King Fahd University of Petroleum and Minerals
College of Computer Science and Engineering
Computer Engineering Department

COE 202: Digital Logic Design (3-0-3)
Term 102 (Spring 2011)
Major Exam II
Thursday April 28, 2011

Time: 120 minutes, Total Pages: 9

Name:__KEY____________________________ ID:_______________ Section: ________

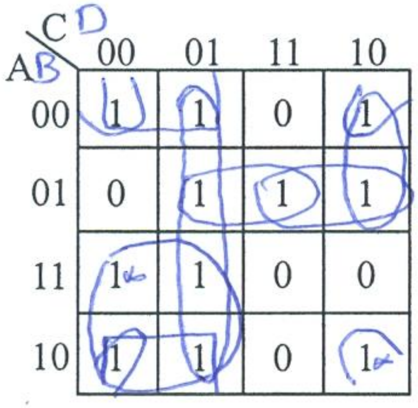
Notes:

- Do not open the exam book until instructed
- **Calculators are not allowed** (*basic, advanced, cell phones, etc*.)
- Answer all questions
- All steps must be shown
- Any assumptions made must be clearly stated

Question	Maximum Points	Your Points
1	**26**	
2	**16**	
3	**12**	
4	**26**	
5	**20**	
Total	**100**	

Question 1. **(26 points)**

(A). For the following Boolean function F(A, B, C, D)=Σ m(0, 1, 2, 5, 6, 7, 8, 9, 10, 12, 13)



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

(i) Identify all the *prime implicants* and the *essential prime implicants* of F. (7+2=9 points)

(ii) Simplify the Boolean function **F** into a minimal sum-of-products expression. (5 points)

(i) Prime Implicants:

$\overline{C}D$, $A\overline{C}$, $\overline{B}\,\overline{C}$, $\overline{B}\,\overline{D}$, $\overline{A}BD$, $\overline{A}BC$, $\overline{A}C\overline{D}$

Essential Prime Implicants:

$A\overline{C}$, $\overline{B}\,\overline{D}$

(ii) $F = A\overline{C} + \overline{B}\,\overline{D} + \overline{A}BC + \overline{C}D$

(B) Consider the following Boolean function **F** together with the don`t care conditions **d**

F(A, B, C, D)=Σ m(0, 2, 5, 8, 10), **d**(A, B, C, D)=Σ m(1, 4, 7, 9, 11, 12, 14, 15)

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

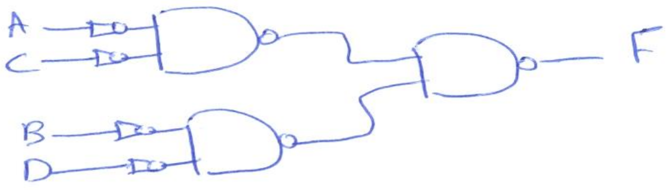
(i) Simplify the Boolean function **F** together with the don`t care conditions **d,** into minimal sum-of-products expression. (4 points)

(ii) Starting with the sum-of-products expression, implement the function using only **NAND** gates and **Inverters**. (4 points)

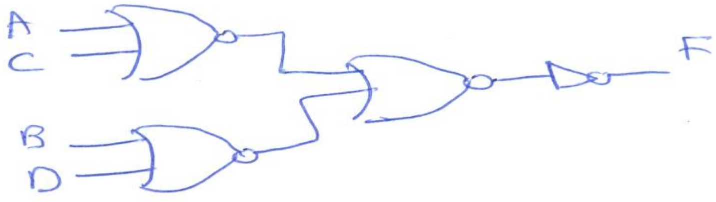
(iii) Starting with the sum-of-products expression, implement the function using only **NOR** gates and **Inverters**. (4 points)

(i) $F = \overline{A}\,\overline{C} + \overline{B}\,\overline{D}$

(ii)



