University of Toronto

Department of Electrical and Computer Engineering

Faculty of Applied Science and Engineering

Final Examination - December 10, 1997

Second Year - Programs 7 and 9

ECE241 - Digital Systems

Examiners: S.D. Brown and J.S. Rose

EXAM Type: D

Last Name: ____________________

First Name: ____________________

Student Number: ____________________

Duration: **2.5 Hours**

You should answer ALL questions except question 5b, which is a bonus question.
All answers should be on these sheets.
No aids permitted.
See Instructions on page 2.

EXAMINER's REPORT

1	______	/16
2	______	/22
3	______	/22
4	______	/18
5	______	/22
TOTAL:	______	/100

Instructions

- READ ALL QUESTIONS CAREFULLY BEFORE ANSWERING.
- Attempt all questions.
- If you need to make any assumptions, state them clearly with your answers.
- Write your answers neatly. Messy work is very hard to read and may cause you to lose marks.
- For questions that specify minterms using the notation in the example below

$$f(x_1, x_2, x_3) = \sum m(1, 3, 5)$$

the minterms are to be interpreted as used in class. Specifically, minterm m_1 represents $x_1x_2x_3 = 001$, minterm m_3 represents $x_1x_2x_3 = 011$, and so on. Answers that fail to interpret the minterms in this way will be considered incorrect.

Question 1 — Short Answers **[16 marks]**

a. Use algebraic manipulation to derive the minimum-cost expressions in sum-of-products form for the following expressions. Be sure to show all of your steps.

i. $f = x \oplus y \oplus xy$

ii. $g = BC\overline{D} + \overline{B} + \overline{C}$

b. Use **one** 4-to-1 multiplexer and NOT gates to implement the following function:

$$f(x_1, x_2, x_3) = \sum m(3, 5, 6)$$

c. The function $f(A, B, C) = \sum m(3, 4, 6)$ can be implemented using **one** 2-input multiplexer, **one** 2-input AND gate, and **one** NOT gate. Design the circuit using exactly these gates.

Question 2 — Short answers II **[22 marks]**

a. In class you learned about carry-lookahead adders. For an 8-bit carry-lookahead adder with inputs called $x_7, \ldots, x_0$ and $y_7, \ldots, y_0$, you are to derive the equation for the carry-lookahead carry-out from the second stage of the adder, c_2. Assume that the carry-in to the adder is called c_0. Draw a block diagram of the first two stages of the adder and label all of the signals, including generate and propagate. Derive the expression for c_2 in terms of x_1, y_1, x_0, y_0 and c_0. Show the steps in your derivation.

Draw the block diagram with signal labels in this space

Derive the expression for c_2 in this space

b. Represent the following numbers in 2's complement notation. Use the least number of bits possible for each number:

i. 37

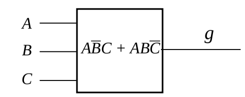
ii. -52

iii. -128

c. Show how the following function can be implemented using 3-input lookup tables (LUTs).

$$f(x_1, x_2, x_3, x_4) = \sum m(1, 2, 4, 7, 8, 11, 13, 14)$$

Use the least number of lookup tables that you can. Draw the circuit containing the LUTS, and indicate the function of each LUT by showing the logic expression that it implements. For example, if a LUT implements the function $g = A\overline{B}C + AB\overline{C}$, then you would draw this LUT in your circuit as shown below:



Show your answer in this space. Use the space on the next page for rough work.

Space for rough work for Question 2c. A truth table is provided for your convenience only.

$x_1 x_2 x_3 x_4$	f
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

d. You are to design a circuit with three inputs and two outputs. The inputs are called $x_2x_1x_0$ and represent a 3-bit positive integer. The output is called s_1s_0, and represents the number of bits in $x_2x_1x_0$ that are equal to 1. What basic circuit that you have seen in class can implement this function?

Answer:

Question 3 — Word problem **[22 marks]**

a. You are to design a circuit that has four inputs called a_1, a_0, b_1, b_0, and produces four outputs called P = p_3, p_2, p_1, p_0. Each pair of input represents a 2-bit positive integer (i.e. $A = a_1a_0$ and $B = b_1b_0$) and the outputs represent the 4-bit arithmetic product (i.e., P = A x B) of those integers.

i. Fill in the truth table below that shows the four outputs.

