EE 109 Homework 5

Name: ____SOLUTIONS_____
Due: **Score: _____**

Show work to get full credit. Remember, use on only one side of the paper and staple them together. Only use a calculator to CHECK your work, not to DO your work.

1.) What are the corresponding decimal representations for the following binary strings: 10110110 , 11011011, if

a. The binary numbers are using *8-bit unsigned* format?
10110110 = 128+32+16+4+2=182
11011011 = 128+64+16+8+2+1=219

b. The binary numbers are using *8-bit 2's complement* format?
10110110 = -128+32+16+4+2 = -74
11011011 = -128+64+16+8+2+1 = -37

2.) For each of the following decimal numbers find the corresponding 8-bit representation using the indicated systems. Note: Some numbers may NOT be representable w/ 8-bits. Also find the minimum bits needed to represent the number in the 2's complement system.

	2's Complement	Minimum bits needed using 2's complement
-128	1000 0000	8 bits
+31	011111	6 bits
+59	00111011	7 bits
-16	11110000	5 bits

3.) Each C declaration of the variable x is initialized to a value in decimal. Show that value represented in hex using the appropriate size indicated by the variable type (e.g. char = 1-byte = 2 hex digits). Do not use a calculator.

a. int x = 256; 0x00000100
b. unsigned char = -128; 0x80
c. unsigned char x = 250; 0xfa
d. char x = -6; 0xfa
e. int x = -1; 0xFFFFFFFF
f. unsigned char x = 'a'; 0x61
g. unsigned char x = 97; 0x61

4. (32 pts.) Perform the following addition and subtraction problems assuming 2's complement numbers. State whether overflow does or does not occur for each problem. Justify your answer for why overflow does or does not occur. (You can easily check your work by converting to decimal.)

a.)
```
 1010 0110
+1101 1011
-------------
 1000 0001
```

n+n=n
$c_{out}=1, c_{in}=1$
No Overflow

b.)
```
 0010 0001
+0111 1001
-------------
 1001 1010
```

p+p=n
$c_{out}=0, c_{in}=1$
Overflow

c.)
```
 1000 0010
-1010 1111
     ||
 1000 0010
+0101 0000
+         1
-------------
 1101 0011
```

n+p=n
No Overflow

d.)
```
 0101 1001
-1010 0101
     ||
 0101 1001
+0101 1010
+         1
-------------
 1011 0100
```

p+p=n
Overflow

5.a

A 3-bit signed-magnitude number can represent a decimal from -3 to +3. Note that we regard "-0" as an illegal input. So we never use "100" as an input. When A is ranging from -3 to +3, Z = A + 2 can represent the decimal number from -1 to +5. We use 4-bit 2's complement number to represent it.

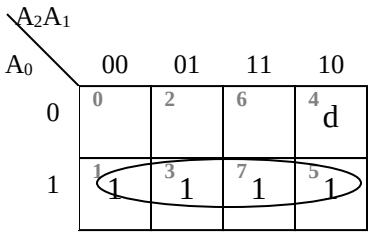
- The block diagram is below:

5.b Truth Table

A	A_2	A_1	A_0	Z_3	Z_2	Z_1	Z_0	Z
+0	0	0	0	0	0	1	0	+2
+1	0	0	1	0	0	1	1	+3
+2	0	1	0	0	1	0	0	+4
+3	0	1	1	0	1	0	1	+5
-0	1	0	0	d	d	d	d	d
-1	1	0	1	0	0	0	1	+1
-2	1	1	0	0	0	0	0	+0
-3	1	1	1	1	1	1	1	-1

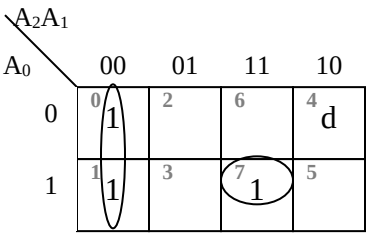
7.b

(1) K-Map of bit Z_0:



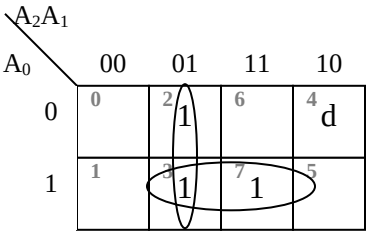
$Z_0 = A_0$

(2) K-Map of bit Z_1:



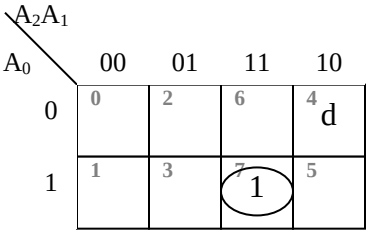
$$Z_1 = \overline{A_2}\,\overline{A_1} + A_2 A_1 A_0$$

(3) K-Map of bit Z_2:



$$Z_2 = \overline{A_2} A_1 + A_1 A_0$$

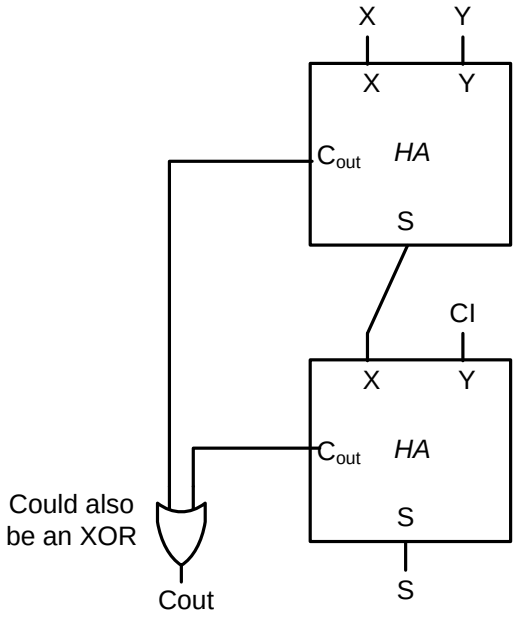
(4) K-Map of bit Z_3:



$$Z_3 = A_2 A_1 A_0$$

6. (10 pts.) Build an equivalent full-adder using two half-adders as building blocks along with additional gate(s) if needed.

Answer:



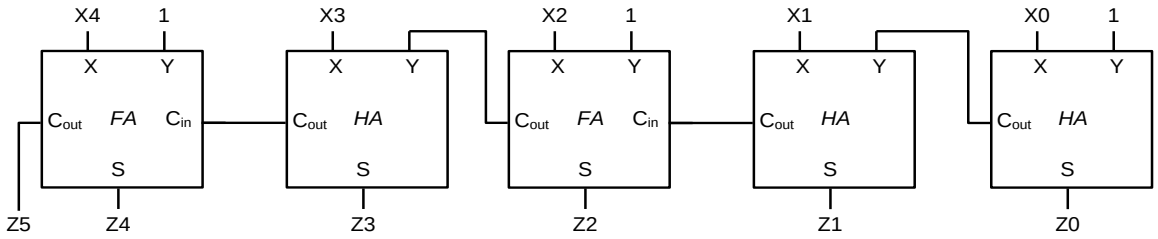
Note: We could use an XOR gate in place of the OR gate since both Cout's will never be 1 at the same time.

7. (10 pts.) Using half-adders and full-adders design a circuit that takes in a 5-bit unsigned number, X ($X_4..X_0$) and produces an output equal to X + 21_{10}. Your design should minimize the area required (i.e. use half-adders where possible.)

Answer:

$$Y = X + 21 = X_4\, X_3\, X_2\, X_1\, X_0 + 10101$$

The first bit addition does not have a carry in so we can use a half-adder ($X_0 + 1$). The second bit only adds the possible carry from previous step since the second bit in 21 is 0 (same for fourth bit). The third bit requires a full adder since ($X_2 + 1 +$ CarryFromBit2), same for fifth bit. In total we use 2 full-adders and 3 half-adders.



8. (20 pts.) Design a circuit that takes in four 4-bit unsigned numbers, A ($A_3..A_0$), B ($B_3..B_0$), C ($C_3..C_0$), and D ($D_3..D_0$) and produces the 6-bit unsigned sum of those numbers. You may use any number of 4-bit adder blocks (74LS283's), single-bit full adders or half adders that we have studied in class. You should organize your adder circuits to perform as many additions in parallel (at the same time) as possible.

ANSWER:

First we will perform two additions in parallel (A+B) and (C+D). Later we add the results of these two additions with another 4-bit adder. One of the carries is attached to this second level adder. The remaining carry from level I and the newly created carry from level II can be added by a half-adder. The result of these carries affects only the higher bits thus it does not affect the lower bits.