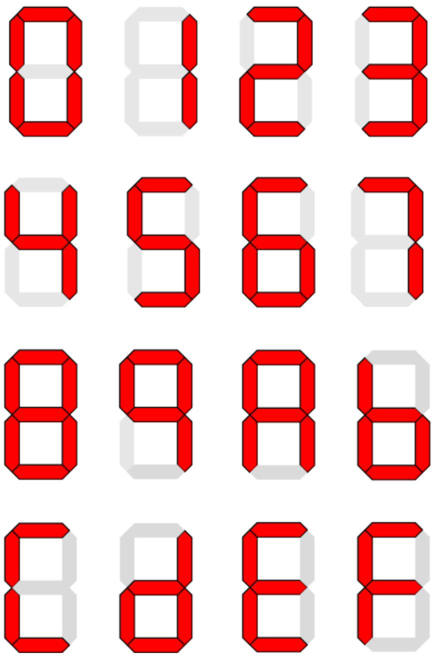
(iii)

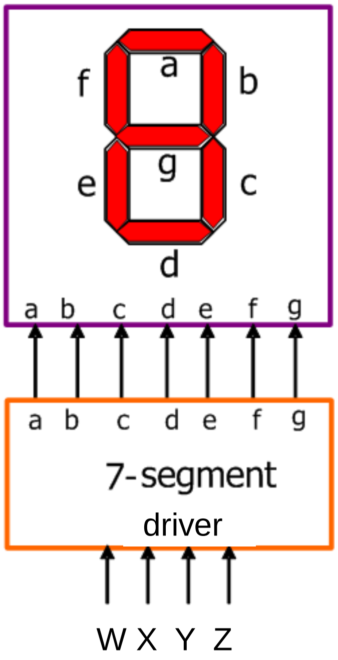


Question 2. **(16 Points)**

Design a 3-bit decrementer using only basic gates (AND, OR, and NOT). The circuit takes a 3-bit unsigned number **$I=I_2I_1I_0$** as input and generates a 3-bit output number **$Z = Z_2Z_1Z_0$** and a **Valid** output **V.** Whenever **$I > 0$** the output **$Z = I-1$** and **V=1**. If **I=0**, the output is invalid which is indicated by an output **V=0.** Derive the simplified Boolean expressions of all outputs.

Question 3. (12 Points)





It is required to design a 7-segment display **driver** whose input is a Hexadecimal digit such that the resulting 7-seg display is as shown above (**Note that** HEX digits larger than 9 are displayed as **A → A**, **B → b**, **C → C**, **D → d**, **E → E**, **F → F**). The driver circuit should generate the 7-segment control signals (**a** to **g**).

If a single **decoder** and number of **OR** gates are used to build this driver circuit;

a. What is the minimum size of the decoder? (3 points)

b. What is the minimum a number of OR gates required to build the 7-segment display driver circuit (3 points)

c. Draw the block diagram of the circuit showing in details how the control signals **g** and **c** are generated. (6 points)

Question 4. (**26** **Points**)

(A)

i. Determine the decimal value of the 7-bit binary number (1011010) when interpreted as:

An unsigned number	A signed-magnitude number	A signed-1's complement number	A signed-2's complement number
90	- 26	-37	-38

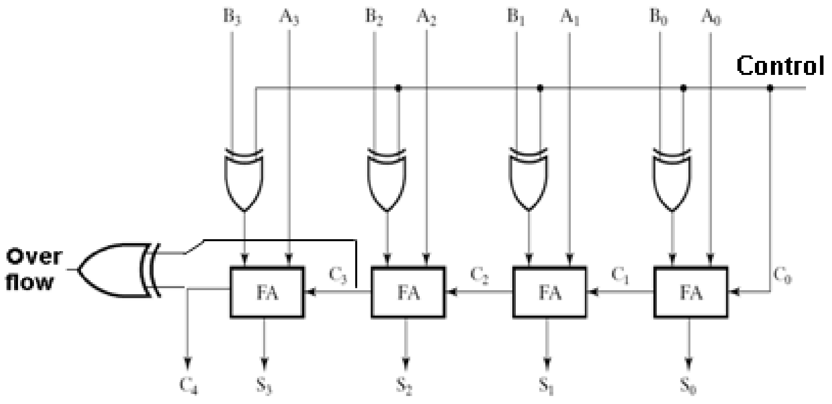
ii. Represent the decimal value (- 21) in binary <u>using a total of 7 bits</u> in the following notations:

A signed-magnitude number	A signed-1's complement number	A signed-2's complement number
1010101	1101010	1101011

iii. Perform the following signed-2's complement arithmetic operations in binary using 5 bits. All numbers given are represented in the signed-2's complement notation. Indicate clearly the <u>carry values from the last two stages</u>. For each of the three operations, check and indicate whether overflow occurred or not.

