



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ «09.03.01 Информатика и вычислительная техника»

---

## ОТЧЕТ по Домашняя работе № 1

**Название:** «Проектирование устройств управления с жесткой логикой»

**Дисциплина:** «Основы проектирования устройств ЭВМ.»

**Вариант № 38**

Студент ИУ6-62Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

А.Е.Медведев  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И. О. Фамилия)

2022 г.

## Цель работы:

В ходе выполнения домашнего задания необходимо разработать устройство управления схемного типа, обрабатывающий входное командное слово  $C=ABCDEF$  и выдающий сигналы управления  $M=M_0, \dots, M_7$  операционному блоку в соответствии с приведенной в индивидуальном задании логикой работы.

По варианту в данном домашнем задании будет графф под номером 2. Название платы — Nexus2 Сам графф приведён на рисунке 1

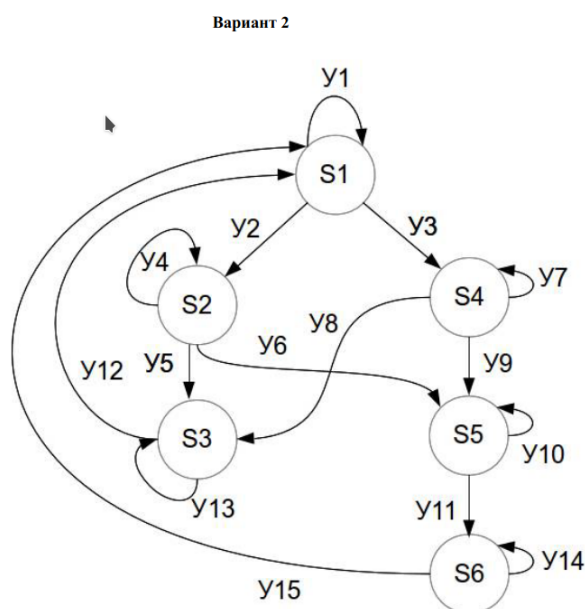


Рисунок 1 – Графф задания

Таблица 1 – Варианты диаграмм и активных сигналов.

| S1 | S2  | S3 | S4 | S5 | S6  |
|----|-----|----|----|----|-----|
| 1  | 3,5 | 6  | -  | 7  | 2,4 |

Таблица 2 – Условия переходов и наименование отладочной платы

| Y1 | Y2 | Y3 | Y4 | Y5   | Y6 | Y7 | Y8  | Y9 | Y10 | Y11  | Y12 | Y13 | Y14 | Y15  |
|----|----|----|----|------|----|----|-----|----|-----|------|-----|-----|-----|------|
| @  | CE | AC | @  | !E+C | F  | @  | !DF | B  | @   | !A!C | A!B | @   | @   | ABCF |

Таблица 3 – Условия переходов и наименование отладочной платы

| Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 | Y8 | Y9 | Y10 | Y11 | Y12 | Y13 | Y14 | Y15 |
|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| -  | -  | -  | -  | -  | -  | -  | -  | -  | -   | -   | -   | -   | -   | -   |

## Этап 1

А. По диаграмме переходов автомата и описанию условий переходов и активных сигналов, определить тип управляющего автомата (автомат Мили или Мура, смешанный). Выбор обосновать. Данный управляющий автомат является автоматом Мура, так как все выходные сигналы зависят только от текущего состояния. В. Произвести кодирование состояний управляющего автомата. Конечный граф представлен на рисунке 2

