|  |  |
| --- | --- |
| **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 (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**ОТЧЁТ**

**по домашней работе**

**Вариант №37**

**Дисциплина:** Основы проектирования устройств ЭВМ

|  |  |  |  |
| --- | --- | --- | --- |
| Студент | ИУ6-62Б |  | И.С. Марчук |
|  | группа | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |
| Преподаватель |  |  | С.В. Ибрагимов |
|  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

**ВВЕДЕНИЕ**

**Цель работы**

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

**ОСНОВНАЯ ЧАСТЬ**

**Условие**

Таблица 1 - Варианты диаграмм и активных сигналов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Вариант** | **Диаграмма**  **переходов** | **Активные сигналы M в состоянии** | | | | | |
| **S1** | **S2** | **S3** | **S4** | **S5** | **S6** |
| 37 | 1 | 2 | 0 | 1,7 | 5,6 | 3 | 4 |

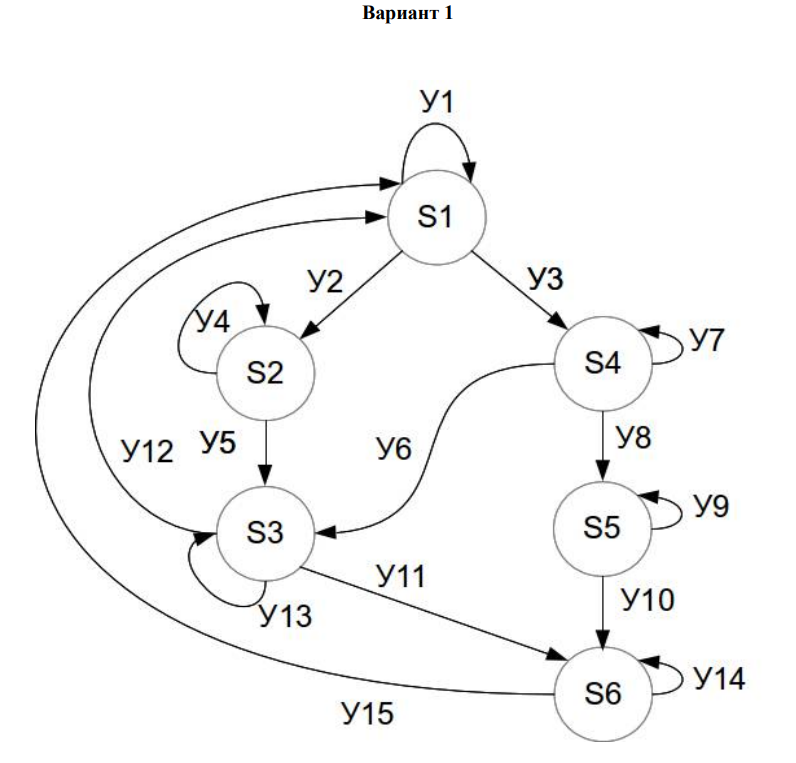


Рисунок 1 – Диаграмма переходов

Таблица 2 - Условия переходов и наименование отладочной платы

(«@» - иначе, «\_X» - НЕ X, «+» - ИЛИ, 1- безусловный переход)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вариант** | **Название**  **отладочной**  **платы** | **Активные сигналы в М состоянии** | | | | | | | | | | | | | | |
| **Y1** | **Y2** | **Y3** | **Y4** | **Y5** | **Y6** | **Y7** | **Y8** | **Y9** | **Y10** | **Y11** | **Y12** | **Y13** | **Y14** | **Y15** |
| 37. | Spartan3 | @ | ABC | A\_B | @ | E+D | F | @ | D\_F | @ | A+\_C | \_A | AB | @ | @ | 1 |

Таблица 3 - Активные сигналы для переходов

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вариант** | **Активные сигналы в М состоянии** | | | | | | | | | | | | | | |
| **Y1** | **Y2** | **Y3** | **Y4** | **Y5** | **Y6** | **Y7** | **Y8** | **Y9** | **Y10** | **Y11** | **Y12** | **Y13** | **Y14** | **Y15** |
| 37 | - | - | - | - | 5,6 | 5,7 | - | 4 | 6,7 | 2 | 5 | - | - | - | - |

**Этап 1**

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.

