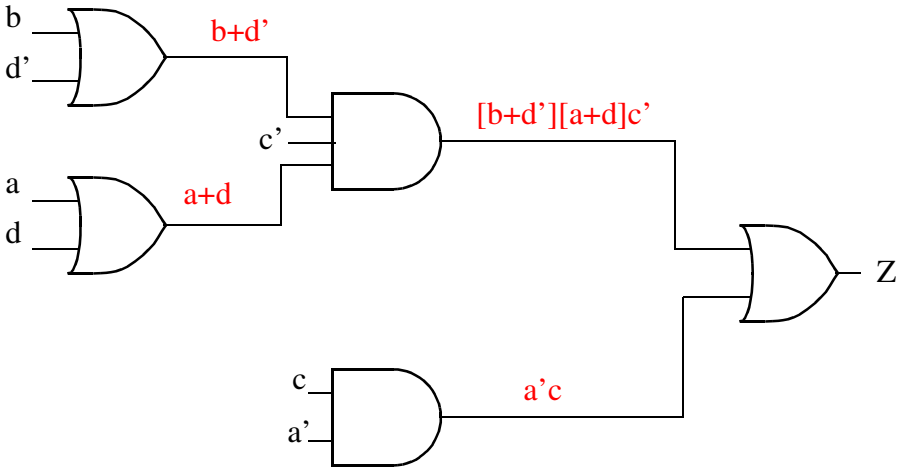
CPE/EE 422/522 **EXAM I** **Name** KEY

Class _____

Spring Semester 2011

Work should be performed systematically and neatly with the final answer being underlined. This exam is closed book, closed notes, closed neighbor. Allowable items on desk include: exam, data manual, pencils, straight edge, and calculator. All other items must be removed from student's desk. Students have approximately 90 minutes (1 1/2 hours) to complete this exam. Best wishes!

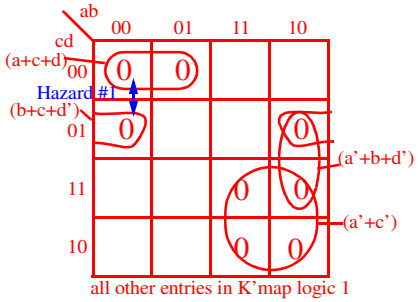
1. [10 points] For the network shown below, find all static 0-hazards. For each 0-hazard found specify the conditions which will cause the hazard to appear at the output (i.e. specify the logic values of the variables which are constant and clearly specify the variable that is assumed to be changing). If there are no 0-hazards found, use a K'map to show why this is the case.



Z=[b+d'][a+d]c'+a'c
=([b+d'][a+d]c'+a')([b+d'][a+d]c'+c)
=(a'+[b+d'])(a'+[a+d]c')(c+[b+d'])(c+[a+d]c')
=(a'+[b+d'])(a'+[a+d])(a'+c')(c+[b+d'])(c+[a+d])(c+c')
=(a'+b+d')(a'~~+a+d~~)(a'+c')(b+c+d')(a+c+d)(~~c+c'~~)

