CS2323 : HOMEWORK-1

ANSWERS:

1.a. addi x8, x5, -5

EXPLANATION: addi is used to add the value of one register and one immediate value. The immediate value can also be a negative number. So, here the operation will be like: x8(destination register) ← x5+(-5)

1.b. slli x5, x3, 3

EXPLANATION: slli(Shift Left Logical Immediate) is used to shift the bits of a source register to the left by specified number of positions. This operation effectively multiplies the value by 2^n where n is the number of bit positions to be shifted. So here, the value of x3 register is shifted to the left by 3 bits, which ultimately means multiplying the value by 2^3 and storing the result in x5 register.

1.c. add x19,x19,x10

EXPLANATION: x19+=x10 ultimately means x19=x19+x10. So directly we can use the add instruction to add the values of x19 and x10 registers and store the result in x19 register.

1.d. addi x15, x15, 1

EXPLANATION: ++x15 means incrementing the value in x15 register by 1. As we are adding the value in register by a constant number, we have to use the addi instruction, to add the x15 register value with 1 and store the result in the same register x15.

1.e. srai x9, x15, 2

EXPLANATION: srai(Shift Right Arithmetic Immediate) is used to shift the bits of a source register to the right by specified number of positions. This operation effectively divides the value by 2^n where n is the positions to be shifted. Here, the n value will be provided by the immediate value that is,2 . sSo, it means the register x15's value will be shifted to right by 2 bits, means, it will be divided by $2^2 = 4$ and stored in register x9.

1.f. li x12, 24

EXPLANATION: li which means load immediate value is used to declare and load the value in the register. Here, we load the value 24 into the register x12.

2.a. ld x9, 160(x5)

addi x7, x9, 100

sd x7, 96(x5)

2.b. addi x9, x9, 1

sd x9, 160(x5)

2.c. ld x9, 96(x5)

ld x10, 40(x5)

sd x9, 40(x5)

sd x10, 96(x5)

2.d. ld x10, 32(x5)

lui x11, 0x00000000FFFFFFFF

and x10, x11, x10 // 32 0's and 32 lsb's

sd x10, 32(x5)

2.e. ld x9, 16(x5)

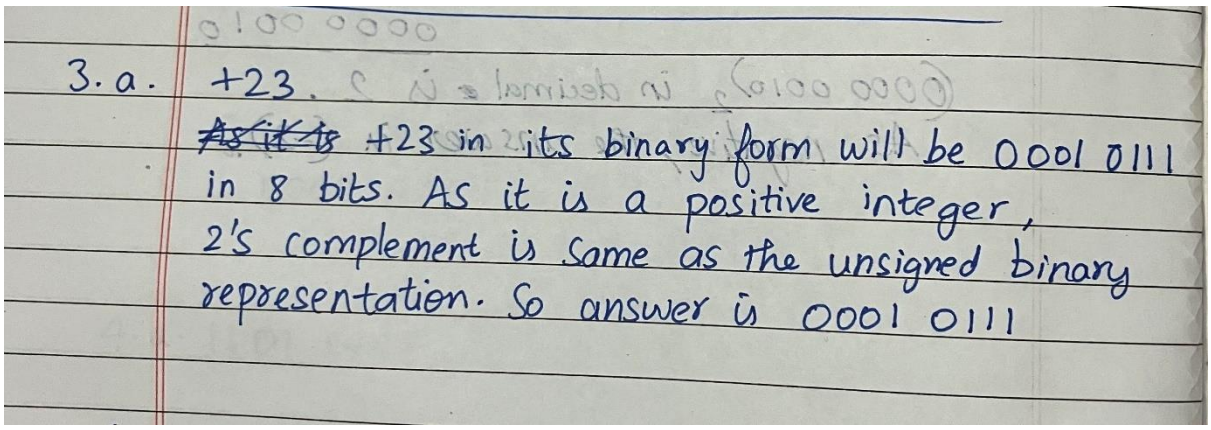
srli x12, x9, 32 // 32 0's and 32 msb's

slli x10, x10, 32 //lsb comes in front

or x9, x10, x12 //swapping

sd x9, 16(x5)

