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 141 (Fall 2014)

Major Exam II

Saturday November 29, 2014

Time: 150 minutes, Total Pages: 11

Name:________________________________ **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	17	
2	14	
3	10	
4	12	
5	12	
Total	65	

Question 1 **(17 Points)**

For the given K-map representing the Boolean function F, answer the following questions:

AB/CD	00	01	11	10
00		1		
01		1	1	1
11	1	1		1
10	1	1		1

(i) Which one of the following is an Implicant of F:

Term	A'C'	A'BD	AC	A'B'C'	BCD'
Implicant (Y/N)					

(ii) Which one of the following is a Prime Implicant (PI) of F:

Term	AC'	A'BC	BC'D	C'D	AD'
PI (Y/N)					

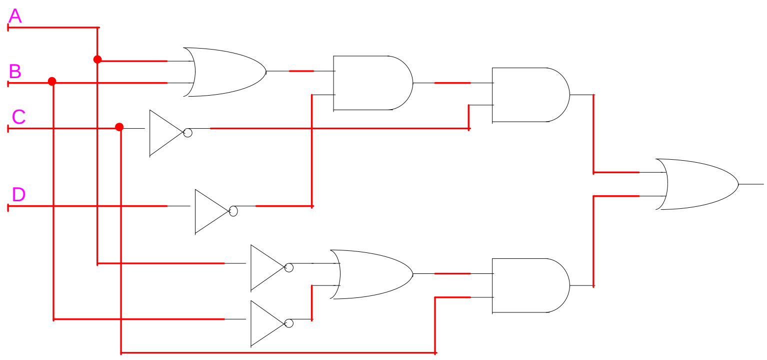
(iii) Which one of the following is an Essential Prime Implicant (EPI) of F:

	C'D	A'BC	A'C'	BC'D	AD'
EPI (Y/N)					

(iv) Obtain a simplified sum-of-product (SOP) expression for F.

(v) The following Boolean expression F= AD + A'C'D' is a simplified version of the expression F= A'B'C'D' + ABCD + AB'C'D. Are there any don`t care conditions? If so, what are they?

(vi) Implement the circuit given below **using only 2-input NAND gates**. Redraw the circuit to obtain a multi-level NAND circuit implementation. Assume that only the true form of each input variable is available.



Question 2. **(14 Points)**

(i) Fill in all blank cells in the two tables below.

Binary	Equivalent decimal value with the binary interpreted as:			
	Unsigned number	Signed-magnitude number	Signed-1's complement number	Signed-2's complement number
10110110				

Decimal	Binary representation in 8 bits:		
	Signed-magnitude representation	Signed-1's complement representation	Signed-2's complement representation
+ 100			
- 100			

(ii) Show how the following arithmetic operations are performed using 5-bit signed 2's-complement system. Check for overflow and mark clearly any overflow occurrences.

(i)
`  01001`
`- 11110`
 	
Overflow: Yes/No
	(ii)
	`  10100`
	`+ 11100`
	
	Overflow: Yes/No

(iii)
`  11111`
`+ 11111`
 	
Overflow: Yes/No
	(iv)
	`  01101`
	`- 11101`
	
	Overflow: Yes/No

Question 3. **(10 points)**

(i) It is required to design a combinational circuit that receives a 4-bit input number, X3X2X1X0, and computes the number of leading zero's in the input. For example, if the input X3X2X1X0=0111 or X3X2X1X0=0100, the output should produce a result indicating that there is a single leading zero. Construct the truth table of the circuit. You do not need to derive the Boolean expressions of the outputs. (5 points)

(ii) Using a block diagram of the design of the 4-bit leading-zero detector circuit in (i) and any other needed MSI blocks (e.g. Adder, Comparator, Multiplexer, Decoder, etc.), design a combinational circuit that receives an 8-bit input number, X7X6X5X4X3X2X1X0, and computes the number of leading zero's in the input. (5 points)

Question 4. **(12 Points)**

Using *only* the following modules:

- One 2-to-4 Decoder with enable,
- One 4-to-1 MUX,
- A maximum of five 1-to-2 DeMUXs /Decoders, and
- The minimum number of 2-input NAND gates (*If needed*)

