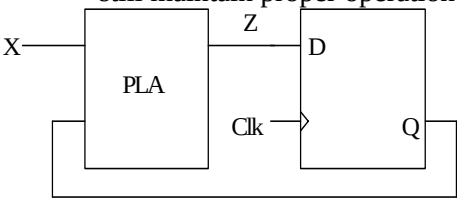
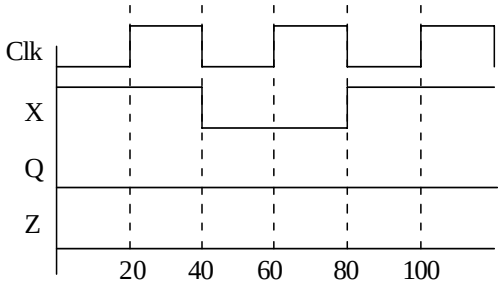
The University of Alabama in Huntsville
ECE Department
CPE/EE 422/522 01
Midterm Exam Solution

1. (10 points) A sequential network consists of a PLA and a D flip-flop, as shown. The propagation delay for the PLA is in the range 5 to 10 ns, and the propagation delay from clock to output of the D flip-flop is 5 to 10 ns. Assuming that X always changes at the same time as the falling edge of the clock, what is the maximum setup and hold time specifications that the flip-flop can have and still maintain proper operation of the network?





For both the setup time and hold time, there are two paths to consider, one from X to the D input of the flip-flop and the other from Q to the D input of the flip-flop. From the timing diagram, tck = 40 ns, tx = 20 ns and ty = 20 ns, where tck is the clock period, tx is the time from a change on X to the active edge of the clock and ty is the time from the active edge of the clock to a change on X. The following equations apply: For Q: (1) $t_{ck} \geq t_{pdmax} + t_{cmax} + t_{su}$, (2) $t_h \geq t_{pdmin} + t_{cmin}$

For X: (3) $t_x \geq t_{su} + t_{cmax}$, (4) $t_h \geq t_y + t_{cmin}$

where t_{pd} is the propagation delay through the flip-flop and t_c is the propagation delay through the combinational circuit (PLA)

So, for setup,

(1) 40 ns ≥ 10 ns + 10 ns + t_{su}, $t_{su} \leq$ 20 ns

(3) 20 ns ≥ t_{su} + 10 ns , $t_{su} \leq$ 10 ns

So, for hold,

(2) $t_h \geq$ 5 ns + 5 ns, $t_h \geq$ 10 ns

(4) $t_h \geq$ 20 ns + 5 ns, $t_h \geq$ 25 ns

For both the setup and the hold times to be always satisfied, we must take the smaller numbers so

$t_{su} = t_h$ = 10 ns

2. (10 points) Write a short VHDL description of a 4-to-1 multiplexer using a VHDL process.

```
entity MUX4_1 is
  port (I3, I2, I1, I0, S1,S0 : in bit;
        F : out bit);
end MUX4_1;
architecture MUX4_1of MUX4_1 is
begin
  process (I3, I2, I1, I0, S1, S0)
  begin
    if (S1 = '0' and S0 = '0') then
      F <= I0;
    elsif (S1 = '0' and S0 = '1') then
      F <= I1;
    elsif (S1 = '1' and S0 = '0') then
      F <= I2;
    else
      F <= I3;
    end if;
end MUX4_1;
```

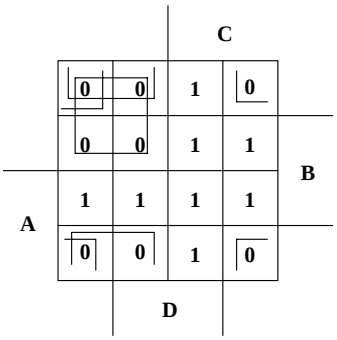
3. (15 points) For the following VHDL, assume that A changes to '1' at 5 ns. Give the values of A, B, C, D, E, and F each time a change occurs. Carry this out until no further change occurs. I

```
entity prob is
  port (D : inout bit);
end prob;

architecture PROB of PROB is
  signal A, B, C, E, F : bit;
begin
  P1: process (A, C)
  begin
    B <= A after 3 ns;
    E <= C after 5 ns;
  end process P1;
  C1: C <= A after 10 ns;
  P2: process (C, E)
  begin
    F <= C and E after 4 ns;
  end process P2;
  C2: D <= A or B or C or F after 1 ns;
end PROB;
```

Time	A	B	C	D	E	F
0 ns	0	0	0	0	0	0
5 ns	1	0	0	0	0	0
6 ns	1	0	0	0	0	0
8 ns	1	1	0	1	0	0
15 ns	1	1	1	1	0	0
20 ns	1	1	1	1	1	0
24 ns	1	1	1	1	1	1

