*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«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)

**Отчет**

**по лабораторной работе №3**

**Вариант 25**

**Дисциплина:**

Основы проектирования устройств ЭВМ

**Название домашней работы:**

Разработка VHDL-приложения на основе конечного автомата

Студент гр. ИУ6-62Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Бурлаков**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2019

# ВВедение

## Цель работы

В ходе выполнения домашнего задания необходимо разработать устройство управления схемного типа, обрабатывающий входное командное слово С={ABCDEF} и выдающий сигналы управления M={M0,…,Mk-1} операционному блоку в соответствии с приведенной в индивидуальном задании логикой работы.

# основная часть

**Этап 1.**

А. По диаграмме переходов автомата (Приложение 1) и описанию условий переходов и активных сигналов (дополнительный файл варианты.pdf), определить тип управляющего автомата (автомат Мили или Мура, смешанный). Выбор обосновать.

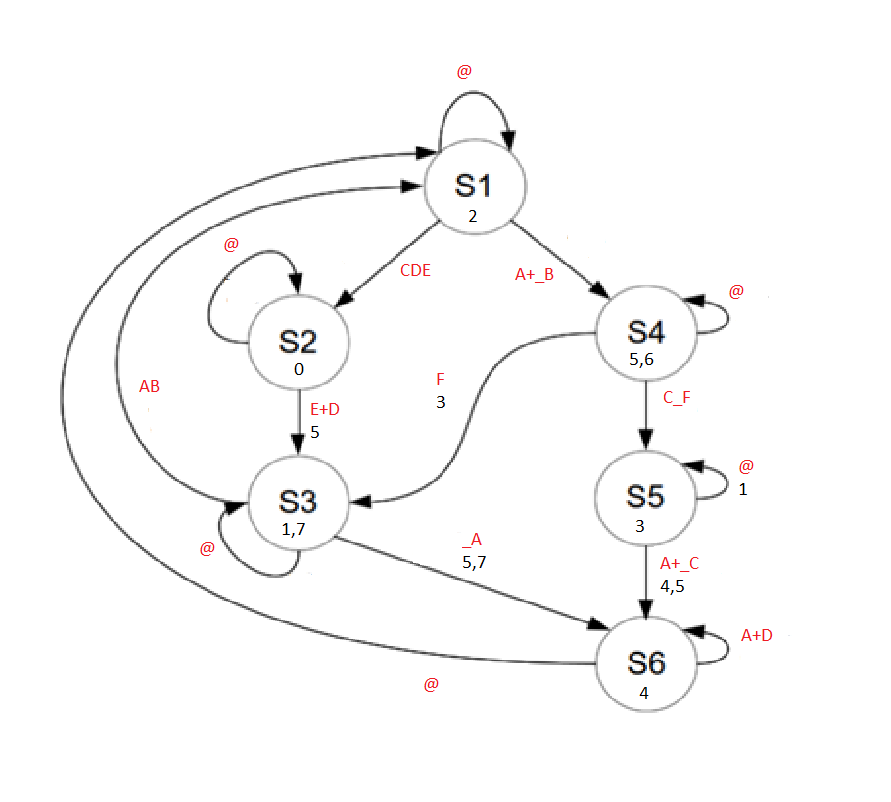


Рисунок 1- Диаграмма переходов автомата





Рисунок 2 – Активные сигналы в состояниях





Рисунок 3 – Таблица переходов





Рисунок 4 – Активные сигналы при переходах

Так как в заданном варианте присутствует выдача выходного сигнала как при нахождении в состоянии, так и при переходах от одного состояния к другому, было принято решение об использовании смешанного управляющего автомата.

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

**Этап 2.** Разработать описание устройства управления на языке VHDL, для чего использовать приведенные в Приложении 2 шаблоны для автоматов Мили и Мура. Разработать тестовое описание для устройства, представляющее собой генератор входных сигналов (см. Приложение 3). Тестовое описание должно обеспечивать проверку всех ветвей автомата.

