

# EE 109 Final “Don’t Make the Same Mistake”

No calculators are allowed. Show all your work to get full credit.

**Note: The following are all INCORRECT answers meant to highlight common misunderstandings! Try to figure out where their answer came from and the misconception that caused it. Correct answers are intentionally not given (and will not be) so that you think about how to solve the problem correctly.**

## 1. Number Systems & Arithmetic

- a. Perform the indicated arithmetic operations given the representation systems below to find the **8-bit (not 9-bit)** results. (Do not use the borrow method for subtraction, use the 2’s complement method of subtraction.) Show intermediate steps. **Finally, state whether overflow has occurred and briefly justify why it has or has not occurred.**

a.) 2’s comp. system

$$\begin{array}{r} 01010111_2 \\ - 10000000_2 \\ \hline \end{array}$$

$$\begin{aligned} 2\text{'s comp. of } 10000000 &= \\ 01111111 + 1 &= 10000000 \end{aligned}$$

$$\begin{array}{r} \text{Add} \\ 01010111 \\ + 10000000 \\ \hline 11010111 \end{array} \quad \text{Correct Sum}$$

Overflow: Y / **(N)** Why?

P + N never gives  
overflow

OR

Last 2 carries are equal

b.) Unsigned binary

$$\begin{array}{r} 11010011_2 \\ + 01010010_2 \\ \hline \end{array}$$

Try this on your own

Overflow: Y / N Why?

Overflow  
Answer is wrong  
because they  
took 2’s comp.  
separately and  
then added. The  
test doesn’t work  
for that case.

2. Perform the following number system conversions (in any order)

$$(6D.4)_{16} = ?_8 = ?_2 = ?_{10}$$

$$(6D.4)_{16} = 0110\ 1011 . 0100_2$$

3. Given the bit string (set of 1's and 0's): 10101011 and the following representation systems, find the string's equivalent decimal value in each system.

System	Decimal value (show work for partial credit)
2's complement	$10101011 = -256 + 64 + 32 + 4 + 1 = -155_{10}$
Signed Magnitude	$10101011 = -32 + 8 + 2 + 1 = -32 + 11 = -21$

4. Find the 2's complement system representation of the following numbers (using the given number of bits).

- a.)  $(-91)_{10}$  given an 8-bit 2's complement system.

a.  $11011011_2$

- b.)  $(-91)_{10}$  given a 16-bit 2's complement system.

a.  $0000000010100101_2$

5. (17 pts.) Bob is throwing a party for his friends: Fred, Greg, Jill and Peter. He can only have the party if at least 2 of his friends attend. He invites them and his friends first check if they are free (able to come). However, some of his friends are also particular about who they associate with. Their conditions for attending are listed below.

- Fred will only attend if at least one boy **besides** Bob is there.
- Greg will only attend if Jill does.
- Peter and Jill love parties and will attend w/o restriction if they are free.

For the different combinations of Bob's friends (1 = free, 0 = busy), indicate which combinations will allow the party to occur (at least 2 attendees even with the restrictions a-c. Then find the minimal **SOP** expression for the party to occur.

F	G	J	P	PARTY
0	0	0	0	0
0	0	0	1	0
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	

		F,G			
		00	01	11	10
J,P	00	0	0	0	1
	01	0	0	1	1
	11	0	1	0	1
	10	0	1	1	1

- Wrong 1 / 0 locations.
- Wrong grouping even with the given 1/0 locations

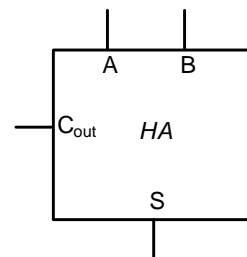
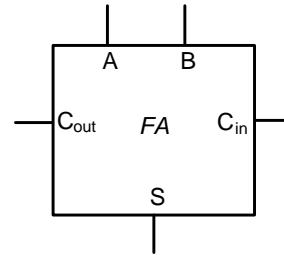
6. Full and Half Adders

