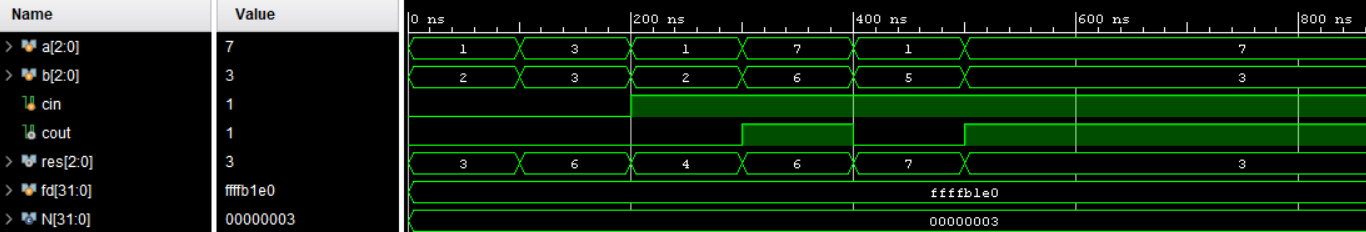


Рис. 6‑2 Результат моделирования

Листинг 6-3. tb1\_sums.v

|  |
| --- |
| `timescale 1ns / 1ps  module tb1\_sums;  localparam N = 3;  reg signed [N-1:0] a, b;  reg cin;  wire cout;  wire signed [N-1:0] res;  sum LAB(a, b, cin, res, cout);  integer fd;  initial begin  cin = 0; a = 1; b = 2;  #100 cin = 0; a = 3; b = 3;  #100 cin = 1; a = 1; b = 2;  #100 cin = 1; a = -1; b = -2;  #100 cin = 1; a = 1; b = -3;  #100 cin = 1; a = -1; b = 3;  #100 $fdisplay(fd, " \*\*\*SUCCESS\*\*\* ");  $display(" \*\*\*SUCCESS\*\*\* ");  $fclose (fd);  end  initial begin  fd = $fopen("log.dat", "w");  $monitor("time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,res,cout);  $fmonitor(fd, "time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,res,cout);  end  endmodule |

Результаты моделирования приведены на Рис. 6-3.

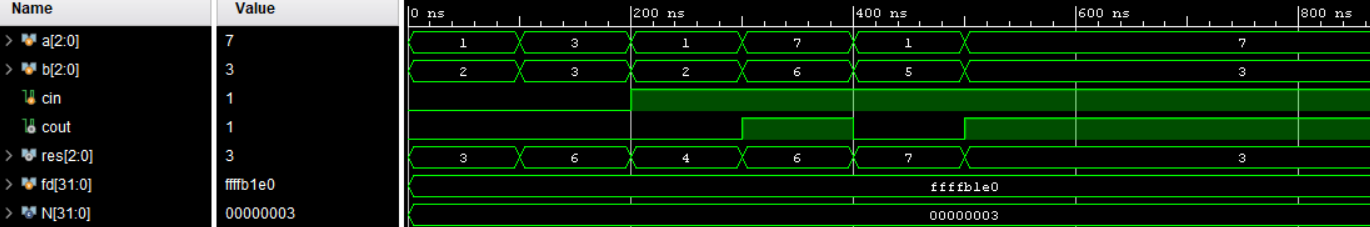


Рис. 6‑3 Результат моделирования

Листинг 6-4. tb2\_sum.v

|  |
| --- |
| `timescale 1 ns / 1 ns  module tb2\_sum;  localparam N = 3;  reg [N-1:0] a, b;  reg cin;  wire cout;  wire [N-1:0] sum;  sum LAB (a, b, cin, sum, cout);  integer fd;  initial begin  test\_sum(1, 2, 0); test\_sum(3, 4, 0);  test\_sum(3, 4, 1); test\_sum(7, 6, 1);  test\_sum(3, 4, 1); test\_sum(7, 6, 1);  $display(" \*\*\*SUCCESS\*\*\* ");  $fdisplay(fd, " \*\*\*SUCCESS\*\*\* "); $fclose (fd);  end  task test\_sum;  input [N-1:0] in\_a, in\_b;  input in\_cin;  reg [N-1:0] exp\_sum;  reg exp\_cout;  begin  a = in\_a;  b = in\_b;  cin = in\_cin;  {exp\_cout, exp\_sum} = a + b + cin;  #1 if (sum != exp\_sum || cout != exp\_cout) begin  $display("Output is wrong"); $fdisplay(fd, "Output is wrong");  $display("Got sum=%d and cout=%d, but expected sum=%d and cout=%d ",sum,cout,exp\_sum,exp\_cout);  $fdisplay(fd, "Got sum=%d and cout=%d, but expected sum=%d and cout=%d ",sum,cout,exp\_sum,exp\_cout);  $fclose (fd); $stop; end  end  endtask  initial begin  fd = $fopen("tb2\_.dat", "w");  $monitor("time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,sum,cout);  $fmonitor(fd, "time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,sum,cout);  end  endmodule |

Результаты моделирования приведены на Рис. 6-4.