Implement the following assuming that *signals are available only in the "True" but not the complement* form:

(i) A 3-to-8 Decoder (*you may use this decoder as a black-box in solving* (ii) and/or (iii) below)

(ii) F1(a,b)= ab+a'b'

(iii) F2(a,b,c)= m0+ m1+m2+m4+m7

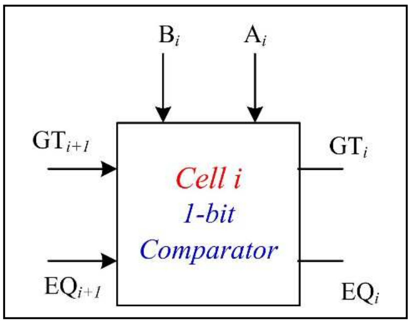
Label all your signals (inside and outside MSI components).

Question 5. **(12 Points)**

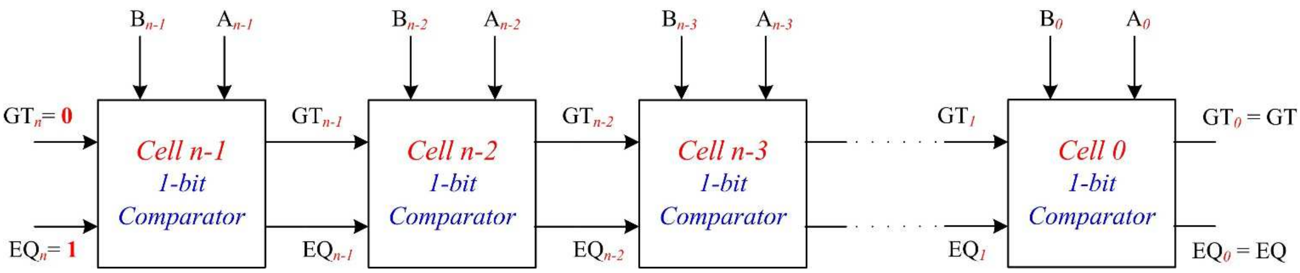
It is required to design an *n*-bit magnitude comparator. The circuit receives two *n*-bit unsigned numbers ***A and B*** and produces two outputs **GT** and **EQ** *as* given in the table to the right.

	GT	EQ
IF A > B	1	0
IF A = B	0	1
IF A < B	0	0

The input operands are processed in a bitwise manner *starting with the most significant bit (MSB)*. The comparator circuit is constructed using *n identical copies* of the basic 1-bit *cell* shown to the right. C*ell **i*** processes the ***i***th input bits (A_i and B_i) together with information passed to it from its predecessor cell (GT_{i+1} and EQ_{i+1}). It produces two output bits (GT_i and EQ_i). The *cell* output GT_i =1 *iff* ($\mathbf{A_{n-1}\, A_{n-2} \ldots A_{i+1}\, A_i > B_{n-1}\, B_{n-2} \ldots B_{i+1}\, B_i}$) and EQ_i =1 *iff* ($\mathbf{A_{n-1}\, A_{n-2} \ldots A_{i+1}\, A_i = B_{n-1}\, B_{n-2} \ldots B_{i+1}\, B_i}$).

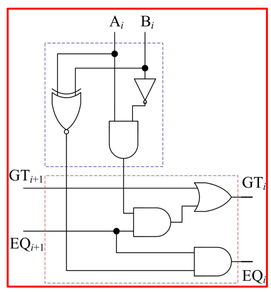


The Figure below shows the *n*-bit comparator circuit implemented using *n* copies of the basic 1-bit cell. *The output of the n-bit comparator is that of the nth cell copy* (*cell 0; the least-significant*). *Note that the inputs* $\mathbf{GT_n}$ *and* $\mathbf{EQ_n}$ *to the first cell* (*cell n-1; the most significant*) *are set to **0 and 1** respectively as there are no more significant bits.*



Boolean expressions of the outputs of ***cell i*** and its gate-level implementation are given below:

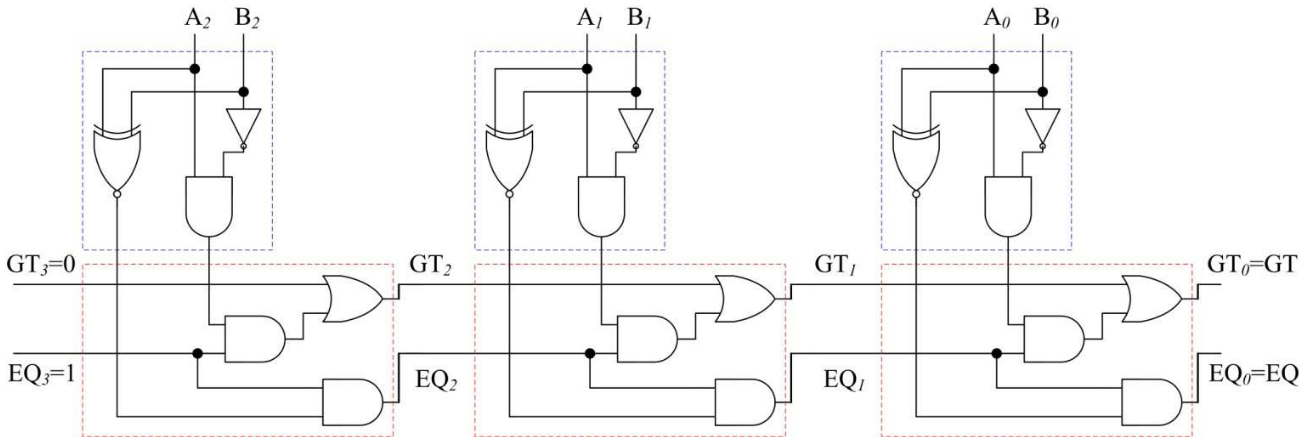
$GT_i = GT_{i+1} + A_i\, \bar{B}_i\, EQ_{i+1}$, and
$\mathbf{EQ_i = (A_i \odot B_i).\, EQ_{i+1}}$



Cell *i* Gate Level Implementation

Assuming that the **XOR** and **XNOR** gates have a delay of **2τ while** *all OTHER gates (including inverters) have a delay of* **1τ**, calculate:

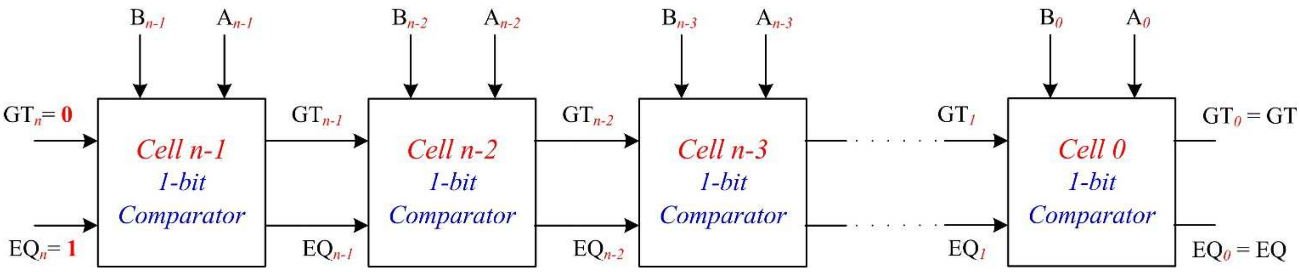
(i) The worst case delay of the 3-bit comparator (as a function of τ) shown below (4 Points)



(ii) The worst case delay of an n-bit comparator (as a function of n and τ) (3 Points)

(iii) Suggest a design for a cascadeable 3-bit comparator with lookahead capability. What is the worst case delay of this unit (using the same delay model of **2τ** for XOR/XNOR gates and **1τ** for all other gates **(**irrespective of their fanin)? (5 Points)

For convenience, the comparator circuit and Boolean expressions of the cell are repeated here.



Boolean expressions of the outputs of ***cell i*** and its gate-level implementation are given below:

$GT_i = GT_{i+1} + A_i \bar{B}_i \, EQ_{i+1}$, and

$EQ_i = (A_i \odot B_i) . EQ_{i+1}$