Time	Event	Process Triggered	Scheduled Transaction	Event?
5 ns	A → '1'	P1	B '1' 8 ns	Yes
		P1	E '0' 10 ns	No
		C1	C '1' 15 ns	Yes
		C2	D '1' 6 ns	Yes
6 ns	D → '1'	none		
8 ns	B → '1'	C2	D '1' 9 ns	No
15 ns	C → '1'	P1	B '1' 18 ns	No
		P1	E '1' 20 ns	Yes
		P2	F '0' 19 ns	No
		C2	D '1' 16 ns	No
20 ns	E → '1'	P2	F '1' 24 ns	Yes
24 ns	F → '1'	C2	D '1' 25 ns	No

4. (5 points) Obtain a minimum product of sums expression for the following function:
f(A, B, C, D) = ΠM(0, 1, 2, 4, 5, 8, 9, 10)



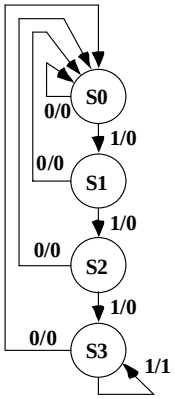
f' = (B'C' + A'C' + B'D')'
= (B'C')'(A'C')'(B'D')
= (B + C)(A + C)(B + D)

5. (15 points) A Mealy sequence detector detects a sequence of four consecutive 1 inputs. The detector has a single binary input, X, and a single binary output, Z. Signal Z should be logic 1 if and only if the last four inputs were all logic 1. Here is an example input-output sequence:

X 01011111110110101111 0
Z 00000011110000000001 0

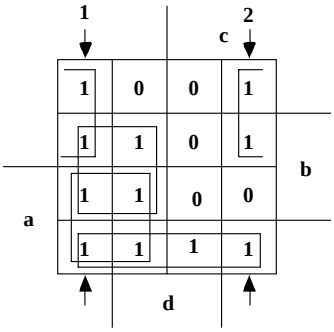
Derive a Mealy state graph and table with a minimum number of states for this sequence detector

	Next State		Output	
Present State	X = 0	X = 1	X = 0	X = 1
S0	S0	S1	0	0
S1	S0	S2	0	0
S2	S0	S3	0	0
S3	S0	S3	0	1



6. (10 points) For the following Ft, find all static-1 hazards. For each hazard, specify the values of the input variables and which variable is changing when the hazard occurs.

Ft = ab' + ac' + bb' + bc' + a'd'

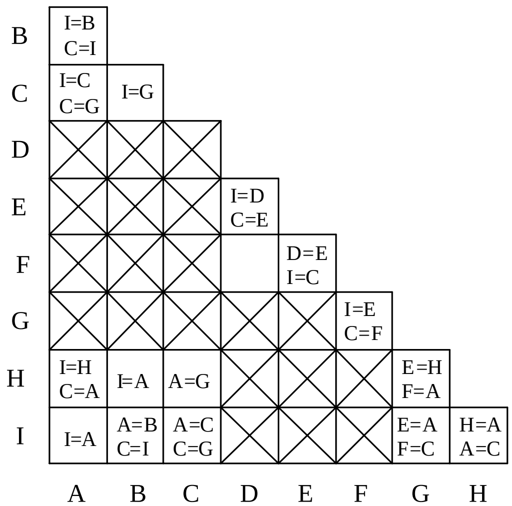


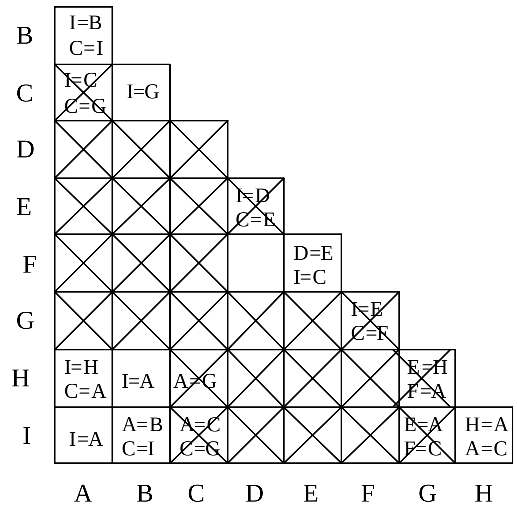
For hazard 1, a is changing and bcd = 000
For hazard 2, a is changing and bcd = 010

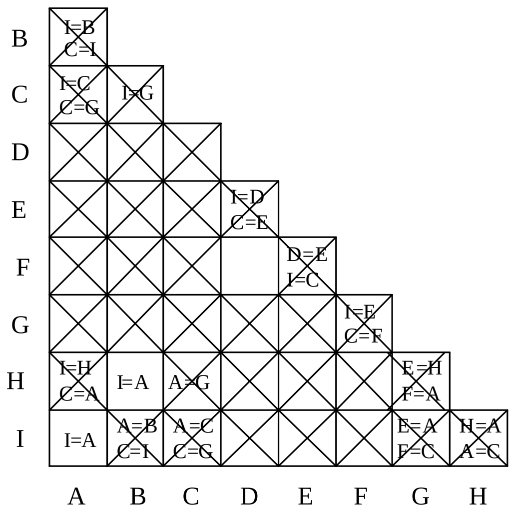
7. (10 points) Reduce the following state table to a minimum number of states. Show all your work in determining the state equivalents