01010	1←1←1← a. 01101 +13 +10110 + -10 00011 +3 ✓	b. 01010 +10 - 11001 00111 -(-7) +17 0←1←1←1← 01010 +00111 10001 ~~+15~~	c. 11010 00110 -6 - 00100 - +4 -10 1←1← 11010 +11100 10110 = - 01010 = -10 ✓
Overflow Occurred? (Yes/No)	No	Yes	NO

(B) Consider the 2's complement 4-bit adder/subtractor hardware shown (**FA** = full adder).



i. Fill in the spaces in the table below.

Inputs			Outputs			
A	**B**	Control	**S** (binary)	C_4	C_3	Overflow
0111	0101	0	1100	0	1	1
1010	1101	1	1101	0	0	0

0←1←1←1←

$$\begin{array}{r} 0111 \\ +0101 \\ \hline 1100 \end{array}$$

0←0←1←

$$\begin{array}{r} 1010 \\ -1101 \end{array} \rightarrow \begin{array}{r} 1010 \\ +0011 \\ \hline 1101 \end{array}$$

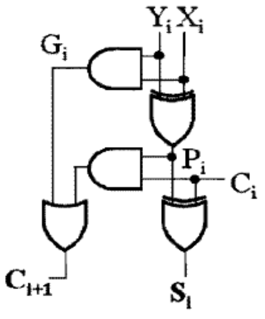
ii. What type of 4-bit adder is used in this design? (Circle the correct answer):

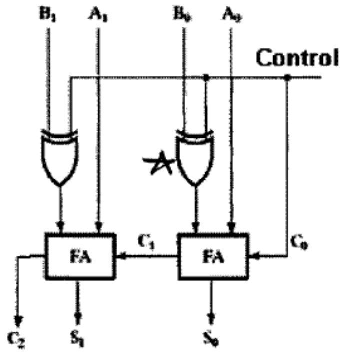
[✓ - Carry-ripple adder]

- Carry-look-ahead adder

b. Consider a 2-bit version of the hardware above which is shown below. Shown also is full adder used. Given that each basic gate (i.e. AND, OR, NOT) has a delay of τ ns and the XOR gate has a delay of 3τ:

The Full Adder (FA)





i. Express, as a function of τ, the longest time interval needed for the hardware to perform an operation on the two 2-bit numbers.

$(3 + 3 + 2 + 3)\tau = 11\tau$

ii. If such an operation must be performed in no longer than 33 ns, calculate the maximum basic gate delay allowed.

