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 151 (Fall 2015-2016)
Major Exam 2
Saturday Nov. 21, 2015

Time: 120 minutes, Total Pages: 12

Name:_____ ID:_____ Section: _____

Notes:

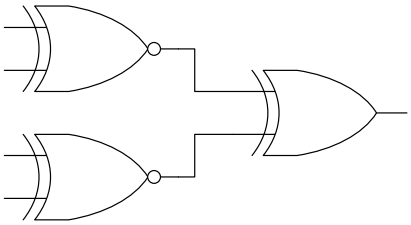
- Do not open the exam book until instructed
- **No Calculators are allowed** (*basic, advanced, cell phones, etc.*)
- Answer all questions
- All steps must be shown
- Any assumptions made must be clearly stated

Question	Maximum Points	Your Points
1	7	
2	12	
3	7	
4	12	
5	16	
6	8	
7	11	
Total	73	

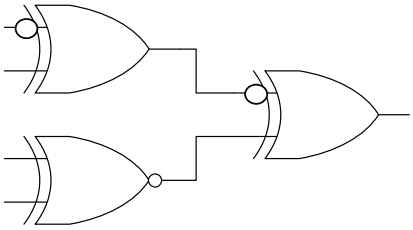
Question 1. Choose the correct answer (one answer only) **(7 Points)**

1) Which of the following represents a 4-input XNOR function?

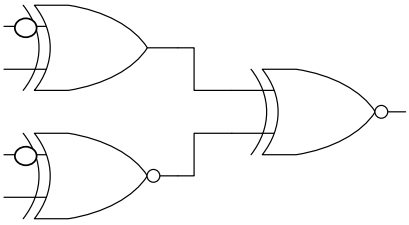
a)



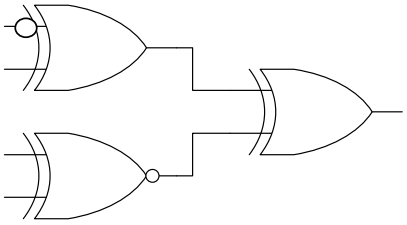
b)



c)



d)



2) NOR-OR (NOR first level and OR second level) function implementation is equivalent to:
   a) NAND-OR
   b) AND-NOR
   c) NOR-AND
   d) OR-NAND

3) Minimizing the shown k-map results in:
   a) 2 terms, 2-variable each
   b) 2 terms, 1-variable each
   c) 1 term with 2 variables
   d) 1 term with 1 variable

X			1
1			1

4) Considering $F(w, x, y, z)$, which of the following represents a <u>single</u> prime implicant having the largest area in a k-map (i.e., the largest group of 1's):
   a) $w + \bar{x} + y + z$
   b) $wx + \overline{yz}$
   c) yz
   d) $\bar{w}x\bar{y}z$

Question 2. **(12 Points)**

1) Represent $F(x, y, z) = (\bar{x} + z)(x + y + \bar{z})(x + \bar{y} + z)$ in the k-map shown below

x \ y z	00	01	11	10
0				
1				

2) Given $F(A, B, C, D)$ shown in the k-map

a) List all essential prime implicants

b) Obtain minimized SOP expression of F

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

3) Given function $F(w,x,y,z) = \sum(2,4,10,12,14)$with don't care conditions $d(w,x,y,z) = \sum(1,5,6,8)$
   a) Use k-maps to provide minimized POS expression for F

   b) Implement F using minimum number of 2-input NOR gates

Question 3. **(7 Points)**

Given an n-bit signed 2's complement number, **X**, it is required to design an iterative combinational circuit to compute the 2's complement of **X**.

(i) Show the inputs and outputs of the 1-bit 2's complement iterative cell to be used for designing the n-bit 2's complement circuit. (2 Points)

(ii) Show the truth table of the 1-bit 2's complement cell. (2 Points)

(iii) Obtain simplified equations for the outputs of the 1-bit 2's complement cell using only the following gate types: NOT, AND, OR, XOR. (2 Points)

(iv) Using the 1-bit 2's complement cell, draw a block diagram for a circuit to compute the 2's complement of a 3-bit number X. (1 Point)

Question 4. **(12 Points)**

(i) It is required to design a circuit to compute the equation **Y=|7*X|**, i.e., Y is equal to the absolute value of 7*X, where X is a **4-bit signed number in 2's complement representation**. Your circuit should be designed using the minimum number and sizes of the following MSI components (Adders, Multiplexers) and additional logic gates if needed. Show clearly the size of all used components.

