This exam will be conducted according to the Georgia Tech Honor Code. I pledge to neither give nor receive unauthorized assistance on this exam and to abide by all provisions of the Honor Code.

Signed (With your Full Legal Name):

_____

Instructions: Instructions: This is a closed book, closed note exam. Calculators are not permitted. If you have a question, raise your hand and I will come to you. Please work the exam in pencil and do not separate the pages of the exam. For maximum credit, show your work.

Good Luck!

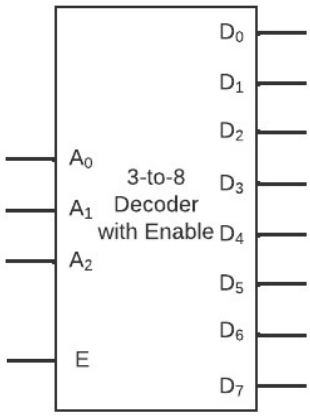
Your Name (please print): _____

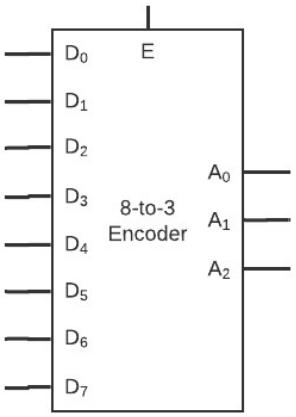
For maximum (and partial) credit, show your work.

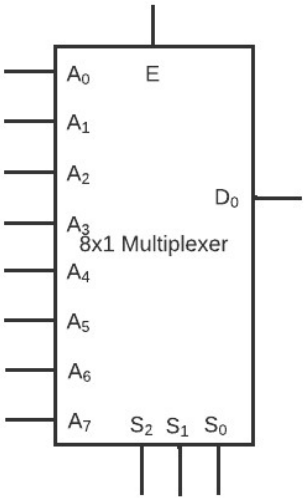
Good Luck!

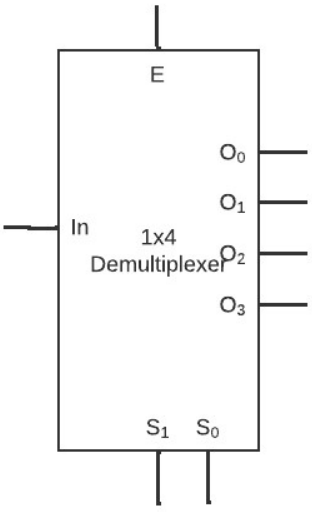
1	2	3	4	total
20	20	40	20	100

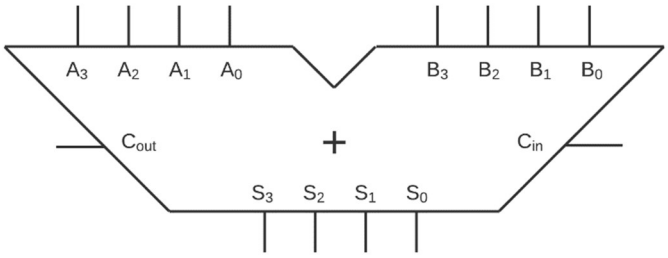
Combinational Building Blocks

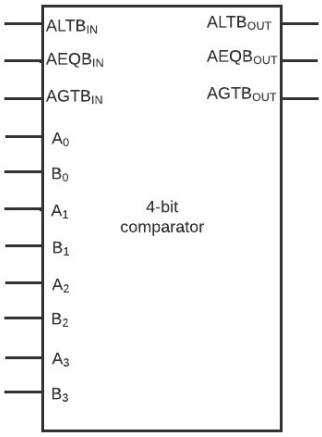












Problem 1 (5 parts, 20 points) Fundamentals

Part A (8 points) Convert the following notations using the minimum required digits.

Binary Notation	Numbering System	Decimal
1101010	Signed-magnitude	-42
10001	1's complement	**-14**
01101	2's complement	13
101110	2's complement	**-18**

Part B (6 points) For the binary representations below, determine the range (minimum to maximum values). **Express all answers in decimal notation.**

Representation	Minimum Value	Maximum Value
Unsigned integer (6 bits)	**0**	**63**
1's complement (5 bits)	**-15**	**15**
2's complement (3 bits)	**-4**	**3**

Part C (2 points) In combinational logic circuits, the outputs are determined by both inputs and the current state (memory) of the system.

a) True

b) False

Part D (2 points) In pass (or transmission) gates, current follows a different path based on whether the input is high or low.

a) True

b) False

Part E (2 points) Which one of the following building blocks cannot be used to cascade (or tie multiple building blocks together to build multi-bit versions of the same function)?

a) Decoder

b) Priority Encoder

c) Demultiplexer

d) Half-adder

Problem 2 (2 parts, 20 points) Computation

Do **NOT** change the order of the operands, show all work, and CIRCLE the final answer.

PART A (10 points) Show how 31 + 9 is done in 6-bit 2's complement. Is your answer correct? EXPLAIN why or why not? Did an overflow occur?

