Course Name: __**Embedded Systems**___

Course Number and Section: **14:332:493:03**

Experiment: [HW#6 – Chapter 11]

Lab Instructor: Professor Phil Southard

Date Submitted: 03/14/2019

Submitted by: [Timothy Langer RUID# 189005424]

Q2)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity question2 is
Port(
clk, X1, X2 : in std_logic;
Z : out std_logic;
Y : out std_logic_vector(1 downto 0)
 );
end question2;

architecture Behavioral of question2 is
type state_type is (STA, STB, STC);
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of state_type: type is "10 11 01";
signal PS, NS : state_type;

begin
sync_proc: process( clk, NS)
begin
if rising_edge(clk) then
PS <= NS;
end if;
end process;

comb_proc: process(X1, X2, PS)
begin
Z<='0';
CASE PS is

when STC =>
      if X2 = '1' then NS <= STB; z <='0';
      else NS <= STA; z <='1';
      end if;

when STA =>
      z<='0';
      if X1 ='1' then NS<= STC;
      else NS <= STA;
      end if;

when STB =>
      if X2 ='1' then NS <= STB; z<='0';
```

```
else NS <= STA; z <='1';
      end if;
when others => NS<=STA;

end case;

end process;

with PS select
Y<= "10" when STA,
      "01" when STC,
      "11" when STB,
      "10" when others;
end Behavioral;
```

Q4)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity question4 is
Port(
clk, X1, X2 : in std_logic;
Z1, Z2 : out std_logic
 );
end question4;

architecture Behavioral of question4 is
type state_type is (STA, STB, STC);

signal PS, NS : state_type;


begin

sync_proc: process( clk, NS)
begin
if rising_edge(clk) then
PS <= NS;
end if;
end process;

comb_proc: process(X1, X2, PS)
begin
Z1<='0'; Z2 <='0';
CASE PS is

when STA =>
     if X1 = '1' then NS <= STB; Z2 <='1';
     else NS <= STC; Z2 <='0';
     end if;

when STB =>
     if X2 ='1' then NS<= STA; Z2<='0';
     else NS <= STC; Z2<='1';
     end if;

when STC =>
     Z2 <='1';
     if X1 ='1' then NS <= STB;
```

```
   else NS <= STA;
      end if;
when others => NS<=STA;

end case;

end process;

with PS select
Z1<= '0' when STA,
      '1' when STB,
      '1' when STC,
      '0' when others;
end Behavioral;
```

Q6

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity question6 is
    Port ( clk : in STD_LOGIC;
           X : in STD_LOGIC;
           Z1, Z2 : out STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (1 downto 0));
end question6;

architecture Behavioral of question6 is
type state_type is (ST0, ST1, ST2, ST3);
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of state_type: type is "00 10 01 11";
signal PS , NS : state_type;
begin

sync_proc: process( clk, NS)
begin
if rising_edge(clk) then
PS <= NS;
end if;
end process;

comb_proc: process(X, PS)
begin
Z1<='0';
CASE PS is

when ST0 =>
    Z1 <= '1';
    Z2 <='0';
    if X = '1' then NS <= ST0;
    else NS <= ST1;
    end if;

when ST1 =>
    Z1 <='1';
    Z2 <='0';
    if X ='1' then NS<= ST0;
    else NS <= ST2;
    end if;
```

```
when ST2 =>
      Z1 <='0';
      Z2 <='0';
      if X ='1' then NS <= ST2;
      else NS <= ST3;
      end if;

when ST3 =>
      Z1 <='0';
      if X ='1' then NS <= ST2; Z2 <='0';
      else NS <= ST0; Z2 <='1';
      end if;

when others => NS<=ST0;

end case;


end process;
with PS select
Y<= "00" when ST0,
      "10" when ST1,
      "01" when ST2,
      "11" when ST3,
      "00" when others;

end Behavioral;
```

Q13)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity question13 is
Port(
clk, X1, X2 : in std_logic;
CS, RD : out std_logic;
Y : out std_logic_vector(2 downto 0)
 );
end question13;

architecture Behavioral of question13 is
type state_type is (STA, STB, STC);
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of state_type: type is "001 010 100";
signal PS, NS : state_type;

begin

sync_proc: process( clk, NS)
begin
if rising_edge(clk) then
PS <= NS;
end if;
end process;

comb_proc: process(X1, X2, PS)
begin
CS <='0'; RD <='0';
CASE PS is

when STA =>
      if X1 = '1' then NS <= STC; CS <='1'; RD <= '0';
      else NS <= STB; CS <='0'; RD <='1';
      end if;

when STB =>
      NS<=STC;
      CS <='1'; RD <='1';
```

```
when STC =>
      if X2 ='1' then NS <= STC; CS<='0'; RD<='1';
      else NS <= STA; CS <='0'; RD<='0';
      end if;
when others => NS<=STA;
end case;
end process;

with PS select
Y<= "001" when STA,
      "010" when STB,
      "100" when STC,
      "001" when others;
end Behavioral;
```