|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«*Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**по лабораторной работе № 2**

**Дисциплина: \_Схемотехника\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Название лабораторной работы: Проектирование цифровых устройств на основе ПЛИС**

**Вариант № 68**

Студент гр. \_ИУ6-53Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_В.К. Залыгин\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_М. Гейне\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2024

**Цель работы**

Закрепление на практике теоретических знаний о способах реализации устройств управления, исследование способов организации узлов ЭВМ, освоение принципов проектирования цифровых устройств на основе ПЛИС.

**Выполнение работы**

На рисунке 1 приведена функциональная схема устройства.

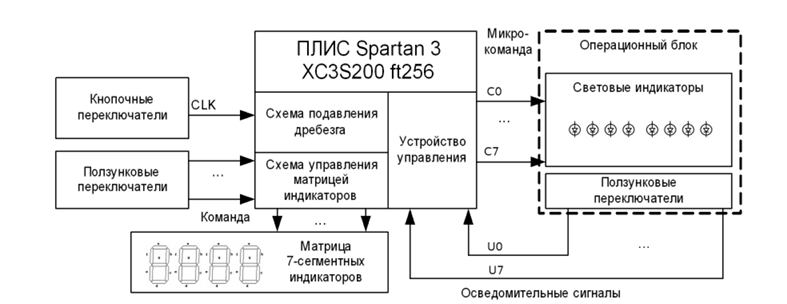


Рисунок 1 – Функциональная схема устройства

На рисунке 2 приведена схема переходов автомата.