Вариант 2

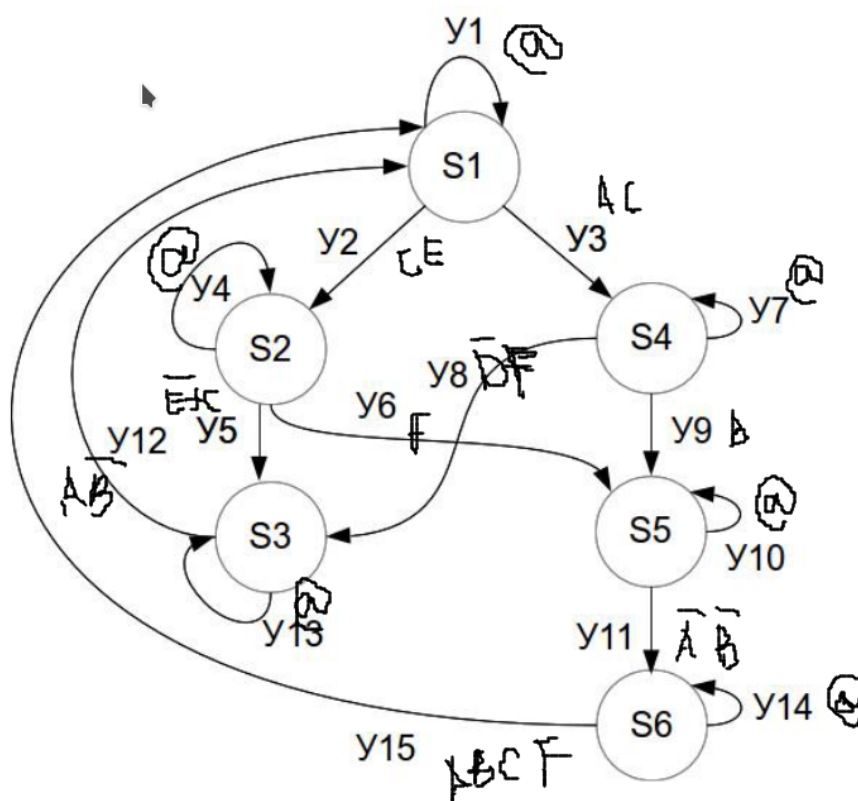


Рисунок 2 – Исправленные граф

## Этап 2

А. Разработать описание устройства управления на языке VHDL, для чего использовать приведенные в Приложении 2 к методическим указаниям шаблоны для автоматов Мили и Мура

Листинг 1 – Описание устройства на языке VHDL

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY control_unit IS
PORT (
    C : IN std_logic_vector (5 DOWNTO 0);
    CLK : IN std_logic;
    RST : IN std_logic;
    M : OUT std_logic_vector (7 DOWNTO 0) );
END control_unit;

ARCHITECTURE rtl OF control_unit IS
    TYPE STATE_TYPE IS (s1, s2, s3, s4, s5, s6);
    SIGNAL current_state: STATE_TYPE := s1;
BEGIN
    PROCESS (clk, rst)
    BEGIN
        IF (rst='0') THEN
            M <= "00000000";
            current_state <= s1;
        ELSIF (CLK'EVENT AND CLK='1') THEN
            CASE current_state IS
                WHEN S1 =>
                    M <= "00000010";
                    IF C(2)='1' AND C(4)='1' THEN
                        current_state <= S2;
                    ELSIF C(0)='1' AND C(2)='1' THEN
                        current_state <= S4;
                    ELSE
```

```

        current_state <= S1;
    END IF;
WHEN S2 =>
    M <= "00101000";
    IF C(5)='1' THEN
        current_state <= S5;
    ELSIF C(4)='0' OR C(2)='1' THEN
        current_state <= S3;
    ELSE
        current_state <= S2;
    END IF;
WHEN S3 =>
    M <= "01000000";
    IF C(0)='1' AND C(1)='0' THEN
        current_state <= S1;
    ELSE
        current_state <= S3;
    END IF;
WHEN S4 =>
    IF C(3)='0' AND C(5)='1' THEN
        current_state <= S3;
    ELSIF C(1)='1' THEN
        current_state <= S5;
    ELSE
        current_state <= S4;
    END IF;
WHEN S5 =>
    M <= "10000000";
    IF C(0)='0' AND C(1)='0' THEN
        current_state <= S6;
    ELSE
        current_state <= S5;
    END IF;
WHEN S6 =>
    M <= "00010100";
    IF C(0)='1' AND C(1)='1' AND C(2)='1' AND C(5)='1' THEN
        current_state <= S1;
    ELSE
        current_state <= S6;

```