(

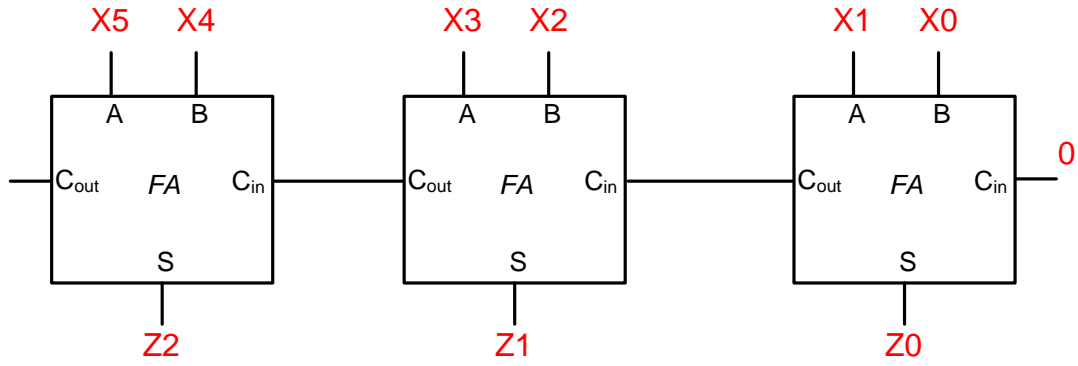
15 pts.)

Tim needs a circuit to count how many 1's are present in a 6-bit number,  $X[5:0]$ . This can be accomplished by adding the six bits of  $X$ . Produce an output binary number,  $Z$ , indicating how many 1's are in the number,  $X$ . (Ex.:  $X = 100110 \Rightarrow$  three 1's  $\Rightarrow Z = 3_{10} = ?_2$ )

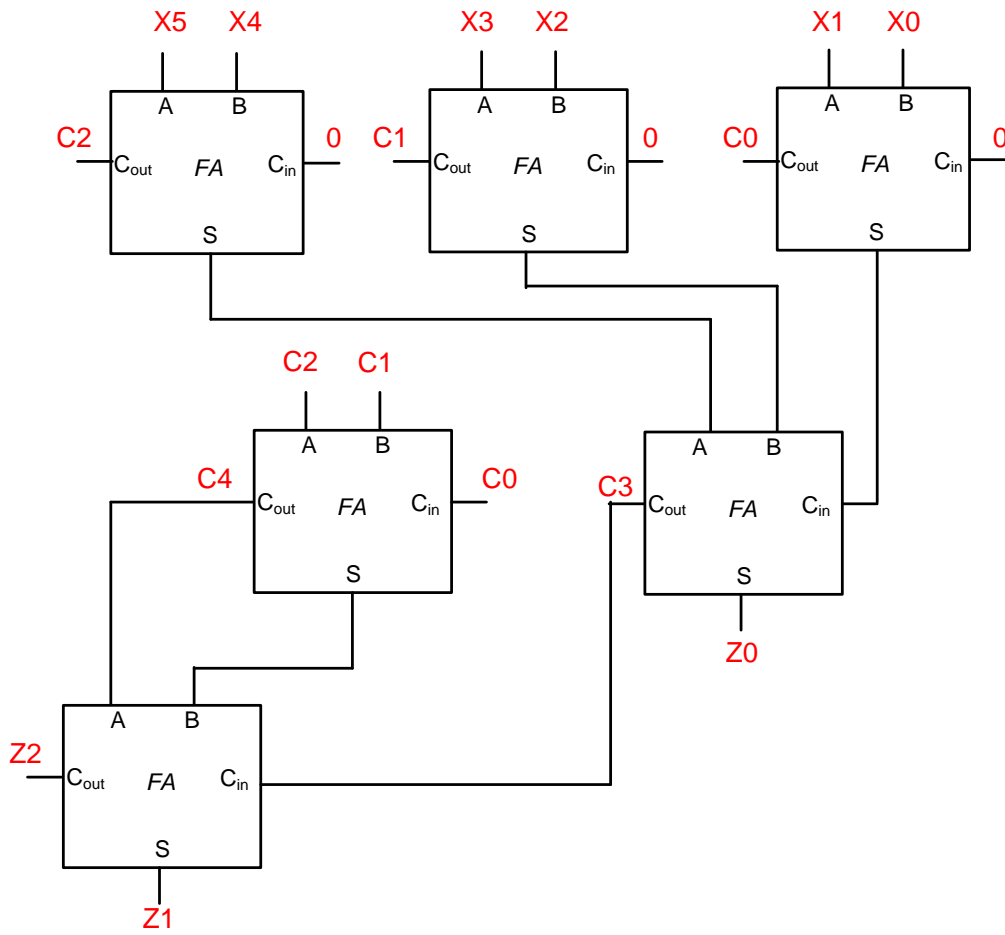
- a. How many bits of output are needed for  $Z$ ? \_\_\_\_\_
- b. Use the **minimal number** of full adders and half adders to implement your design, though your selection of either full or half adders will not be graded (i.e. If only a HA is needed and you use a FA, no points will be deducted.) **We recommend writing out what column/place value each bit and sum/carry belong to. This will ensure correct connections.** A full and half adder have been drawn for you that you can feel free to use. You will need to add more...



