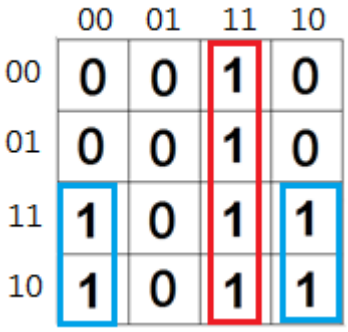
NTU CISE Digital System and Labs 2014 MIDTERM (2014-04-23)

1. (12%) Convert the following Boolean expressions or K-map into **Sum of Products** with the indicated number of literals. Recall that a literal is either a variable or its negation.

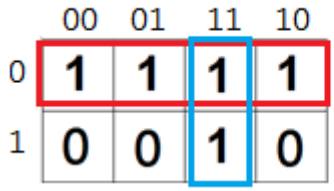
(Hint: You may use K-map to speed up the computing)

i) F (A, B, C, D) = Σ (2, 3, 10, 11, 12, 13, 14, 15) (reduce to 4 literals)



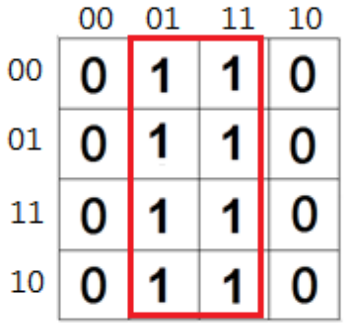
AB+B'C

ii) A'C' + ABC + AC' (reduce to 3 literals)



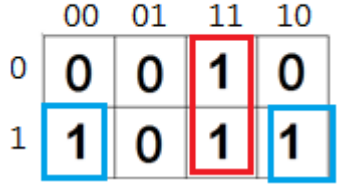
AB+C'

iii) A'B(D' + C'D) + B(A + A'CD) (reduce to 1 literal)
= A'BD' + A'BC'D + AB + A'BCD



B

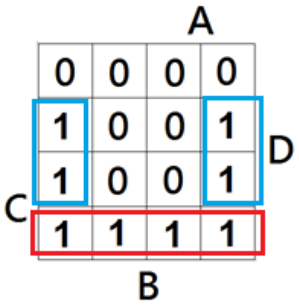
iv) (B' + A) • (B + C) (reduce to 4 literals)
=~~BB'~~+B'C+AB+AC



AB+B'C

v)

CD \ AB	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

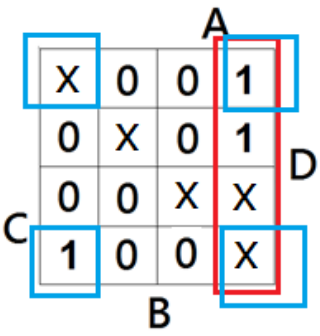


CD'+B'D

(reduce to 4 literals)

vi)

CD \ AB	00	01	11	10
00	X	0	0	1
01	0	X	0	1
11	0	0	X	X
10	1	0	0	X



AB'+B'D'

(reduce to 4 literals)

2. (10%) Consider the 5-bit binary equation "comparator" : The first 2-bit is input A, the third bit is operator bit C where 0 means "add" while 1 means "subtract", and the last 2-bit is input B. Both A and B are binary signed integer. If the equation occurs overflow (cannot be represented by 2-bit binary signed integer), the ouput K will be 1, otherwise, K will be 0.

For example: 11 0 10 = -1 + (-2) = -3 (overflow), output = 1
00 1 11 = 0 – (-1) = 1 (safe), output = 0

i) Draw the K-map for K

C = 0

B1B0 \ A1A0	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	0	0	1
10	0	0	1	1

C = 1

B1B0 \ A1A0	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	1	0	0
10	1	1	0	0

ii) Write the Boolean expressions for the output K

$K = A_1'A_0'B_1C' + A_1B_1B_0'C' + A_1A_0'B_1'B_0C' + A_1A_0'B_1'B_0C + A_1'A_0B_1C + A_1'B_1B_0'C$

3. (3%) (a) Demonstrate that a 2-bit NOR gate is a universal logic element. You can do this by showing how they can be used to make: NOR, AND, OR, and XOR gates.