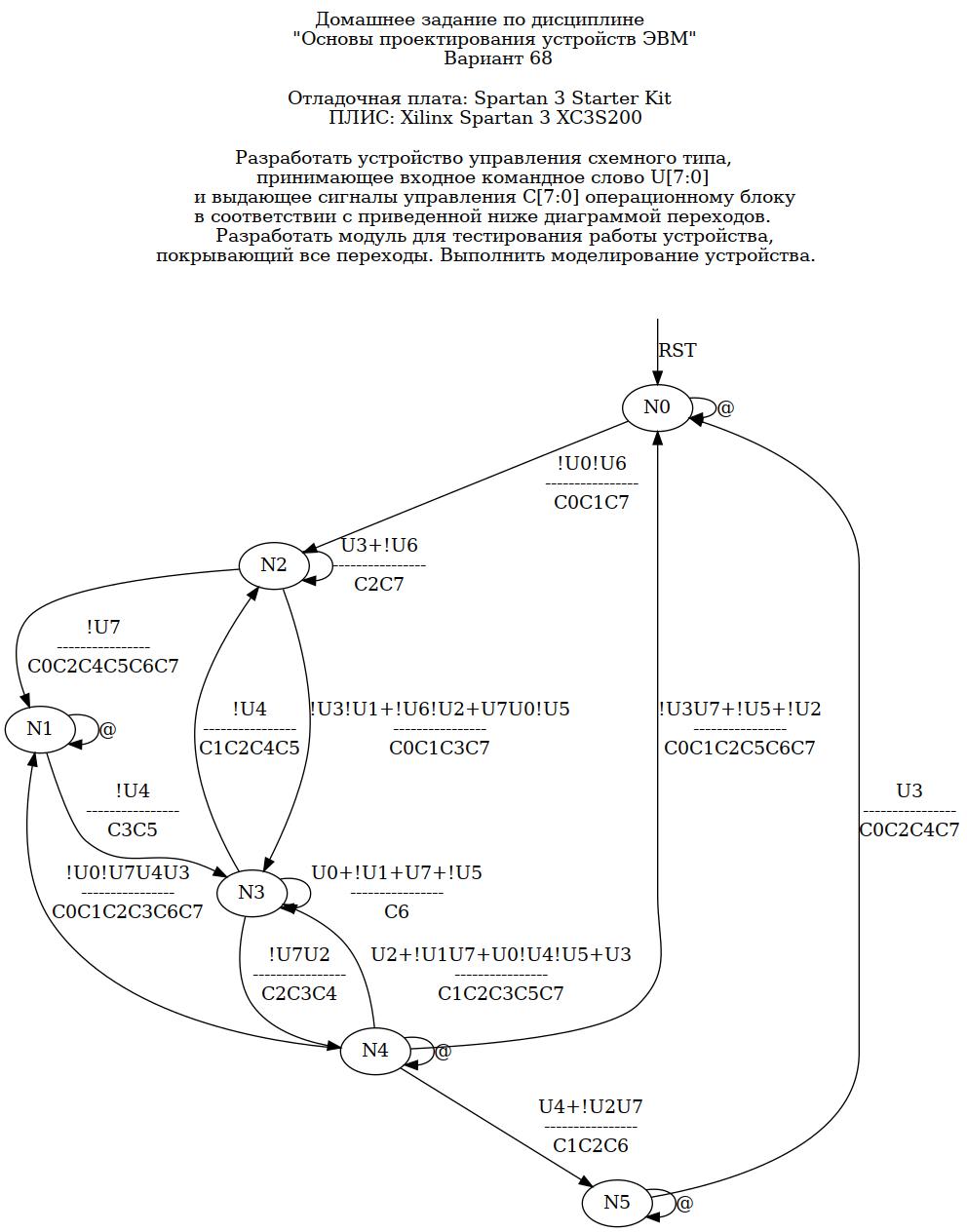


Рисунок 2 – Схема переходов автомата

Был открыт проект домашнего задания, в него были добавлены модули из первой ЛР. В первой ЛР, файл main.v был изменен.

Листинг 1 – Описание модуля подавления дребезга

|  |
| --- |
| module main (  input clk,  input count,  input [7:0] U,  input rst,  output [3:0] a,  output [7:0] led,  output [7:0] C\_to\_print,  output [1:0] state,  output [2:0] state\_to\_print  );  //cnt -  wire cnt;  lab2\_example lab2\_example\_inst (  .clk(clk),  .rst(rst),  .count(count),  .cnt(cnt)  //.s\_out(state)  );    //reg [7:0] C;  //wire [2:0] state\_to\_print = 3'b000;  lab3 lab3\_inst(  .clk(clk),  .rst(rst),  .en(en),  .U(U),  .C(C\_to\_print),  .state(state\_to\_print)  );  //  reg [16:0] counter;  wire counter\_ovf = (counter == 2\*\*20);  always @(posedge clk)  if(rst || counter\_ovf)  counter <= 0;  else  counter <= counter + 1;  // CNT - CNT ( )  reg cnt\_ff;  wire cnt\_rise = (cnt==1'b1) && (cnt\_ff==1'b0);  always @(posedge clk)  if(rst)  cnt\_ff <= 1'b0;  else begin  cnt\_ff <= cnt;    end  //  assign en = cnt\_rise;  reg [15:0] main\_counter;  always @(posedge clk)  if(rst)  main\_counter <= 1'b0;  else  if(cnt\_rise) begin  //main\_counter <= main\_counter + 1;  main\_counter <= {13'b0000000000000,state\_to\_print};    end    /\* \*/  wire [3:0] driver\_to\_decoder;  seven\_segment\_driver ssd\_inst (  .clk(clk),  .rst(rst),  .q(state\_to\_print),  .clk\_div(counter\_ovf),  .d(driver\_to\_decoder),  .a(a)  );    led\_decode led\_decode\_inst (  .dh(driver\_to\_decoder),  .seg\_data(led)  );  endmodule |

Код теста приведен в листинге 2.