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

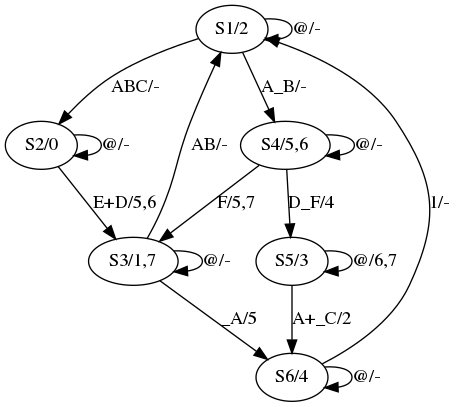


Рисунок 2 - Схема автомата

Предложенный автомат является смешанным, так как некоторые выходные сигналы зависят только от текущего состояния (признака автомата Мура), и некоторых от двух состояний – текущего и прошлого (автомат Мили)

**Этап 2**

Разработать описание устройства управления на языке VHDL, для чего использовать приведенные в Приложении 2 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 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='1') THEN

M <= "00000000";

current\_state <= s1;

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

CASE current\_state IS

WHEN S1 =>

M <= "00000100";

IF (C(0)='1' AND C(1)='1' AND C(2)='1') THEN

current\_state <= S2;

ELSIF (C(0)='1' AND C(1)='0') THEN

current\_state <= S4;

ELSE

current\_state <= S1;

END IF;

WHEN S2 =>

M <= "00000001";

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

M <= "01100000";

current\_state <= S3;

ELSE

current\_state <= S2;

END IF;

WHEN S3 =>

M <= "10000010";

IF (C(0)='1' AND C(1)='1') THEN

current\_state <= S1;

ELSIF (C(0)='0') THEN

M <= "00100000";

current\_state <= S6;

ELSE

current\_state <= S3;

END IF;

WHEN S4 =>

M <= "01100000";

IF (C(3)='1' AND C(5)='0') THEN

M <= "00010000";

current\_state <= S5;

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

M <= "10100000";

current\_state <= S3;

ELSE

current\_state <= S4;

END IF;

WHEN S5 =>

M <= "00001000";

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

M <= "00000100";

current\_state <= S6;

ELSE

M <= "11000000";

current\_state <= S5;

END IF;

WHEN S6 =>

M <= "00010000";

current\_state <= S1;

END CASE;

END IF;

END PROCESS;

END arch\_control\_unit;

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 |
| 0 | - | - | - | - | - | S1 -> S1 | 2 |
| 1 | 1 | 1 | - | - | - | S1 -> S2 | 2 |
| - | - | - | 0 | 0 | - | S2 -> S2 | 0 |
| - | - | - | - | 1 | - | S2 -> S3 | 5,6 |
| 1 | 0 | - | - | - | - | S3 -> S3 | 1,7 |
| 1 | 1 | - | - | - | - | S3 -> S1 | 1,7 |
| 1 | 0 | - | - | - | - | S1 -> S4 | 2 |
| - | - | - | 0 | - | 0 | S4 -> S4 | 5,6 |
| - | - | - | 1 | - | 0 | S4 -> S5 | 4 |
| 0 | - | 1 | - | - | - | S5 -> S5 | 6,7 |
| 1 | - | - | - | - | - | S5 -> S6 | 2 |
| - | - | - | - | - | - | S6 -> S1 | 4 |
| 1 | 0 | - | - | - | - | S1 -> S4 | 2 |
| - | - | - | - | - | 1 | S4 -> S3 | 5,7 |
| 0 | - | - | - | - | - | S3 -> S6 | 5 |

Пропущенные значения в таблице могут принимать любой значение 0, или 1.

Листинг 2 – Тестовое описания для устройства

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 arch\_test OF test 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;

WAIT FOR CLK\_PERIOD;

C <= "000111"; --S1 -> S2

WAIT FOR CLK\_PERIOD;

C <= "000000"; --S2 -> S2

WAIT FOR CLK\_PERIOD;

C <= "010000"; --S2 -> S3

WAIT FOR CLK\_PERIOD;

C <= "000001"; --S3 -> S3

WAIT FOR CLK\_PERIOD;

C <= "000011"; --S3 -> S1

WAIT FOR CLK\_PERIOD;

C <= "000001"; --S1 -> S4

WAIT FOR CLK\_PERIOD;

C <= "000000"; --S4 -> S4

WAIT FOR CLK\_PERIOD;

C <= "001000"; --S4 -> S5

WAIT FOR CLK\_PERIOD;

C <= "000100"; --S5 -> S5

WAIT FOR CLK\_PERIOD;

C <= "000001"; --S5 -> S6

WAIT FOR CLK\_PERIOD;

C <= "000000"; --S6 -> S1

WAIT FOR CLK\_PERIOD;

C <= "000001"; --S1 -> S4

WAIT FOR CLK\_PERIOD;

C <= "100000"; --S4 -> S3

WAIT FOR CLK\_PERIOD;

C <= "000000"; --S3 -> S6

WAIT FOR CLK\_PERIOD;

WAIT;

END PROCESS SIM\_PROC;

END ARCHITECTURE;

**Этап 3**

Установить ПО ModelSim PE (или аналогичный продукт: Xilinx ISE, Altera Quartus).