	Next State		
Present State	X = 0	X = 1	Output
A	I	C	1
B	B	I	1
C	C	G	1
D	I	C	0
E	D	E	0
F	I	C	0
G	E	F	0
H	H	A	1
I	A	C	1

	Next State		
Present State	X = 0	X = 1	Output
A	A	C	1
B	B	A	1
C	C	G	1
D	A	C	0
E	D	E	0
G	E	D	0





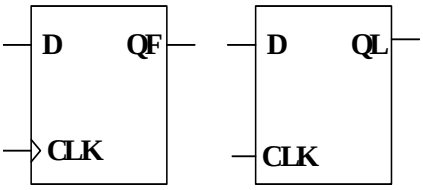


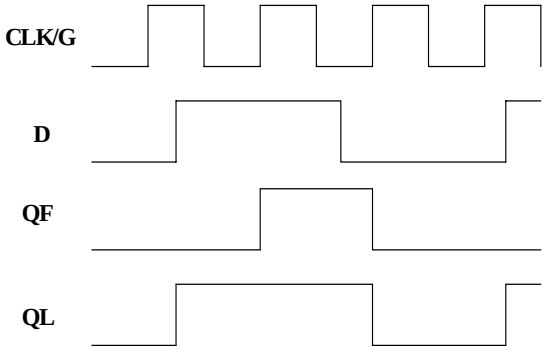
8. (5 points) Write out the truth table for the following equation.

$$F = (A \bullet B') + C$$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

9. (5 points) Draw a timing diagram that illustrates the difference between a D flip-flop and a D latch.





10. (1 point) __Synchronous_ design is a technique that uses a clock to coordinate the operation of all flip-flops, registers and counters in the system.

11. (1 point) The value of a ___variable__ changes instantaneously in VHDL

12. (1 point) A process with a sensitivity list is activated whenever _an event occurs on any signal_ __in the sensitivity list_.

13. (1 point) VHDL is case-sensitive (True/False) __False_

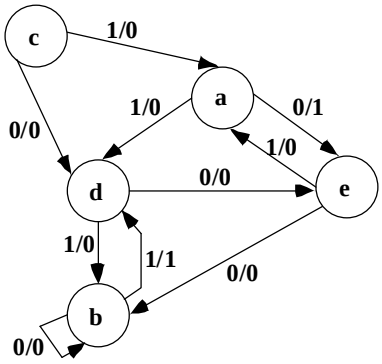
14. (1 point) __Sequential__ networks commonly use flip-flips as storage devices.

15. (10 points) Draw the state diagram for the following state machine. Is it a Moore machine or a Mealy machine?

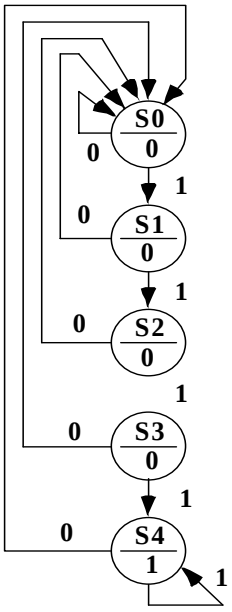
```
ENTITY state_machine IS
   PORT (sig_in ; IN BIT; clk : IN BIT;
         sig_out : OUT BIT);
END state_machine;

ARCHITECTURE state_machine OF state_machine IS
   TYPE state_type IS (a, b, c, d, e);
   SIGNAL current_state, next_state : state_type;
BEGIN
   PROCESS (sig_in, current_state)
   BEGIN
      sig_out <= '0'; next_state <= b;
      CASE current_state
      WHEN a =>
         IF sig_in = '0' THEN next_state <= e; sig_out <= '1';
         ELSE next_state <= d;
         END IF;
      WHEN b =>
         IF sig_in = '0' THEN next_state <= b;
         ELSE next_state <= d; sig_out <= '1';
         END IF;
     WHEN c =>
       IF sig_in = '1' THEN next_state <= a;
```

```
        ELSE next_state <= d;
        END IF;
      WHEN d =>
        IF sig_in = '0' THEN next_state <= e;
        END IF;
      WHEN e =>
        IF sig_in = '1' THEN next_state <= a;
        END IF;
       END CASE;
   END PROCESS;
   PROCESS (clk)
   BEGIN
      IF (clk'EVENT AND clk = '1') THEN
          current_state <= next_state;
      END IF;
   END PROCESS;
END state_machine;
```

Mealy



Extra Credit (5 points): Rework problem 6 using a Moore machine.



	Next State		
Present State	X = 0	X = 1	Output
S0	S0	S1	0
S1	S0	S2	0
S2	S0	S3	0
S3	S0	S4	0
S4	S0	S4	1