McGill University
Department of Electrical and Computer Engineering

Course: ECSE-323 Digital System Design **Fall 2008**

Assignment #3

TOPIC: VHDL for Combinational Circuits

Tutorial Session 1 (Monday)

1.- Write a *complete* VHDL description for the circuit that implements the function $f = (ab + bc + cd + ad) \oplus (b\bar{c}d) + a\bar{b}\bar{c}d$ using only ***a single selected signal assignment statement.***

```
entity Q1 is
     port(a,b,c,d  : in std_logic;
          f : out std_logic);
   end Q1;

   architecture A1 of Q1 is
   begin
     WITH a SELECT
          f <= ((b or(b and c)or d))xor(b and not c and d))or(not
   b and not c and d) WHEN '1',
          f <= ((b and c)or(c and d))xor(b and not c and d) WHEN
   '0',
               `0' WHEN others;
   end A1;
```

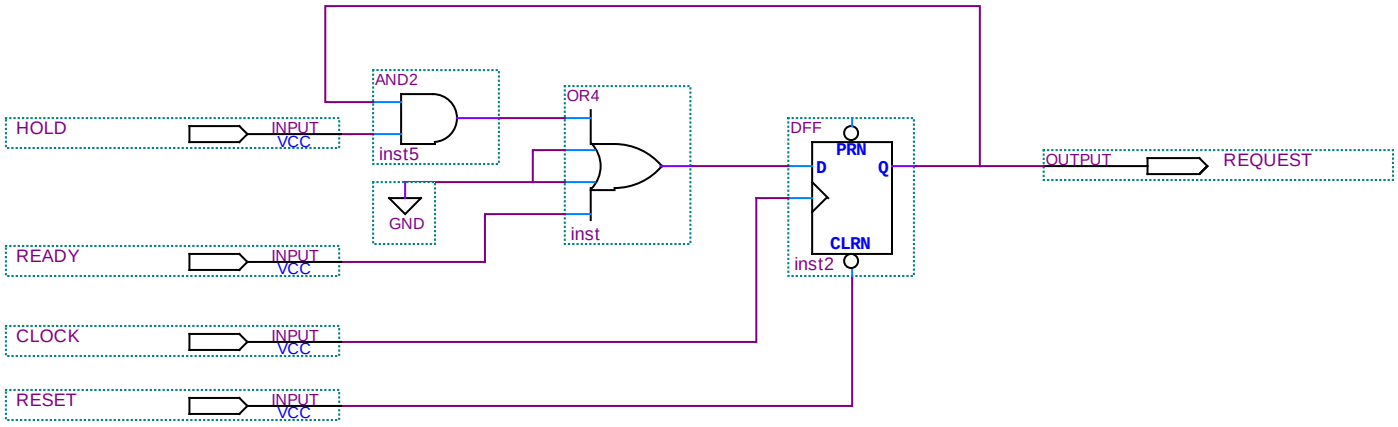
---

2. – Repeat question 1, but use only simple concurrent assignment statements.

```
   entity Q2 is
     port(a,b,c,d  : in std_logic;
          f : out std_logic);
   end Q2;

   architecture A2 of Q2 is
   begin
     f <= (((a and b)or(b and c)or(c and d))xor(b and not c and
d))or(a and not b and not c and d);
   end A2;
```

3.- Using component instantiation statements write a *complete* VHDL design entity for the system shown in the following diagram. It is not necessary for you to supply the design entities for the blocks.




```
entity Q3 is
   port(HOLD, READY, CLOCK, RESET : in std_logic;
        REQUEST    : out std_logic;
end Q3;

architecture A3 of Q3 is
component AND2
   port(a,b : in std_logic;
        f : out std_logic);
end component;
component OR4
   port(a,b,c,d : in std_logic;
        f : out std_logic);
end component;
component DFF
   port(D,clock,PRN,CLRN : in std_logic;
        Q : out std_logic);
end component;
signal Iand, Ior, Iq : std_logic;
begin
   G1 : AND2 port map (a=>Iq,b=HOLD,f=>Iand);
   G2 : OR4 port map (a=>Iand,b=>'0',c=>'0',d=>READY,f=>Ior);
   G3 : DFF port map (D=>Ior,clock=>CLOCK,CLRN=>RESET,Q=>Iq);
   REQUEST <= Iq;
end A3;
```

note: it would be an error to connect the output port REQUEST directly to the input of the AND gate, since out ports cannot be read. Thus we connect the output of the DFF to an internal signal, Iq, and connect Iq to both the output port REQUEST and the input of the AND gate.