not able to map

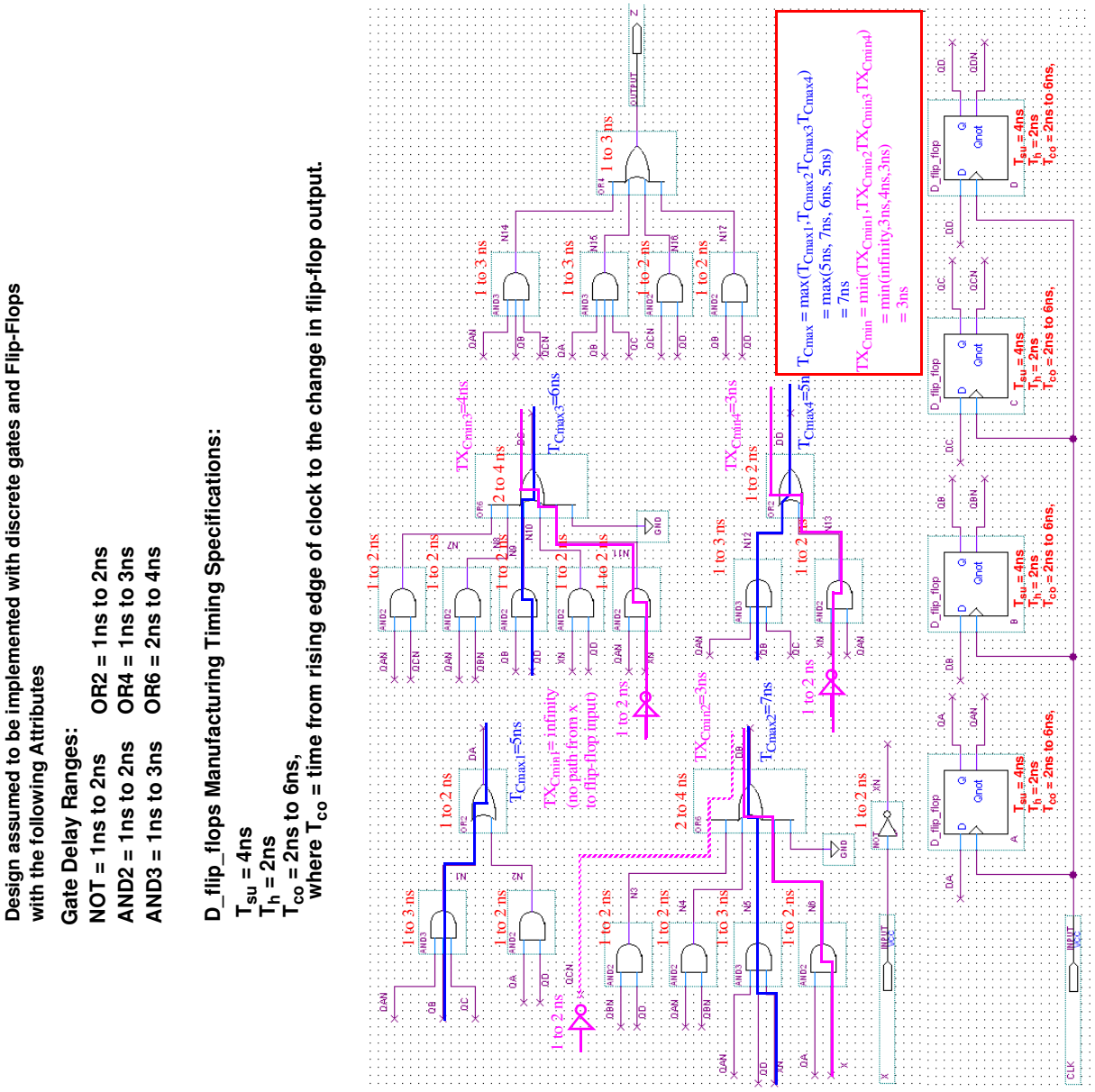
not able to map



One 0-hazard found

Hazard #1:
d changing,
while a = 0, b = 0, c = 0

2.[10 points] For the synchronous sequential network shown below, determine the following:.

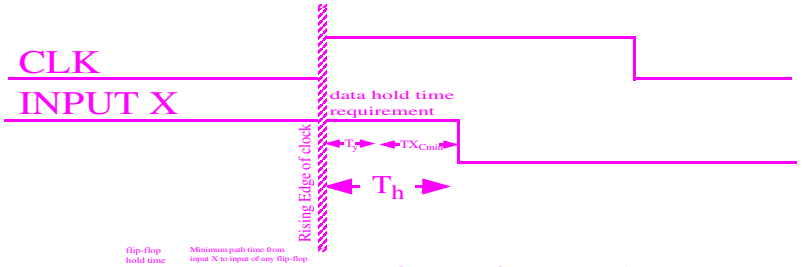


a) What is the maximum clock frequency for proper operation of the network if input X is driven in the appropriate manner by the external circuitry?