(6%) (b) Is an XOR gate a universal logic element? Why or why not?

(3%) (c) What about a 2-input NAND gate?

Note that each input of the NOR gate must be used, it cannot be left unconnected.

4. (12%) You are a testing assistant in the bulb factory. The factory has just produced 256 new bulbs numbered from 0 to 255 in binary number and needs you to examine them. So you design a series of tests shown as below. If the bulb is checked in some round, the entry of the bulb should be set as 1, or 0 if unchecked. But this list is a waste of paper, the boss ask you to simplify the list to Boolean function. Please satisfy his requirement.

No.	Binary number $[N_7 \ldots N_0]$	R_0	R_1	……	R_n
0	00000000	1			
1	00000001	1			
2	00000010	1			
3	00000011	1			
. .	. .	. .	. .	. .	. .
254	11111110	0			
255	11111111	0			

For instance, you have tested the first 4 bulbs (No.0 ~ No.3) bulbs in Round 0, so you write down the Boolean function as $R_0 = \overline{N_7} \cdot \overline{N_6} \cdot \overline{N_5} \cdot \overline{N_4} \cdot \overline{N_3} \cdot \overline{N_2}$

Please finish the rest (as simple as possible):

i) R_1 = Test the even numbers
If $N_0 = 0$, then R = 1. So R = $\overline{N_0}$

ii) R_2 = Test the multiples of 8
If $N_2N_1N_0 = 000$, then R = 1. So R = $\overline{N_2N_1N_0}$

iii) R_3 = Divide the bulbs into quarters. Test the 2^{nd} and the 3^{rd} quarter
If $N_7N_6 = 01$ or 10, then R = 1. So R = $\overline{N_6}N_7 + N_6\overline{N_7}$

iv) R_4 = Test all except for the last 32 bulbs
If $N_7N_6N_5 = 111$, then R = 0. So R = $\overline{N_7N_6N_5} = \overline{N_7} + \overline{N_6} + \overline{N_5}$

v) R_5 = Test all except for the multiple of 16
If $N_3N_2N_1N_0 = 0000$, then R = 0. So R = $\overline{\overline{N_3} \cdot \overline{N_2} \cdot \overline{N_1} \cdot \overline{N_0}} = N_3 + N_2 + N_1 + N_0$

vi) R_6 = Test No.129 and No. 219 only
$129_{(10)} = 10000001_{(2)}$, $219_{(10)} = 11011011_{(2)}$. So R = $N_7\overline{N_6N_5N_4N_3N_2N_1}N_0 + N_7N_6\overline{N_5}N_4N_3\overline{N_2}N_1N_0$

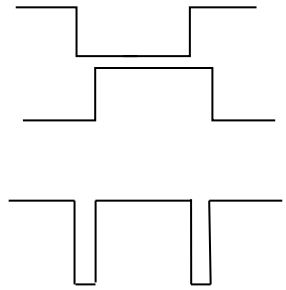
5. For a circuit with an inverter and an exclusive or gate, suppose that the exclusive or gate has no time delay:

(4%) (a) What the output signal Y will be like with the input signal X?

X:

$\overline{X}$:

Y:

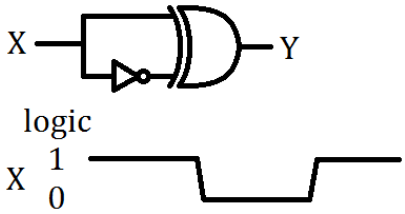


(4%) (b) Why does Y have such relationship with X?

