Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных средств

Лабораторная работа №4

«Описание и моделирование триггеров и конечных автоматов»

Вариант 8

Выполнила студент Проверил:

группы 450701 Бибило П.Н.

Никитин Г.Ю.

Минск, 2016

1. Задание:

**Часть 1**. С использованием девятизначного алфавита STD\_LOGIC составить VHDL-модель и провести моделирование **триггера** двумя способами:

* По логической схеме триггера (структурное описание);
* По таблице функционирования триггера (алгоритмическое описание).

Сравнить результаты моделирования.

**Часть 2**. С использованием перечислимых типов составить VHDL-модель **конечного автомата** и провести моделирование двумя способами:

* С помощью моделирующей программы;
* С помощью скрипта (в последнем случае предполагается использование системы моделирования ModelSim).

Часть 1.

Двухтактный сканируемый T-триггер со сбросом

|  |  |  |
| --- | --- | --- |
| Имя | Входы | Выходы |
| FQTR\_1 | C, R, Ti, Te | Q |

## FQTR_1

## Таблица истинности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R | Te | Ti | C | Q |
| 1 | - | - | [--] | 0 |
| 0 | 1 | - | [01] | (Ti) |
| 0 | 0 | - | [01] | ^(Q) |
| 0 | - | - | [1-] | N |
| 0 | - | - | [00] | N |

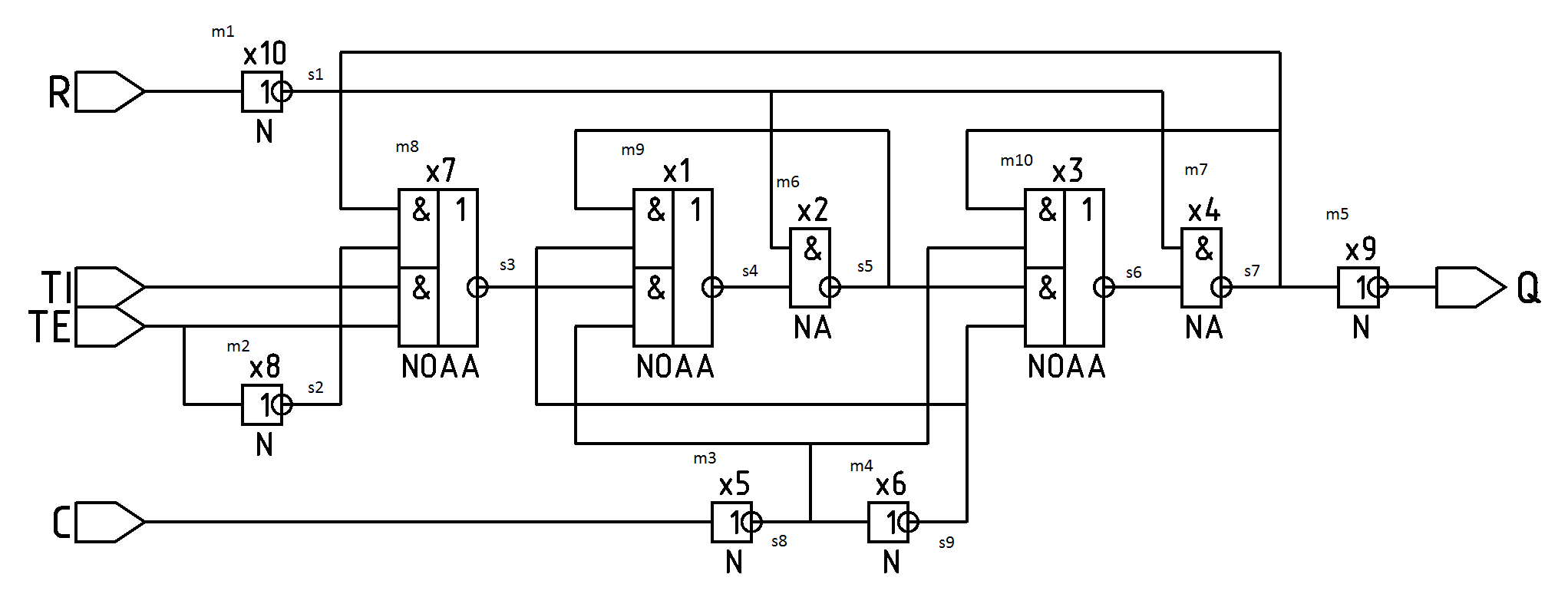


Рис. 1 – Логическая схема триггера (Вариант 8)

## Обозначения в таблице функционирования триггера

Через “-” обозначено *любое* из (0,1) значение сигнала; в квадратных скобках показывается *изменение* сигнала (обычно синхросигнала), например, через [01] обозначается передний фронт сигнала (сигнал меняется из 0 в 1); через [10] обозначается задний фронт сигнала (сигнал меняется из 1 в 0); символ N обозначает *неизменяемое* (предыдущее) значение сигнала; символ ^ является знаком инверсии (отрицания).