$T_{min} = T_{su} + T_{Cmax} + T_{co(max)} = 4ns + 7ns + 6ns = 17ns$

$f_{max} = 1/T_{min} = 1/17ns = 58.8Mhz$

b) What is the earliest time <u>after</u> the rising clock edge that input X is allowed to change?

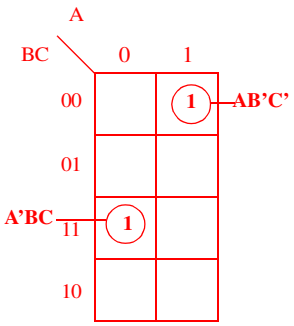


$T_y = T_h - TX_{Cmin} = 2ns - 3ns = -1\ ns$

meaning Input X can change 1 ns <u>before</u> the active edge of the clock!!!

3. [5 points] Obtain a minimum sum-of-products expression for the following function:

$F(A, B, C) = \prod M(0, 1, 2, 5, 6, 7)$



$F(A, B, C) =$ A'BC + AB'C'

4. [5 points] What is the primary purposes of hardware description languages such as VHDL? What three areas can hardware description languages be employed to facilitate the design process (hint: Design Entry is one area, what are two more?)?

Hardware description languages such as VHDL allow one to more easily create, debug and maintain complex digital designs. They allow one to express the functionality of a digital circuit in a highly abstract manner, while also permitting low-level design.

Areas where HDLs are extremely useful are

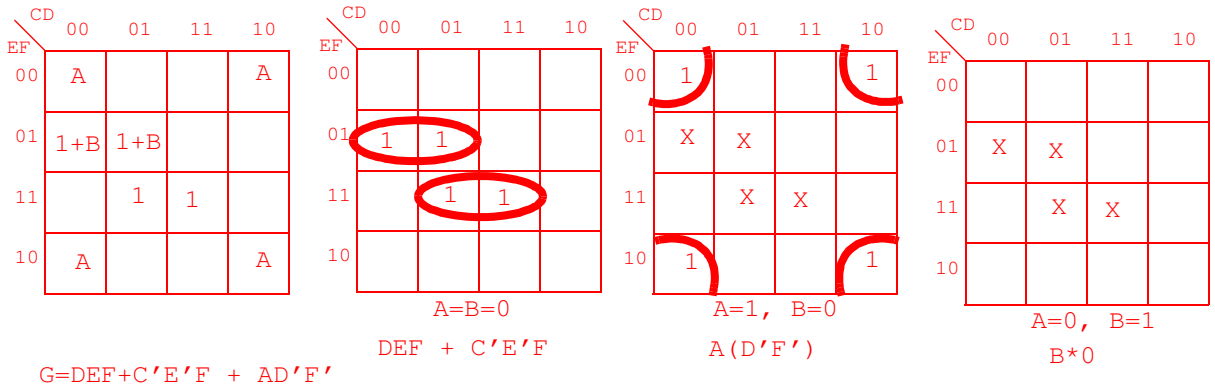
- Design Entry -- allowing the design to be entered in a manner that supports a wide range of abstraction.
- Design Documentation -- allowing the implicit and explicit support for the design to be documented.
- Design Simulation -- creating a framework from which the functionality of the design can be evaluated before it is implemented.
- Design Verification and Timing analysis -- the HDL often supports constructs that can be used by automated tools that will verify that design specifications and internal timings parameters are met.
- Design Synthesis -- automatic creation of the design implementation from the design that is captured in the HDL.

5. [5 points] What is the major differences between transport and inertial delays in VHDL?