Tutorial Session 2 (Wednesday)

1.- Write a *complete* VHDL description for the circuit that implements the function $f = (abcd + abd) \oplus (b + \overline{c}d)$ using only ***a single selected signal assignment statement.***

```
entity Q1 is
     port(a,b,c,d  : in std_logic;
          f : out std_logic);
  end Q1;

  architecture A1 of Q1 is
  begin
     WITH a SELECT
          f <= ((b and c and d) or (b or d))xor(b or (not c and
  d)) WHEN '1',
          f <= (b or (not c and d)) WHEN '0',
               `0' WHEN others;
  end A1;
```

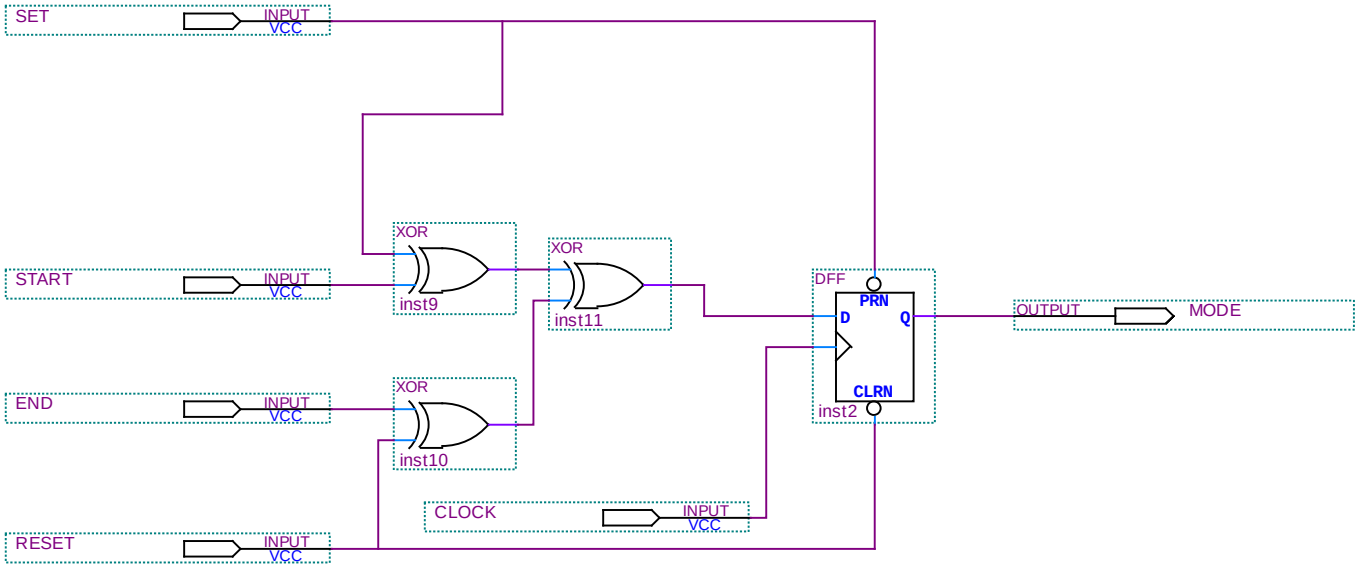
---

2. – Repeat question 1, but use only simple concurrent assignment statements.

```
  entity Q2 is
     port(a,b,c,d  : in std_logic;
          f : out std_logic);
  end Q2;

  architecture A2 of Q2 is
  begin
     f <= ((a and b and c and d) or (a and b and d)) xor (b or
(not c and d));
  end A2;
```

3.- Using component instantiation statements write a *complete* VHDL design entity for the system shown in the following diagram. It is not necessary for you to supply the design entities for the blocks.




```
entity Q3 is
   port(SET, START, END, CLOCK, RESET : in std_logic;
        MODE    : out std_logic;
end Q3;

architecture A3 of Q3 is
component XOR
   port(a,b : in std_logic;
        f : out std_logic);
end component;
component DFF
   port(D,clock,PRN,CLRN : in std_logic;
        Q : out std_logic);
end component;
signal Ix1, Ix2, Ix3 : std_logic;
begin
   G1 : XOR port map (a=>SET,b=START,f=>Ix1);
   G2 : XOR port map (a=>END,b=RESET,f=>Ix2);
   G3 : XOR port map (a=>Ix1,b=Ix2,f=>Ix3);
   G4 : DFF port map
(D=>Ix3,clock=>CLOCK,PRN=>SET,CLRN=>RESET,Q=>MODE);
end A3;
```