|  |  |  |
| --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | |
| Факультет компьютерных систем и сетей | | |
| Кафедра электронных вычислительных средств | | |
| Отчет по лабораторной работе №3 по курсу «Микропроцессорные системы» | | |
| Выполнил  студент гр. |  | Проверил  Ключеня В. В. |
| Минск 2015 | | |

# Цель работы

Изучить основы составления VHDL проектов в среде Xilinx ISE для Spartan-3 FPGA.

# Задание

Составить VHDL описание секундомера c выводом на семисегментный индикатор в среде Xilinx ISE для Spartan-3 FPGA.

# Ход работы

Структурная схема устройства представлена на рисунке 1.

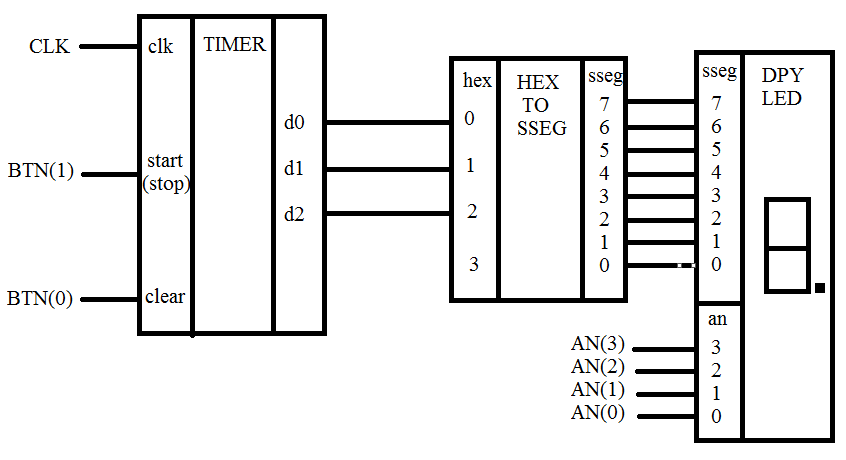


Рисунок 1 – Структурная схема

Ниже приведено описание секундомера:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity stop\_watch is

port(

clk: in std\_logic;

go, clr: in std\_logic;

d2, d1, d0: out std\_logic\_vector(3 downto 0)

);

end stop\_watch;

architecture cascade\_arch of stop\_watch is

constant DVSR: integer:=5000000;

signal ms\_reg, ms\_next: unsigned(22 downto 0);

signal d2\_reg, d1\_reg, d0\_reg: unsigned(3 downto 0);

signal d2\_next, d1\_next, d0\_next: unsigned(3 downto 0);

signal d1\_en, d2\_en, d0\_en: std\_logic;

signal ms\_tick, d0\_tick, d1\_tick: std\_logic;

begin

-- register

process(clk)

begin

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

ms\_reg <= ms\_next;

d2\_reg <= d2\_next;

d1\_reg <= d1\_next;

d0\_reg <= d0\_next;

end if;

end process;

-- next-state logic

-- 0.1 sec tick generator: mod-5000000

ms\_next <=

(others=>'0') when clr='1' or

(ms\_reg=DVSR and go='1') else

ms\_reg + 1 when go='1' else

ms\_reg;

ms\_tick <= '1' when ms\_reg=DVSR else '0';

-- 0.1 sec counter

d0\_en <= '1' when ms\_tick='1' else '0';

d0\_next <=

"0000" when (clr='1') or (d0\_en='1' and d0\_reg=9) else

d0\_reg + 1 when d0\_en='1' else

d0\_reg;

d0\_tick <= '1' when d0\_reg=9 else '0';

-- 1 sec counter

d1\_en <= '1' when ms\_tick='1' and d0\_tick='1' else '0';

d1\_next <=

"0000" when (clr='1') or (d1\_en='1' and d1\_reg=9) else

d1\_reg + 1 when d1\_en='1' else

d1\_reg;

d1\_tick <= '1' when d1\_reg=9 else '0';