An inertial delay only delays the effects of a transition on a signal, if that transition is present for a period that is equal to are greater than the delay value that is specified. In this way short pulses are filtered out with intertial delays. A transport delay will delay the effects of a transition on a signal regardless of the time the transistion is present. (A reject clause can be added to a transport delay to allow it to reject signals less than a specified value that is independent of the delay value, but this is not the default.) Intertial delays are the default and tend to represent the electronic effect of gates, whereas the transport delays tend to represent the effect of busses or wires.

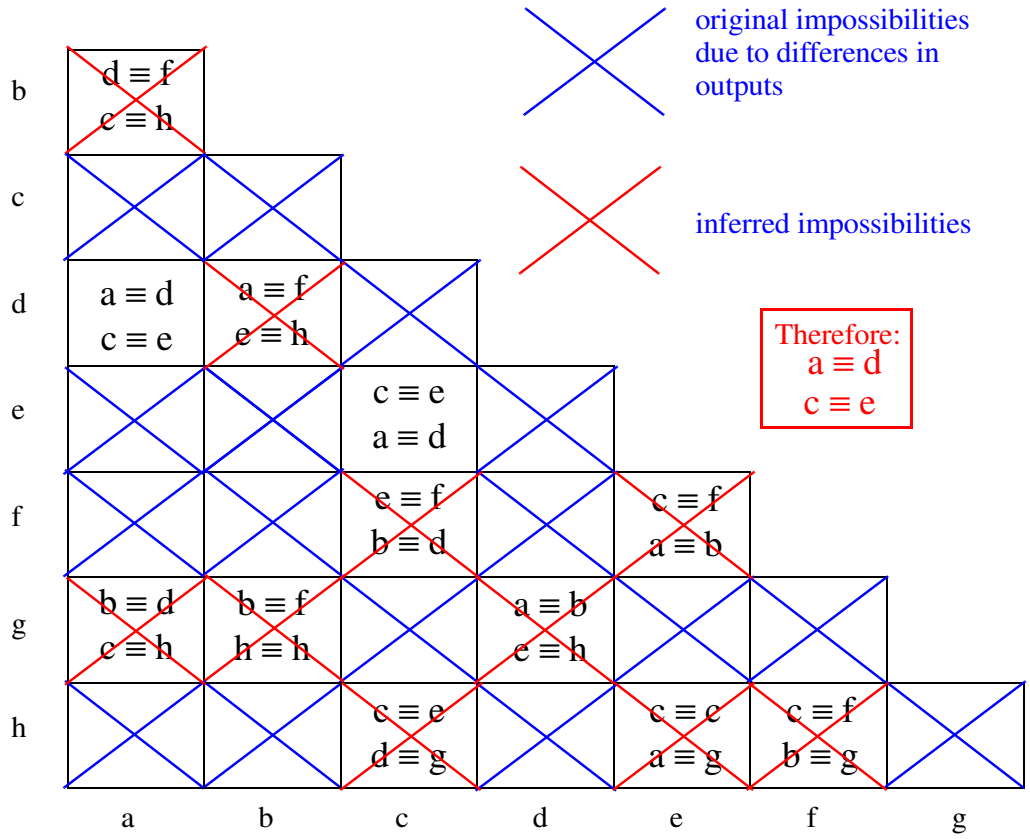
6. [10 points] For the Boolean function shown below find the minimum sum of products (SOP) expression using 4-variable Knaugh map(s) with map-entered variables.

G = DEF + C'E'F + AD'EF' + AD'E'F' + BC'E'F



7. [10 points] Reduce the following state table to a minimum number of states

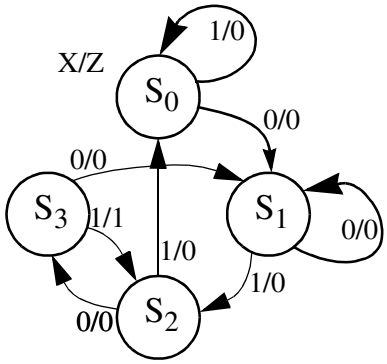
present state	next state X=0	1	present output
a	d	c	0
b	f	h	0
c	e	d	1
d	a	e	0
e	c	a	1
f	f	b	1
g	b	h	0
h	c	g	1



