



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 (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по домашнему заданию № 1

Название: Проектирование устройств управления с жесткой логикой

Дисциплина: Основы проектирования устройств ЭВМ

Студент

ИУ6-62Б

(Группа)

(Подпись, дата)

С.В. Астахов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Вариант 20

Введение

Цель работы: разработать устройство управления схемного типа, обрабатывающий входное командное слово $C=\{ABCDEF\}$ и выдающий сигналы управления $M=\{M_0,...,M_{k-1}\}$ операционному блоку в соответствии с приведенной в индивидуальном задании логикой работы.

Условие:

Индивидуальные условия приведены в таблицах 1-3.

Таблица 1 - варианты диаграмм и активных сигналов

Вариант	Диаграмма переходов	Активные сигналы М в состоянии					
		S1	S2	S3	S4	S5	S6
20	4	2	0	1, 7	5, 6	3	4

Таблица 2 - условия переходов и наименование отладочной плат

Вар.	Плата	Условия переходов														
		Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
20	Nexus2	@	EF	CD	AC	ABC	D	AF	@	@	@	@	AB	@	ABC	EF+A

Таблица 3 - активные сигналы для переходов

Вар.	Условия переходов														
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
20	-	-	4,3,2	0	5,7	-	5	-	-	-	-	-	-	-	-

Этап 1

Задание:

По диаграмме переходов автомата и описанию условий переходов и активных сигналов, определить тип управляющего автомата (автомат Мили или Мура, смешанный). Выбор обосновать.

Произвести кодирование состояний управляющего автомата. Составить схему переходов/состояний полученного автомата. Схему представить в отчете.

Ход работы:

Данный управляющий автомат является смешанным автоматом по причине зависимости некоторых выходных сигналов только от текущего состояния (признака автомата Мура), и некоторых от двух состояний – текущего и прошлого (автомат Мили).

Схема переходов/состояний полученного автомата с учетом значений из индивидуального задания приведена на рисунке 1.

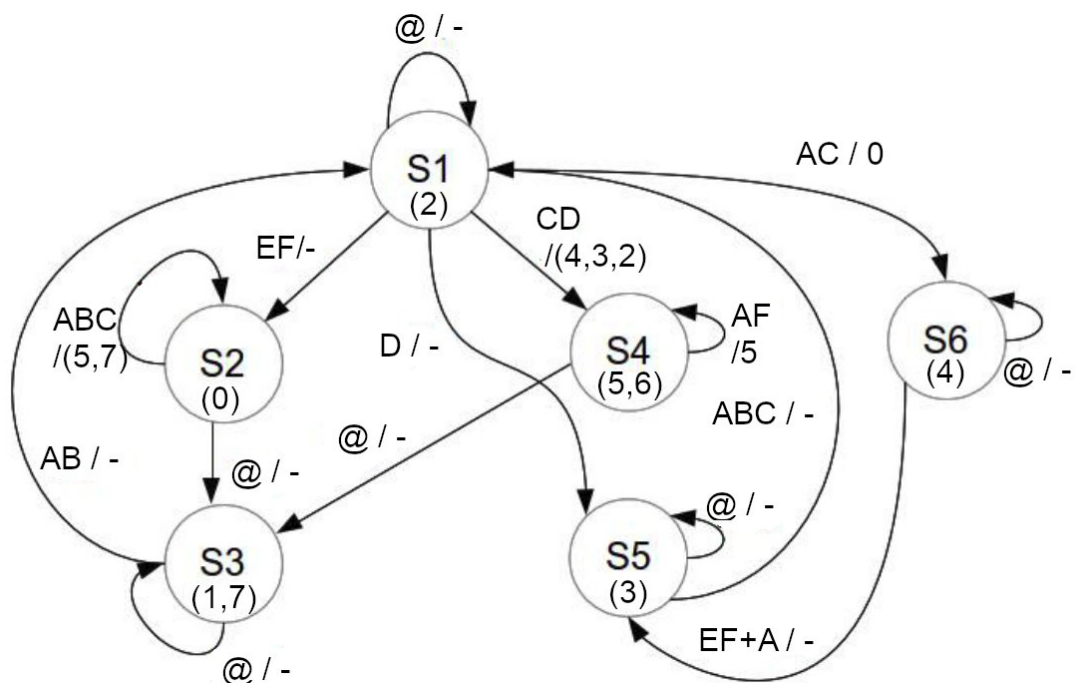


Рисунок 1 - схема переходов/состояний автомата

Этап 2

Задание:

Разработать описание устройства управления на языке VHDL, для чего использовать шаблоны для автоматов Мили и Мура. Разработать тестовое описание для устройства, представляющее собой генератор входных сигналов. Тестовое описание должно обеспечивать проверку всех ветвей автомата.

Ход работы:

Исходный код описания разработанного устройства приведен в листинге

1.

Листинг 1 – описание устройства

```
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 DOWNT0 0);
    CLK : IN std_logic; RST : IN std_logic;
    M : OUT std_logic_vector (7 DOWNT0 0) );
END control_unit;

ARCHITECTURE arch_control_unit 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 <= "00000100";
            current_state <= s1;
        ELSIF (CLK'EVENT AND CLK='1') THEN
            CASE current_state IS

                WHEN S1 =>
                    M <= "00000100";
                    IF (C(4)='1' AND C(5)='1') THEN
                        current_state <= S2;
                    ELSIF (C(2)='1') AND (C(3)='1') THEN
                        M <= "00011100";
                        current_state <= S4;
                    ELSIF (C(3)='1') THEN
                        current_state <= S5;
                    ELSIF (C(0)='1') AND (C(2)='1') THEN
                        M <= "00000001";
                        current_state <= S6;
                    ELSE
                        current_state <= S1;
                    END IF;

                WHEN S2 =>
                    M <= "00000001";
                    IF (C(0)='1' AND C(1)='1' AND C(2)='1') THEN
                        M <= "10100000";
                        current_state <= S2;
```

```

        ELSE
            current_state <= S3;
        END IF;

    WHEN S3 =>
        M <= "10000010";
        IF (C(0)='1' AND C(1)='1') THEN
            current_state <= S1;
        ELSE
            current_state <= S3;
        END IF;

    WHEN S4 =>
        M <= "01100000";
        IF (C(0)='1') AND (C(5)='1') THEN
            M <= "00100000";
            current_state <= S4;
        ELSE
            current_state <= S3;
        END IF;

    WHEN S5 =>
        M <= "00001000";
        IF (C(0)='1' AND C(1)='1' AND C(2)='1') THEN
            current_state <= S1;
        ELSE
            current_state <= S5;
        END IF;

    WHEN S6 =>
        M <= "00010000";
        IF ((C(4)='1' AND C(5)='1') OR C(0)='1') THEN
            current_state <= S5;
        ELSE
            current_state <= S6;
        END IF;
    END CASE;
END IF;
END PROCESS;
END arch_control_unit;

```

Тестовое описание, обеспечивающее проверку всех ветвей приведено в листинге 2.

Листинг 2 – тестовое описание

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

```

```

ENTITY control_unit_tb IS
END control_unit_tb;

ARCHITECTURE behavior OF control_unit_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT control_unit
    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 COMPONENT;

    --Inputs
    signal C : std_logic_vector(5 downto 0) := (others => '0');
    signal CLK : std_logic := '0';
    signal RST : std_logic := '0';

    --Outputs
    signal M : std_logic_vector(7 downto 0);

    -- Clock period definitions
    constant CLK_period : time := 10 ns;

BEGIN

    -- Instantiate the Unit Under Test (UUT)
    uut: control_unit PORT MAP (
        C => C,
        CLK => CLK,
        RST => RST,
        M => M
    );

    -- Clock process definitions
    CLK_process :process
    begin
        CLK <= '0';
        wait for CLK_period/2;
        CLK <= '1';
        wait for CLK_period/2;
    end process;