```

        END IF;
    END CASE;
END IF;
END PROCESS;
END rtl;

```

Б. Разработать тестовое описание для устройства, представляющее собой генератор входных сигналов. Тестовое описание должно обеспечивать проверку всех ветвей автомата.

Таблица 4 – Обход всех ветвей

| A | B | C | D | E | F | Дуга     | Сигналы |
|---|---|---|---|---|---|----------|---------|
| - | - | - | - | - | - | S1 -> S1 | 1       |
| 1 | - | 1 | - | - | - | S1 -> S4 | -       |
| - | - | - | - | - | - | S4 -> S4 | -       |
| - | 1 | - | - | - | - | S4 -> S5 | 7       |
| - | - | - | - | - | - | S5 -> S5 | 7       |
| 0 | 0 | - | - | - | - | S5 -> S6 | 2.4     |
| - | - | - | - | - | - | S6 -> S6 | 2.4     |
| 1 | 1 | 1 | - | - | 1 | S6 -> S1 | 1       |
| - | - | - | - | - | - | S1 -> S2 | 3.5     |
| - | - | - | - | - | - | S2 -> S2 | 3.5     |
| - | - | 1 | - | 0 | - | S2 -> S3 | 6       |
| - | - | - | - | - | - | S3 -> S3 | 6       |
| 1 | 0 | - | - | - | - | S3 -> S1 | 1       |
| 1 | - | 1 | - | - | - | S1 -> S4 | -       |
| - | - | - | 0 | - | 1 | S4 -> S3 | 6       |
| 1 | 0 | - | - | - | - | S3 -> S1 | 1       |
| - | - | 1 | - | 1 | - | S1 -> S2 | 3.5     |
| - | - | - | - | - | 1 | S2 -> S5 | 7       |

Пропущенные значения в таблице могут принимать значения 0 или 1.

Листинг теста показан в листинге 2

## Листинг 2 – Описание устройства на языке VHDL

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.NUMERIC_STD.ALL;

ENTITY test_dz IS
END test_dz;

ARCHITECTURE RTL OF test_dz IS
COMPONENT control_unit PORT (
    M : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
    CLK : IN STD_LOGIC;
    RST : IN STD_LOGIC;
    C : IN STD_LOGIC_VECTOR(5 DOWNTO 0)
);
END COMPONENT;

SIGNAL M : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL CLK : STD_LOGIC := '0';
SIGNAL RST : STD_LOGIC := '0';
SIGNAL C : STD_LOGIC_VECTOR(5 DOWNTO 0) := (OTHERS => '0');
CONSTANT CLK_PERIOD : TIME := 10NS;

BEGIN

    UUT : control_unit
    PORT MAP(
        M => M,
        CLK => CLK,
        RST => RST,
        C => C
    );

    CLK_PROCESS : PROCESS
    BEGIN
        CLK <= '0';
        WAIT FOR CLK_PERIOD/2;
        CLK <= '1';
        WAIT FOR CLK_PERIOD/2;
```

```

END PROCESS CLK_PROCESS;

SIM_PROC : PROCESS
BEGIN
    RST <= '0';
    WAIT FOR CLK_PERIOD;
    C <= "000000"; --S1 -> S1
    RST <= '1';
    WAIT FOR CLK_PERIOD;
    C <= "000000"; --S1 -> S1
    WAIT FOR CLK_PERIOD;
    C <= "000101"; --S1 -> S4
    WAIT FOR CLK_PERIOD;
    C <= "000000"; --S4 -> S4
    WAIT FOR CLK_PERIOD;
    C <= "000110"; --S4 -> S5
    WAIT FOR CLK_PERIOD;
    C <= "111111"; --S5 -> S5
    WAIT FOR CLK_PERIOD;
    C <= "000000"; --S5 -> S6
    WAIT FOR CLK_PERIOD;
    C <= "111111"; --S6 -> S1
    WAIT FOR CLK_PERIOD;
    C <= "000000"; --S1 -> S2
    WAIT FOR CLK_PERIOD;
    C <= "010000"; --S2 -> S2
    WAIT FOR CLK_PERIOD;
    C <= "000100"; --S2 -> S3
    WAIT FOR CLK_PERIOD;
    C <= "000010"; --S3 -> S3
    WAIT FOR CLK_PERIOD;
    C <= "000001"; --S3 -> S1
    WAIT FOR CLK_PERIOD;
    C <= "000101"; --S1 -> S4
    WAIT FOR CLK_PERIOD;
    C <= "100000"; --S4 -> S3
    WAIT FOR CLK_PERIOD;
    C <= "000001"; --S3 -> S1
    WAIT FOR CLK_PERIOD;

