

<p style="text-align: center;">ECE2030A Fall 2008 Introduction to Computer Engineering Exam #2</p>

50 min in-class exam.
Close books, close notes.
NO calculator.

This exam is given under the Georgia Tech Honor Code System. You must observe and sign the Honor Pledge: "I have neither given nor received aid on this exam." Your print name and signature below signifies your compliance with this honor code.

Name (Print): _____ **SOLUTIONS** _____

Signature: _____

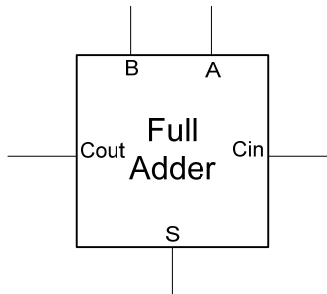
1. _____ (30 pts)
2. _____ (20 pts)
3. _____ (30 pts)
4. _____ (20 pts)

Total (100 pts) _____

1. (30%) **Adder Design.**

1.1. (5%) Derive the truth table for a "Full Adder".

Let the basic full adder be represented as:



The Truth Table will be:

Cin	A	B	S(um)	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

- 1.2. (5%) Simplify the outputs of your Full Adder using Karnaugh maps and derive their Boolean expressions. (Do not convert your equations using XOR operators. AND, OR and Inversion will suffice as you need them for problem 1.3.)

		AB			
		00	01	11	10
Cin	0	0	1	0	1
	1	1	0	1	0

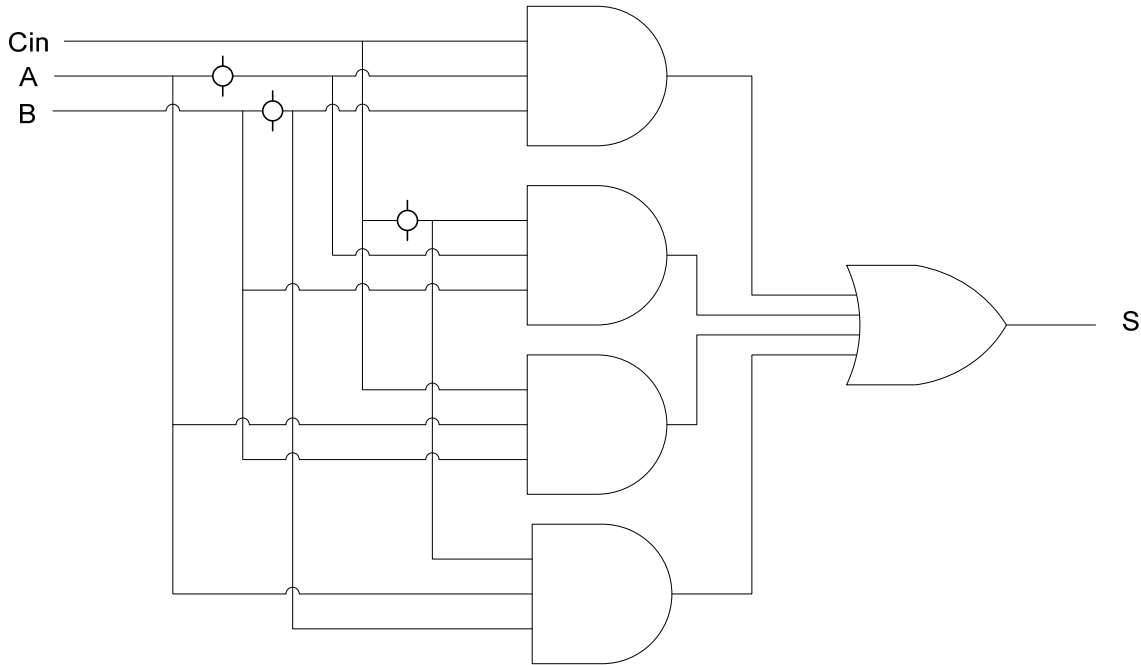
$$S = \overline{Cin}\overline{A}\overline{B} + \overline{Cin}A\overline{B} + \overline{Cin}A\overline{B} + \overline{Cin}A\overline{B}$$