Because NOT gate has delay

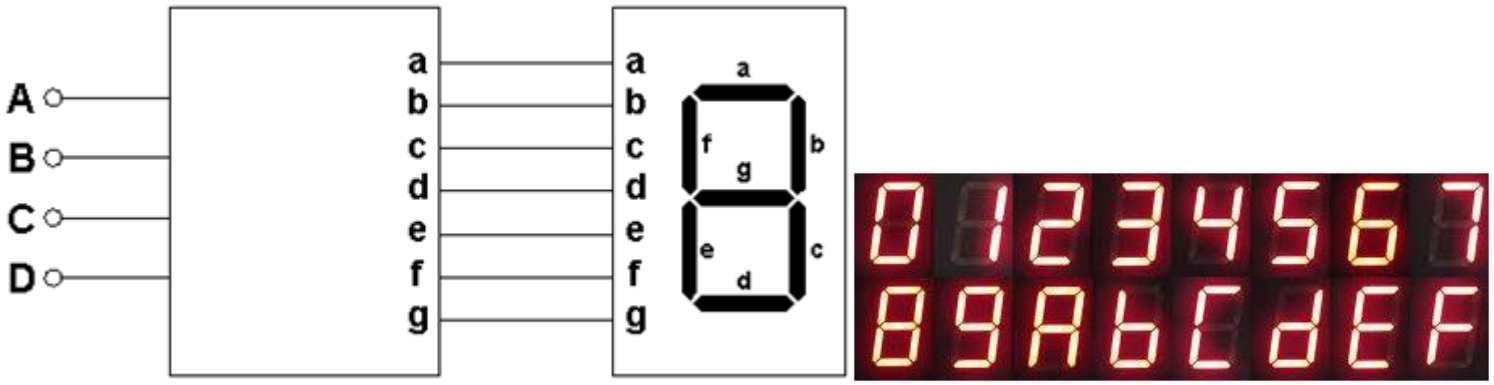
(4%) (c) What can the circuit do in real life?

Whatever example would be accepted



6. (12%) The following figure shows how the hexadecimal 7-segment display decoder works. DCBA is a 4-bit binary number input, then the decoder will transfer the binary number into segment signal. Assume that 1 is for segment "on" while 0 is segment "off".

For instance, if DCBA represent the digit "0", which means all the segment except for "g" should be on. So the decoder will send (1, 1, 1, 1, 1, 1, 0) to the 7-segment display. Answer the following questions:



i) If this decoder can only display decimal number (0 - 9), how many bits are needed for the input?
Because $10 > 8(2^3)$, we need 4 bits

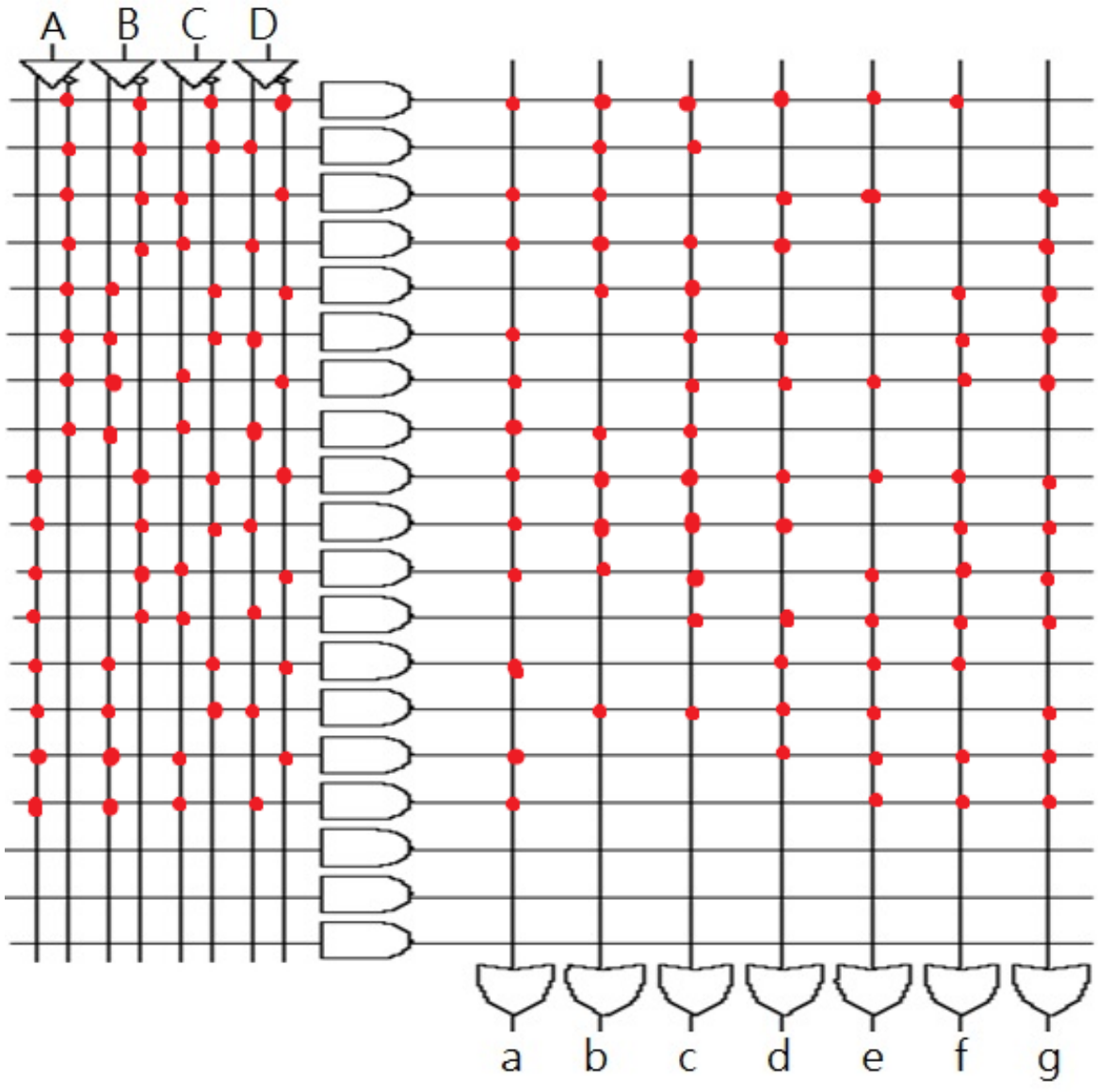
ii) If the output is unique, How many bits for the input of 7-segment are valid **at most**?
There are total 2^7 kinds of output, so we may have 2^7 inputs at most, that is, 7 bits