Листинг 2 – Код теста

|  |
| --- |
| `timescale 10ns/1ns  module main\_tb();  reg clk=1'b0, count=1'b0, rst=1'b1;  wire [3:0] a;  wire [7:0] led;  wire [1:0] state;  reg [7:0] U;  wire [2:0] st\_to\_p;  wire [7:0] C\_to\_p;  main uut (  .clk,  .count,  .rst,  .a,  .C\_to\_print(C\_to\_p),  .state,  .U,  .state\_to\_print(st\_to\_p)  );  task click();  begin  @(posedge clk) count = #1 1'b1;  @(posedge clk) count = #1 1'b0;  @(posedge clk) count = #1 1'b1;  @(posedge clk) count = #1 1'b0;  @(posedge clk) count = #1 1'b1;  @(posedge clk) count = #1 1'b0;  @(posedge clk) count = #1 1'b1;  #1000  @(posedge clk) count = #1 1'b0;  @(posedge clk) count = #1 1'b1;  @(posedge clk) count = #1 1'b0;  @(posedge clk) count = #1 1'b1;  @(posedge clk) count = #1 1'b0;  @(posedge clk) count = #1 1'b1;  @(posedge clk) count = #1 1'b0;  end  endtask  always #10 clk = ~clk;    initial begin  #200 rst = 1'b0;  #1000;  U = 'b00000001; // N0 -> N0  click;  U = 'b00000000; // N0 -> N2  click;  U = 'b10001100; // N2 -> N2  click;  U = 'b01000110; // N2 -> N1  click;  U = 'b00010000; // N1 -> N1  click;  U = 'b00000000; // N1 -> N3  click;  U = 'b00010000; // N3 -> N3  click;  U = 'b00110110; // N3 -> N4  click;  U = 'b00010000; // N4 -> N5 ??  click;  U = 'b00000000; // N5 -> N5  click;  U = 'b00001000; // N5 -> N0  click;    $finish;  end  endmodule |

После создания распиновки и заливки проекта на ПЛИС работа была протестирована в соответствии с таблицей переходов из ДЗ. Переходы представлены в таблице 1.

Таблица 1 – Таблица переходов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дуга |  |  |  |  |
|  | 00000001 | 1 | 00000000 | 0 |
|  | 00000000 | 0 | 10000011 | 131 |
|  | 10001100 | 140 | 10000100 | 132 |
|  | 01000110 | 6 | 11110101 | 245 |
|  | 00010000 | 16 | 00000000 | 0 |
|  | 00000000 | 0 | 00101000 | 40 |
|  | 00010000 | 16 | 01000000 | 64 |
|  | 00110110 | 54 | 00011100 | 28 |
|  | 00010000 | 16 | 01000110 | 70 |
|  | 00000000 | 0 | 00000000 | 0 |
|  | 00001000 | 8 | 10010101 | 149 |

Результаты тестирования были удовлетворительными – логика работы соответствует предполагаемой. Временная диаграмма представлена на рисунке 3.

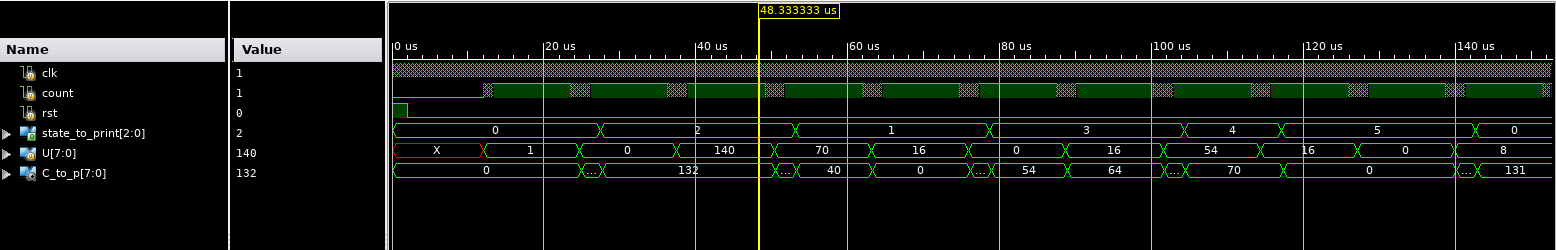


Рисунок 3 – Временная диаграмма

Некоторые тесты приведены в таблице 2.

Таблица 2 – Тестирование макета

|  |  |  |
| --- | --- | --- |
| Номер теста | Ожидаемый результат | Полученный результат |
| 1 | Остались в S0 при U =00000001 | Остались в S0 |
| 2 | Перешли в S2 по U=00000000 | Перешли в S2 |
| 3 | Из S2 перешли в S1 по U=01000110 | перешли в S1 |

**Вывод**

В ходе лабораторной работы были закреплены на практике теоретические сведения, полученных при изучении методики проектирования цифровых устройств на основе программируемых логических интегральных схем (ПЛИС).