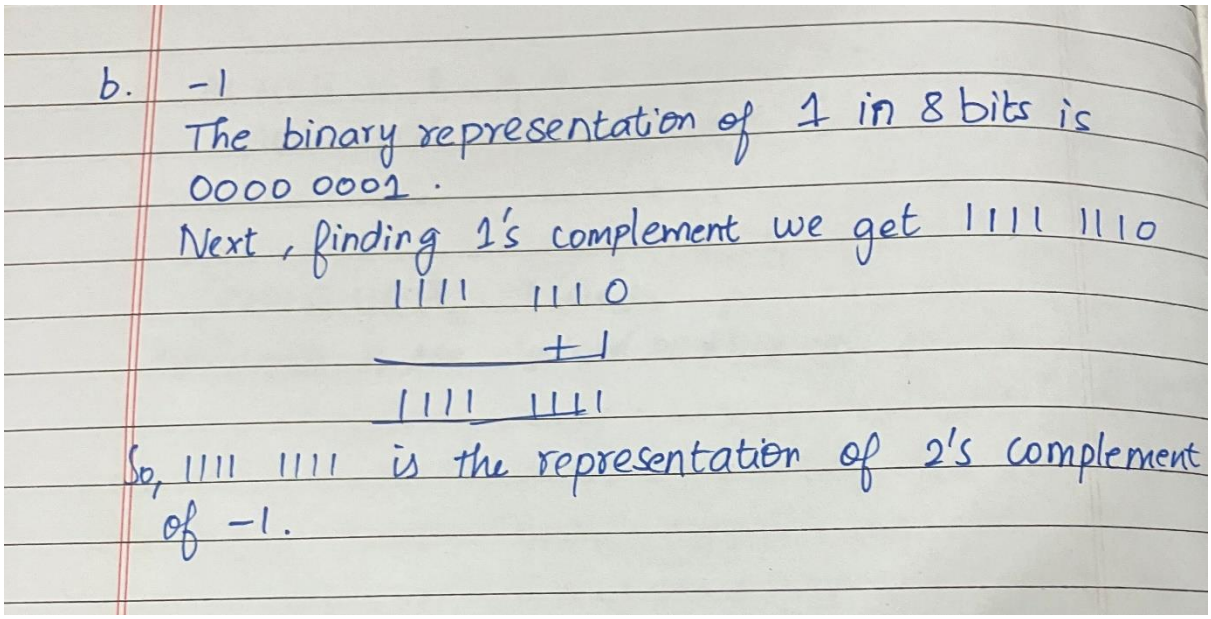
3.a.



3. a. +23.

~~As it is~~ +23 in its binary form will be 0001 0111 in 8 bits. As it is a positive integer, 2's complement is same as the unsigned binary representation. So answer is 0001 0111

3.b.



