ECE 341 Midterm Exam

Time allowed: 90 minutes Name: _____

Total Points: 75

Points Scored: _____

Problem No. 1 (11 points)

For parts (a) through (d), indicate whether the statement is TRUE or FALSE. For parts (e) and (f) encircle the correct answer(s).

(a) Asynchronous counters are usually faster than synchronous counters. **FALSE**

(b) A shift register is capable of reading/writing data in a "serial" fashion. **TRUE**

(c) CISC instructions typically have a higher average CPI than RISC instructions. **TRUE**

(d) A sequential multiplier requires more adders than an array multiplier. **FALSE**

(e) A 16-bit **blocked carry-lookahead adder (CLA)** composed of four 4-bit CLA blocks is used to add two numbers X ($x_{15}x_{14}x_{13}\ldots\ldots x_1x_0$) and Y ($y_{15}y_{14}y_{13}\ldots.y_1y_0$). Under which of the following conditions is the carry-out c_{12} equal to 0? Encircle all the correct options:

i. $c_0 = 0$, all the CLA blocks generate a carry, but none of the CLA blocks propagate a carry

ii. $c_0 = 0$, all the CLA blocks propagate a carry, but none of the CLA blocks generates a carry

iii. $c_0 = 1$, all the CLA blocks propagate a carry, but none of the CLA blocks generates a carry

(f) **(Extra credit problem: 3 points)**: You are required to implement a logic function $f = z + xy\bar{z}$ in its **minimal** form. You can use any combination of 2-input logic gates. What is the **minimum** number of logic gates required to implement f?

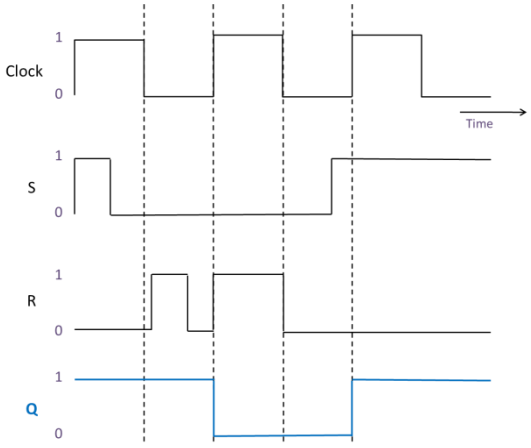
i. 1 gate

ii. 4 gates

iii. 2 gates

Problem No. 2 (13 points)

(a) **(5 points)** The input waveforms for a **gated SR latch** are shown in the following figure. Plot the Q output as a function of time.



(b) **(8 points)** You are required to design a logic function f for comparing two 3-bit numbers A ($a_2a_1a_0$) and B($b_2b_1b_0$), as follows:

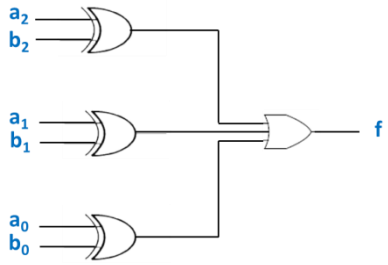
i. When A is equal to B, the output f should be equal to "0"

ii. When A is not equal to B, the output f should be equal to "1".

Derive the logic expression for f and draw the logic circuit implementation.

Solution:

$$f = (a_2 \oplus b_2) + (a_1 \oplus b_1) + (a_0 \oplus b_0)$$



Problem No. 3 (15 points)

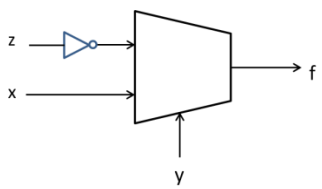
(a) **(7 points)** Implement the following 3-input function by using a 2-input multiplexer and any number of NOT gates.

$$f = \bar{x}\,\bar{y}\,\bar{z} + x\,\bar{y}\,\bar{z} + x\,y\,\bar{z} + x\,y\,z$$

Solution:

$$f = \bar{x}\,\bar{y}\,\bar{z} + x\,\bar{y}\,\bar{z} + x\,y\,\bar{z} + x\,y\,z$$
$$= \bar{y}\,\bar{z}\,(\bar{x} + x) + x\,y\,(\bar{z} + z)$$
$$= \bar{y}\,\bar{z} + x\,y$$

