A. CHEHAB DEPARTMENT OF ELECTRICAL AND COMPUTER EGINEERING
M. MANSOUR AMERICAN UNIVERSITY OF BEIRUT

FALL TERM 2004-2005

MIDTERM EE/CCE 2007

EECE320 - DIGITAL SYTEMS DESIGN

NOVEMBER 23, 2003

NAME: _____ ID: _____

COURSE SECTION: SECTION 1 (PROF. CHEHAB)

SECTION 2 (PROF. MANSOUR)

INSTRUCTIONS:

- **THE EXAM IS CLOSED BOOK/CLOSED NOTES. THE DURATION IS TWO HOURS.**
- **CALCULATORS ARE NOT ALLOWED.**
- **WRITE YOUR NAME AND ID NUMBER IN THE SPACE PROVIDED ABOVE.**
- **INDICATE THE SECTION YOU ARE REGISTERED IN.**
- **PROVIDE YOUR ANSWERS IN THE SPACE PROVIDED ON THE QUESTION SHEET.**
- **THE SCRATCH BOOKLET WILL NOT BE CONSIDERED IN GRADING.**
- **BE AS NEAT AND CLEAR AS POSSIBLE.**

Problem	Total Points	Earned Points
1	20	
2	8	
3	6	
4	10	
5	10	
6	10	
7	8	
8	10	
9	8	
10	16	
11	16	
Total	122	

Problem 1: Short questions [**20 points**]

Answer the following questions:

(CORRECT answer = +1point, NO answer = 0 point, WRONG answer = -1 point)

1. The acronym **VHDL** stands for _____ _____.

2. In logic minimization, a *minimal* sum can be the *canonical* sum. **True / False** (Circle one).

3. Any Boolean function can be built using NAND gates. **True / False** (Circle one).

4. State the *consensus* theorem: _____.

5. Consider the binary number 1010111 represented in signed-magnitude notation. The corresponding 2's complement representation is _____.

6. The sum of $AB.C_{16}$ and 56.7_{8} in **base 4** is _____.

7. The decimal value of the smallest *n*-bit signed-magnitude integer is smaller than the decimal value of the smallest *n*-bit 2's complement integer. **True / False** (Circle one).

8. Consider the unsigned *n*-bit integer *X*. Let *Y* be the unsigned integer obtained by shifting *X p* positions to the left and inserting ones on the right. Determine *Y* in terms of *X* and *p*. **Answer**; *Y* = _____.

9. In Boolean algebra, if $X = X + Y$, then $Y = 0$. **True / False** (Circle one).

10. Within a *process* in VHDL, statements execute sequentially. **True / False** (Circle one).

11. A *with-select* construct can be used within a *process* statement in VHDL. **True / False** (Circle one).

12. How many pairs of Boolean functions $F(a,b,c,d)$ and $G(a,b,c,d)$ are there that are complements of each other? (Don't count a pair more than once.) **Answer**: _____.

13. Write the `entity` declaration of a 2-to-4 decoder with input *a* and output *b* assuming standard logic vectors.

14. Complete the following architecture definition for the prime-number detector using **`select`** signal assignment, where *N* is a 4-bit input and *F* is the output that is asserted if *N* is prime:

```
ARCHITECTURE prime4_arch OF prime IS
BEGIN




END prime4_arch;
```

15. Complete the following architecture definition for decoder with output *Y*:

```
ARCHITECTURE V2to4dec_b OF V2to4dec IS
      SIGNAL Y_s : STD_LOGIC_VECTOR(0 TO 3) ;
BEGIN
      PROCESS(IN, EN)
      BEGIN
            CASE  IN  IS
                  WHEN "00"  =>  Y_s  <=  "1000" ;
                  _____________________________________;
                  _____________________________________;
                  _____________________________________;
                  _____________________________________;

            END CASE ;

            IF EN = '1'   THEN   _______________;
            ELSE   _______________;
            END IF ;
      END PROCESS;
END V2to4dec_b;
```

