

**Department of Electrical Engineering**

**Optional 2: BCD to 7 Segments Display**

Name: Socheath Sok, ID: 014470701

Professor: Dr. Ray Wang

Class: EE 301

Date Due: December 11, 2017

**Explanation**

The goal of the lab is to learn how to show BCD (binary coded decimal) on a 7 segments display. The binary input are control manually by manipulating the switches and will be send through a converter to be change to its equivalent decimal value. The result will, then, be displayed on one of the 7 segments given. Knowing that all the segment are connected to a common anode, a low value (‘0’) is needed to light up each LEDs, so to make the converter, the number of LEDs needed to light up to represent the decimal values were determined and assigned to each binary numbers from one to nine and a letter ‘E’ will be displayed for anything higher. However, with this set up, all four segments will light up and display the same value. To turn three of them off, three more inputs were created and connected to three of the anodes. Next, the design was saved and used as a component along with a counter in a structural code, so that the output can be control by a clock input.

**Code for J-K flip flop**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity jkff\_2 is

Port ( j : in STD\_LOGIC;

k : in STD\_LOGIC;

clr : in STD\_LOGIC;

clk : in STD\_LOGIC;

q,qn: out STD\_LOGIC);

end jkff\_2;

architecture Behavioral of jkff\_2 is

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

begin

Process (clk,j,k,s,clr)

begin

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

if clr = '0' then s <= '0';

else

if j = '0' and k = '0' then s <= s;

elsif j = '0' and k = '1' then s <= '0';

elsif j = '1' and k = '0' then s <= '1';

Else s <= not s;

end if;

end if;

end if;

q <= s;

qn <= not s;

end process;

end Behavioral;

**Code for Up Counter**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity counter is

Port ( clock,clr : in STD\_LOGIC;

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

end counter;

architecture Behavioral of counter is

component jkff\_2

Port ( j : in STD\_LOGIC;

k : in STD\_LOGIC;

clr : in STD\_LOGIC;

clk : in STD\_LOGIC;

q,qn: out STD\_LOGIC);

end component;

signal s : std\_logic\_vector(3 downto 0) := "0000";

signal a,b,c,d: std\_logic ;

begin

R1: jkff\_2 port map('1','1',clock,clr,s(0),a);

R2: jkff\_2 port map('1','1',a,clr,s(1),b);

R3: jkff\_2 port map('1','1',b,clr,s(2),c);

R4: jkff\_2 port map('1','1',c,clr,s(3),d);

Q <= s;

end Behavioral;

**Code for Converter**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

entity L is

Port ( x : in STD\_LOGIC\_VECTOR (3 downto 0);

z1,z2,z3 : in std\_logic;

y : out STD\_LOGIC\_VECTOR (6 downto 0));

end L;

architecture Behavioral of L is

begin

with x select y <=

not "0111111" when "0000", --0

not "0000110" when "0001", --1

not "1011011" when "0010", --2

not "1001111" when "0011", --3

not "1100110" when "0100", --4

not "1101101" when "0101", --5

not "1111100" when "0110", --6

not "0000111" when "0111", --7

not "1111111" when "1000", --8

not "1100111" when "1001", --9

not "1111001" when others; --Error

end Behavioral;

**Structural Code**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity bcd\_counter is

Port ( clock,z1,z2,z3,clr : in STD\_LOGIC;

y : out STD\_LOGIC\_VECTOR (6 downto 0));

end bcd\_counter;

architecture Behavioral of bcd\_counter is

component counter

Port ( clock,clr : in STD\_LOGIC;

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

end component;

component L

Port ( x : in STD\_LOGIC\_VECTOR (3 downto 0);

z1,z2,z3 : in STD\_LOGIC;

y : out STD\_LOGIC\_VECTOR (6 downto 0));

end component;

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

begin

T1: counter port map (clock,clr,Q);

R1: L port map (Q,z1,z2,z3,y);

end Behavioral;

**TestBench**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

ENTITY bcd\_counter\_test IS

END bcd\_counter\_test;