Листинг 1 – Код автомата

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.std\_logic\_arith.all;

ENTITY control\_unit IS

PORT(

U : IN std\_logic\_vector ( 5 DOWNTO 0 );

clk : IN std\_logic;

rst : IN std\_logic;

V : OUT std\_logic\_vector ( 7 DOWNTO 0 ) );

END control\_unit;

ARCHITECTURE moore OF control\_unit IS

TYPE STATE\_TYPE IS (s1, s2, s3, s4, s5, s6);

SIGNAL current\_state : STATE\_TYPE;

BEGIN

clocked\_proc : PROCESS (clk, rst)

BEGIN

IF (RST = '1') THEN

current\_state <= s1;

V <= "00000000";

ELSIF (clk'EVENT AND clk ='1') THEN

CASE current\_state IS

WHEN s1 =>

V <= "00100000";

IF (U(4 downto 2) = "111") THEN

current\_state <= s2;

ELSIF (U(0) = '1' OR U(1) = '0') THEN

current\_state <= s4;

ELSE

current\_state <= s1;

END IF;

WHEN s2 =>

IF (U(4) = '1' OR U(3) = '1') THEN

current\_state <= s3;

V <= "10000100";

ELSE

current\_state <= s2;

V <= "10000000";

END IF;

WHEN s3 =>

IF (U(1 downto 0) = "11") THEN

current\_state <= s1;

V <= "01000001";

ELSIF (U(0) = '0') THEN

current\_state <= s6;

V <= "01000101";

ELSE

current\_state <= s3;

V <= "01000001";

END IF;

WHEN s4 =>

IF (U(2) = '1' AND U(5) = '0') THEN

current\_state <= s5;

V <= "00000110";

ELSIF (U(5) = '1') THEN

current\_state <= s3;

V <= "00010110";

ELSE

current\_state <= s4;

V <= "00000110";

END IF;

WHEN s5 =>

IF (U(0) = '1' OR U(2) = '0') THEN

current\_state <= s6;

V <= "00011100";

ELSE

current\_state <= s5;

V <= "01010000";

END IF;

WHEN s6 =>

V <= "00001000";

IF (U(0) = '1' OR U(3) = '1') THEN

current\_state <= s6;

ELSE

current\_state <= s1;

END IF;

WHEN OTHERS =>

current\_state <= s1;

END CASE;

END IF;

END PROCESS clocked\_proc;

END moore;

Листинг 2 – Код тестирующей программы

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.std\_logic\_unsigned.all;

USE ieee.numeric\_std.ALL;

ENTITY test IS

END test;

ARCHITECTURE behavior OF test IS

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

COMPONENT control\_unit

PORT(

U : IN std\_logic\_vector(5 downto 0);

clk : IN std\_logic;

rst : IN std\_logic;

V : OUT std\_logic\_vector(7 downto 0)

);

END COMPONENT;

--Inputs

signal U : std\_logic\_vector(5 downto 0) := (others => '0');

signal clk : std\_logic := '0';

signal rst : std\_logic := '0';

--Outputs

signal V : std\_logic\_vector(7 downto 0);

-- Clock period definitions

constant clk\_period : time := 10ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: control\_unit PORT MAP (

U => U,

clk => clk,

rst => rst,

V => V

);

-- 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

-- hold reset state for 100ns.

rst<='1';

wait for 100ns;

rst<='0';

wait for clk\_period/4;

-- insert stimulus here

U<="111010";

wait for clk\_period;

U<="111110";

wait for clk\_period;

U<="100110";

wait for clk\_period;

U<="101110";

wait for clk\_period;

U<="101101";

wait for clk\_period;

U<="101111";

wait for clk\_period;

U<="101011";

wait for clk\_period;

U<="001011";

wait for clk\_period;

U<="001111";

wait for clk\_period;

U<="001110";

wait for clk\_period;

U<="001111";

wait for clk\_period;

U<="000111";

wait for clk\_period;

U<="000110";

wait for clk\_period;

U<="000011";

wait for clk\_period;

U<="100011";

wait for clk\_period;

U<="100010";

wait for clk\_period;

U<="000000";

wait;

end process;

END;

