University of Toronto

Department of Electrical and Computer Engineering

ECE241- Digital Systems

Final Examination

December 2001

TYPE A (No Aids)

Last Name: _____

First Name: _____

Student Number: _____

Signature: _____

Duration: **2.5 hours**

Answer ALL questions on this test paper.

There is extra space at the end if you need it.

EXAMINER's REPORT

1	_____	/20
2	_____	/24
3	_____	/22
4	_____	/18
5	_____	/16
TOTAL	_____	/100

Question 1 - [20 marks]

1. Implement the following function by using exactly one 4-to-1 multiplexer, one NAND gate and one NOR gate.

$f(x_1,x_2,x_3,x_4)= \sum m(0,1,2,4,8)$

[4 marks]

2. Implement the following function by using exactly one 4-to -1 multiplexer, one NAND gate and one NOR gate.

$g(x_1,x_2,x_3,x_4)= \sum m(2,3,5,7,11,13)$

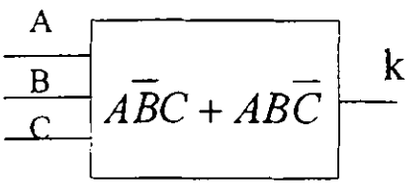
[4 marks]

3. Show how the following functions can be implemented using 3-input lookup tables (LUT).

$f(x_1,x_2,x_3,x_4)= \sum m(0,1,2,4,8)$

$g(x_1,x_2,x_3,x_4)= \sum m(2,3,5,7,11,13)$

Use the **least number of LUTs** that you can. Draw the circuit containing the LUTs and indicate the function of each LUT by showing the logic function that it implements. For instance, if a LUT implements the function $k = A\bar{B}C + AB\bar{C}$, then you would draw the LUT in your circuit as shown below.



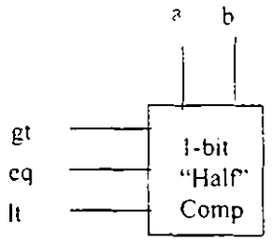
[Correct: 6 marks]

[Minimal: 6 marks]

Question 2 – [20 marks]

A circuit has two inputs ***a*** and ***b*** and three outputs ***gt***, ***eq*** and ***lt***. The outputs correspond to the results of the comparison between ***a*** and ***b***. Each output is equal to 1 when the result of the comparison is true and 0 otherwise. This circuit could be called a "half comparator".

gt = (a>b)
eq = (a=b)
lt = (a<b)



1. Give the expression for ***gt***, ***eq*** and ***lt*** by using AND, OR, NOT and XOR operators.

[3 marks]

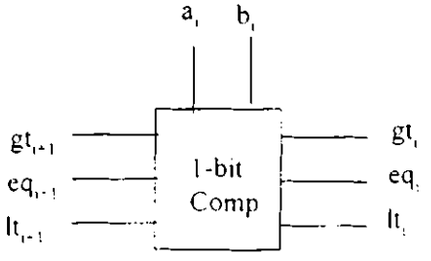
ANSWER

gt =

eq =

lt =

2. We now want to compare two 4-bit unsigned numbers A ($a_3a_2a_1a_0$) and B ($b_3b_2b_1b_0$) by using a "ripple comparator". The basic block is the "1-bit comparator" which has five inputs a_i, b_i, gt_i, eq_i and lt_i and three outputs gt_{i+1}, eq_{i+1} and lt_{i+1}. It delivers the results of the comparison for stage i (gt_{i+1}, eq_{i+1} and lt_{i+1}) according to a_i, b_i and the results of the comparisons for the previous stage (gt_i, eq_i and lt_i).
gt, ***eq*** and ***lt*** correspond respectively to greater than, equal; and less than.



Give the expressions for gt_{i+1}, eq_{i+1} and lt_{i+1} by using AND, OR, NOT and XOR operators.