(7 Points)

(ii) It is required to design a circuit to compute the equation **Y=X mod 5**, i.e. Y is the remainder of dividing X by 5, where is **X is a 4-bit unsigned number**. For example, 9 mod 5=4 and 10 mod 5=0. Your circuit should be designed using the minimum number and sizes of the following MSI components (Decoder, Encoder) and additional logic gates if needed. Show clearly the size of all used components.

(5 Points)

Question 5. **(16 Points)**

(i) Fill in all blank cells in the table below. [4 points]

Binary (6-bits)	Equivalent decimal value with the binary interpreted as:			
	Unsigned integer	Signed-magnitude number	Signed-1's complement number	Signed-2's complement number
110110				

(ii) Fill in all blank cells in the table below. [6 points]

Decimal	Binary representation in **6 bits:**		
	Signed-magnitude representation	Signed-1's complement representation	Signed-2's complement representation
+ 29			
- 29			
	Binary representation in **8 bits:**		
	Signed-magnitude representation	Signed-1's complement representation	Signed-2's complement representation
- 29			

(iii) Show how the following arithmetic operations are performed using 6-bit signed 2's-complement system. Check for overflow and mark clearly any overflow occurrences. [6 points]

(1) 110100 - 111110 Overflow: Yes/No	(2) 110111 +111000 Overflow: Yes/No
(3) 111110 + 111111 Overflow: Yes/No	(4) 001101 - 111101 Overflow: Yes/No

Question 6. **(8 Points)**

(i) Show the implementation of the function $F(x,y,z) = \sum m(0,2,5,6,7)$ [6 points]

(a) Using a single MUX of minimum size

(b) Using a minimum size decoder and a single gate with minimum number of inputs

(ii) Implement a 4-to-1 MUX using a minimum number of 2-to-1 MUXs. Clearly mark and label all inputs. [2 points]

Question 7. **(11 Points)**

(iii) A piece of hardware is described as a Verilog module. One of the given below Verilog codes is the correct description of this piece:

a. Indicate which of these codes is valid and which is invalid fully justifying your answer? (3 Points)

b. Give the logic diagram of this piece of hardware? (3 Points)

	module V_Q7A (**output reg** z, u, **input** x, w, t, q);
	wire v, y;
	assign y = (q==1'b1) ? v : w; //if q=1 y=v else y=w
	always @(x,y,t)
	begin
	z = (x^y)^t;
	u = (x&y) | (x&t) | (y&t);
	end
	not (v , w); //inverter gate instance
	endmodule

	module V_Q7B (**output reg** z, u, **input** x, w, t, q);
	wire v;
	reg y;
	always @(x,q,t)
	begin
	assign y=(q==1'b1)? v:w; //if q=1 y=v else y=w
	z = (x^y)^t;
	u = (x&y) | (x&t) | (y&t);
	end
	not (v , w); //inverter gate instance
	endmodule

	module V_Q7C (**output reg** z, u, **input** x, w, t, q);
	reg v, y;
	always @(x,q,t, v)
	begin
	if (q ==1'b1) y = v ; else y = w;
	z = (x^y)^t;
	u = (x&y) | (x&t) | (y&t);
	not (v , w); //inverter gate instance
	end
	endmodule

(iv) You are to write a test bench for the 4-bit adder module which has the following declaration: (5 Points)

module adder4 **(output reg [3** : 0**]** sum , **output reg** cout , **input [3** : 0**]** A , B**) ;**

Use the shown test patterns

Time Unit	**A**	**B**
0	5	6
10	15	9
20	9	3
30	13	14

Verilog Primitives

- Basic logic gates only
  - and
  - or
  - not
  - buf
  - xor
  - nand
  - nor
  - xnor

These gates are expandable: 1st node is O/P node, followed by 1, 2, 3 … number of input nodes

Verilog Operators

{ }	concatenation
+ - * / **	arithmetic
%	modulus
> >= < <=	relational
!	logical NOT
&&	logical AND
||	logical OR
==	logical equality
!=	logical inequality
===	case equality
!==	case inequality
? :	conditional

~	bit-wise NOT
&	bit-wise AND
|	bit-wise OR
^	bit-wise XOR
^~ ~^	bit-wise XNOR
&	reduction AND
|	reduction OR
~&	reduction NAND
~|	reduction NOR
^	reduction XOR
~^ ^~	reduction XNOR
<<	shift left
>>	shift right