



«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 \_\_\_\_\_

## О Т Ч Е Т

по лабораторной работе № 2

Дисциплина: Схемотехника

Название лабораторной работы: Проектирование цифровых устройств на  
основе ПЛИС

Вариант № 68

Студент гр. ИУ6-53Б

\_\_\_\_\_  
(Подпись, дата)

В.К. Залыгин

\_\_\_\_\_  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

М. Гейне

\_\_\_\_\_  
(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.

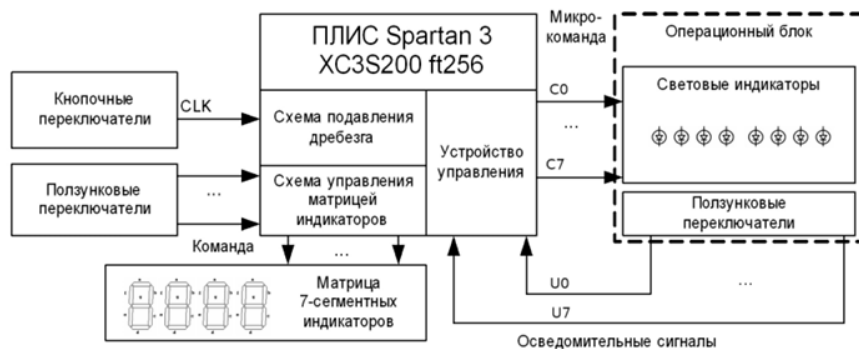


Рисунок 1 – Функциональная схема устройства

На рисунке 2 приведена схема переходов автомата.

Домашнее задание по дисциплине  
"Основы проектирования устройств ЭВМ"  
Вариант 68

Отладочная плата: Spartan 3 Starter Kit  
ПЛИС: Xilinx Spartan 3 XC3S200

Разработать устройство управления схемного типа, принимающее входное командное слово  $U[7:0]$  и выдающее сигналы управления  $C[7:0]$  операционному блоку в соответствии с приведенной ниже диаграммой переходов. Разработать модуль для тестирования работы устройства, покрывающий все переходы. Выполнить моделирование устройства.

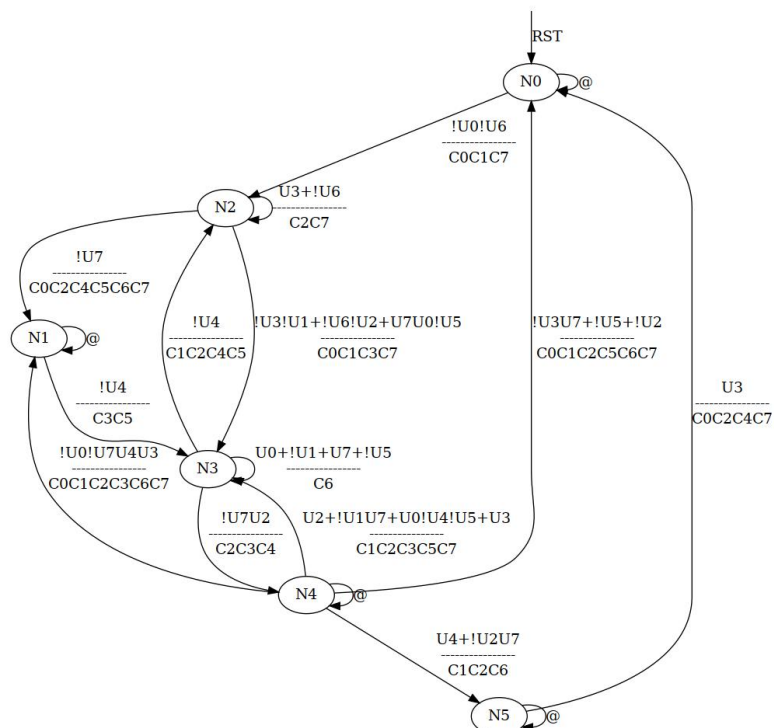


Рисунок 2 – Схема переходов автомата

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
endmodule
```

```

        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'b00000000000000, state_to_print};
            end

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	$U_2[7:0]$	$U_{10}$	$C_2[7:0]$	$C_{10}$
$N_0 \rightarrow N_0$	00000001	1	00000000	0
$N_0 \rightarrow N_2$	00000000	0	10000011	131
$N_2 \rightarrow N_2$	10001100	140	10000100	132
$N_2 \rightarrow N_1$	01000110	6	11110101	245
$N_1 \rightarrow N_1$	00010000	16	00000000	0
$N_1 \rightarrow N_3$	00000000	0	00101000	40
$N_3 \rightarrow N_3$	00010000	16	01000000	64
$N_3 \rightarrow N_4$	00110110	54	00011100	28
$N_4 \rightarrow N_5$	00010000	16	01000110	70
$N_5 \rightarrow N_5$	00000000	0	00000000	0
$N_5 \rightarrow N_0$	00001000	8	10010101	149

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

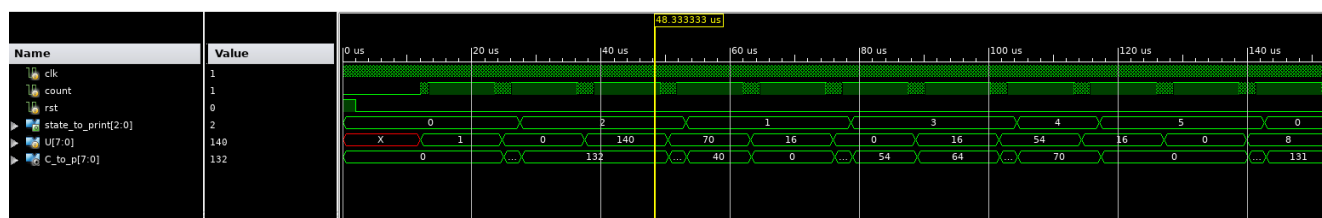


Рисунок 3 – Временная диаграмма

Некоторые тесты приведены в таблице 2.

Ы

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).