		11111
31	=>	011111
+ 9		+ 001001
40		101000

Overflow? **YES**

The answer is incorrect because an overflow occurred. The sign bit was pushed into the 6th bit so that the answer is saved as -24 instead of +40.

PART B (10 points) Show how 15 – 28 is done in 6-bit 2's complement. Is your answer correct? EXPLAIN why or why not? Did an overflow occur?

		11
15	=>	001111
+ -28		+ 100100
-13		110011

Overflow? **NO**

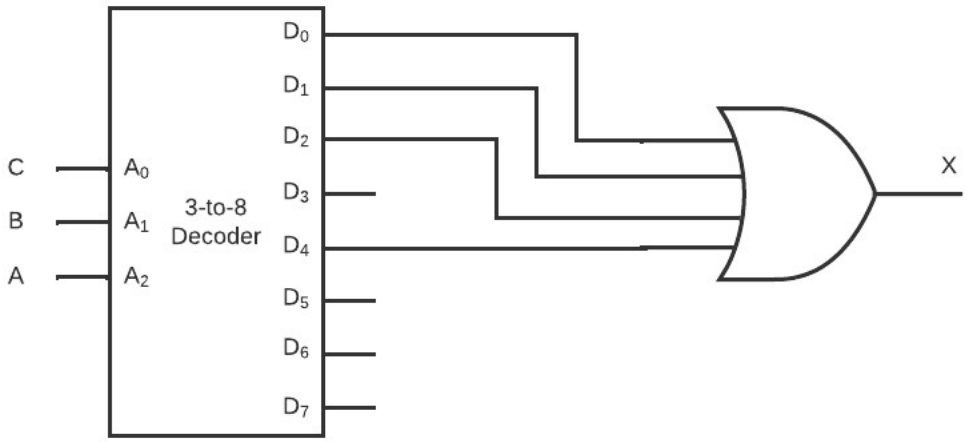
The answer is correct because the final result is a negative number (no overflow has occurred) and is equal to -13 (110011 -> -001101->-13)

Problem 3 (4 parts, 40 points) Decoder / Multiplexer Design

Solve the following questions given the truth table below.

A	B	C	X	Y
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	0
1	1	1	0	1

PART A (10 points) Design output **X** using a single 3-to-8 decoder (without an enable) and the minimum number of gates. You may use any logic gate with no limit on the number of gate inputs. Do not simplify.



PART B (10 points) How many transistors would you need to implement the circuit above? Assume that the decoder is implemented using canonical sum-of-products equations (no minimization), complemented inputs are provided inside the decoder, and logic gates have no limit on the number of gate inputs.

Implementing decoders using canonical SOP equations requires eight 3-input AND gates. If you assumed that complemented inputs are provided, you only need 6 transistors per 3-input AND gate. I also accepted you using 8 transistors per 3-input AND gate (6 transistors for a NAND and 2 transistors for a NOT gate).

The total number of transistors needs to include 10 transistors to account for the 4-input OR gate.

$$\#\,\text{transistors, decoder} = 6\text{T} \times 8\,\text{AND}_{3-\text{inp}} = \mathbf{48\ transistors} \text{ or } 8\text{T} \times 8\,\text{AND}_{3-\text{input}} = \mathbf{64\ transistors}$$

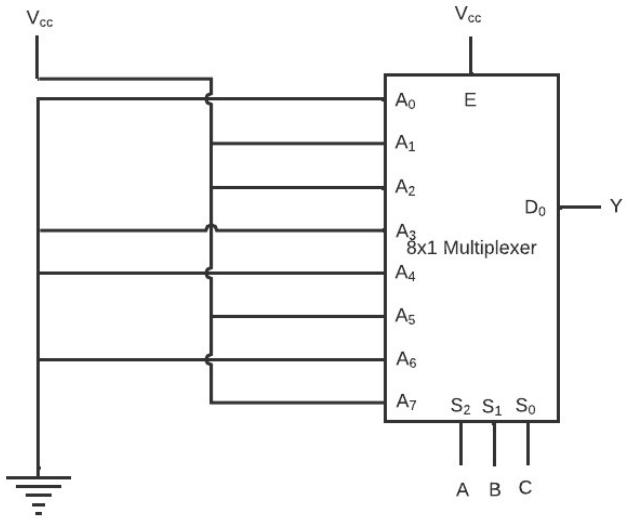
$$\#\,\text{transistors, total} = 48\ \text{transistors} + 10\text{T} \times 1\,\text{OR}_{4-\text{inp}} = \mathbf{58\ transistors} \text{ or } \mathbf{74\ transistors}$$

Problem 3 Continued (4 parts, 40 points) Decoder / Multiplexer Design

Solve the following questions given the truth table below.

A	B	C	X	Y
0	0	0	1	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	0
1	1	1	0	1

PART C (10 points) Design output **Y** using a single 8-to-1 multiplexer (without an enable) and the minimum number of gates. You may use any logic gate with no limit on the number of gate inputs. Do not simplify.



