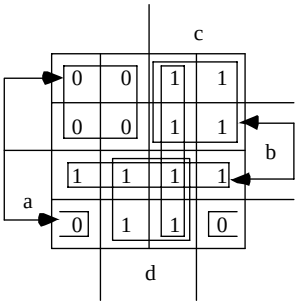
The University of Alabama in Huntsville
Electrical and Computer Engineering
CPE/EE 422/522
Spring 2005
Homework #3 Solution

1.5 (20 points) (a) Find all the static hazards in the following network. For each hazard, specify the values of the input variables and which variable is changing when the hazard occurs. For one of the hazards, specify the order in which the gate outputs must change.
(b) Design a NAND-gate network that is free of static hazards to realize the same function.

(a) F = ((ab)'(a + c)' + (a' + d)')'
= ab + (a + c)(a' + d)
= ab + aa' + ad + ac' + cd

For zero hazards

F = ab + (a + c)(a' + d) Using X + YZ = (X +Y)(X + Z), X = ab, Y = a+c, Z=a'+d
= (ab + a + c)(ab + a' + d) Using X + YZ = (X +Y)(X + Z), X = a'+d, Y=a, Z=b
= (a(b + 1) + c)(a' + d + a)(a' + d + b)
= (a + c) (a' + d + a)(a' + d + b)



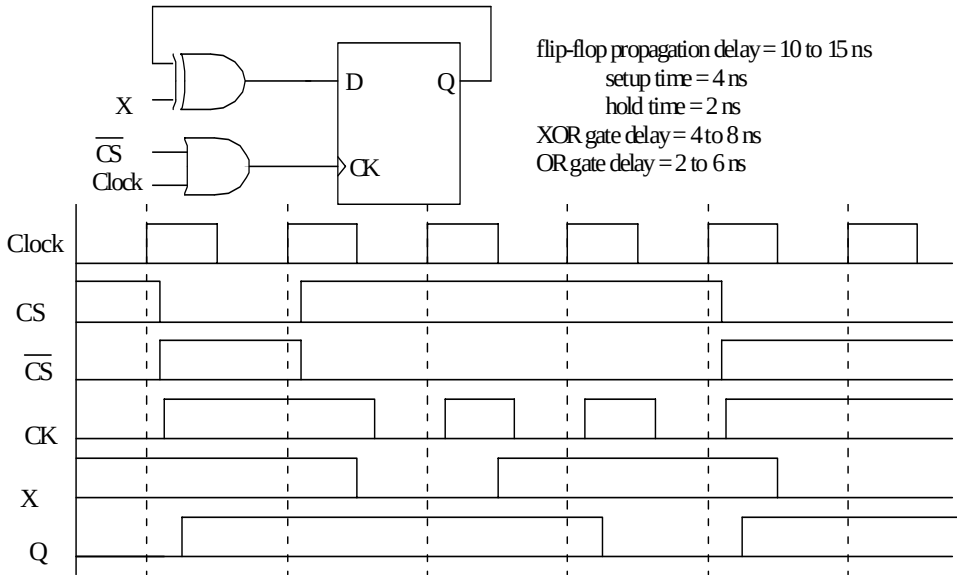
1-hazard
bcd = 110, a changing
0-hazard
bcd = 000, a changing

1.9 (15 points) A sequential network has one input (X) and two outputs (Z1 and Z2). An output Z1 =1 occurs every time the sequence 010 is completed provided that the sequence 100 has never ooccurred. An output Z2 =1 occurs every time the input sequence 100 is completed. Note that once a Z2=1 output has occurred, Z1 =1 can never occur, but *not* vice versa.
(a) Derive a Mealy state graph and table with a minimum number of states (8 states).

	NS		Z1Z2	
PS	X = 0	X = 1	X = 0	X = 1
S0	S1	S3	00	00
S1	S1	S2	00	00
S2	S4	S3	10	00
S3	S4	S3	00	00
S4	S5	S2	01	00
S5	S5	S6	00	00
S6	S7	S6	00	00
S7	S5	S6	01	00

1.17 (15 points) Assume that CS (and also) change 2 ns after the rising edge of the clock.

(a) Plot CK and Q on the timing diagram. A precise plot is not required; just show the relative times at which the signals change.
(b) If X changes at the falling edge of Clock, as shown, what is the maximum clock frequency?
(c) With respect to the rising edge of Clock, what is the earliest that X can change and still satisfy the hold-time requirement?



(b) Check both X and Q paths:

For Q: $t_{ck} \geq t_{pmax} + t_{cmax} + t_{su}$ = 15 ns + 8 ns + 4 ns = 27 ns

For X: $t_{ck}/2 \geq t_{cmax} + t_{su}$

$t_{ck} \geq 2(t_{cmax} + t_{su})$ = 2(8 ns + 4 ns) = 24 ns

Must use the longer, f_{clk} <= 1/27 ns

(c) The setup time is relative to CK, but the question asks relative to the rising edge of clock. So, the max value for the OR gate must be used so the input is definitely held long enough.