iii) Write the truth table for this **hexadecimal** 7-segment display system

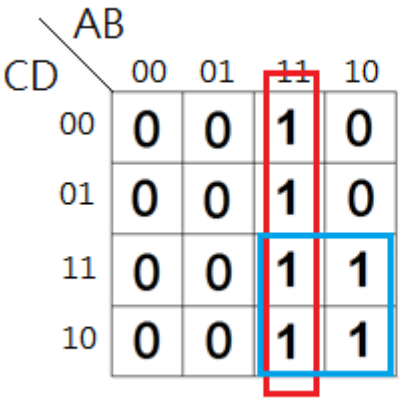
iv) Draw the PLA implementation of this decoder

Here is the example of PLA implementation:

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1



7. (10%) Find a function to detect an error in the representation of a decimal digit in BCD. In other words, write an equation with value 1 when the input are any one of the six unused bit combinations in the BCD code (10, 11, 12, 13, 14, 15), and value 0 otherwise. Please implement it with only NAND gates and inverters.



$AB + AC$

A • B = not (A nand B)

$A + B = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A} \cdot \overline{B}}$

The rest is your work :"D

Problem B:

(a) (20%) (Verilog HDL – Code Debugging)

Identify syntax errors and inappropriate code then correct them and explain: 2pts for each error.

```
module 16x16-MAC (out, rst, op1, op2)
input clk, rst;
input [15:0] op1, op2;
output [39:0] out;
reg [31:0] product;
assign product = op1 * op2;
//40-bit accumulator after 16x16 multiplier
always @(posedge clk || negedge rst)
  if(rst) out= 0;
  else out = out + product;
endmodules
```

```
module 16x16-MAC (out, clk, reset, op1, op2) ;

// module name cannot be started with number, and ended with ";"

input clk, reset;

input [15:0] op1, op2;

output [39:0] out;

reg [39:0] out;

wire [31:0] product;

assign product = op1 * op2;

//40-bit accumulator after 16x16 multiplier

always @(posedge clk or negedge reset)

  if(!reset) out <= 0;

  else out <= out + product;

endmodule
```