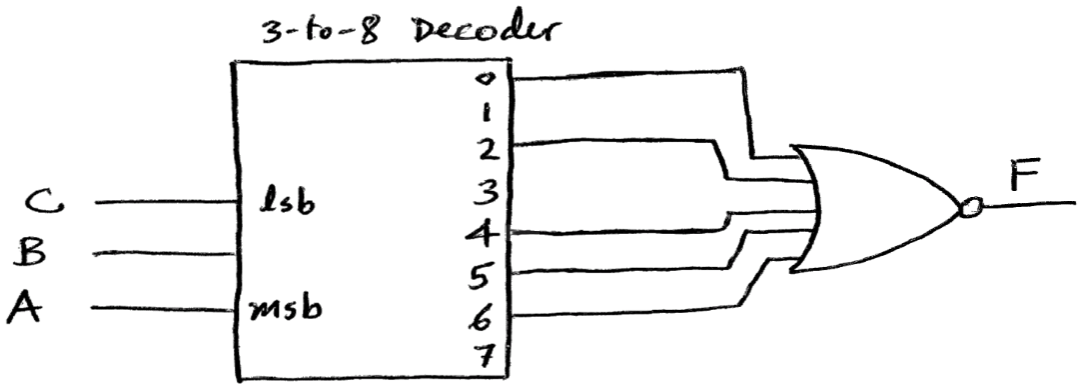
$33 \text{ ns} / 11 = 3 \text{ ns}$

Question 5. **(20 Points)**

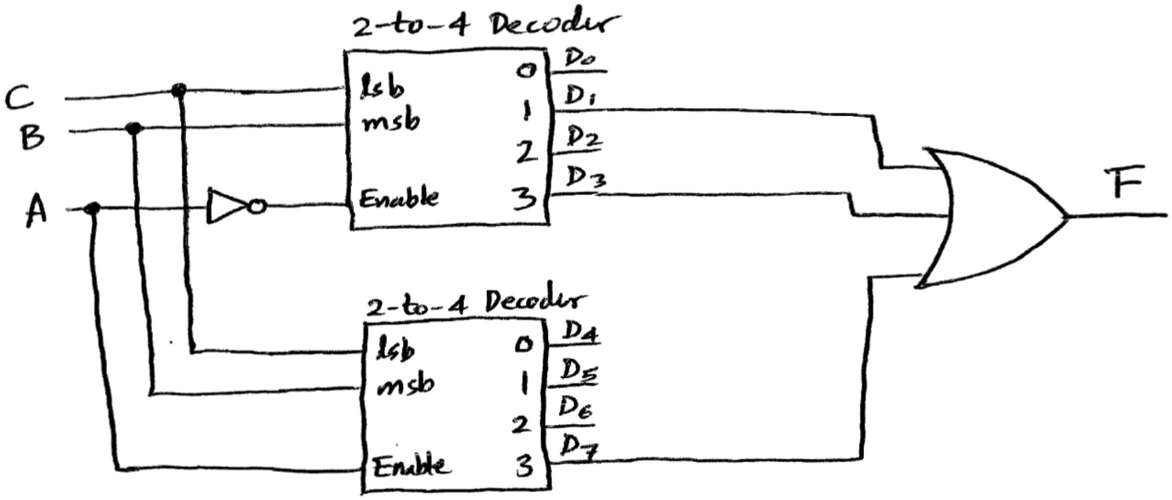
(A) Given the function

F (A,B,C) = Π M (0,2,4,5,6)

i. Implement F using one (1) 3-to-8 decoder, and one (1) **NOR** gate. Properly label all input and output lines.



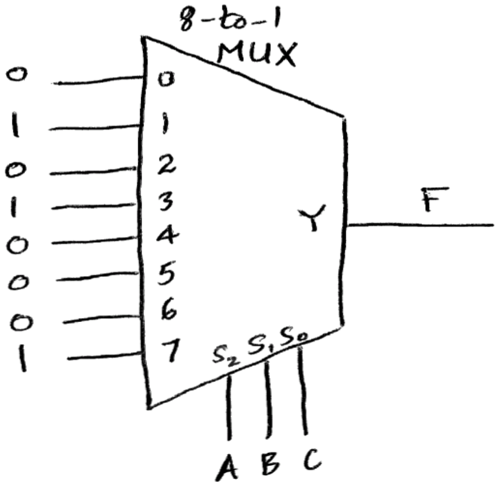
ii. Implement F using two (2) 2-to-4 decoders with enable, one (1) inverter, and one (1) **OR** gate. Properly label all input and output lines.



(B) Given the function

F (A,B,C) = Σ m (1,3,7)

i. Implement F using an 8-to-1 MUX. Properly label all input and output lines.



ii. Implement F using a 4-to-1 MUX. Show how you obtained your solution, and properly label all input and output lines.

A	B	C	F	
0	0	0	0	$I_0 = C$
0	0	1	1	
0	1	0	0	$I_1 = C$
0	1	1	1	
1	0	0	0	$I_2 = \phi$
1	0	1	0	
1	1	0	0	$I_3 = C$
1	1	1	1	

OR

	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB
$\bar{C}$	0	2	4	6
C	①	③	5	⑦
	$I_0 = C$	$I_1 = C$	$I_2 = \phi$	$I_3 = C$