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

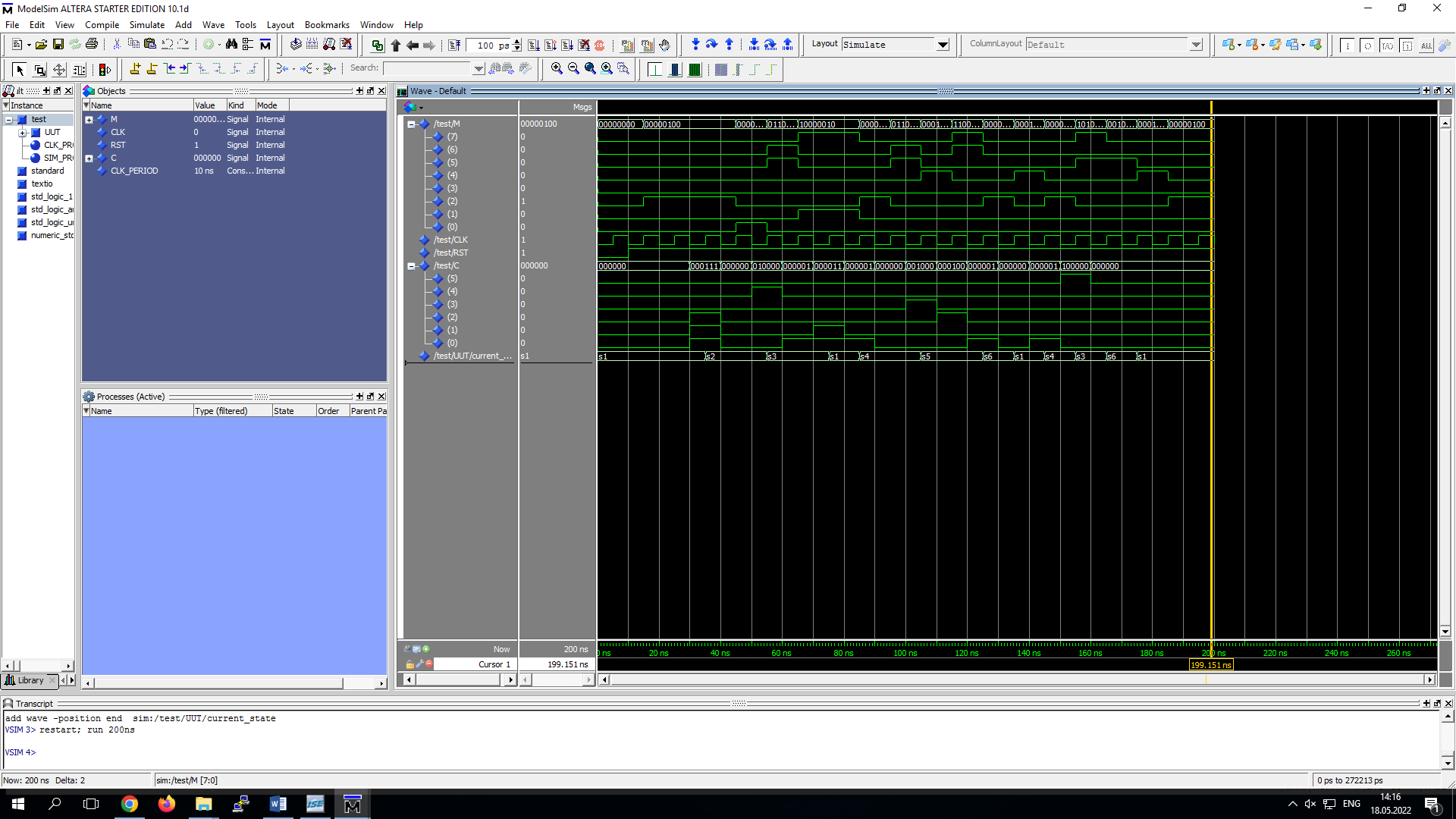


Рисунок 3 – Результаты моделирования

Из полученных диаграмм можно увидеть, что формируются активные выходные сигналы для каждого состояния. Достигнуты все конечные состояния автомата.

**Вывод:**

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