**Часть 2. Описание и моделирование конечного автомата**

*Составить VHDL-описание конечного автомата Мили, заданного совмещенной таблицей переходов.*

Алфавит состояний A = {a1, a2, a3, a4}. Начальное состояние автомата - a1. Входной алфавит Z образуют сигналы z1, z2, z3,  т.е. Z = {z1 , z2 , z3}. Выходной алфавит W образуют сигналы w1, ..., w5, т.е. W = {w1, w2 ,w3 ,w4,w5}. На пересечении строки zn и столбца aq в таблице находится состояние as , в которое должен перейти автомат из состояния aq под воздействием сигнала zn. После косой черты в этой же графе таблицы указывается выходной сигнал, выдаваемый автоматом в состоянии aq при поступлении на его вход сигнала zn .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица задания функционирования конечного автомата | | | | |
| Входные сигналы | Состояния | | | |
| a1 | a2 | a3 | a4 |
| z1 | a4/w4 | a2/w1 | a1/w2 | a4/w4 |
| z2 | a4/w5 | a1/w3 | a3/w4 | a3/w3 |
| z3 | a1/w2 | a2/w4 | a1/w4 | a4/w1 |

1. VHDL-модель логической схемы:

А) Часть 1. Триггер.

Файл n:

library ieee;

use ieee.std\_logic\_1164.all;

entity n is

port (x:in std\_logic;

y:out std\_logic);

end;

architecture beh of n is

begin

y<= not x after 1 ns;

end;

Файл na:

library ieee;

use ieee.std\_logic\_1164.all;

entity na is

port (x1, x2:in std\_logic;

y:out std\_logic);

end;

architecture beh of na is

begin

y<=x1 nand x2 after 2 ns;

end;

Файл noaa

library ieee;

use ieee.std\_logic\_1164.all;

entity noaa is

port (x1, x2, x3, x4:in std\_logic;

y:out std\_logic);

end;

architecture beh of noaa is

begin

y<=((x1 and x2) nor (x3 and x4)) after 4 ns;

end;

Файл tfun:

library ieee;

use ieee.std\_logic\_1164.all;

entity tfun is

port (r, ti, te, c:in std\_logic;

q:out std\_logic);

end;

architecture beh of tfun is

begin

p0:process(r, ti, te, c)

begin

if (r='0') then

if (c'event and c='1') then

if (te='0') then q<=not ti;

elsif (te='1') then q<=ti;

end if;

end if;

else q<='0';

end if;

end process;

end;

Файл tstr:

library ieee;

use ieee.std\_logic\_1164.all;

entity tstr is

port (r, ti, te, c:in std\_logic;

q:out std\_logic);

end;

architecture beh of tstr is

component n

port (x:in std\_logic;

y:out std\_logic);

end component;

component na

port (x1, x2:in std\_logic;

y:out std\_logic);

end component;

component noaa

port (x1, x2, x3, x4:in std\_logic;

y:out std\_logic);

end component;

signal s: std\_logic\_vector(1 to 9);

begin

m1:n port map(r, s(1));

m2:n port map(te, s(2));

m3:n port map(c, s(8));

m4:n port map(s(8), s(9));

m5:n port map (s(7), q);

m6: na port map(s(1), s(4), s(5));

m7: na port map(s(1), s(6), s(7));

m8: noaa port map(s(7), s(2), ti, te, s(3));

m9: noaa port map(s(5), s(9), s(3), s(8), s(4));

m10: noaa port map(s(7), s(8), s(5), s(9), s(6));

end;