[3 marks]

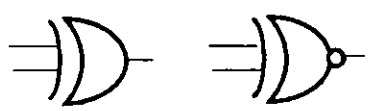
ANSWER

gt_{i+1} =

eq_{i+1} =

lt_{i+1} =

3. Design the corresponding circuits using NOT, NAND, XOR and XNOR gates. NAND gates can have any number of inputs. XOR and XNOR gates have two inputs. The output of a XNOR gate is 1 when the two inputs are equal; and 0 otherwise. The symbols for XOR and XNOR gates are shown below.



XOR **XNOR**

[4 marks]

ANSWER

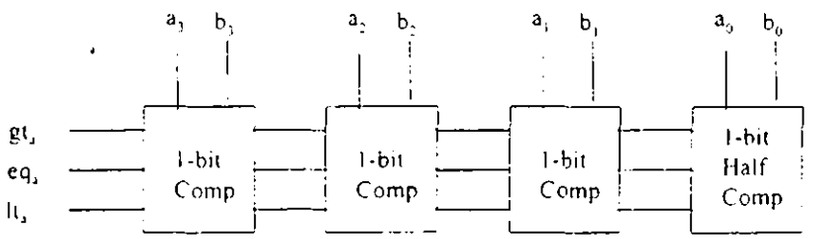
The propagation delay of NOT and NAND gates is $(p+1)/2$ ns, where p is the number of inputs. Not gate propagation delay is 1 ns. A 2-input NAND gate propagation delay is 1.5 ns, etc. The propagation delay of XNOR and XOR gates is *2* ns.

4. Fill out the following table which gives the propagation delay between the inputs and the output for the 1-bit comparator

[3 marks]

INPUT	TO	OUTPUT	Propagation delay (ns)
a_i or b_i (worst case)		gt_{i+1}	
a_i or b_i (worst case)		eq_{i+1}	
a_i or b_i (worst case)		lt_{i+1}	
gt_i		gt_{i+1}	
eq_i		eq_{i+1}	
lt_i		lt_{i+1}	

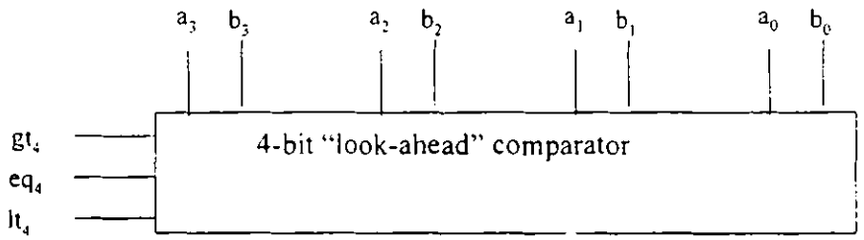
5. What is the worst case propagation delay for the 4-bit ripple comparator between the inputs and the final results of the comparison (gt_4, eq_4 and lt_4) ? The scheme of the 4-bit ripple comparator is shown below



ANSWER (Show what is the critical path in the 4-bit comparator and give the value of the propagation delay in ns)

[4 marks]

We now use the "look-ahead" technique to implement the 4 bit comparator (shown below)



6. Give the expressions of gt_4, eq_4 and lt_4 for the 4-bit look-ahead comparator.

[3 marks]

gt_4 =

eq_4=

lt_4 =

7. What is the worst case propagation delay using the same assumptions as for the ripple comparator?

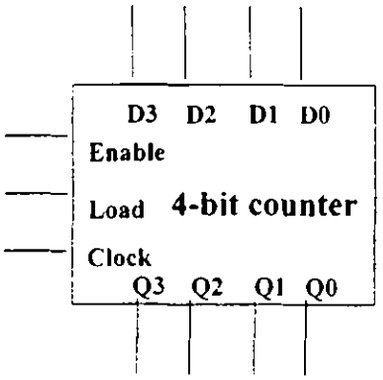
ANSWER (Show what is the critical path in the 4-bit comparator and give the value of the propagation delay in ns)

[4 marks]

Question 3 - [22 marks]

1 Design a modulo-11 counter with an Enable input by using

- the 4-bit counter below, which has Enable and Synchronous Load inputs
- logic gates (you can use NOT, AND, OR, NAND, NOR gates with any number of inputs).