16. What is the dual of $F=\Sigma_{X,Y,Z}(1,3,5,7)$? **Answer**: _____.

17. Define a static 1 hazard: **Answer**: _____
_____.

18. A function of *n* variables that includes all the 2^n minterms is equal to _____.

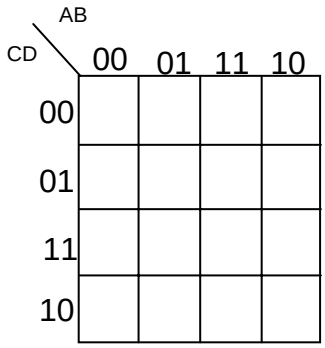
19. State the *Prime-Implicant* theorem: _____.

20. What property does a *Gray* code have? _____
_____.

Problem 2 [8 points]

Consider the function: $F = \Sigma_{A,B,C,D}(1, 3, 4, 5, 10, 11, 12, 13, 14, 15)$

a) What are the prime implicants of F?



_____

b) What are the essential prime implicants of F?

_____

c) What is a minimum SOP expression for F?

_____

d) What is a minimum POS expression for F?

_____

Problem 3 [6 points]

Draw the gate implementation of the expression $F = (X_1+X_2')(X_2+X_3)$

a) What type of hazard is possible with this implementation?

_____

b) Indicate which transition causes such hazard

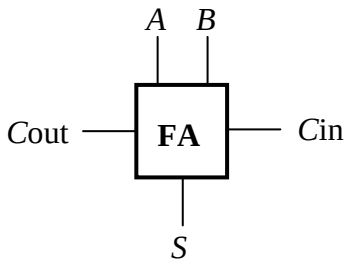
_____

c) How do you modify the implementation to obtain a hazard-free circuit?

Problem 4: Boolean Function Implementations [**10 points**]

a) Implement the Boolean function $F = A$ XOR B using NAND gates. You should use the minimum number of NAND gates in your implementation. Label all inputs and outputs. First use the scratch booklet, and then write your final answer clearly below.

b) Consider the 1-bit full adder gate (FA) shown below. It takes as inputs ***A*** and ***B*** and a carry input ***C*in**, and generates the sum ***S*** of ***A***, ***B***, and ***C*in**, and a carry output ***C*out**. Implement each of the following functions using one FA gate. Label all your inputs and outputs clearly according to the function to be implemented. Any unused outputs should be connected to ground.



$F_1 = X.Y$	$F_2 = X + Y$	$F_3 = X$ XOR Y
$F_4 = X$ XNOR Y	$F_5 = X'$	$F_6 = X$ XNOR Y XNOR Z

Problem 5: Shannon's Theorem [**10 points**]

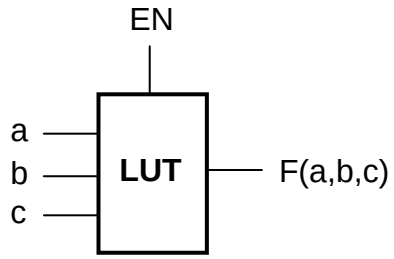
Consider the Boolean function $F(a,b,c,d,e,f,g,h) = a'b'c'd + ag'h + ag + a'b'ce + a'bf''$. Using Boolean algebra, show that F can be implemented with 2-to-1 multiplexers only (no other gates or inverters). Inputs to the circuit are the variables a, b, c ,... and the constants 0 and 1. Draw the resulting logic circuit.

Problem 6: Majority Function [**10 points**]

An *n*-bit majority function generates a 1 if more than half the input bits are one (i.e., *n*/2 or more bits are one assuming *n* even). Design a 4-bit majority function *F* using a single 4-to-1 multiplexer and minimal extra gates.

Problem 7: Look-Up Tables [**8 points**]

A look-up table (LUT) implements a Boolean function by memorizing its truth table. For a given input, a LUT simply 'looks up' the corresponding output. A 3-input LUT with enable is shown below. For example, a LUT that implements the function F(a,b,c)=a+bc produces F(0,1,1)=1 when a=0, b=1,c=1.