		AB			
		00	01	11	10
Cin	0	0	0	1	0
	1	0	1	1	1

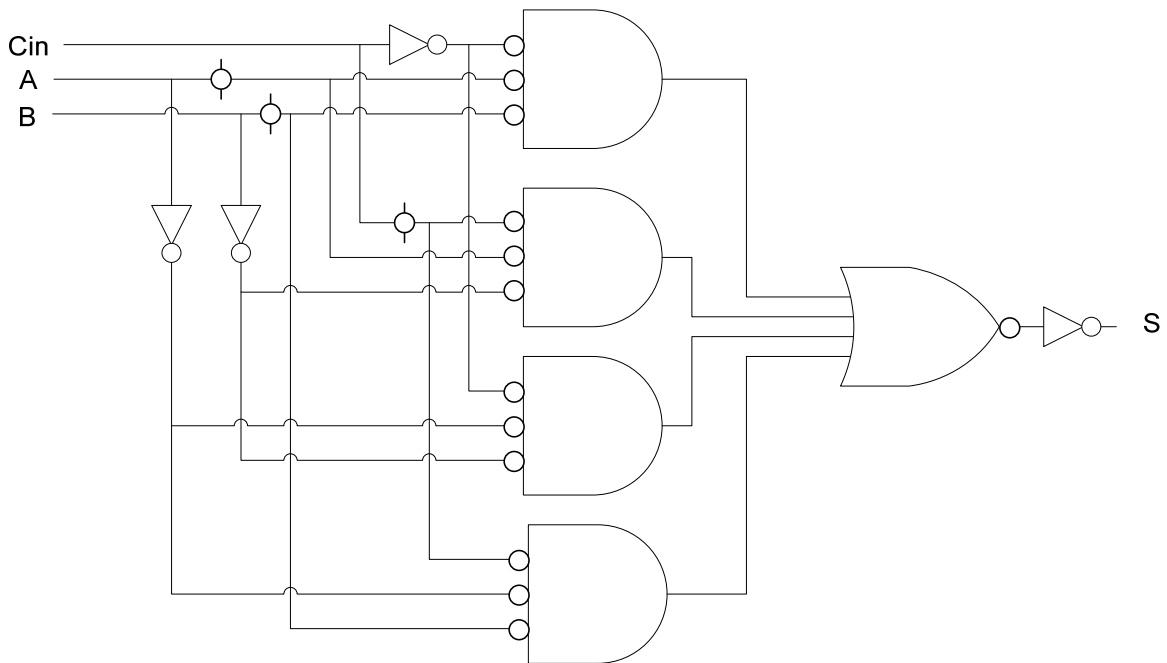
$$Cout = CinB + CinA + AB$$

- 1.3. (10%) Implement your outputs using NOR gates only based on Mixed Logic method. Please try to minimize the number of gates used in your logic circuits whenever you can share gates. You can assume your NOR gates can have any arbitrary number of inputs. You will need this result for problem 1.4.