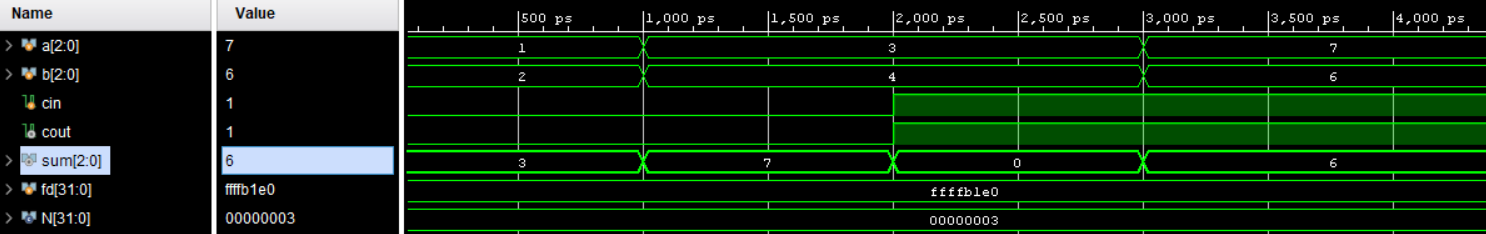
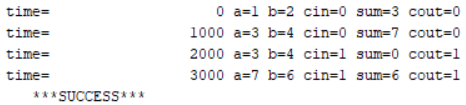


Рис. 6‑4 Результат моделирования

Вывод в файл и консоль:



Листинг 6-5. tb2\_sums.v

|  |
| --- |
| `timescale 1 ns / 1 ns  module tb2f\_sums;  localparam N = 3;  reg signed [N-1:0] a, b;  reg cin;  wire cout;  wire signed [N-1:0] sum;  sum LAB (a, b, cin, sum, cout);  integer fd;  initial begin  test\_sum(1, 2, 0); test\_sum(3, 4, 0);  test\_sum(3, 4, 1); test\_sum(7, 6, 1);  $display(" \*\*\*SUCCESS\*\*\* ");  $fdisplay(fd, " \*\*\*SUCCESS\*\*\* "); $fclose (fd);  end  task test\_sum;  input [N-1:0] in\_a, in\_b;  input in\_cin;  reg [N-1:0] exp\_sum;  reg exp\_cout;  begin  a = in\_a;  b = in\_b;  cin = in\_cin;  {exp\_cout, exp\_sum} = a + b + cin;  #1 if (sum != exp\_sum || cout != exp\_cout) begin  $display("Output is wrong"); $fdisplay(fd, "Output is wrong");  $display("Got sum=%d and cout=%d, but expected sum=%d and cout=%d ",sum,cout,exp\_sum,exp\_cout);  $fdisplay(fd, "Got sum=%d and cout=%d, but expected sum=%d and cout=%d ",sum,cout,exp\_sum,exp\_cout);  $fclose (fd); $stop; end  end  endtask  initial begin  fd = $fopen("tb2\_.dat", "w");  $monitor("time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,sum,cout);  $fmonitor(fd, "time=%t a=%d b=%d cin=%b sum=%d cout=%b",$time,a,b,cin,sum,cout);  end  endmodule |

Результаты моделирования приведены на Рис. 6-5.

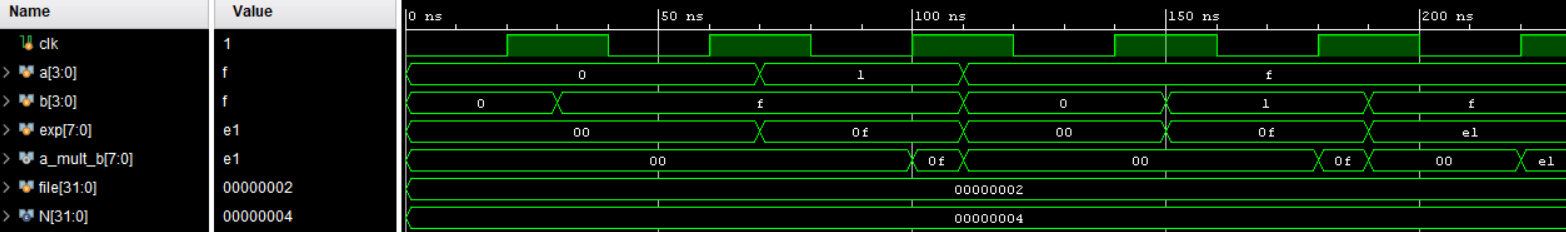


Рис. 6‑5 Результат моделирования

Вывод в файл:

time= 0 a= 1 b= 2 cin=0 sum= 3 cout=0

time= 1000 a= 3 b=-4 cin=0 sum=-1 cout=0

time= 2000 a= 3 b=-4 cin=1 sum= 0 cout=1

time= 3000 a=-1 b=-2 cin=1 sum=-2 cout=1

\*\*\*SUCCESS\*\*\*

## Назначение выводов СБИС

Назначение выводов СБИС и стандартов ввода вывода приведено на Рис. 6‑3.



Рис. 6‑3 Назначение выводов

## Тестирование на плате Nexys4 DDR

Для тестирования проекта на плате использовались тесты, описанные в разделе 6.4

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 задании, и были написаны тесты, требуемые в задании.