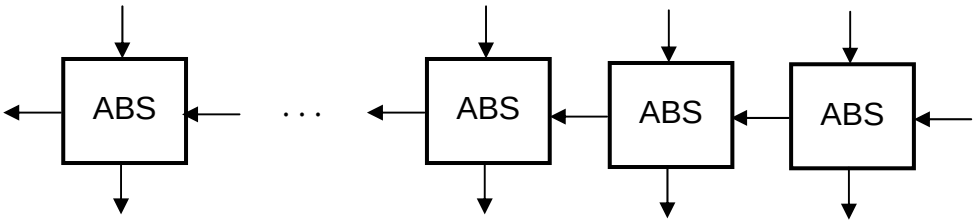
a) Show how to implement a Boolean function G(a,b,c,d,e) of five variables using 3-input LUTs with enable and a multiplexer. Label all inputs and outputs.

b) Repeat part a) but now using 3-input LUTs with enable and a decoder (with minimal extra gates).

Problem 8: Absolute Value Computer [**10 points**]

In this problem, we are interested in designing a logic circuit that determines the absolute value of an *n*-bit 2's complement number $A_{n-1} \cdots A_2 A_1 A_0$. Since *n* can be large, it is not practical to implement the circuit by first constructing a truth table. Hence, we'll build the circuit using identical simple blocks that connect to each other as shown below.

Design the simple block ABS so that the circuit computes the absolute value of its input. Try to use as few logic gates as possible in your design. Clearly label all inputs and outputs. Write down Boolean equations for all outputs of the block ABS.



Problem 9: Algebraic Manipulations [**8 points**]

Prove the identity of the following Boolean equation using algebraic manipulations. Justify each simplifying step you make by stating the corresponding theorem or axiom you use.

$$WY + W'YZ' + WXZ + W'XY' = WY + W'XZ' + X'YZ' + XY'Z$$

Proof:

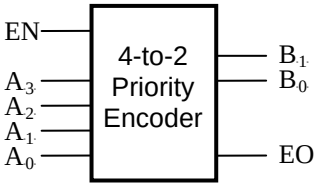
Problem 10: Dual Priority Encoder [**16 points**]

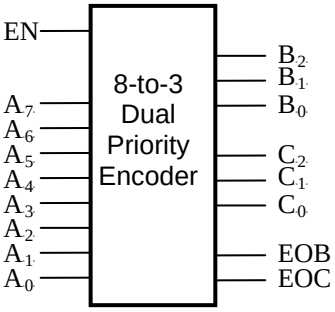
In this problem, you are asked to design an 8-to-3 priority encoder that identifies the inputs with *highest* and *second* highest priorities using regular 4-to-2 priority encoders with enable. The inputs of the 4-to-2 encoder are defined as follows:

EN: When asserted, block functions as an encoder; otherwise, outputs are zero.
A_3,A_2,A_1,A_0: Data inputs where **A_3** has highest priority.
B_1,B_0: Outputs identifying highest priority input.
EO: Asserted if EN=1 and at least one is A_i one.

The output signals **B_2, B_1, B_0** of the 8-to-3 dual priority encoder identify the inputs with highest priority respectively. The output signals **C_2, C1, C0** identify the inputs with *second* highest priority. The signal EOB is asserted if a highest priority input is detected, and EOC is asserted if a second-highest priority input is detected.

In your design, you can use standard logic gates (AND, OR, NOT) as well as 2-to-4 decoders with enable.





Problem 11: VHDL [**16 points**]

We would like to design a comparator that compares two 8-bit inputs and produces one output that is set to one if the numbers are equal and 0 otherwise. The design is to be iterative. So, first design a 1-bit comparator that takes two input data bits and one cascaded input, and produces one cascaded output. Find the expression of the output in terms of the inputs.
Write a VHDL entity and architecture that implements the 1-bit comparator. Then, write a VHDL entity and architecture that implements the 8-bit comparator by cascading 1-bit comparators. Finally, write a testbench that tests the implementation with the following inputs:

A = 0x23 and B = 0x32
A = 0x34 and B = 0x34
A = 0xFF and B = 0x00