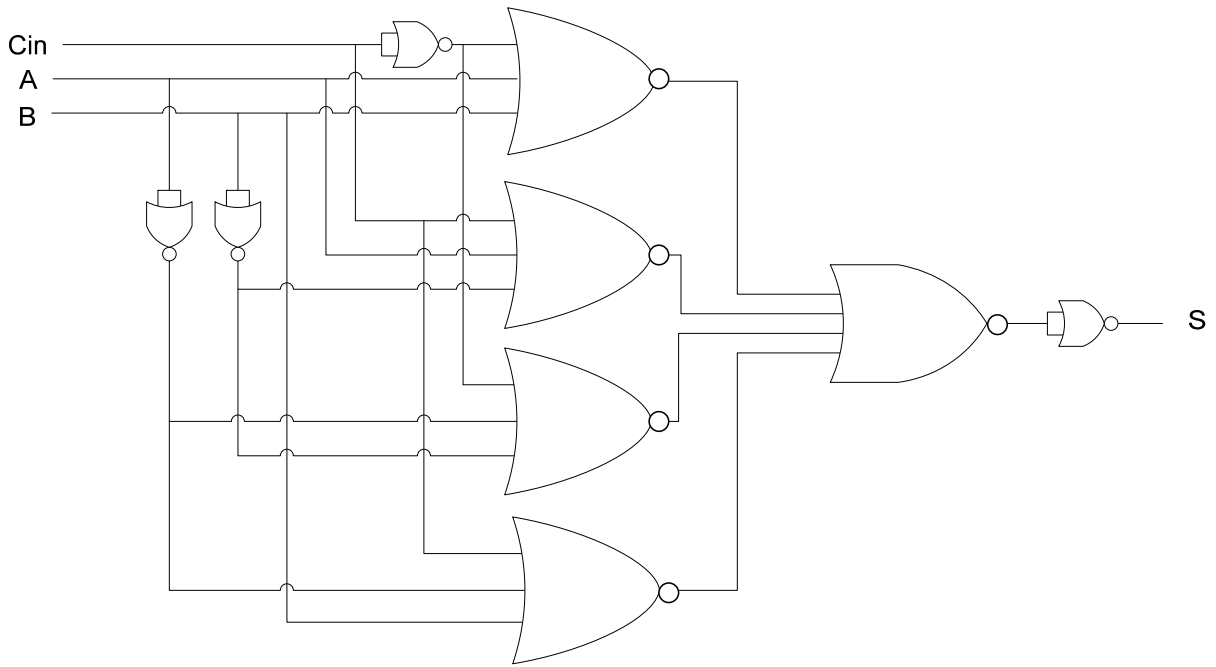
Starting with the sum logic:



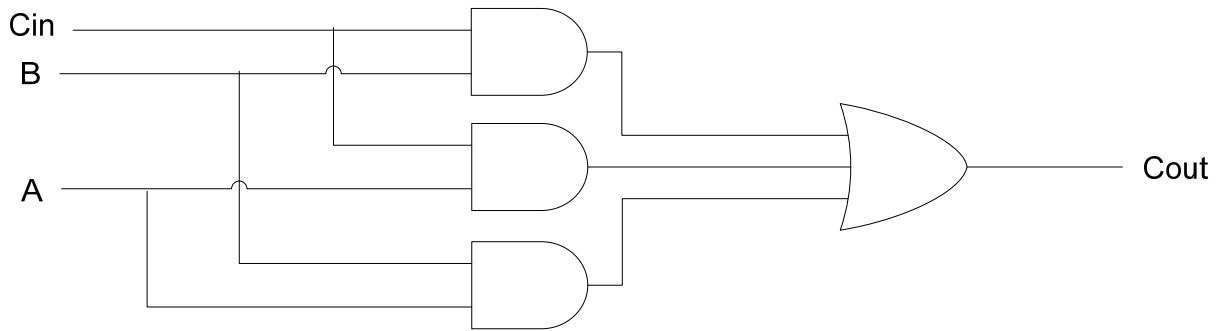
Convert to NOR gates and balance with inverters:



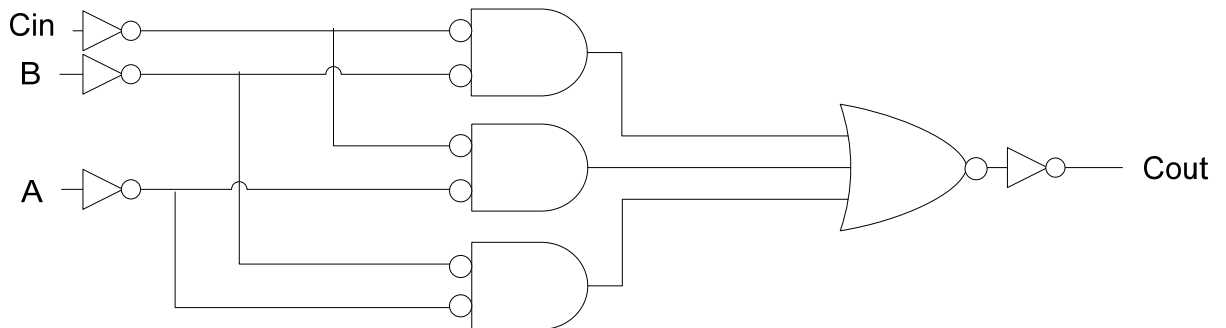
Final version:



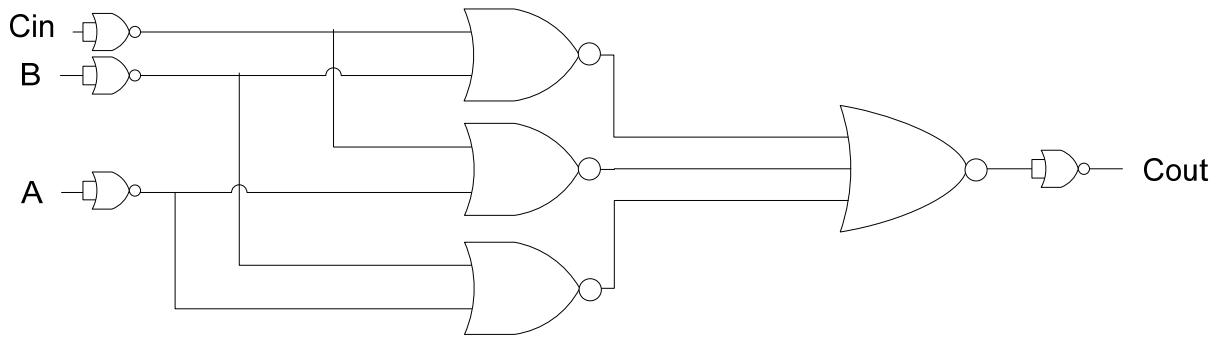
For the Cout:



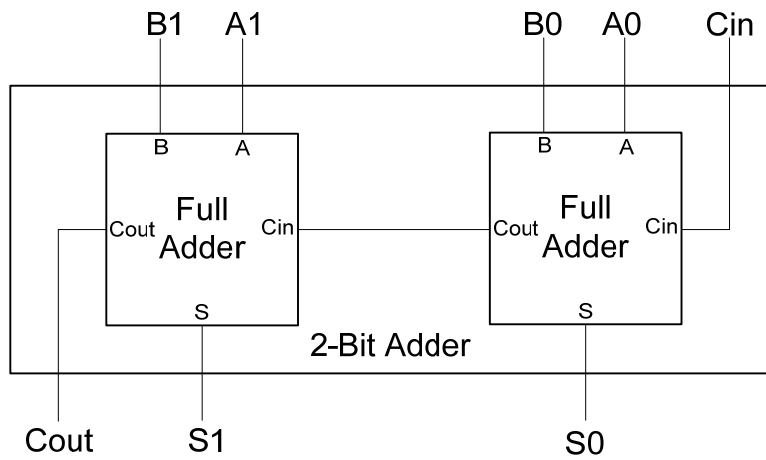
Convert to NOR gates and balance with inverters:



Final Cout circuit:

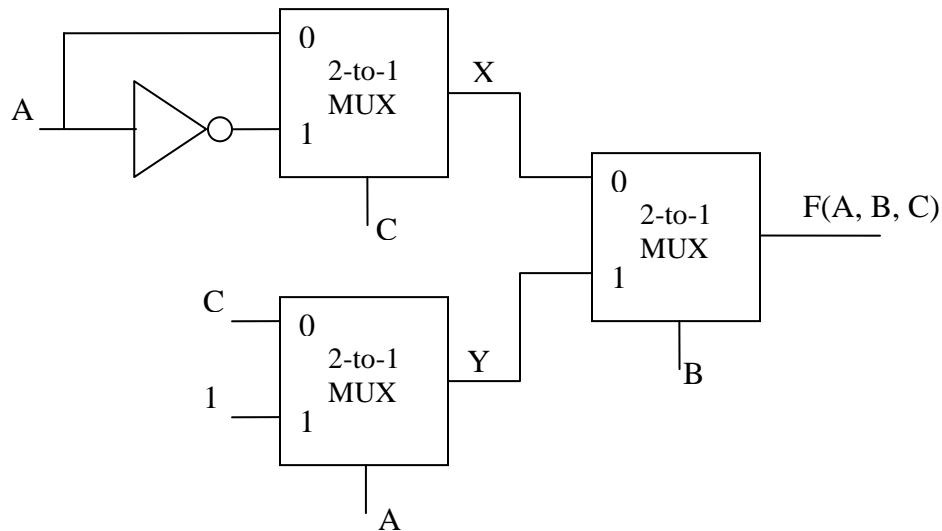


1.4. (10%) Design a 2-bit adder using your Full Adder, what is the critical path (in number of NOR gates based on 1.3) of your 2-bit adder? (You don't need to worry about the overflow detection logic.)



Delay = 4 gates per basic adder. Therefore, for a 2-bit adder, there is a total delay of 8 gates.

2. (20%) **Multiplexor.** Given the following logic circuits implemented with three 2-to-1 multiplexors and inputs A, B, and C, please reverse-engineer the $F(A, B, C)$ back to its “Canonical SOP form” in minterms.



$$X = \bar{C} \cdot A + C \cdot \bar{A}$$

$$\begin{aligned} Y &= \bar{A} \cdot C + A \cdot 1 \\ &= \bar{A} \cdot C + A \\ &= C + A \end{aligned}$$

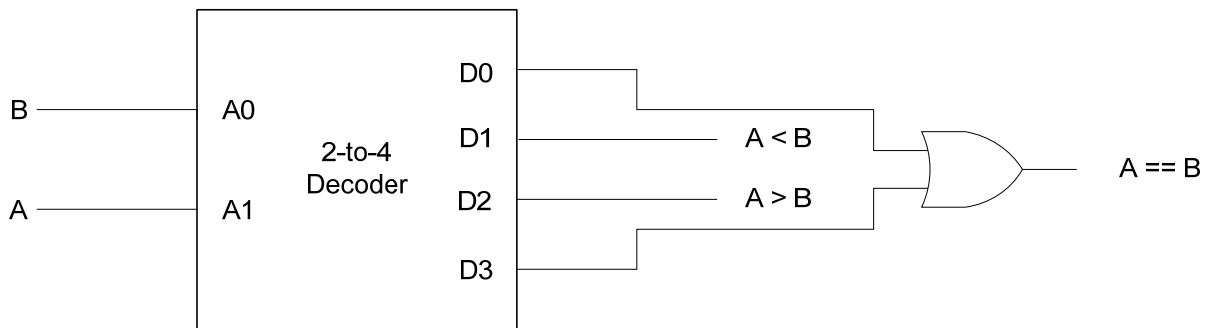
$$\begin{aligned} F &= \bar{B} \cdot X + B \cdot Y \\ &= \bar{B} \cdot (\bar{C} \cdot A + C \cdot \bar{A}) + B \cdot (C + A) \\ &= A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + B \cdot C + A \cdot B \\ &= A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + (A + \bar{A}) \cdot B \cdot C + A \cdot B \cdot (\bar{C} + C) \\ &= A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot B \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C \\ &= m(4) + m(1) + m(7) + m(3) + m(7) + m(6) \\ &= \sum m(1,3,4,6,7) \end{aligned}$$

3. (30%) **Magnitude comparators.**

3.1. (10%) Construct the Truth Table for a one-bit unsigned magnitude comparator. There are two inputs: A and B with three outputs: $A > B$, $A < B$, and $A == B$.