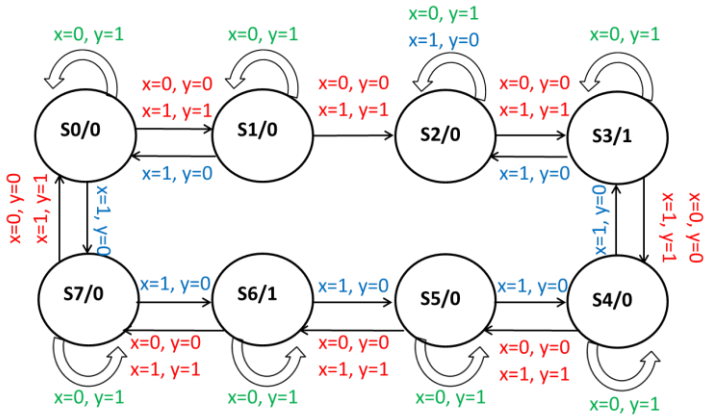
The above equation leads to the following 2-input multiplexer based implementation:



(b) **(8 points)** You are required to design a 3-bit synchronous counter by using a finite state machine. The counter has two external input signals *x* and *y*, which dictate the operation of the counter as follows: (i) If the two inputs have identical values, the counter counts *up*, (ii) If x = 0, y = 1, the counter holds its value, (ii) If x = 1, y = 0, the counter keeps counting *down* until it reaches "2" and then stops there. The counter also has an output signal *z*, which is equal to 1 only if the present value of the counter is a non-zero multiple of 3.

Draw the state diagram for this state machine.

Solution:



Problem No. 4 (20 points)

(a) **(5 points)** A 4-bit carry-lookahead adder (CLA) is used to add two numbers $X(x_3x_2x_1x_0)$ = 0110 and $Y(Y_3Y_2Y_1Y_0)$ = 1001 with an external carry-in c_0 = 1. Use the CLA carry-out equation to compute the values of following outputs: (i) carry-out c_3, (ii) sum s_3.

Solution:

X = 0110, Y = 1001, c_0 = 1

$G_0 = x_0$ AND y_0 = 0 AND 1 = 0, $G_1 = x_1$ AND y_1 = 1 AND 0 = 0

$G_2 = x_2$ AND y_2 = 1 AND 0 = 0, $G_3 = x_3$ AND y_3 = 0 AND 1 = 0

$P_0 = x_0$ OR y_0 = 0 OR 1 = 1, $P_1 = x_1$ OR y_1 = 1 OR 0 = 1

$P_2 = x_2$ OR y_2 = 1 OR 0 = 1, $P_3 = x_3$ OR y_3 = 0 OR 1 = 1

$c_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0c_0$

= 0 OR (1 AND 0) OR (1 AND 1 AND 0) OR (1 AND 1 AND 1 AND 1)

= 0 OR 0 OR 0 OR 1 = 1

$$s_3 = x_3 \oplus y_3 \oplus c_3 = 0 \oplus 1 \oplus 1 = 0$$

(b) **(7 points)** A multiplier circuit employing Booth algorithm is used to multiply two numbers *A* and *B*. Assume that the circuit can choose either of the two operands (A or B) as "multiplicand" and "multiplier", respectively. The choice needs to be made in such a way that the multiplication operation results in the least number of non-zero summands.
If A = +6, B = -15, which of the two operands should be chosen as the "multiplier"? Why?

Solution:

A = +6 = $(00110)_2$
Booth recording for A = 0 +1 0 -1 0

B = -15 = $(10001)_2$
Booth recording for B = -1 0 0 +1 -1

Since the Booth recording for A has more 0s 3 as compared to 2 for B), using A as the "multiplier" will result in fewer summands. Therefore "A" should be chosen as the "multiplier".

(c) **(8 points)** Consider a program running on a processor P1 with a clock period of 0.5 nanoseconds. The instructions in the program can be categorized into two types:

(i) Type-1: Each of these instructions takes 3 cycles to execute on P1.
(ii) Type-2: Each of these instructions takes 1 cycle to execute on P1.

Assume that x% of the total instructions in the program are of Type-1, while the remaining instructions are of Type-2. Also assume that processor is required to have an instruction execution rate of at least 1 billion instructions per second. What is the maximum value of "x" (percentage of Type-1 instructions), which will allow the processor to meet the desired execution rate target? Show all the necessary calculations.