$ty_{Clock} = ty_{CK}$ + 6 ns = -2ns + 6 ns = 4 ns

CK($t_y = t_h - t_{cmin}$ = 2 ns – 4 ns = -2 ns)

2.2 (15 points) (a) Write VHDL code for a full subtractor using logic equations.

```
entity FULL_SUB is
  port (X, Y, BIN : in bit;
        SUM, BOUT : out bit);
end FULL_SUB;

architecture EQUATIONS of FULL_SUB is
begin
  SUM <= (X and not Y and not BIN) or (not X and not Y and BIN) or
         (X and Y and BIN) or (not X and Y and not BIN);
  BOUT <= (not X and BIN) or (not X and Y) or (Y and BIN);
end EQUATIONS;
```

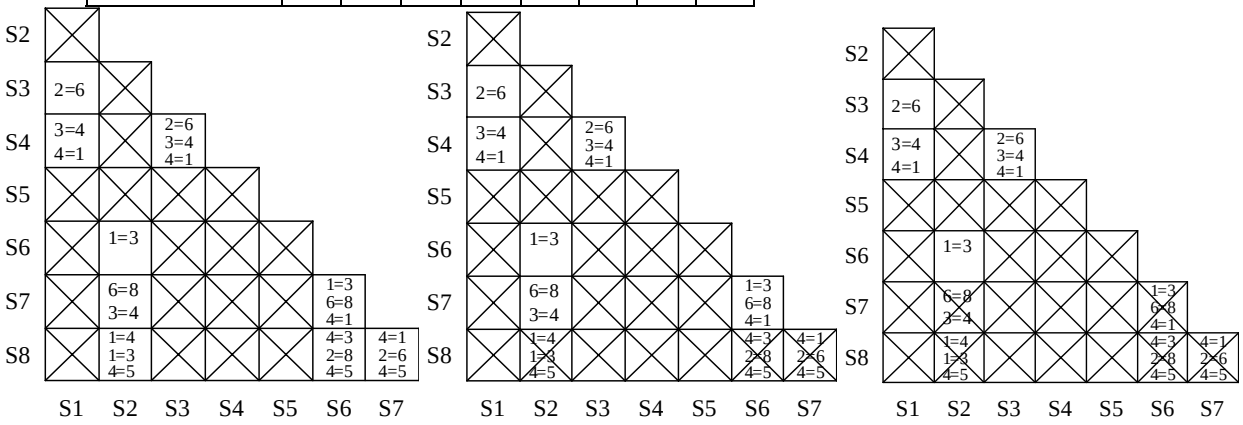
(b) Write VHDL code for a 4-bit subtractor using the module defined in (a) as a component.

```
entity SUB4 is
  port (A, B : in bit_vector (3 downto 0); BIN : in bit;
        S: out bit_vector (3 downto 0); BOUT : out bit);
end SUB4;
```

```
architecture STRUCTURAL of SUB4 is
  component FULL_SUB
    port (X, Y, BIN : in bit;
          SUM, BOUT : out bit);
  end component;
  for ALL : FULL_SUB use entity work.FULL_SUB(EQUATIONS);
  signal BORROW : bit_vector (3 downto 1);
begin
  FS0 : FULL_SUB port map (A(0), B(0), BIN, S(0), BORROW(1));
  FS1 : FULL_SUB port map (A(1), B(1), BORROW(1), S(1), BORROW(2));
  FS2 : FULL_SUB port map (A(2), B(2), BORROW(2), S(2), BORROW(3));
  FS3 : FULL_SUB port map (A(3), B(3), BORROW(3), S(3), BOUT);
```

(20 points) Reduce the following state table to a mininum number of states using the implication technique.

Present State	Next State, X1X2				Output Z, X1X2			
	00	01	11	10	00	01	11	10
S1	S2	S5	S4	S1	0	1	1	0
S2	S1	S8	S3	S5	1	0	1	1
S3	S6	S5	S4	S1	0	1	1	0
S4	S2	S5	S3	S4	0	1	1	0
S5	S2	S5	S3	S7	1	1	0	0
S6	S3	S8	S1	S5	1	0	1	1
S7	S1	S6	S4	S5	1	0	1	1
S8	S4	S2	S1	S4	1	0	1	1



So, state S2 is equivalent to S6 and States S1, S3, and S4 are all equivalent. The reduced state table is as follows:

Present State	Next State, X1X2				Output Z, X1X2			
	00	01	11	10	00	01	11	10
S1	S2	S5	S1	S1	0	1	1	0
S2	S1	S8	S1	S5	1	0	1	1
S5	S2	S5	S1	S7	1	1	0	0
S7	S1	S2	S1	S5	1	0	1	1
S8	S1	S2	S1	S1	1	0	1	1