A	B	$A > B$	$A < B$	$A == B$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

3.2. (10%) Use “one 2-to-4 decoder” and some “logic gates” (i.e., OR, AND, and such) to implement this one-bit comparator.

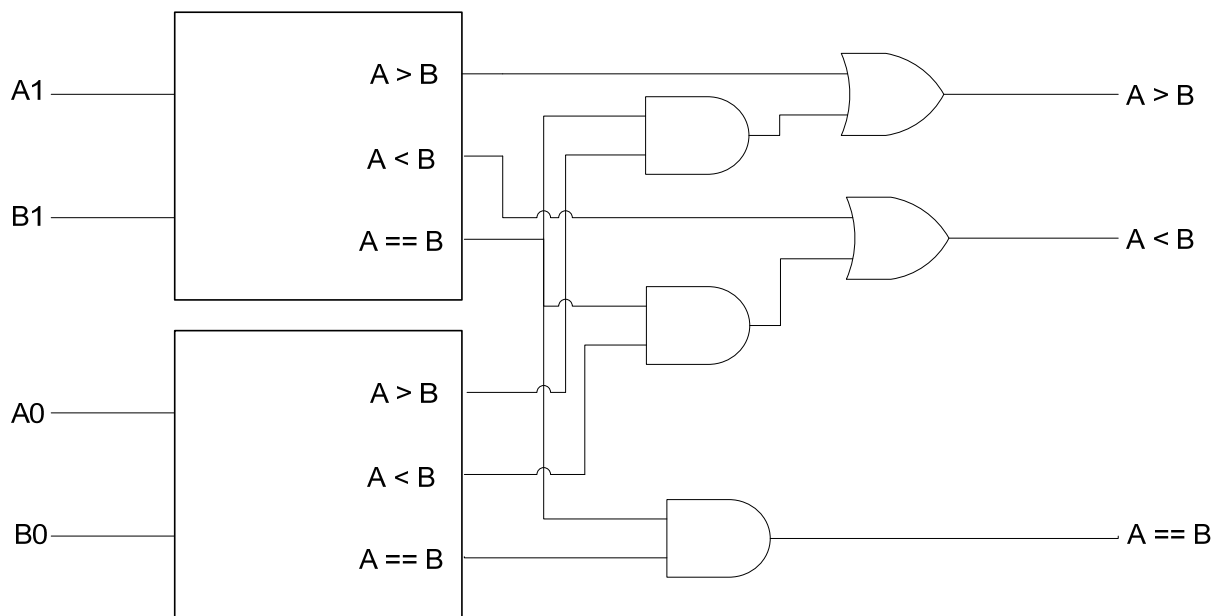


3.3. (10%) Take the 1-bit comparator you design as a building block (i.e., draw it as a box with 2 inputs and 3 outputs), now use two of these and other logic gates to design a 2-bit unsigned magnitude comparator. The new comparator consists of 4 inputs: A[1:0] and B[1:0], and 3 outputs: A>B, A<B, and A==B. Please show your logic circuits schematic and label the inputs and outputs properly.

$$A > B = (A1 > B1) + (A1 == B1) \cdot (A0 > B0)$$

$$A < B = (A1 < B1) + (A1 == B1) \cdot (A0 < B0)$$

$$A == B = (A1 == B1) \cdot (A0 == B0)$$



4. (20%) **Combinational Logic.** Given a fixed function $F(A) = 5 \cdot A$ where input A is an 8-bit signed number. Please demonstrate a logic circuit using only shifter(s) and adder(s) to carry out function F . You only need to draw the shifter(s) and adder(s) using block diagrams labeled with their inputs and outputs, i.e., no need to draw the logic gates inside the block. You need to specify what type of shifter you use inside the block.

$$F = 4A + A$$

$$F = A \ll 2 + A$$