1.)



2.)

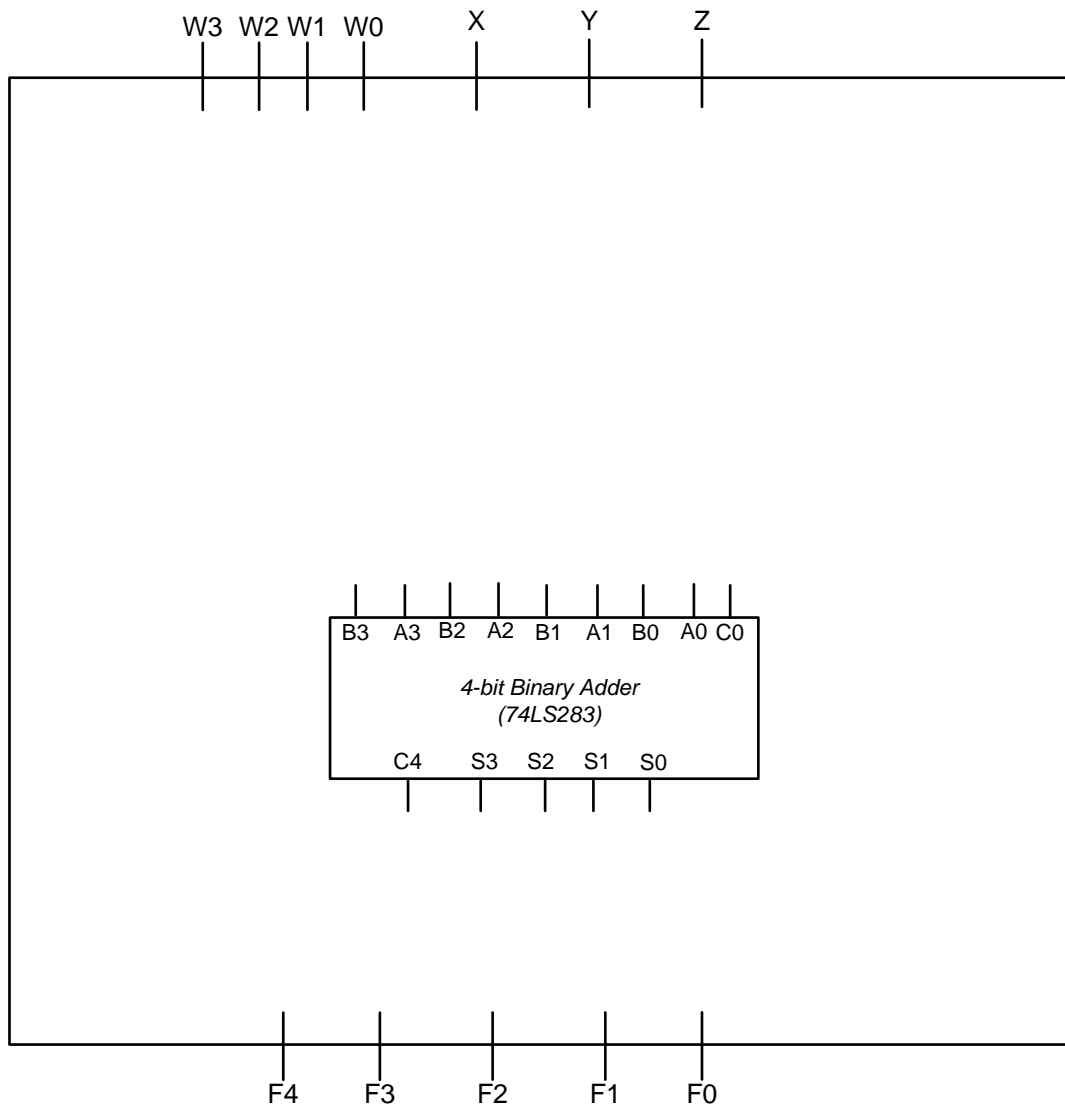
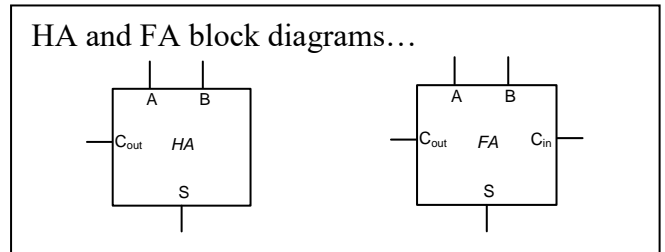