Файл testtrigger:

library ieee;

use ieee.std\_logic\_1164.all;

entity testtrigger is

end;

architecture atest of testtrigger is

component tstr

port (r, ti, te, c:in std\_logic;

q:out std\_logic);

end component;

signal r, ti, te, c, q: std\_logic;

begin

m1:tstr port map(r=>r, ti=>ti, te=>te, c=>c, q=>q);

r<='1',

'0' after 200 ns;

c<='0',

'1' after 50 ns,

'0' after 100 ns,

'1' after 150 ns,

'0' after 200 ns,

'1' after 250 ns,

'0' after 300 ns,

'1' after 350 ns,

'0' after 400 ns,

'1' after 450 ns,

'0' after 500 ns,

'1' after 550 ns,

'0' after 600 ns,

'1' after 650 ns,

'0' after 700 ns,

'1' after 750 ns,

'0' after 800 ns,

'1' after 850 ns,

'0' after 900 ns,

'1' after 950 ns,

'0' after 1000 ns,

'1' after 1050 ns,

'0' after 1100 ns,

'1' after 1150 ns,

'0' after 1200 ns,

'1' after 1250 ns,

'0' after 1300 ns,

'1' after 1350 ns,

'0' after 1400 ns;

te<='0',

'1' after 400 ns;

ti<='0',

'1' after 100 ns,

'0' after 200 ns,

'1' after 300 ns,

'0' after 400 ns,

'1' after 500 ns,

'0' after 600 ns,

'1' after 700 ns,

'0' after 800 ns;

end;

Файл testtrigger будет идентичным для функционального и структурного описания триггера, за тем лишь исключением, что необходимо поменять tstr на tfun в описании компонентов.

В итоге у нас получились следующие временные диаграммы (Рис. 2 и 3).

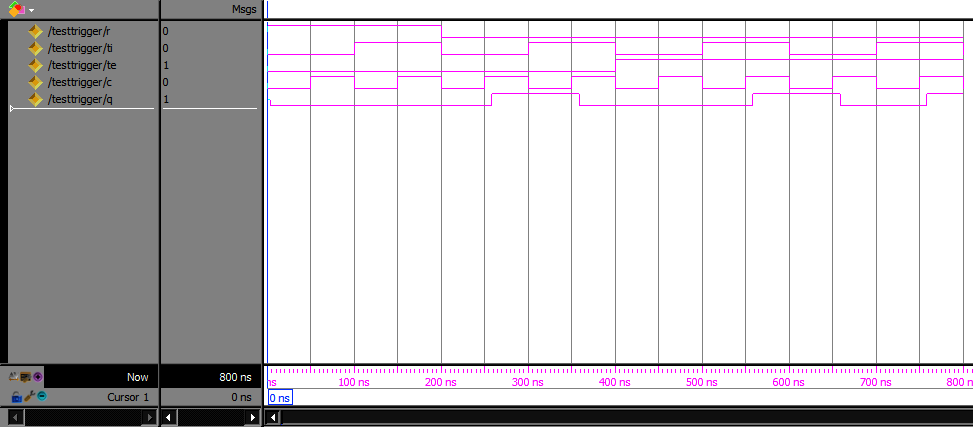


Рис. 2 – Временная диаграмма для структурного описания

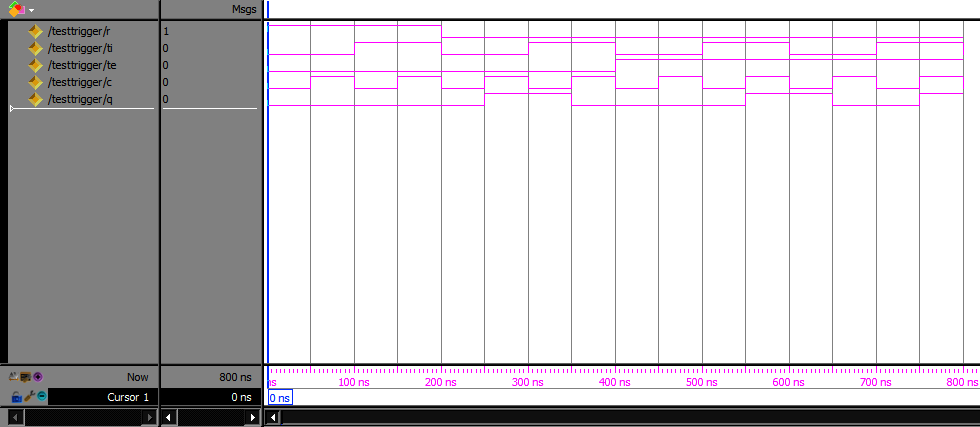


Рис. 3 – Временная диаграмма для функционального описания

Б) Часть 2. Конечный автомат.

