**First Term Exam.**

**15/Sep/2015**

# Name:

Gerardo Daniel Naranjo Gallegos. A01209499.

# VHDL code for 74LS42:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity First\_Term\_Exam is

Port ( A : in STD\_LOGIC;

B : in STD\_LOGIC;

C : in STD\_LOGIC;

D : in STD\_LOGIC;

Outputs : out STD\_LOGIC\_VECTOR (0 to 9));

end First\_Term\_Exam;

architecture Behavioral of First\_Term\_Exam is

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

begin

Ent <= D & C & B & A;

Outputs <= "0111111111" when Ent <= x"0" else

"1011111111" when Ent <= x"1" else

"0101111111" when Ent <= x"2" else

"0110111111" when Ent <= x"3" else

"0111011111" when Ent <= x"4" else

"0111101111" when Ent <= x"5" else

"0111110111" when Ent <= x"6" else

"0111111011" when Ent <= x"7" else

"0111111101" when Ent <= x"8" else

"0111111110" when Ent <= x"9" else

"1111111111" when Ent <= x"10" else

"1111111111" when Ent <= x"11" else

"1111111111" when Ent <= x"12" else

"1111111111" when Ent <= x"13" else

"1111111111" when Ent <= x"14" else

"1111111111";

end Behavioral;

# VHDL TestBench for 74LS42:

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY First\_Term\_Exam\_tb IS

END First\_Term\_Exam\_tb;

ARCHITECTURE behavior OF First\_Term\_Exam\_tb IS

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

COMPONENT First\_Term\_Exam

PORT(

A : IN std\_logic;

B : IN std\_logic;

C : IN std\_logic;

D : IN std\_logic;

Outputs : OUT std\_logic\_vector(0 to 9)

);

END COMPONENT;

--Inputs

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

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

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

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

--Outputs

signal Outputs : std\_logic\_vector(0 to 9);

-- No clocks detected in port list. Replace <clock> below with

-- appropriate port name

-- constant <clock>\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: First\_Term\_Exam PORT MAP (

A => A,

B => B,

C => C,

D => D,

Outputs => Outputs

);

-- Clock process definitions

-- <clock>\_process :process

-- begin

-- <clock> <= '0';

-- wait for <clock>\_period/2;

-- <clock> <= '1';

-- wait for <clock>\_period/2;

-- end process;

**-- Stimulus process**

**stim\_proc: process**

**begin**

**-- hold reset state for 100 ns.**

**wait for 100 ns;**

**-- wait for <clock>\_period\*10;**

**-- insert stimulus here**

**A<='0'; B<='0'; C<='0'; D<='0'; wait for 100 ns;**

**A<='1'; B<='0'; C<='0'; D<='0'; wait for 100 ns;**

**A<='0'; B<='1'; C<='0'; D<='0'; wait for 100 ns;**

**A<='1'; B<='1'; C<='0'; D<='0'; wait for 100 ns;**

**A<='0'; B<='0'; C<='1'; D<='0'; wait for 100 ns;**

**A<='1'; B<='0'; C<='1'; D<='0'; wait for 100 ns;**

**A<='0'; B<='1'; C<='1'; D<='0'; wait for 100 ns;**

**A<='1'; B<='1'; C<='1'; D<='0'; wait for 100 ns;**

**A<='0'; B<='0'; C<='0'; D<='1'; wait for 100 ns;**

**A<='1'; B<='0'; C<='0'; D<='1'; wait for 100 ns;**

**A<='0'; B<='1'; C<='0'; D<='1'; wait for 100 ns;**

**A<='1'; B<='1'; C<='0'; D<='1'; wait for 100 ns;**

**A<='0'; B<='0'; C<='1'; D<='1'; wait for 100 ns;**

**A<='1'; B<='0'; C<='1'; D<='1'; wait for 100 ns;**

**A<='0'; B<='1'; C<='1'; D<='1'; wait for 100 ns;**

**A<='1'; B<='1'; C<='1'; D<='1'; wait for 100 ns;**

**wait;**

**end process;**

END;

# Screen capture of the simulation:

