

EE2000 2018/19 Assignment 4 (Due: Dec 2, Sunday, 2018 12:00 midnight)

Question 1: (VHDL Statement)

Given the concurrent VHDL statements.

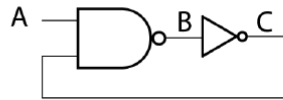
B <= A and C after 3 ns;

C <= not B after 2ns;

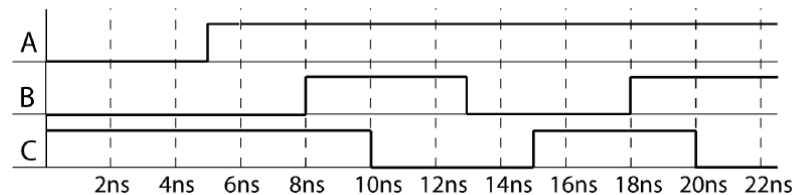
- Draw the circuit the statements represent.
- Draw a timing diagram if initially A = B = '0' and C = '1', and A changes to '1' at time 5 ns. Draw the timing diagram until 22 ns.

Suggested Answer:

(a)



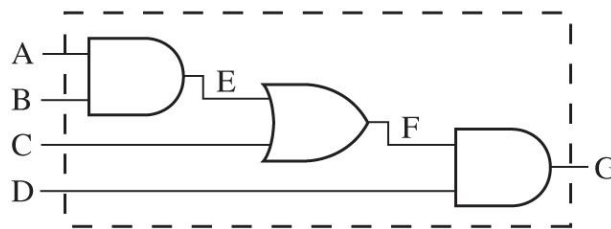
(b)



Question 2: (Basic VHDL Statement)

Write VHDL code for the following circuit. Assume that the gate delays are negligible (no delay).

- Using concurrent statements.
- Using a process with sequential statements.



Suggested Answer:

```
(a) entity circuit is
    port(A, B, C, D: in bit;
          G: out bit);
end circuit;

architecture internal of circuit is
    signal E, F: bit;
begin
    E <= A and B;
    F <= E or C;
    G <= D and F;
end internal;
```

```

(b) entity circuit is
    port(A, B, C, D: in bit;
          G: out bit);
end circuit;

architecture internal of circuit is
    signal E, F: bit;
begin
    process(A, B, C, D, E, F)
    begin
        E <= A and B;
        F <= E or C;
        G <= D and F;
    end process;
end internal;

```

Question 3: (Block Diagram)

Consider the following VHDL Code:

```

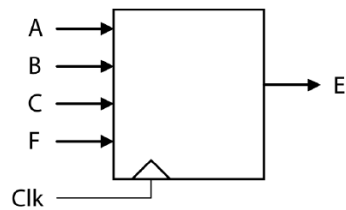
5 entity Q3 is
6     port(
7         A, B, C, F, Clk: in bit;
8         E: out bit
9     );
10 end Q3;
11
12 architecture Behavioral of Q3 is
13     signal D, G: bit;
14     begin
15         process(Clk)
16         begin
17             if (clk'event) and (clk = '1') then
18                 D <= A and B and C;
19                 G <= not A and not B;
20                 E <= D or G or F;
21             end if;
22         end process;
23     end Behavioral;

```

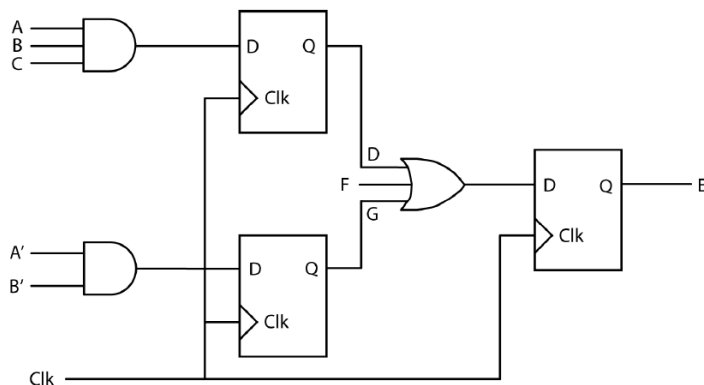
- Draw a block diagram for the circuit (no gates – only at the block level)
- Draw the circuit generated by the above code (at the gate level)

Suggested Answer:

(a)



(b)



Question 4: (DFF with Synchronous Clear)

In the lecture, we have covered how to design a D-type Flip-Flop with Asynchronous Clear, as shown below. Now, 1) modify the design to turn it into a DFF with Synchronous Clear, 2) provide a testbench for this design, 3) simulated waveform to demonstrate the design is working properly.

```
34 entity DFF_AC is
35   -- Port ( );
36   port (
37       CLK, D, ClrN: in std_logic;
38       Q: out std_logic
39   );
40 end DFF_AC;
41
42 architecture Behavioral of DFF_AC is
43 begin
44     process (CLK, ClrN)
45     begin
46         if CLRn = '0' then Q <= '0';
47         else if CLK'event and CLK = '1'
48             then Q <= D;
49         end if;
50     end if;
51 end process;
52 end Behavioral;
```

Suggested Answer:

Design source:

entity DFF_SC is

Port (CLK,D,CLRn:IN STD_LOGIC;

Q: OUT STD_LOGIC);

end DFF_SC;

architecture Behavioral of DFF_SC is

begin

process(CLK)

begin

if(rising_edge(Clk)) then

if(CLRn='0') then

Q <= '0';

else

```
        Q <= D;
    end if;
end if;
end process;
end Behavioral;
```

Testbench:

(In order to test the synchronous reset, we should assert a reset signal after the rising edge of “clk” and keep low when next rising edge of “clk” comes.)

```
entity DFF_SC_tb is
```

```
end DFF_SC_tb;
```

```
architecture Behavioral of DFF_SC_tb is
```

```
component DFF_SC is
```

```
    Port (CLK,D,CLRN:IN STD_LOGIC;
```

```
    Q: OUT STD_LOGIC );
```

```
end component;
```

```
signal CLK: std_logic;
```

```
signal CLRN: std_logic;
```

```
signal Q: std_logic;
```

```
signal D: std_logic;
```

```
constant clk_period:time:=10ns;
```

```
begin
```

```
    uut: DFF_SC
```

```
        port map(CLK,D,CLRN,Q);
```

```
    clk_gen: process
```

```
    begin
```

```
        CLK<='0';
```

```

wait for clk_period/2;

CLK<='1';

wait for clk_period/2;

end process;

```

```

stim_proc: process

```

```

begin

```

```

    wait for 10ns;

```

```

    CLRN<='0';

```

```

    D<='1';

```

```

    wait for 10ns;

```

```

    CLRN<='1';

```

```

    wait for 5ns;

```

```

    D<='0';

```

```

    wait for 10ns;

```

```

    D<='1';

```

```

    wait;

```

```

end process;

```

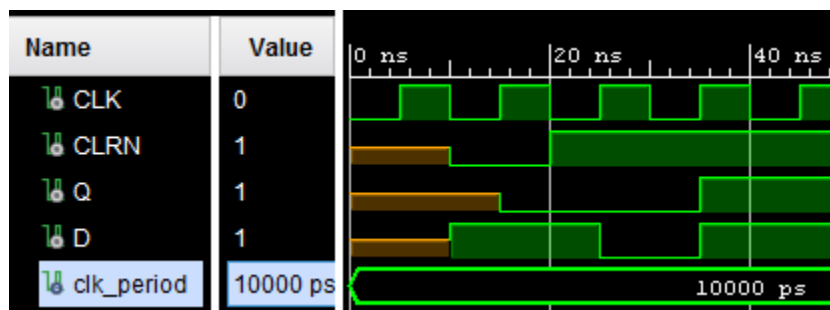
```

end Behavioral;

```

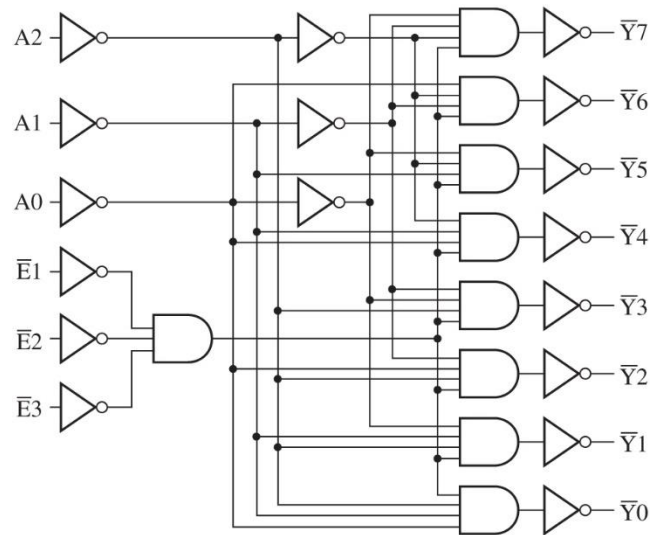
Simulation waveform:

It is clear that, the DFF will not reset unless it meets a rising edge of "CLK", even though CLRN signal is valid before the rising edge of "CLK".



Question 5: (Decoder)

The 74HC138 is a 3-to-8 decoder with a logic diagram as shown below. Please write a dataflow VHDL model for this circuit using the corresponding logical equations.



Suggested Answer:

```
Entity decoder_3to8_dataflow is
    port(A0, A1, A2, E1_bar, E2_bar, E3: in bit;
          Y0_bar, Y1_bar, Y2_bar, Y3_bar, Y4_bar, Y5_bar, Y6_bar, Y7_bar: out bit := '1'
    );
End decoder_3to8_dataflow;
```

Architecture dataflow of decoder_3to8_dataflow is

```
signal enable: bit;
begin
    enable<=(E3) and (not E2_bar) and (not E1_bar);
    Y0_bar<=not((enable)and(not A2)and(not A1)and(not A0));
    Y1_bar<=not((enable)and(not A2)and(not A1)and(A0));
    Y2_bar<=not((enable)and(not A2)and(A1)and(not A0));
    Y3_bar<=not((enable)and(not A2)and(A1)and(A0));
    Y4_bar<=not((enable)and(A2)and(not A1)and(not A0));
    Y5_bar<=not((enable)and(A2)and(not A1)and(A0));
    Y6_bar<=not((enable)and(A2)and(A1)and(not A0));
    Y7_bar<=not((enable)and(A2)and(A1)and(A0));
End dataflow;
```

End