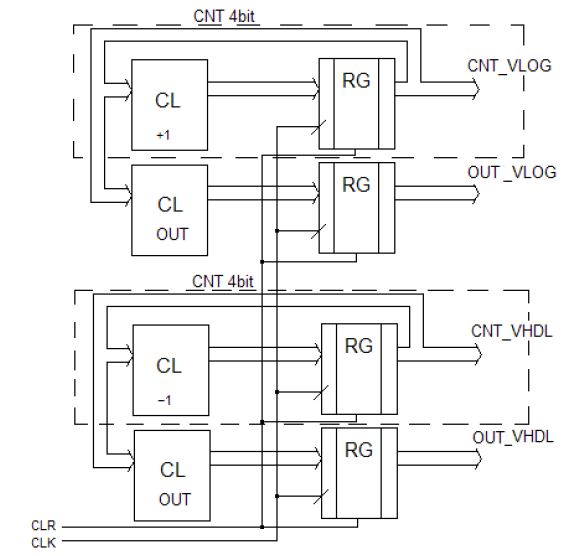
|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования"Московский технологический университет" **МИРЭА** | |
| Институт информационных технологий (ИТ) | |
| Кафедра вычислительной техники (ВТ) | |

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ**  **О выполнении лабораторной работы №2** | | |
| **по дисциплине** | | |
| **«**Моделирование**»**  *Вариант 39* | | |
|  | | |
| Выполнил студент группы ИВБО-03-14 | *Кривошея М.С.* | |
| Преподаватель | *Казанцева Л.В.* | |

**Постановка задачи**

В лабораторной работе требуется построить цифровой узел с элементами памяти, представляющий собой генератор последовательности. Последовательность генерируется в виде бесконечного циклического набора 4 разрядных слов. Генератор представляет собой счетчик имеющий модуль счета **16**. Каждому состоянию ставится в соответствие некоторая выходная комбинация.

Структурная схема генератора последовательности:



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CNT\_VLOG** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **A** | **B** | **C** | **D** | **E** | **F** |
| **OUT\_VLOG** | E | C | 0 | E | 4 | 5 | F | C | 6 | 6 | 9 | 9 | 3 | 1 | B | 8 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CNT\_VHDL** | **F** | **E** | **D** | **C** | **B** | **A** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| **OUT\_VHDL** | 8 | B | 1 | 3 | 9 | 9 | 6 | 6 | C | F | 5 | 4 | E | 0 | C | E |

**Решение**

**1. ТOP-модуль на Verilog**

`timescale 1ns / 1ps

module TOP(

input CLK,

input RST,

output [3:0] CNT\_VHDL,

output [3:0] OUT\_VHDL,

output [3:0] CNT\_VLOG,

output reg [3:0] OUT\_VLOG

);

reg [3:0] STATE;

always @ (posedge CLK, posedge RST)

begin

if(RST) begin

STATE <= 0;

OUT\_VLOG <= 4'hE;

end else

STATE <= STATE + 1;

case (STATE)

4'h0 : OUT\_VLOG <= 4'hC;

4'h1 : OUT\_VLOG <= 4'h0;

4'h2 : OUT\_VLOG <= 4'hE;

4'h3 : OUT\_VLOG <= 4'h4;

4'h4 : OUT\_VLOG <= 4'h5;

4'h5 : OUT\_VLOG <= 4'hF;

4'h6 : OUT\_VLOG <= 4'hC;

4'h7 : OUT\_VLOG <= 4'h6;

4'h8 : OUT\_VLOG <= 4'h6;

4'h9 : OUT\_VLOG <= 4'h9;

4'hA : OUT\_VLOG <= 4'h9;

4'hB : OUT\_VLOG <= 4'h2;

4'hC : OUT\_VLOG <= 4'h1;

4'hD : OUT\_VLOG <= 4'hB;

4'hE : OUT\_VLOG <= 4'h8;

4'hF : OUT\_VLOG <= 4'hE;

endcase

end

assign CNT\_VLOG = STATE;

VHDL INT\_MOD(

.CLK(CLK),

.RST(RST),

.CNT\_O(CNT\_VHDL),

.VAR\_O(OUT\_VHDL));

endmodule

**2. Модуль INT MOD на VHDL**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity VHDL is

Port ( CLK : in STD\_LOGIC;

RST : in STD\_LOGIC;

CNT\_O : out STD\_LOGIC\_VECTOR (3 downto 0);

VAR\_O : out STD\_LOGIC\_VECTOR (3 downto 0));

end VHDL;

architecture Behavioral of VHDL is

signal STATE : STD\_LOGIC\_VECTOR (3 downto 0);

begin

CNT : process(CLK, RST) begin

if(RST = '1') then

STATE <= "1111";

VAR\_O <= "1000";

elsif(CLK'event and CLK = '1') then

STATE <= STATE - 1;

case STATE is

when "1111" => VAR\_O <= "1011";

when "1110" => VAR\_O <= "0001";

when "1101" => VAR\_O <= "0011";

when "1100" => VAR\_O <= "1001";

when "1011" => VAR\_O <= "1001";

when "1010" => VAR\_O <= "0110";

when "1001" => VAR\_O <= "0110";

when "1000" => VAR\_O <= "1100";

when "0111" => VAR\_O <= "1111";

when "0110" => VAR\_O <= "0101";

when "0101" => VAR\_O <= "0100";

when "0100" => VAR\_O <= "1110";

when "0011" => VAR\_O <= "0000";

when "0010" => VAR\_O <= "1100";

when "0001" => VAR\_O <= "1110";

when "0000" => VAR\_O <= "1000";

when others => null;

end case;

End if;

end process;

CNT\_O <= STATE;

end Behavioral;

**3. Тестовое окружение**

`timescale 1ns / 1ps

module Test\_top;

reg CLK;

reg RST;

wire [3:0] CNT\_VHDL;

wire [3:0] OUT\_VHDL;

wire [3:0] CNT\_VLOG;

wire [3:0] OUT\_VLOG;

TOP uut (

.CLK(CLK),

.RST(RST),

.CNT\_VHDL(CNT\_VHDL),

.OUT\_VHDL(OUT\_VHDL),

.CNT\_VLOG(CNT\_VLOG),

.OUT\_VLOG(OUT\_VLOG)

);

always begin

CLK = 0; #50;

CLK = 1; #50;

end

initial begin

CLK = 0; RST = 1; #50;

CLK = 0; RST = 0;

end

endmodule

**4. Результаты верификации**