$a_1\ a_0\ b_1\ b_0$	p_3	p_2	p_1	p_0
0 0 0 0				
0 0 0 1				
0 0 1 0				
0 0 1 1				
0 1 0 0				
0 1 0 1				
0 1 1 0				
0 1 1 1				
1 0 0 0				
1 0 0 1				
1 0 1 0				
1 0 1 1				
1 1 0 0				
1 1 0 1				
1 1 1 0				
1 1 1 1				

ii. Derive a minimal sum-of-products expression for each of p_3, p_2, p_1 and p_0, using the K-maps given below.

K-map for p_3:

a_1a_0 / b_1b_0

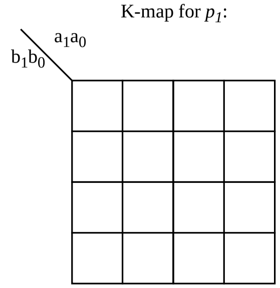
Minimum-cost expression for p_3:

K-map for p_2:

a_1a_0 / b_1b_0

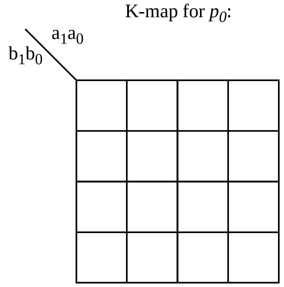
Minimum-cost expression for p_2:

K-map for p_1:



Minimum-cost expression for p_1:

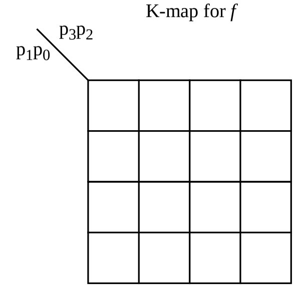
K-map for p_0:



Minimum-cost expression for p_0:

b. For this part you are to design another combinational circuit. The inputs to the circuit are p_3, p_2, p_1 and p_0 *directly from the outputs of the circuit of part* a. The circuit has one output, called f. The function f should be logic 1 only if $p_3p_2p_1p_0$ have the value 0001, 0100, or 1001. Design a circuit for f using the minimum number of NOR gates. You can use only NOR gates, but they can be of any size. Use the K-map given below. Use the space on the following page for any rough work.

K-map for f

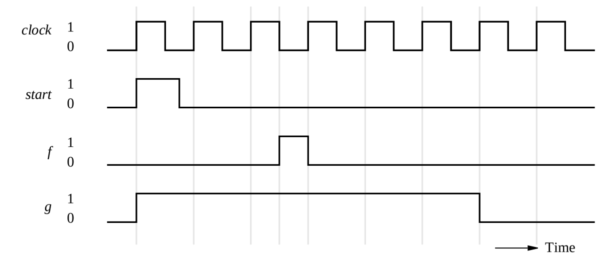


Show the circuit for f in the space below:

Space for rough work for Question 3b.

Question 4 — Sequential Circuits **[18 marks]**

a. A logic circuit has two inputs, *clock* and *start*, and two outputs, *f* and *g*. The behavior of the circuit is described by the timing diagram below. When a pulse is received on the start input, the circuit produces pulses on the *f* and *g* outputs as shown in the timing diagram. Design a suitable circuit using only the following components: 3-bit resettable positive-edge triggered synchronous counter, and basic logic gates. Assume that the delays through all logic gates and the counter are negligible.



Show your answer in this space. Use the space on the next page for rough work.

Space for rough work for Question 4a.

b. Consider the VHDL architecture body shown below.

```
ARCHITECTURE Structure OF example IS
    SIGNAL x: STD_LOGIC;
BEGIN
    tff_0: tff PORT MAP (
        t=>e, clk=>clock, clrn=>Resetn, prn=>Presetn, q=>q0);
    tff_1: tff PORT MAP (
        t=>q0, clk=>clock, clrn=>Resetn, prn=>Presetn, q=>q1);
    tff_2: tff PORT MAP (
        t=>x, clk=>clock, clrn=>Resetn, prn=>Presetn, q=>q2);
    x <= NOT(NOT q0 OR NOT q1);
END Structure;
```

Note that the corresponding ENTITY declaration is not shown, but it would define signals *e*, *clock*, *Resetn*, and *Presetn* as inputs, and signals *q0*, *q1*, and *q2* as outputs. The tff module is a positive-edge-triggered T flip-flop. The meaning of the statement

```
tff_0: tff PORT MAP (
    t=>e, clk=>clock, clrn=>Resetn, prn=>Presetn, q=>q0);
```

is to create a T flip-flop with the following connections to signals: the *t* input is connected to the *e* signal, *clk* is connected to the *clock* signal, *clrn* to *Resetn*, *prn* to *Presetn*, and *q* to *q0*.

i. In the space below, draw the circuit represented by the VHDL code.

ii. What is the purpose of this circuit (i.e., what does the circuit do)?

Answer:

iii. Assume the following timing parameters: propagation delay through a logic gate is 2 ns, time from clock edge to change of flip-flop output is 1 ns, and flip-flop setup time is 2 ns. What is the maximum clock frequency for which the circuit will work properly?

Answer:

Question 5 — Finite State Machine **[20 marks]**

a. Consider a Moore-type finite-state machine with one input, x, and one output, z. The machine should produce the output $z = 1$ if the values of x on successive clock cycles have the following pattern: 10101. Otherwise, z should be 0.

Draw a state diagram for the machine. Use the least possible number of states.

b. The state table given below represents a Moore-type finite-state machine.

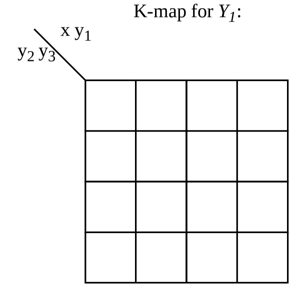
Present State	Next State		Output
	x = 0	x = 1	z
A	A	B	0
B	C	B	0
C	A	D	0
D	C	E	0
E	C	B	1

i. Using a sequential state assignment, fill in the state-assigned table below. By sequential state assignment, we mean that state A is encoded as 000, state B as 001, state C as 010, and so on. Note that the state flip-flop outputs are called y_1, y_2 and y_3.

Present State		Next State				Output
		x = 0		x = 1		z
Label	Code $y_1y_2y_3$	Label	Code $Y_1Y_2Y_3$	Label	Code $Y_1Y_2Y_3$	

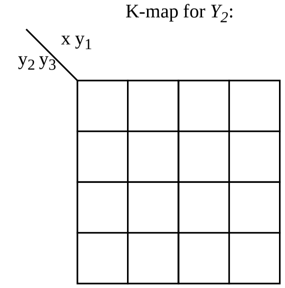
ii. Use the K-maps given below to derive minimal expressions (sum-of-products) for the state flip-flops and output to implement the machine. Make use of don't-cares if they exist.

K-map for Y_1:



Minimum-cost expression for Y_1:

K-map for Y_2:



Minimum-cost expression for Y_2:

Answer to Question 5b, part ii continued

K-map for Y_3:

$x\,y_1$ / $y_2\,y_3$

Minimum-cost expression for Y_3:

K-map for z:

$y_1\,y_2$ / y_3

Minimum-cost expression for z

iii. Draw the complete circuit for the finite-state machine.

c. BONUS QUESTION. You do not have to do this question, but extra marks will be given for it if you do. The state table below defines a finite-state machine that has one input, x, and one output, z. Using the formal procedure described in class, determine if the number of states in the machine can be reduced. Show your steps to derive the minimum possible number of states.

Present	Next State		Output
State	$x = 0$	$x = 1$	z
A	A	B	0
B	A	C	0
C	D	C	0
D	A	E	1
E	A	F	0
F	D	F	0

Answer (use the back of this page for rough work if extra space is needed.):

EXTRA SPACE — USE ONLY IF NEEDED

EXTRA SPACE — USE ONLY IF NEEDED