ANSWER

[4 marks]

2. Design a ***modulo-12*** counter with four D Flip-Flops.
Give the minimal sum of products expression for inputs D_3, D_2, D_1 and D_0 of the flip-flops (the flip-flop outputs are Q_3, Q_2, Q_1 and Q_0; where Q_3 is the most significant and Q_0 is the least-significant bit).

[Correct: 4 marks]
[Minimal: 2 marks]

ANSWER

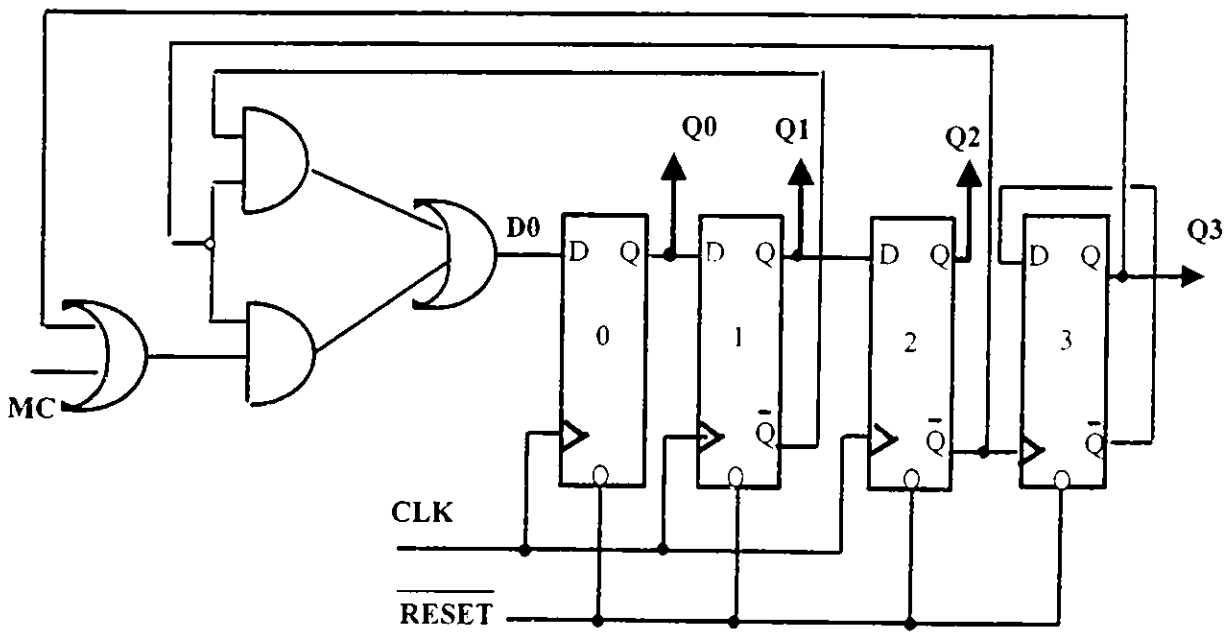
D_3=

D_2=

D_1=

D_0=

3. The circuit below has 4 D Flip Flops, 1 input MC (Mode Count) and a clock CLK.



Give the sum of products expression for D_0 as a function of the FF outputs when MC=0 and when MC=1.

ANSWER

[2 marks]

MC=0

D_0 =

MC=1

D_0 =

4. Assume that all the FFs are initially reset by the asynchronous RESET signal.

What are the successive values of Q_0 Q_1 Q_2 Q_3 after each clock cycle when MC=0?

ANSWER on the NEXT PAGE **[4 marks]**

5. All the FFs are again reset by the asynchronous RESET signal.

What are the successive values of Q_0 Q_1 Q_2 Q_3 after each clock cycle when MC=1?

ANSWER on the NEXT PAGE **[4 marks]**

	MC	= 0				MC	=1	
Q0	Q1	Q2	Q3		Q0	Q1	Q2	Q3
0	0	0	0		0	0	0	0

What is the function of the circuit?.