ARCHITECTURE behavior OF bcd\_counter\_test IS

COMPONENT bcd\_counter

PORT(

clock : IN std\_logic;

z1 : IN std\_logic;

z2 : IN std\_logic;

z3 : IN std\_logic;

clr : IN std\_logic;

y : OUT std\_logic\_vector(6 downto 0) );

END COMPONENT;

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

signal z1 : std\_logic:= '0';

signal z2 : std\_logic:= '0';

signal z3 : std\_logic:= '0';

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

signal y : std\_logic\_vector(6 downto 0);

constant clock\_period : time := 10 ns;

BEGIN

uut: bcd\_counter PORT MAP (

clock => clock,

z1 => z1,

z2 => z2,

z3 => z3,

clr => clr,

y => y );

clock\_process :process

begin

clock <= '0';

wait for clock\_period/2;

clock <= '1';

wait for clock\_period/2;

end process;

stim\_proc: process

begin

clr <= '0';

wait for 5 ns;

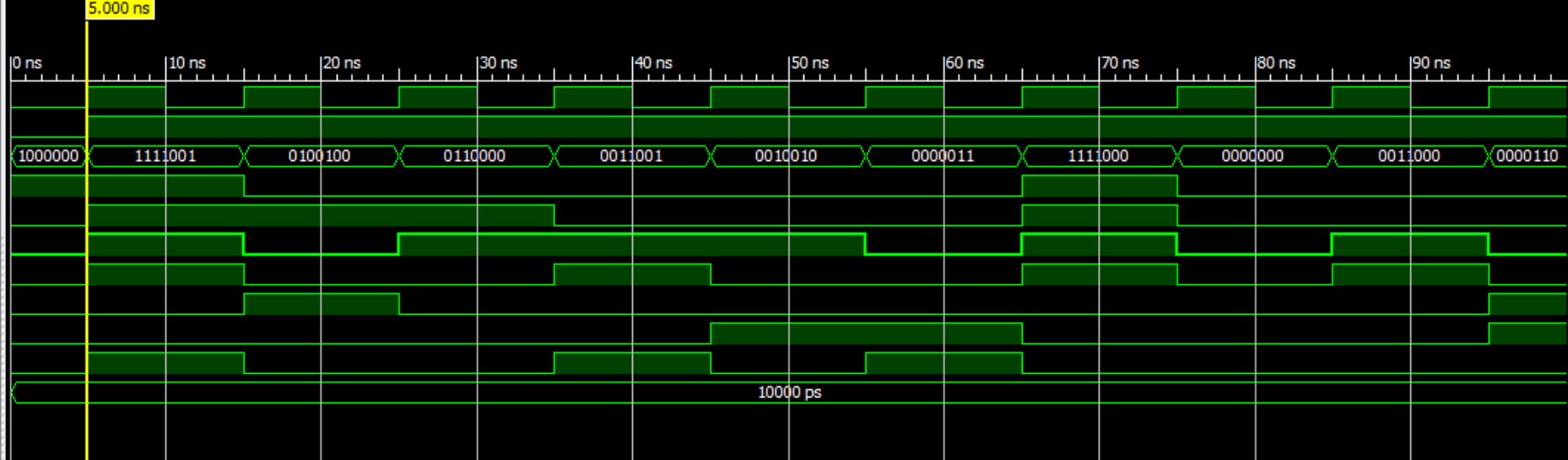
clr <='1';

wait ;

end process;

end ;

**WaveForm**



**Constraint File**

NET "clock" LOC = "G12";

NET "clock" clock\_dedicated\_route = false;

NET "clr" LOC = "P11";

NET "z1" LOC = "K14";

NET "z2" LOC = "M13";

NET "z3" LOC = "J12";

NET "y(0)" LOC = "L14";

NET "y(1)" LOC = "H12";

NET "y(2)" LOC = "N14";

NET "y(3)" LOC = "N11";

NET "y(4)" LOC = "P12";

NET "y(5)" LOC = "L13";

NET "y(6)" LOC = "M12";