Solution:

CPI = (x * 3) + ((1 - x)*1) = 2x + 1
Clock Period = 0.5 ns = 0.5 * 10^9 s
Clock Rate = 1 / Clock Period = 2 GHz
Required Instruction Execution Rate = 10^9 instructions per second

Since: Execution time = (Number of Instructions * CPI) / Clock Rate
Therefore: Instruction Execution Rate = Number of Instructions/Execution Time = Clock Rate/CPI

In order to meet the desired execution rate target:
Clock Rate / CPI >= 10^9
2 * 10^9 / (2x + 1) >= 10^9
Solving for x yields: x <= 0.5

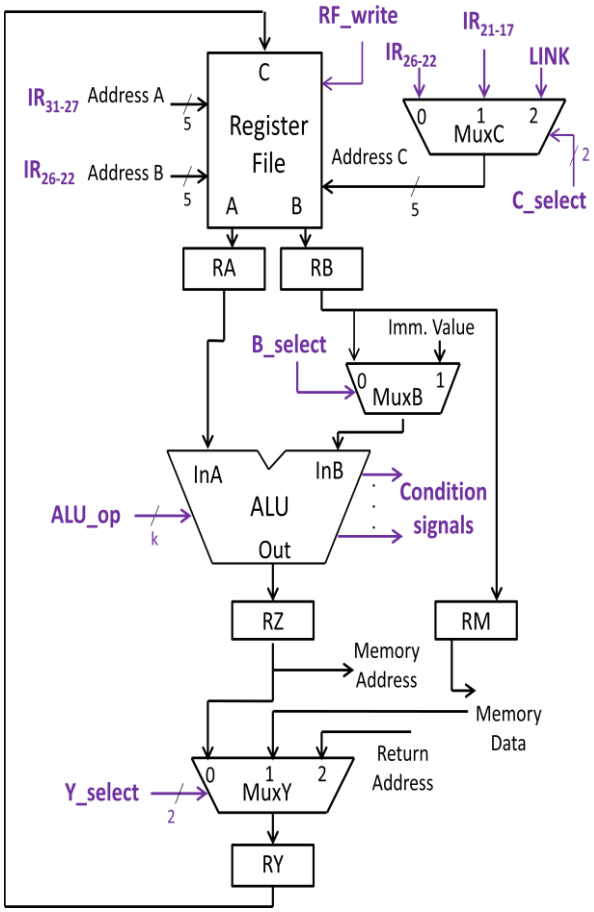
Therefore the percentage of Type-1 instructions in the program must be less than 50%.

Problem No. 5 (16 points)

The 5-stage RISC processor discussed in class is used to execute the following sequence of instructions, one after the other:

Instruction 1: STORE R1, 200(R2)
Instruction 2: LOAD R1, 100(R2)
Instruction 3: SUBTRACT R4, R1, R3

The processor data path with all the control signals is shown in the following figure:



(a) **(5 points)** Write down the register transfer notation (RTN) expression for each of the three instructions.

Solution:

Instruction 1: [R2] + 200 ← [R1]
Instruction 2: R1 ← [[R2] + 100]
Instruction 3: R4 ← [R1] – [R3]

(b) **(5 points)** For **each** of the three instructions in the program, answer the following questions:

(i) Which of the two inputs ("0" or "1") is selected by MuxB?

Solution:

MuxB selects inputs **1**, **1**, and **0** for instructions 1, 2, and 3, respectively.

(ii) Which of the three inputs ("0", "1" or "2") is selected by MuxY?

Solution:

For instruction 1, the input selection for *MuxB* does not matter (any answer is correct), because no writeback is carried out in stage-5.
For instructions 2 and 3, *MuxB* selects inputs **1** and **0**, respectively.

(c) **(6 points)** The following table shows the initial values of the relevant registers and memory locations. Complete the table by filling in the column corresponding to the final values of each register and memory location, after the above code sequence has been executed by the processor.

Register/Memory Location	Initial Contents	Final Contents
R1	46	**64**
R2	30000	**30000**
R3	30	**30**
R4	10	**34**
Memory Address 30100	64	**64**
Memory address 30200	20	**46**