7. Adders (10 pts.)

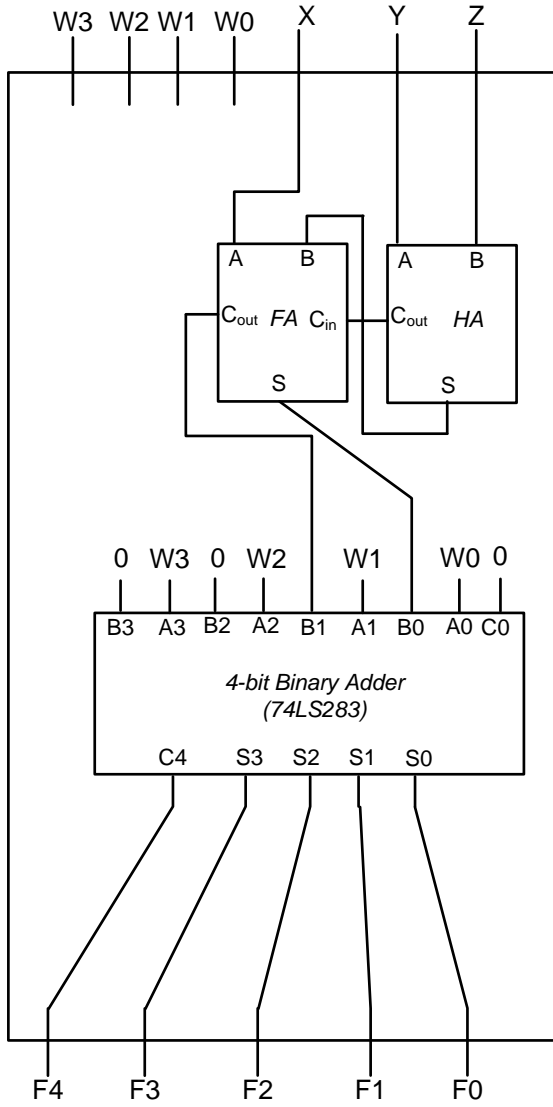
You are given a 4-bit unsigned number,  $W[3:0]$ , representing how many people have voted 'yes' on a proposition. However, three people vote late. Their vote is represented by three single-bit values:  $X$ ,  $Y$ , and  $Z$  ('1' = yes / '0' = no). Design a circuit to produce the total sum of yes votes (i.e. design a circuit that adds three one-bit numbers,  $X$ ,  $Y$ , and  $Z$  to a 4-bit number,  $W[3:0]$ ). The output should be a 5-bit number  $F[4:0] = W[3:0] + X + Y + Z$ .

To implement this circuit you will need a 4-bit 74LS283 adder drawn below. You may use additional half- and full-adders (shown below to jog your memory) as necessary. Implement your circuit with a **minimal** amount of logic beyond the '283 adder (i.e. use fewer building blocks and also try to substitute a single HA in place of an FA where possible).

*Hint: It may help to write out the column(s) of addition you need to perform.*



Wrong solution 1



Wrong Solution 2

