Q1) Explain the organization of Von Neumann computer with schematic digram.

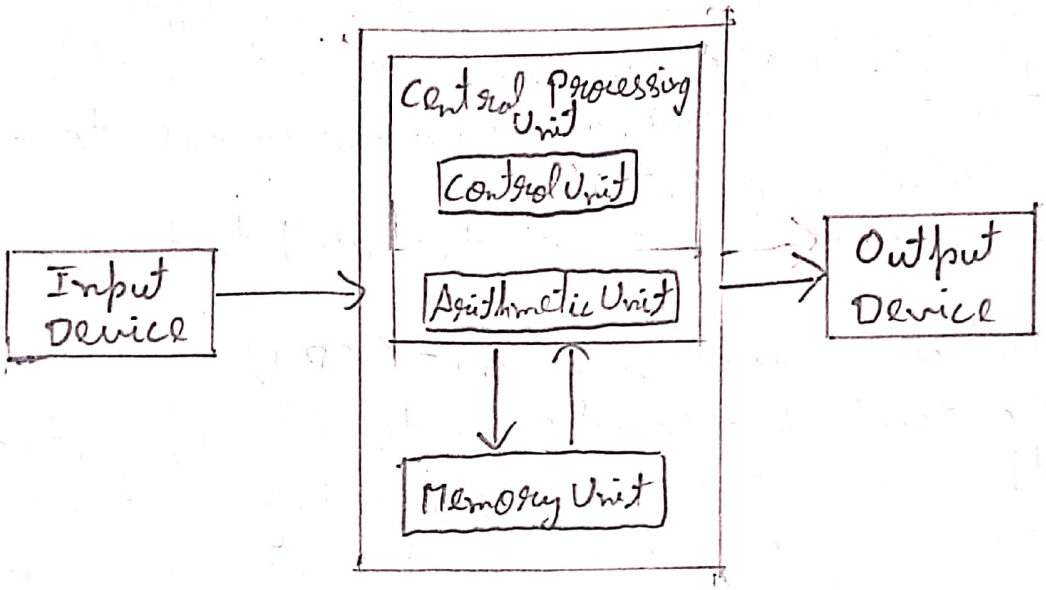
A1) The Von Neumann architecture is a foundational model for modern computers, conceptualized by mathematician John Von Neumann in 1945. It defines a system where the computer's hardware and software components are integrated into a structure, allowing stored programs to be executed sequentially.
The key components of the architecture are:

1) Central Processing Unit (CPU)
- Control Unit (CU): Directs the operations of the computer by interpreting instructions from memory and issuing control signals to execute them
- Arithmetic Logic Unit (ALU): Performs arithmetic and logical operations
- Registers: Small, fast storage locations in the CPU used to hold data and instructions temporarily.

2) Memory Unit (RAM)
- Stores both data and instructions. In the Von Neumann architecture, memory is unified, meaning that same memory stores both program code and data it operates on.

3) Input / Output Devices

- Input devices (like keywords, mouse etc) provide data to the computer
- Output devices (like monitors, printers etc) display or send processed data to the user.



Q2) Represent +68 and -68 signed number in 1's & 2's complement form

A2) +68 Binary 01000100_2

For 1's complement

$+68 > 0$

∴ 1's complement

$= 01000100_2$

For 2's complement

$= 01000100_2$

-68 Binary 01000100_2

For 1's complement

$-68 < 0$

Reverse the 0's with 1's

$= 10111011_2$

For 2's

1's + 1

$= 10111011 + 1$

$= 10111100_2$

Q3) Determine the decimal value of 1101010, if they are in (i) sign-magnitude form (ii) 2's complement form, and (iii) 1's complement form.