```



```

        C <= "010100"; --S1 -> S2
        WAIT FOR CLK_PERIOD;
        C <= "100000"; --S2 -> S5
        WAIT FOR CLK_PERIOD; WAIT;
    PROCESS SIM_PROC;
END ARCHITECTURE;

```

### Этап 3:

Моделирование в САПР Xilinx ISE 14.7 Результаты моделирования приведены на рисунках 3. Из них следует, что описанное устройство работает корректно.

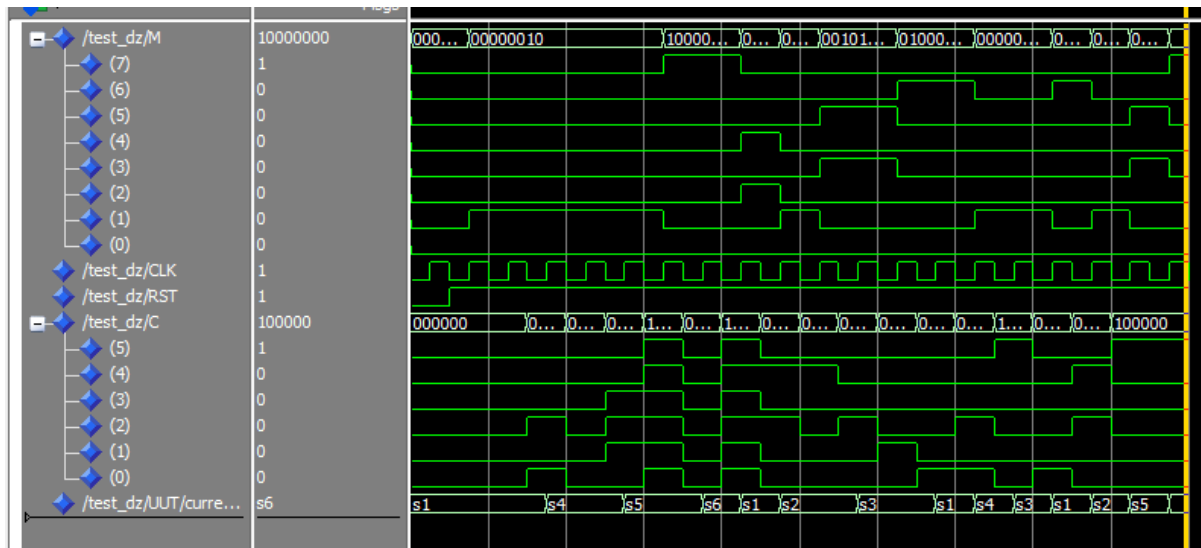


Рисунок 3 – Переключения состояний программы

### Вывод:

В ходе выполнения домашнего задания разработано устройство управления схемного типа, обрабатывающее входное командное слово  $C=ABCDEF$  и выдающее сигналы управления  $M = M_0, \dots, M_{k-1}$  операционному блоку в соответствии с приведенной в индивидуальном задании логикой работы.