PART D (10 points) How many transistors would you need to implement the circuit above? Assume that the multiplexer is implemented using pass gates and complemented inputs are provided.

Multiplexers are built using banks of pass gates to downselect to a single output signal. The first bank will need 8 pass gates, the second bank will need 4 pass gates, and the third and final bank needs 2 pass gates.

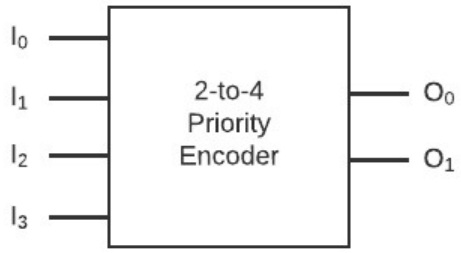
The total number of transistors is the same as the number needed for the multiplexer design.

$$\#\ \text{transistors, decoder} = 2\text{T} \times 14\ \text{Pass} = 28\ \text{transistors}$$

$$\#\ \text{transistors, total} = 28\ \text{transistors}$$

Problem 4 (3 parts, 20 points) Encoders, Adders, and Comparators

PART A (4 points) Consider a priority encoder with the following behavior:

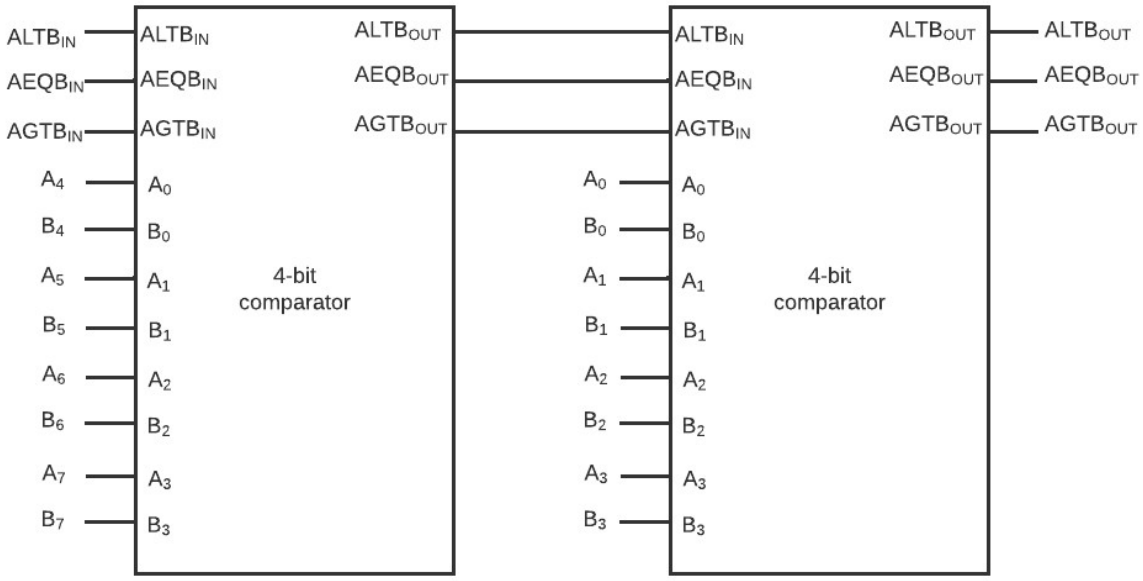


I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0	0	X	X
X	X	X	1	1	0	0
0	0	1	0	1	0	1
X	1	X	0	1	1	0
1	0	X	0	1	1	1

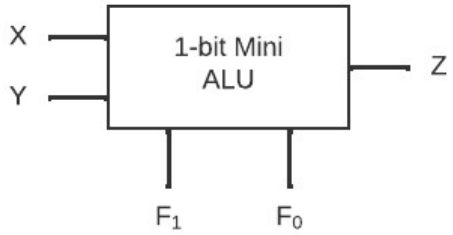
What input priority has been assigned to the four inputs (I_3, I_2, I_1, and I_0)?

I_0 > I_2 > I_3 > I_1

PART B (6 points) Build an 8-bit magnitude comparator using the two 4-bit building blocks below. You should label inputs ($A_{0\text{-}7}$, $B_{0\text{-}7}$, $AEQB_{IN}$, $ALTB_{IN}$, and $AGTB_{IN}$) and outputs ($AEQB_{OUT}$, $ALTB_{OUT}$, and $AGTB_{OUT}$) as well as show connections between the chips.



PART C (10 points) Use at least two types of building blocks (e.g., decoders, encoders, multiplexers, adders, comparators) and logic gates to design the 1-bit Mini Arithmetic Logic Unit (ALU) described below.



X	Y	F_1	F_0	Z
X	Y	0	0	X + Y (OR)
X	Y	0	1	X · Y (AND)
X	Y	1	0	X = Y (Equal)
X	Y	1	1	X != Y (Not Equal)