Problem 1:

Conversions:

a. Convert the following hexadecimal numbers to their representation in binary:
   i. FACD
   ii. A39B
   iii. 821F
   iv. 7065
   v. F1C4

b. Convert the following binary numbers to their representation in hexadecimal:
   i. 1010000000111011
   ii. 0101101110100101
   iii. 1011010011110100
   iv. 1111010100100111
   v. 0011010110011010

Solution:

a. From hexadecimal to binary:
   i. FACD
   1111101011001101
   ii. A39B
   1010001110011011
   iii. 821F
   1000001000011111
   iv. 7065
   0111000001100101
   v. F1C4
   1111000111000100

b. From binary to hexadecimal:
   i. 1010000000111011
   A03B
   ii. 0101101110100101
   5BA5
   iii. 1011010011110100
   B4F4
   iv. 1111010100100111
   F527
   v. 0011010110011010
   359A

Problem 2:

Given an architecture with a register size of 16 bits. State each of the numbers from exercise 1 as positive or negative if stored in registers of this size.

Solution:

a.

i. FACD (negative)
ii. A39B (negative)
iii. 821F (negative)
iv. 7065 (positive)
v. F1C4 (negative)

b.

i. 1010000000111011 (negative)
ii. 0101101110100101 (positive)
iii. 1011010011110100 (negative)
iv. 1111010100100111 (negative)
v. 0011010110011010 (positive)

Problem 3:

Convert the following negative numbers to their representation in binary in an 8-bit register using 2's complement:

i. -5
ii. -6
iii. -2
iv. -1 (Remember this representation!)
v. -8 (Could 4-bit register +8?)

Solution:

i. -5
00000101 (+5)
11111010 (1's complement)
11111011 (-5 in 2's complement)

ii. -6
00000110 (+6)
11111001 (1's complement)
11111010 (-6 in 2's complement)

iii. -2
00000010 (+2)
11111101 (1's complement)
11111110 (-2 in 2's complement)

iv. -1 (Remember this representation!)
00000001 (+1)
11111110 (1's complement)
11111111 (2's complement)

v. -8 (Could 4-bit register +8?)
00001000 (+8)
11110111 (1's complement)
11111000 (2's complement)

+8 can not be represented in a 4-bit register, however, -8 could be represented.

Problem 4:

Given an architecture with a register size of 4 bits. Perform the following mathematical operations in binary. State when an overflow occurs.

i. 2+5
ii. 5+3
iii. 3+(-2)
iv. 5+(-3)
v. 4+(-1)
vi. -3+(-2)
vii. -5+(-3)

Solution:

i. 2+5

```
  0010
+ 0101
----------
  0111        (no overflow)
```

ii. 5+3

```
  0101
+ 0011
------
  1000        (overflow occurred)
```

iii. 3+(-2)

```
  0011
+ 1110
------
  0001        (no overflow)
```

iv. 5+(-3)

```
  0101
+ 1101
------
  0010        (no overflow)
```

v. 4+(-1)

```
  0100
+ 1111
------
  0011        (no overflow)
```

vi. -3+(-2)

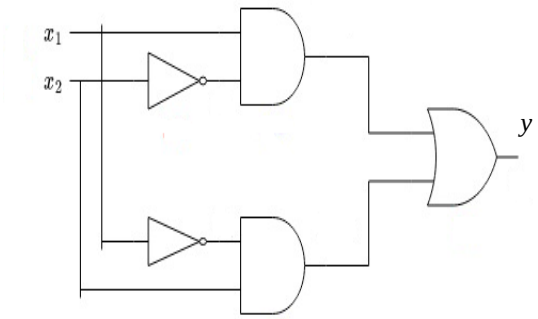
```
  1101
+ 1110
-------
  1011    -5 in 2's complement  (no overflow)
```

vii. -5+(-3)

```
  1011
+ 1101
  ---------
  1000    -8 in 2's complement  (no overflow)
```

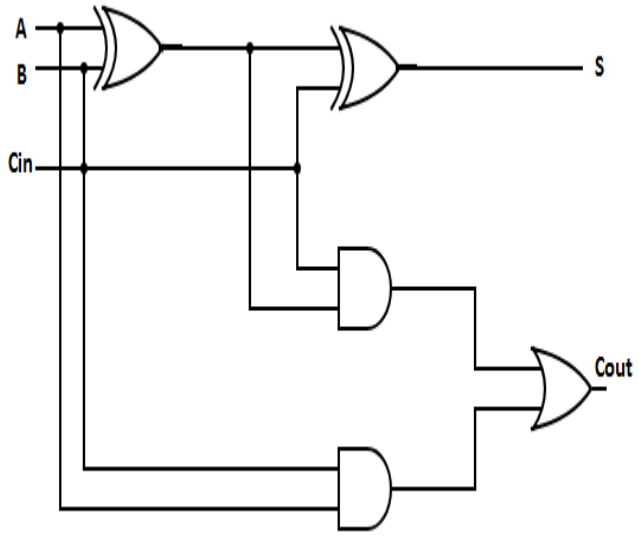
Problem 5:

a. Given the below simple digital circuit, fill the below table with the value of output y given the inputs x1, and x2 as stated in the table.



x1	x2	y
0	0	
0	1	
1	0	
1	1	

b. Given the below simple digital circuit, fill the below table with the values of the outputs S and Cout given the inputs A, B, and Cin as stated in the table.



A	B	Cin	S	Cout
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Solution:

a.

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

b.

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

Problem 6:

For each of the following multiplexers, state the number of selection lines:

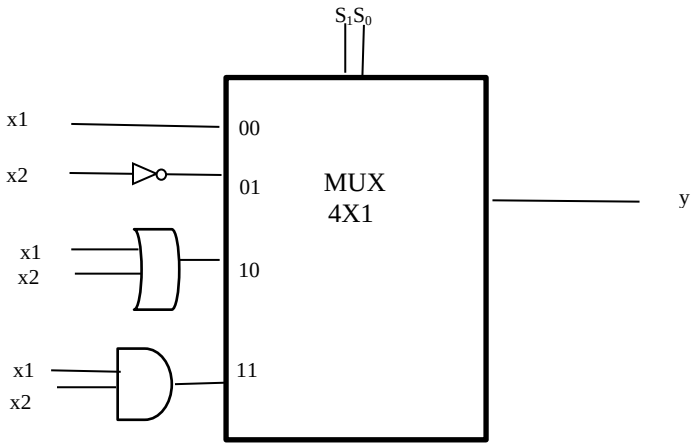
i. 16 × 1 multiplexer.
ii. 32× 1 multiplexer.
iii. 128× 1 multiplexer.
iv. 256× 1 multiplexer.

Solution:

i. 4 selection lines.
ii. 5 selection lines.
iii. 7 selection lines.
iv. 8 selection lines.

Problem 7:

Given the below design, fill the below table with the values of the output y given the inputs x1, x2, and the selection lines as stated in the table.



x1	x2	S1	S0	y
0	0	0	0	
0	1	0	1	
1	0	1	0	
1	1	1	1	

Solution:

x1	x2	S1	S0	y
0	0	0	0	0
0	1	0	1	0
1	0	1	0	1
1	1	1	1	1