Reduced State Table

present state	next state X=0	1	present output
a	a	c	0
b	f	h	0
c	c	a	1
f	f	b	1
g	b	h	0
h	c	g	1

8. [20 points] Develop a two process VHDL model for the state graph shown below:




```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity SM is
   port(X,CLK: in STD_LOGIC;
            Z : out STD_LOGIC;
   }
end SM;

architecture BEHAVIORAL of SM is
   signal STATE, NEXTSTATE: integer range 0 to 3 :=0;
   begin
      P1:process(STATE,X)
      begin
         case STATE is
            when 0 =>
              if X='0' then Z<='0'; NEXTSTATE<=1; end if;
              if X='1' then Z<='0'; NEXTSTATE<=0; end if;
            when 1 =>
              if X='0' then Z<='0'; NEXTSTATE<=1; end if;
              if X='1' then Z<='0'; NEXTSTATE<=2; end if;
            when 2 =>
              if X='0' then Z<='0'; NEXTSTATE<=3; end if;
              if X='1' then Z<='0'; NEXTSTATE<=0; end if;
            when 3 =>
              if X='0' then Z<='0'; NEXTSTATE<=1; end if;
              if X='1' then Z<='1'; NEXTSTATE<=2; end if;
             when others => null;
         end case;
      end process P1;

      P2:process(CLK)
      begin
         if (CLK='1) then
            STATE <= NEXTSTATE;
         end if;
       end process P2;

end BEHAVIORAL;
```

9. [15 points] In the following VHDL model fragment (which corresponds to a portion of the architecture section of a VHDL file) A, B, C, and D are all integers that have a value of 0 at time = 10ns. If E changes from '0' to '1' at time 20 ns, specify the times (s) at which each signal will change and the value to which it will change. List these changes in chronological order. List any delta delays as a separate entry.

```
P1: process
    begin
       wait on E;
       A <= transport 2 after 5 ns;
       B <= A + 1;
       wait for 0 ns;
       A <= transport A + 5 after 5 ns;
       B <= B + 2;
     end process P1;

C <= A after 10 ns;

P2: process(B)
    begin
       D <= B after 5 ns;
    end process P2;
```

Time	A	B	C	D	E
10 ns	0	0	0	0	0
20 ns	0	0	0	0	1
20 ns + Δ	0	1	0	0	1
20 ns + 2Δ	0	3	0	0	1
25 ns	5	3	0	3	1
35 ns	5	3	5	3	1

10. [10 points] For the keyboard interface design you implemented in Laboratory 2, answer the following:
Describe at least two main differences between the CLOCK_27 and the PS2_CLK signals.

- The CLOCK_27 is a much faster than the PS2_CLK (about 1890 times faster)

- The PS2_CLK clock is not continuous. It stops when data is not being sent. The CLOCK_27 runs continuously.

In general, what type of problems might occur if we have two or more disjoint clocks at different speeds simultaneously clock different portions of a sequential design? What bad thing or things might happen in this design?

The design then becomes asynchronous at the interface between the circuitry that uses the different clocks, which can result in setup and hold violation and metastability. This can result in race conditions which often results in unreliable data transfer.

Would it be possible for us to entirely eliminate the PS2_CLK from our design if we knew the speed at which the data is begin sent from the keyboard and the protocol that is being used? Explain.

Yes, it would. The PS2 port actually utilizes an asynchronous serial data protocol. This protocol has the concepts of a start bit and stop bit and a normal resting logical value of the serial line. Circuitry can be built (which is commonly called a Universal Asynchronous Receiver/Transmitter, UART) that utilizes the start bit and the known transfer rate, to sync up with the data transfer asynchronously in a manner that only requires a single internal clock.