САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

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

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет

по заданию lab6

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 lab6

## Задание

Изображение выглядит как текст

Автоматически созданное описание

Вариант 10

– Фибоначчи XOR

## Описание на языке System Verilog

Описание разрабатываемого устройства на языке Verilog приведено ниже листингах 1–3.

Листинг 1. lab6.sv

|  |
| --- |
| `timescale 1ns / 1ps  module lab6(  input bit clk,  input bit rst,  input bit ena,  output bit [7:0] mem\_out  );  bit [7:0] lfsr\_out;  bit [7:0] d\_in;  assign d\_in = lfsr\_out;  lfsr\_8\_7\_6\_5\_4\_1\_f LFSR(.\*);  histogram\_unit HU(.\*);  endmodule |

Листинг 2. lfsr\_f.sv

|  |
| --- |
| `timescale 1ns / 1ps  module lfsr\_f (  input bit clk,  input bit rst,  input bit ena,  output bit [8:1] lfsr\_out  );  always\_ff @(posedge clk, posedge rst)  if (rst) lfsr\_out <= 10'd1;  else if (ena)  if (lfsr\_out == '0) lfsr\_out <= 10'd1;  else lfsr\_out <= {lfsr\_out[7:1], lfsr\_out[8]^lfsr\_out[4]^lfsr\_out[3]^lfsr\_out[2]};// 8 4 3 2 // variant 10  endmodule |

Листинг 3. histogram\_unit.sv

|  |
| --- |
| `timescale 1ns/1ns  module histogram\_unit (  input bit clk,  input bit [7:0] d\_in,  input bit rst,  input bit ena,  output bit [7:0] mem\_out  );  bit [7:0] mem\_arr [0:1023];  bit [7:0] mem\_in;  bit [7:0] adr\_in, adr\_clear;  initial begin:initial\_val  for (int i = 0; i < 1024; i++) mem\_arr[i] = 0;  end:initial\_val  assign mem\_in = (rst)? 0 : ((ena)? (mem\_out + 10'd1): mem\_out);  assign adr\_in = (rst)? adr\_clear : d\_in;  always @(posedge clk)  begin:building\_historgram  mem\_arr[adr\_in] <= mem\_in;  mem\_out <= mem\_arr[adr\_in];  end:building\_historgram  always\_ff @(posedge clk, negedge rst)  begin :clearing\_array  if(~rst) adr\_clear<=0;  else adr\_clear<= adr\_clear+ 1'b1;  end:clearing\_array  endmodule |

Листинг 4. db\_lab6.sv

|  |
| --- |
| `timescale 1ns / 1ps  module db\_lab6(input logic clk);  bit rst, ena;  bit [7:0] mem\_out;  lab6 LAB(.\*);  vio\_0(.clk, .probe\_out0(rst), .probe\_out1(ena));  ila\_0(.clk, .probe0(mem\_out));  endmodule |