b. -1

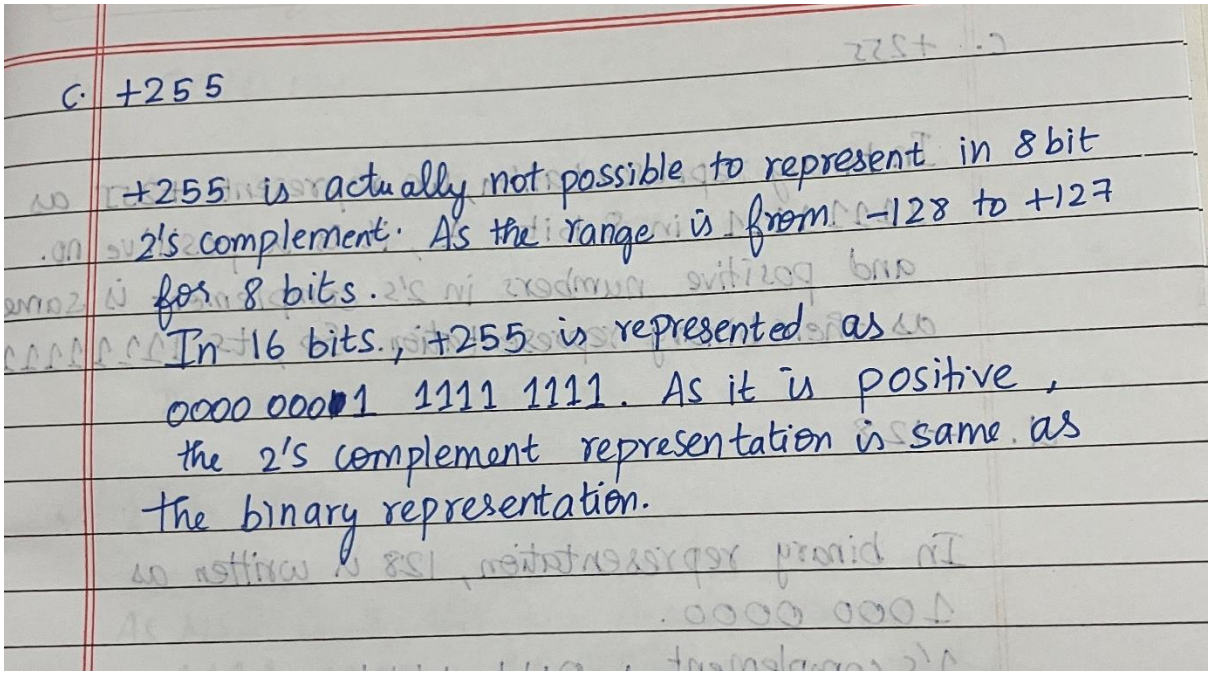
The binary representation of 1 in 8 bits is 0000 0001.

Next, finding 1's complement we get 1111 1110

```
  1111 1110
+        1
-----------
  1111 1111
```

So, 1111 1111 is the representation of 2's complement of -1.

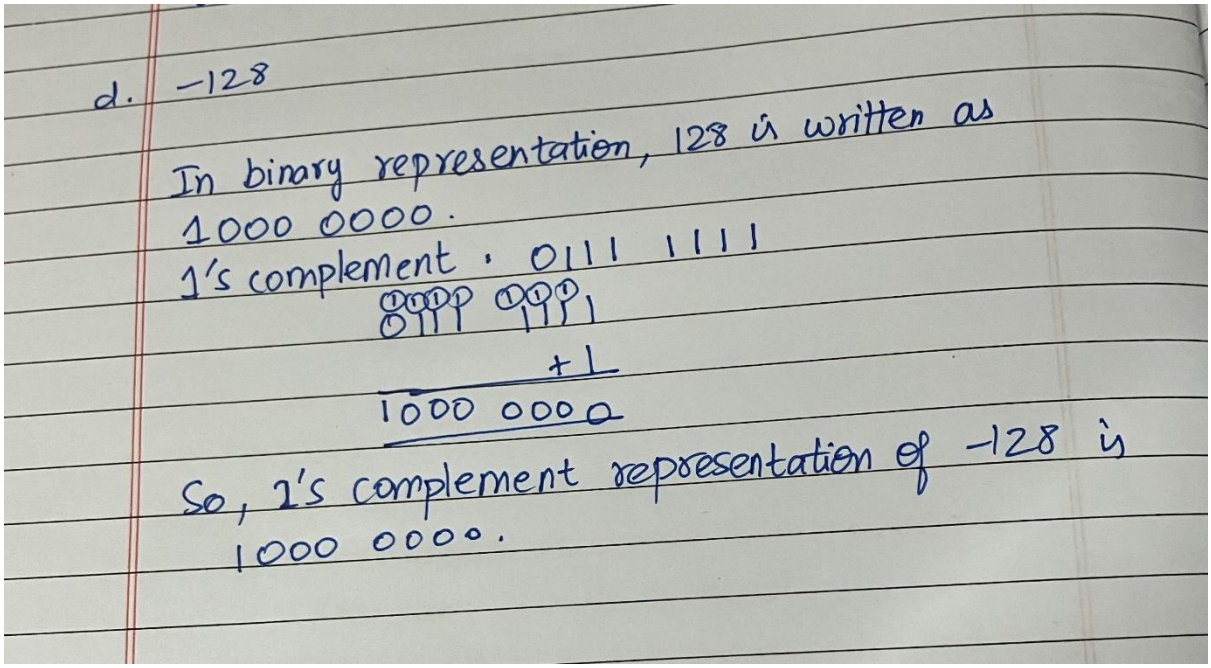
3.c.



c. +255

+255 is actually not possible to represent in 8 bit 2's complement. As the range is from -128 to +127 for 8 bits.
In 16 bits., +255 is represented as 0000 0000 1111 1111. As it is positive, the 2's complement representation is same as the binary representation.