-- 10 sec counter

d2\_en <=

'1' when ms\_tick='1' and d0\_tick='1' and d1\_tick='1' else

'0';

d2\_next <=

"0000" when (clr='1') or (d2\_en='1' and d2\_reg=5) else

d2\_reg + 1 when d2\_en='1' else

d2\_reg;

-- output logic

d0 <= std\_logic\_vector(d0\_reg);

d1 <= std\_logic\_vector(d1\_reg);

d2 <= std\_logic\_vector(d2\_reg);

end cascade\_arch;

Далее приведен код, позволяющий вывести результат работы на семисегментные индикаторы:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.numeric\_std.all;

entity disp\_hex\_mux is

port(

clk, reset: in std\_logic;

hex3, hex2, hex1, hex0: in std\_logic\_vector(3 downto 0);

dp\_in: in std\_logic\_vector(3 downto 0);

an: out std\_logic\_vector(3 downto 0);

seg: out std\_logic\_vector(7 downto 0)

);

end disp\_hex\_mux ;

architecture arch of disp\_hex\_mux is

-- each 7-seg led enabled (2^18/4)\*25 ns (40 ms)

constant N: integer:=18;

signal q\_reg, q\_next: unsigned(N-1 downto 0);

signal sel: std\_logic\_vector(1 downto 0);

signal hex: std\_logic\_vector(3 downto 0);

signal dp: std\_logic;

begin

-- register

process(clk,reset)

begin

if reset='1' then

q\_reg <= (others=>'0');

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

q\_reg <= q\_next;

end if;

end process;

-- next-state logic for the counter

q\_next <= q\_reg + 1;

-- 2 MSBs of counter to control 4-to-1 multiplexing

sel <= std\_logic\_vector(q\_reg(N-1 downto N-2));

process(sel,hex0,hex1,hex2,hex3,dp\_in)

begin

case sel is

when "00" =>

an <= "1110";

hex <= hex0;

dp <= dp\_in(0);

when "01" =>

an <= "1101";

hex <= hex1;

dp <= dp\_in(1);

when "10" =>

an <= "1011";

hex <= hex2;

dp <= dp\_in(2);

when others =>

an <= "0111";

hex <= hex3;

dp <= dp\_in(3);

end case;

end process;

-- hex-to-7-segment led decoding

with hex select

seg(7 downto 1) <=

"0000001" when "0000",

"1001111" when "0001",

"0010010" when "0010",

"0000110" when "0011",

"1001100" when "0100",

"0100100" when "0101",

"0100000" when "0110",

"0001111" when "0111",

"0000000" when "1000",

"0000100" when "1001",

"0001000" when "1010", --a

"1100000" when "1011", --b

"0110001" when "1100", --c

"1000010" when "1101", --d

"0110000" when "1110", --e

"0111000" when others; --f

-- decimal point

seg(0) <= dp;

end arch;

Далее приведен код тестирующей программы:

library ieee;

use ieee.std\_logic\_1164.all;

entity stop\_watch\_test is

port(

clk: in std\_logic;

btn: in std\_logic\_vector(3 downto 0);

an: out std\_logic\_vector(3 downto 0);

seg: out std\_logic\_vector(7 downto 0)

);

end stop\_watch\_test;

architecture arch of stop\_watch\_test is

signal d2, d1, d0: std\_logic\_vector(3 downto 0);

begin

disp\_unit: entity work.disp\_hex\_mux

port map(

clk=>clk, reset=>'0',

hex3=>"0000", hex2=>d2,

hex1=>d1, hex0=>d0,

dp\_in=>"1101", an=>an, seg=>seg);

watch\_unit: entity work.stop\_watch(cascade\_arch)

port map(

clk=>clk, go=>btn(1), clr=>btn(0),

d2 =>d2, d1=>d1, d0=>d0 );

end arch;

# Вывод

В ходе выполнения данной лабораторной работы для секундомера было cоставлено структурное VHDL-описание, был сгенерирован файл прошивки для Spartan-3 FPGA и оценен результат работы непосредственно на устройстве.