## Результат синтеза (RTL)

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

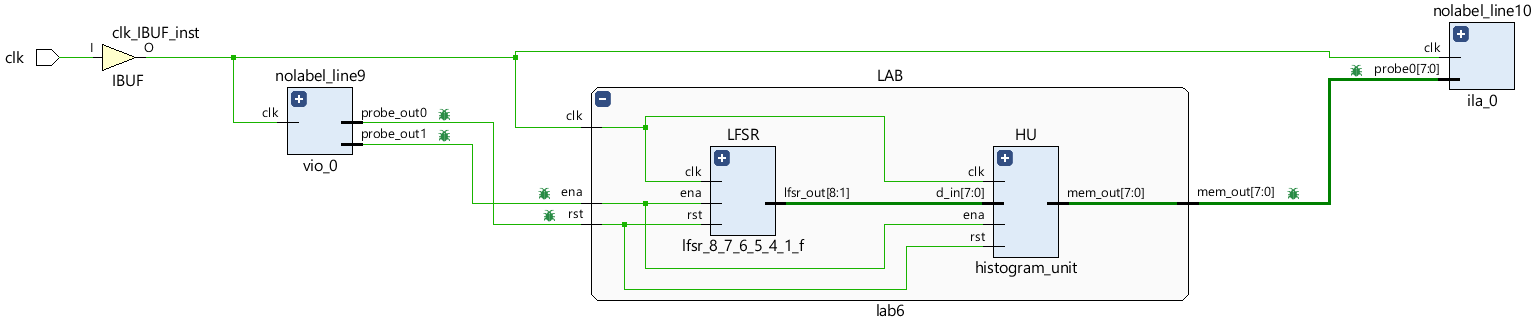


Рис. ‑1 Синтезированная схема

## Моделирование

Для проверки правильности работы созданного Verilog описания использовались тесты представленные в листингах 4-5:

Листинг 4. tb\_lfsr.sv

|  |
| --- |
| `timescale 1ns / 1ps  module tb\_lfsr();  bit clk = '0;  bit [8:1] lfsr\_out = '0;  bit [8:1] lfsr\_out\_f = '0;  bit [9:1] cnt\_int = '0;  bit rst = '1;  bit ena = '0;    lfsr\_8\_7\_6\_5\_4\_1\_f LFSR\_8\_7\_6\_5\_4\_1\_F(.lfsr\_out(lfsr\_out), .clk, .ena, .rst);  always #10 clk = ~clk;  initial begin  #15;  rst = '0;  #25;  ena = '1;  end  initial begin  #45;  @(negedge clk);  lfsr\_out\_f = lfsr\_out;  forever begin @(negedge clk);  cnt\_int++;  if (lfsr\_out\_f == lfsr\_out) break;  end  #20;  $stop;  end  endmodule |

Листинг 5. tb\_lab6.sv

|  |
| --- |
| `timescale 1ns / 1ps  module tb\_lab6();  bit clk = '1;  bit rst = '0;  bit ena = '0;  bit [7:0] mem\_out = '0;  lab6 LAB6(.\*);  always #10 clk = ~clk;  initial begin  ena = '1;  #5180;  $stop;  end  endmodule |