3.d.



d. -128

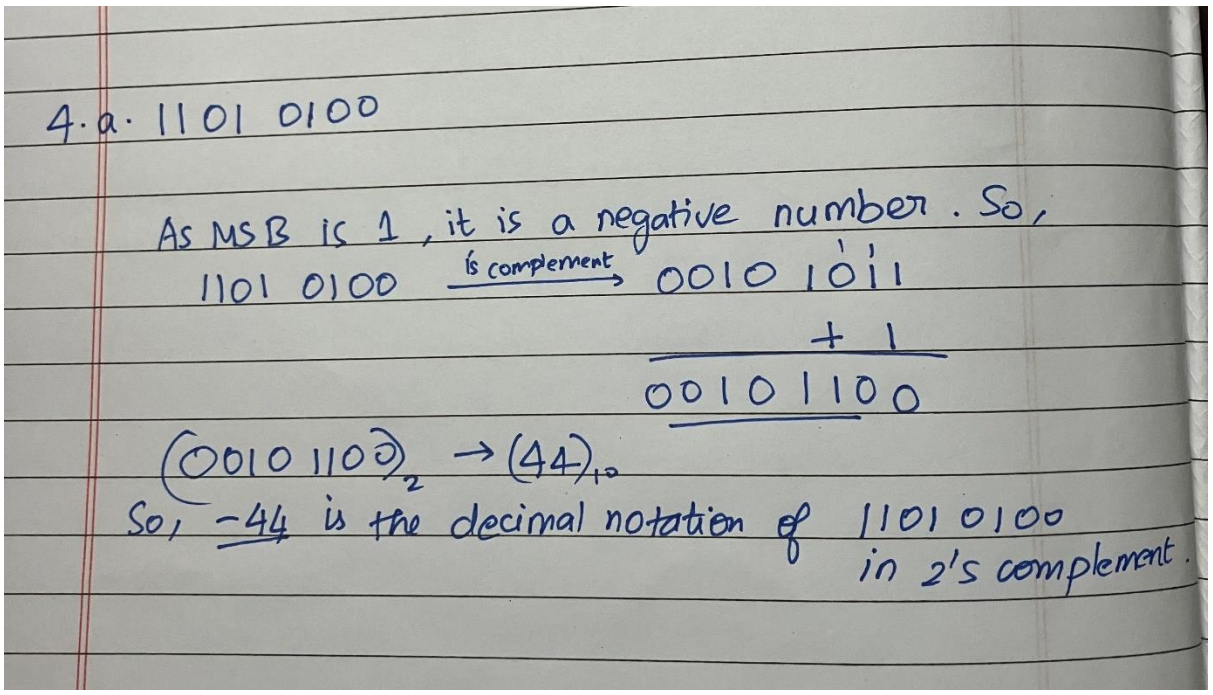
In binary representation, 128 is written as 1000 0000.
1's complement : 0111 1111

+1

1000 0000

So, 2's complement representation of -128 is 1000 0000.

4.a.