ANSWER: **[3 marks]**

Question 4 - [18 marks]

A synchronous sequential circuit has two input lines, x_1 and x_0, and two output lines, z_1 and z_0. At each clock cycle, the combination $x_1\ x_0$ constitutes a 2-bit binary number (x_1 is the most significant bit and x_0 is the least significant bit). If the present value of the input number is less than the immediately preceding value, than the outputs are $z_1\ z_0 = 10$. If the present value of the input number is greater than the immediately preceding values, than the outputs are $z_1\ z_0 = 01$. If it is the same, $z_1\ z_0 = 00$.
The FSM can be implemented as a Mealy machine with 2 D flip-flops.

1. Complete the state table below that corresponds to the Mealy implementation.

[8 marks]

PS	NS	NS	NS	NS	$z_1\ z_0$	$z_1\ z_0$	$z_1\ z_0$	$z_1\ z_0$
	$x_1\ x_0$	$x_1\ x_0$	$x_1\ x_0$	$x_1\ x_0$	$x_1\ x_0$	$x_1\ x_0$	$x_1\ x_0$	$x_1\ x_0$
Q_1Q_0	00	01	11	10	00	01	11	10
00								
01								
11								
10								

2. Give the minimal sum of products expressions for the inputs of the D flip-flops and for the outputs.

ANSWER: (Use next page for rough work)

[Correct: 4 marks]
[Minimal: 4 marks]

D_1

D_0

z_1

z_0

Question 5 - [20 marks]

The VHDL code given below is supposed to implement a finite state machine. It contains a significant mistake.

[5 marks]

1. Correct the mistake by giving the correct code in place of the wrong code.

CORRECT VERSION OF THE CODE

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY fe2001 IS
        PORT ( Clock, Resetn, X : IN     STD_LOGIC ;
                        Z               : OUT   STD_LOGIC ) ;
END fe2001;

ARCHITECTURE Behavior OF fe2001 IS
        TYPE State_type IS (A, B, C, D) ;
        SIGNAL state, next_state : State_type ;
BEGIN
        PROCESS ( Resetn, Clock )
        BEGIN
                IF Resetn = '0' THEN
                        state <= A ;
                ELSIF (Clock = '1') THEN
                        state <= next_state          ;
                END IF;
                        END PROCESS ;
        PROCESS ( X, state )
        BEGIN
                CASE state IS
                        WHEN A =>
                                IF X='1' THEN
                                        next_state <= B ;
                                        ELSE
                                        next_state <= A ;
                                END IF;
                        WHEN B =>
                                IF X='1' THEN
                                        next_state <= C ;
                                        ELSE
                                        next_state <= A ;
                                END IF;
                        WHEN C =>
                                IF X='1' THEN
                                        next_state <= C ;
                                        ELSE
                                        next_state <= D ;
                                END IF;
                        WHEN D =>
                                IF X='1' THEN
                                        next_state <= B ;
                                        ELSE
                                        next_state <= A ;
```

```
                        END IF;
            END CASE ;
        END PROCESS ;

        PROCESS ( X. state )
        BEGIN
            CASE state IS
                WHEN A =>
                    Z <='0' ;
                WHEN B =>
                    Z <='0' ;
                WHEN C =>
                    Z <='0' ;
                WHEN D =>
                    IF X='1' THEN
                        Z <='1' ;
                        ELSE
                        Z <='0' ;
                    END IF;
            END CASE ;
        END PROCESS ;
END Behavior ;
```

2. Draw the state diagram for the machine defined by this code.

ANSWER

[6 marks]

3. Is the FSM a Moore or a Mealy machine?

ANSWER **[1 mark]**

4. What does the FSM do?

ANSWER.

[4 marks]

EXTRA SPACE - USE ONLY IF NEEDED

EXTRA SPACE - USE ONLY IF NEEDED

EXTRA SPACE - USE ONLY IF NEEDED