    -- Stimulus process
    stim_proc: process

```

```

begin
  C <= "000000"; --S1 -> S1
  RST <= '0';
  WAIT FOR CLK_PERIOD;
  RST <= '1';

  -- Loop 1 --
  C <= "000000"; --S1 -> S1
  WAIT FOR CLK_PERIOD;
  C <= "001100"; --S1 -> S4
  WAIT FOR CLK_PERIOD;
  C <= "100001"; --S4 -> S4
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S4 -> S3
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S3 -> S3
  WAIT FOR CLK_PERIOD;
  C <= "000011"; --S3 -> S1
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S1 -> S1
  WAIT FOR CLK_PERIOD*5;

  -- Loop 2 --

  C <= "110000"; --S1 -> S2
  WAIT FOR CLK_PERIOD;
  C <= "000111"; --S2 -> S2
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S2 -> S3
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S3 -> S3
  WAIT FOR CLK_PERIOD;
  C <= "000011"; --S3 -> S1
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S1 -> S1
  WAIT FOR CLK_PERIOD*5;

  -- Loop 3 --

  C <= "000101"; --S1 -> S6
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S6 -> S6
  WAIT FOR CLK_PERIOD;
  C <= "110000"; --S6 -> S5
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S5 -> S5
  WAIT FOR CLK_PERIOD;
  C <= "000111"; --S5 -> S1
  WAIT FOR CLK_PERIOD;
  C <= "000000"; --S1 -> S1
  WAIT FOR CLK_PERIOD*5;

```

```

-- Loop 4 --

C <= "001000"; --S1 -> S5
WAIT FOR CLK_PERIOD;
C <= "000111"; --S5 -> S1
WAIT FOR CLK_PERIOD;
C <= "000000"; --S1 -> S1
WAIT FOR CLK_PERIOD*5;

wait;
end process;

END;

```

Этап 3

Задание:

Выполнить моделирование полученного теста в ПО ModelSim PE. Результаты моделирования представить в отчете.

Ход работы:

Результаты моделирования приведены на рисунках 2-3. Из них следует, что описанное устройство работает корректно.

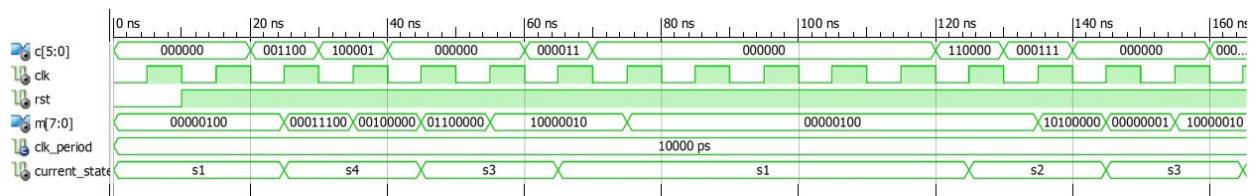


Рисунок 2 – временная диаграмма тестирования

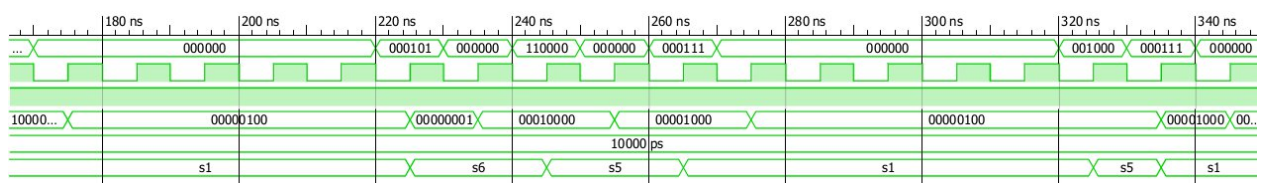


Рисунок 3 – временная диаграмма тестирования

Вывод: в ходе выполнения домашнего задания были закреплены навыки разработки и тестирования проектов устройств на языке VHDL (в данном случае – устройства управления с жесткой логикой на основе цифровых автоматов).