4.a. 1101 0100

As MSB is 1, it is a negative number. So,

1101 0100 —(1's complement)→ 0010 1011

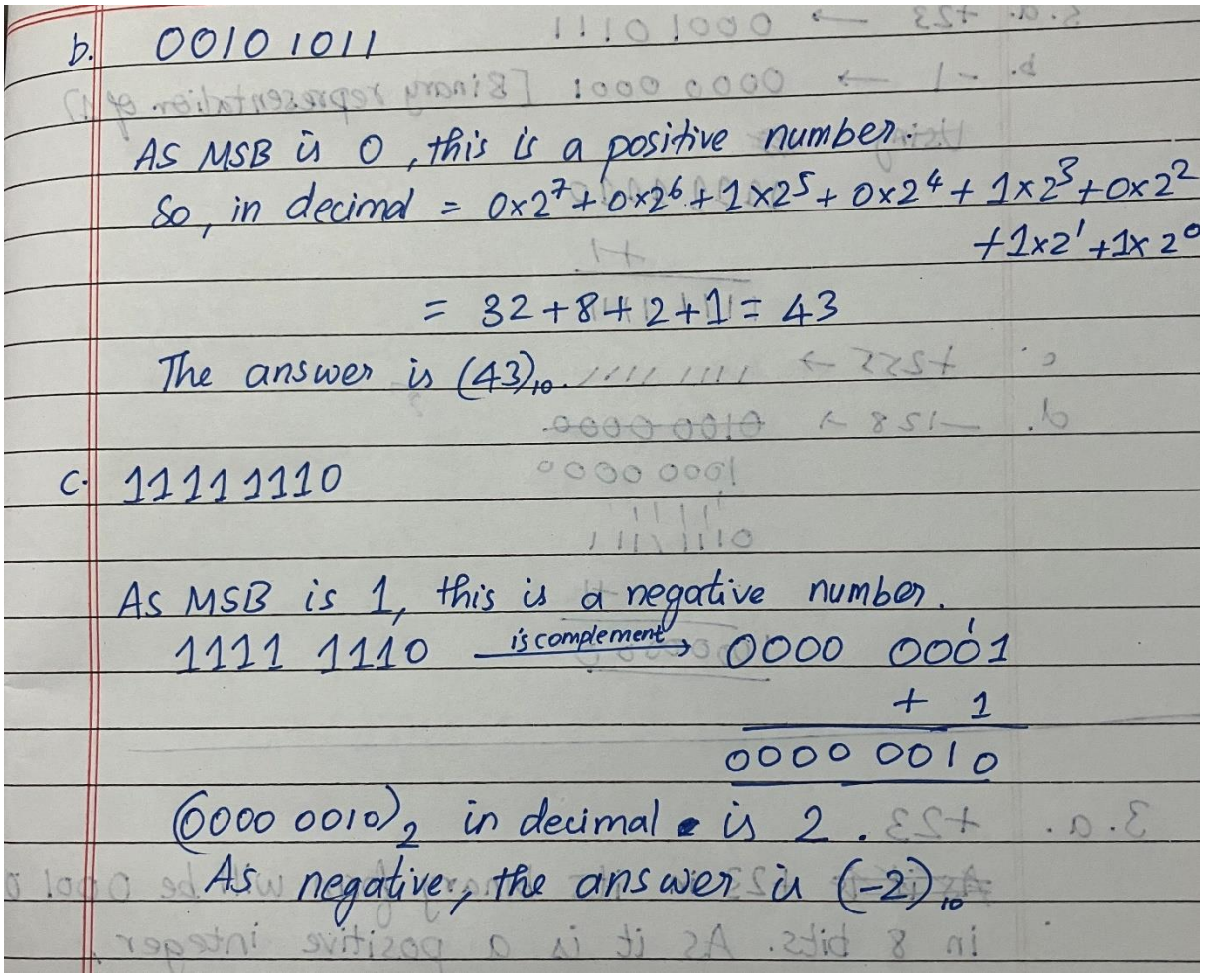
+ 1

0010 1100

$(0010\ 1100)_2 \rightarrow (44)_{10}$

So, -44 is the decimal notation of 1101 0100 in 2's complement.

4.b. and 4.c.



b. 0010 1011

As MSB is 0, this is a positive number.

So, in decimal $= 0\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$

$= 32 + 8 + 2 + 1 = 43$

The answer is $(43)_{10}$.

c. 1111 1110

As MSB is 1, this is a negative number.

1111 1110 —(1's complement)→ 0000 0001

+ 1

0000 0010

$(0000\ 0010)_2$ in decimal is 2.

As negative, the answer is $(-2)_{10}$