Результаты моделирования приведены на Рис. 1‑4

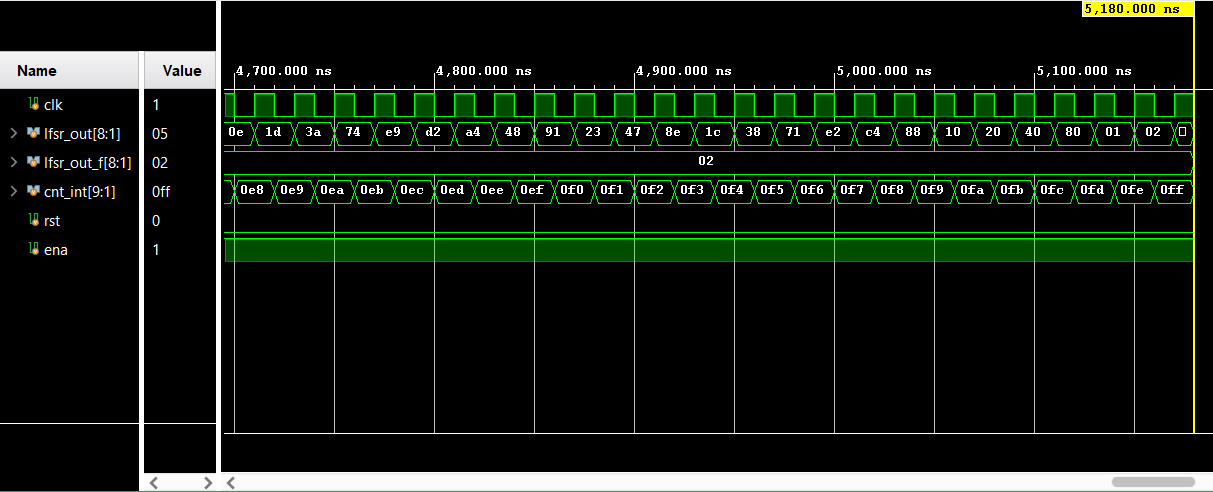


Рис. ‑2 Результат моделирования tb\_lfsr

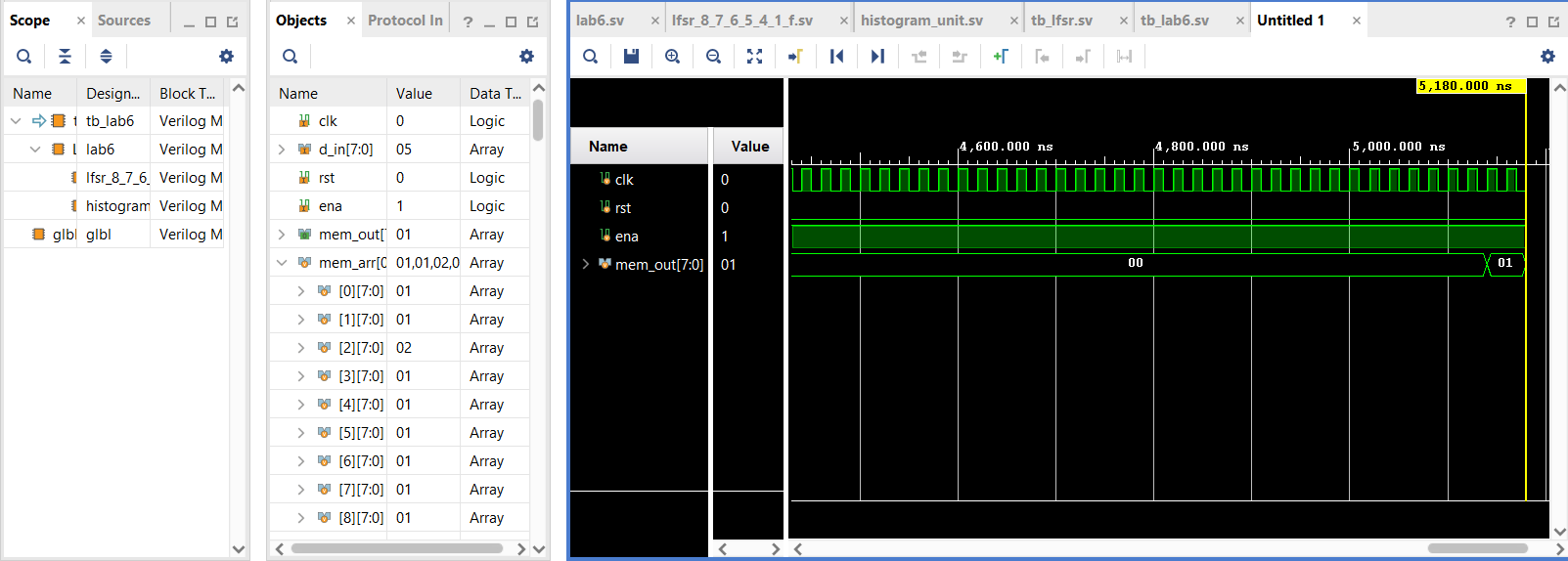


Рис. ‑3 Результат моделирования tb\_lab6

## Назначение выводов СБИС

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

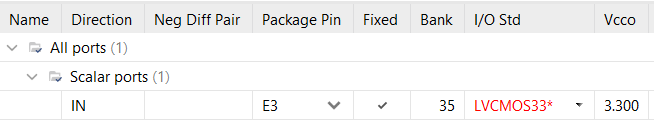


Рис. ‑5 Назначение выводов

## Тестирование на плате Nexys4 DDR

Для тестирования проекта на плате использовались тесты, описанные в разделе 1.4. Также для тестирования использовались ip-ядра VIO и ILA для логического анализатора. 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.