Файл types

package types is

type mili\_in\_type is (z1, z2, z3);

type mili\_out\_type is (w1, w2, w3, w4, w5);

type T\_state is (a1, a2, a3, a4);

end;

Файл mili

library work;

use work.types.all;

entity mili is

port(z: in mili\_in\_type;

clock: in bit;

w: out mili\_out\_type;

next\_state1, state1:out T\_state);

end;

architecture beh of mili is

signal state, next\_state: T\_state;

begin

state1<=state;

next\_state1<=next\_state;

NS: process (state, z)

begin

case state is

when a1=> if (z=z1)then next\_state <= a4; w<=w4;

elsif (z=z2) then next\_state <= a4; w<=w5;

elsif (z=z3) then next\_state <= a1; w<=w2;

end if;

when a2=> if (z=z1) then next\_state <= a2; w<=w1;

elsif (z=z2) then next\_state <= a1; w<=w3;

elsif (z=z3) then next\_state <= a2; w<=w4;

end if;

when a3=> if (z=z1) then next\_state <= a1; w<=w2;

elsif (z=z2) then next\_state <= a3; w<=w4;

elsif (z=z3) then next\_state <= a1; w<=w4;

end if;

when a4=> if (z=z1) then next\_state <= a4; w<=w4;

elsif (z=z2) then next\_state <= a3; w<=w3;

elsif (z=z3) then next\_state <= a4; w<=w1;

end if;

end case;

end process;

clk: process(clock)

begin

if (clock'event and clock='1') then

state<=next\_state;

end if;

end process;

end;

Файл test

library work;

use work.types.all;

entity test is

end;

architecture atest of test is

component mili

port(z: in mili\_in\_type;

clock: in bit;

w: out mili\_out\_type;

next\_state1, state1:out T\_state);

end component;

signal z: mili\_in\_type;

signal clock: bit;

signal w: mili\_out\_type;

signal next\_state1, state1: T\_state;

begin

m1: mili port map

(z=>z, clock=>clock, w=>w, next\_state1=>next\_state1, state1=>state1);

z<=z1,

z2 after 100 ns,

z3 after 200 ns,

z2 after 300 ns,

z3 after 400 ns,

z1 after 500 ns,

z2 after 600 ns,

z2 after 700 ns;

clock<=not clock after 50 ns;

end;

Файл script.tcl

vsim -gui work.test

add wave sim:/test/\*

force z z1 100, z2 200, z3 300, z2 400, z3 400, z1 500, z2 600

force clock 0 50, 1 100 -repeat 50

view vawe

run 1600 ns

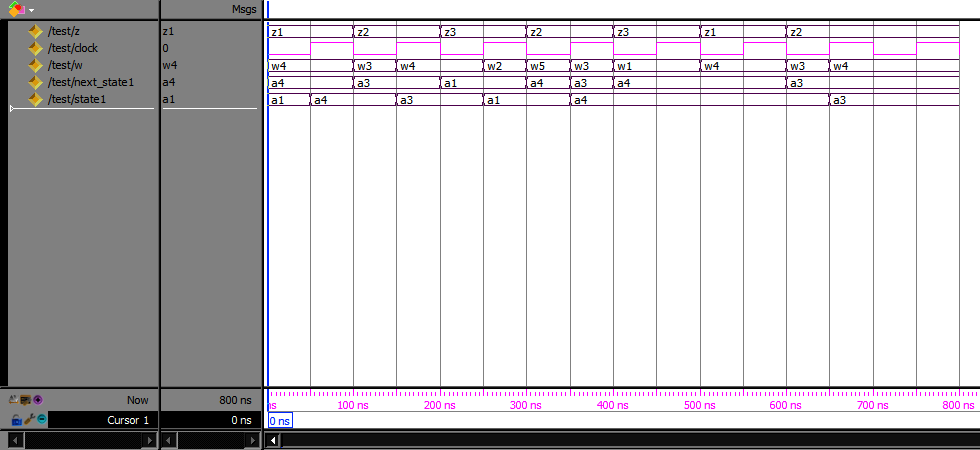


Рис. 4 – Временная диаграмма для конечного автомата

3.Анализ результатов и выводы.

Мы провели моделирование (структурное и функциональное для триггера) заданного триггера и конечного автомата в среде Modelsim SE и успешно выполнили выданное задание. В результате была получены верные временные диаграммы работы устройств.