**Этап 3.** Выполнить моделирование полученного теста в ПО ModelSim PE. Результаты моделирования представить в отчете.

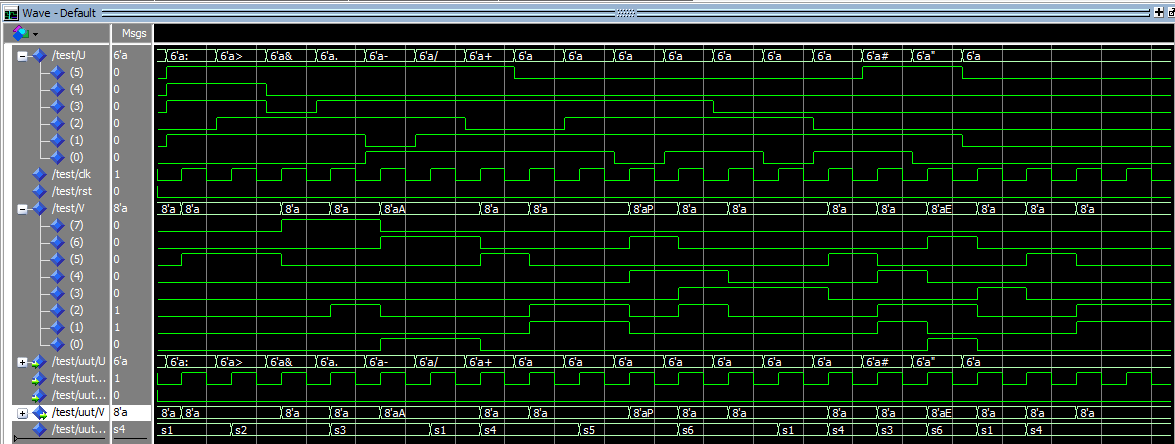


Рисунок 5 – Схема работы алгоритма

Таблица 1 – Таблица переходов, входов и выходов теста

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | A | B | C | D | E | F | cur\_St | next\_St | V\_next | V\_test |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | s1 | s1 | 2 | 2 |
| 2 | 0 | 1 | 1 | 1 | 1 | 1 | s1 | s2 | 2 | 2 |
| 3 | 0 | 1 | 1 | 0 | 0 | 1 | s2 | s2 | 0 | 0 |
| 4 | 0 | 1 | 1 | 1 | 0 | 1 | s2 | s3 | 05 | 05 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | s3 | s3 | 17 | 17 |
| 6 | 1 | 1 | 1 | 1 | 0 | 1 | s3 | s1 | 17 | 17 |
| 7 | 1 | 1 | 0 | 1 | 0 | 1 | s1 | s4 | 2 | 2 |
| 8 | 1 | 1 | 0 | 1 | 0 | 0 | s4 | s4 | 56 | 56 |
| 9 | 1 | 1 | 1 | 1 | 0 | 0 | s4 | s5 | 56 | 56 |
| 10 | 0 | 1 | 1 | 1 | 0 | 0 | s5 | s5 | 13 | 13 |
| 11 | 1 | 1 | 1 | 1 | 0 | 0 | s5 | s6 | 345 | 345 |
| 12 | 1 | 1 | 1 | 0 | 0 | 0 | s6 | s6 | 4 | 4 |
| 13 | 0 | 1 | 1 | 0 | 0 | 0 | s6 | s1 | 4 | 4 |
| 14 | 1 | 1 | 0 | 0 | 0 | 0 | s1 | s4 | 2 | 2 |
| 15 | 1 | 1 | 0 | 0 | 0 | 1 | s4 | s3 | 356 | 356 |
| 16 | 0 | 1 | 0 | 0 | 0 | 1 | s3 | s6 | 157 | 157 |

**Этап 4.**  В САПР Xilinx ISE 9.1 Web Pack создать проект описания устройства на основе ПЛИС Spartan 3 типа XC3S200 FT256 (набор XC3S200) или ПЛИС Spartan 3E500 FG320 (набор Nexys 2)

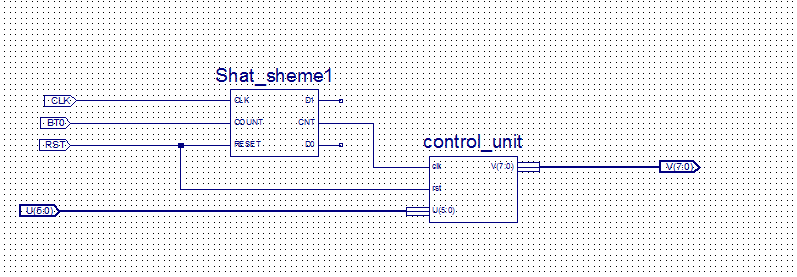


Рисунок 6 – Схема проектируемого автомата

Проведено программирование макетной ПЛИС и проведено тестирование разработанного устройства. Тестирование показало корректность работы разработанного устройства.

# Заключение

В ходе данной лабораторной работы были закреплены на практике теоретические знания о способах реализации устройств управления, исследованы способы организации узлов ЭВМ, освоены принципы проектирования цифровых устройств на основе ПЛИС.