CPE 487: Digital System Design

HW6
Due: 5/2/18

1. Synthesis (20 points)

Angela needs to build a small module that takes 4 inputs*: x, y*, *p* and *q*. If *p='0'* then the output of the module should be *(x NOR y)*. Otherwise if *q='0'*, the output should be *(x XOR y)*. If neither *p* nor *q* is set to *'0'*, then it doesn't matter what the output is. Angela has written the following code:

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

entity module is
port( x,y,p,q: in std_logic;
        sdata: out std_logic);
end module;

architecture behavioral of module is

begin
        process(x,y,p,q)
        begin
                if p='0' then sdata <= x nor y;
                elsif q='0' then sdata <= x xor y;
                end if;
        end process;
end behavioral;
```

(a) Why will this code not produce a good synthesized result? (3 points)

(b) Enter the code into the Xilinx ISE software and run the synthesis utility. Check the RTL schematic of what is actually synthesized. Show the RTL schematic (5 points)

(c) How would you improve the code (show the code)? (5 points)

(d) Enter your new code and re-synthesize. Show the RTL schematic (5 points)

(e) What has changed? (2 points)

2. Synthesis (15 points)

Draw the logic inferred by the following model:

```
entity xyz  is
   Port ( A, B, C, D : in  STD_LOGIC;
           Z : out  STD_LOGIC);
end xyz;

architecture RTL of xyz is
signal S: STD_LOGIC_VECTOR (3 downto 0);
signal P: STD_LOGIC;
begin
        S(0) <= A and B when D = ‘1’ else
                A nor B;
SP:     process
        variable V: STD_LOGIC;
        begin
                wait until rising_edge(C);
                for i in 1 to 3 loop
                        V := S(i-1) or D;
                        S(i) <= not V;
                end loop;
                P <= not D;
        end process;
        Z <= S(3) and S(2) and P when C = ‘0’;
end RTL;
```