

**Department of Electrical Engineering**

**Optional 3: Selectable Counter**

Name: Socheath Sok, ID: 014470701

Professor: Dr. Ray Wang

Class: EE 301

Date Due: December 11, 2017

**Explanation**

For this lab, the goal is to change the asynchronous up counter, created in lab 5, into a selectable counter that can perform both up and down counting modes. The mode of the counter will control by an input and that input signal will be combined with the output Q into and AND gate. The inverse signal will combine with the output Qn (the inverse of Q) into another AND gate. The outputs of both gates will then become the inputs to a third AND gate with its output connected to the clock input of the second J-K flip flop. The output of the third AND is also one of the main output listed in the structural code. The same set up will be connected to the clocks of the third and fourth flip flops. For the first flip flip, the clock input will be connected to the main clock signal and as for the last flip flop, Q will not be connected to anything and will be last output. Qn, on the other hand, will remains open.

**Code for JK 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;

**Structural Code**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity UC\_2 is

Port ( clock, m : in STD\_LOGIC;

clr : in STD\_LOGIC;

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

end UC\_2;

architecture Behavioral of UC\_2 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,x,y,z: std\_logic :='0';

begin

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

x <= (s(0) and not m) or (a and m);

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

y <= (s(1) and not m) or (b and m);

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

z <= (s(2) and nott m) or (c and m);

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

Q <= s;

end Behavioral;

**TestBench**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_unsigned.ALL;

USE ieee.std\_logic\_arith.ALL;

ENTITY UC\_2T IS

END UC\_2T;

ARCHITECTURE behavior OF UC\_2T IS

COMPONENT UC\_2

Port ( clock, m : in STD\_LOGIC;

clr : in STD\_LOGIC;

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

END COMPONENT;

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

signal clr : std\_logic ;

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

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

constant clock\_period : time := 10 ns;

BEGIN

uut: UC\_2 PORT MAP (

clock => clock,

Q => Q,

m => m,

clr => clr );

clock\_process :process

BEGIN

clock <= '1';

wait for clock\_period/2;

clock <= '0';

wait for clock\_period/2;

end process;

stim\_proc: process

begin

clr <= '0';

wait for 20 ns;

clr <= '1'; m <= '0';

wait for 50 ns;

clr <= '0'; m <= '1';

wait for 20 ns;

clr <= '1'; m <= '1';

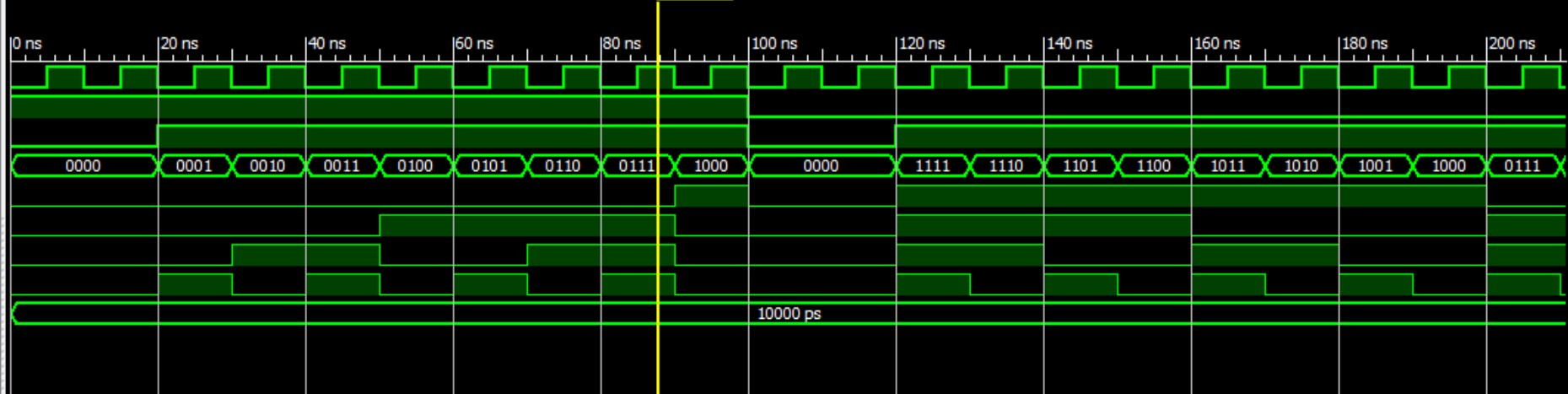
wait for 50 ns;

wait ;

end process;

END;

**WaveForm**



**Constraint File**

NET "clock" CLOCK\_DEDICATED\_ROUTE = FALSE;

NET "clock" LOC = "G12";

NET "Q(0)" LOC = "M5" ;

NET "Q(1)" LOC = "M11" ;

NET "Q(2)" LOC = "P7" ;

NET "Q(3)" LOC = "P6" ;

NET "m" LOC = "P11" ;

NET "clr" LOC = "N3";