A3) i) Sign-magnitude form

In Sign-magnitude form MSB represents the sign (0 for positive & 1 for negative)

∴ 1101010 will represent −42

MSB ↙ ↘ Mag

ii) 2's Complement form

Step1: 1101010 (Take the full binary number)

Step2: 0010101 (1's complement)

Step3: 0010101 + 1 (2's complement)

Sign, 0010110

Final, −22 in Decimal

iii) 1's complement form

Step1: 1101010 (Take full binary number)

step2: 0010101 (1's Complement)

∴ Final, −21 in Decimal

Q4) Considering a 2's complement-based 8 bit processor, perform the following arithmetic operations. Also, check the existence of overflow in each operation

a) (-100) + (15)

1) Represent -100 in 2's complement

as MSB is

Binary for 100 : 01100100

2's of -100

1's + 1 = 10011011 + 1

= 10011100

2) Represent 15 in 2's complement

Binary: 00001111

2's : 00001111

-85 fits the 8-bit range

Here, no overflow

```
  10011100
+ 00001111
------------
  10100011
```

↙

MSB (negative)

Takes 2's complement of result

1's : 01010100

2's : 01010100 + 1

```
  01010101
```

Decimal - 85

Result: -85

b) (-32) + (-97)

1) Represent -32 in 2's complement

Binary: 00100000

2's of -32

1's + 1 = 00100000 11011111 + 1

= 11100000

2) Represent -97 in 2's complement

10011111

```
  11100000
+ 10011111
------------
 (1)01111111
```

↙ Discard ↓

overflow Decimal: 127

In reality, -32 + (-97) should equal -129 but this can't be represented in 8 bit 2's complement

Here, the overflow

Q5) Express 85.125 as a floating point number using IEEE double precision

A5) Step1: Convert number into binary

2	85	
2	42	1
2	21	0
2	10	1
2	5	0
2	2	1
2	1	0
	0	1

$0.125 \times 2 = 0.250$ 0
$0.250 \times 2 = 0.500$ 0
$0.500 \times 2 = 1.0$ 1

$85.125_{10} = 1010101.001_2$

Step2: Normalize the binary Number

$85.125_{10} = 1.010101 \times 2^6$

Step3: Add 1023 to the Exponent and find its binary

Exponent $= 6 + 1023 = 1029 = 10000000101_2$

Step4: Assemble (Final Representation

0 10000000101 01010100 & upto 52 bits

↑ Sign ↑ Exponent ↑ Mantissa

Q6) Find out the floating point number from the following representation

0 10000011 00111000000000000000000

A6) Step 1:

Sign: 0

Exponent: $10000011_2 = 131_{10}$

Mantissa: 00111000000000000000000

Step 2:

Calculate the exponent

Decimal of 10000011_2 is 131_{10}

This is excess 127 form

To get actual: $131 - 127 = 4_{10}$

Step 3:

Construct Mantissa

$1.00111000000000000000000_2$

Step 4:

Value $= 1.00111_2 \times 2^4$

$1.00111_2 = 1 + 2^{-3} + 2^{-4} + 2^{-5}$

$= 1 + 0.125 + 0.0625 + 0.03125$

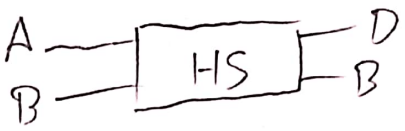
$= 1.21875_{10}$

Step 5: Multiply with Exponent

$1.21875 \times 2^4 = 19.5_{10}$

Q7) In an arithmetic logic unit, two half adders are connected to form a full subtractor. Show the connections between half subtractors. Also determine the required Boolean logic for outputs

A7) For Half Subtractor,

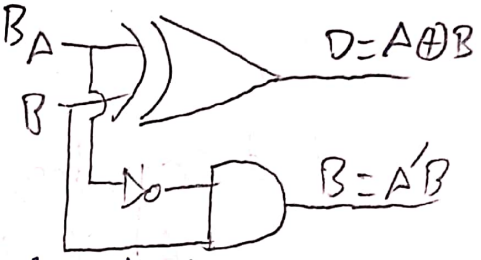


Truth Table

A	B	Diff	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = A'B + AB' = A \oplus B$

$B = A'B$



For Full Subtractor Using Two Half Subtractors

1) First Half subtractor

Inputs: A & B

Output: $D_1 = A \oplus B$

$B_1 = \bar{A}B$

2) Second Half Subtractor

Inputs: D_1 & B_{in}

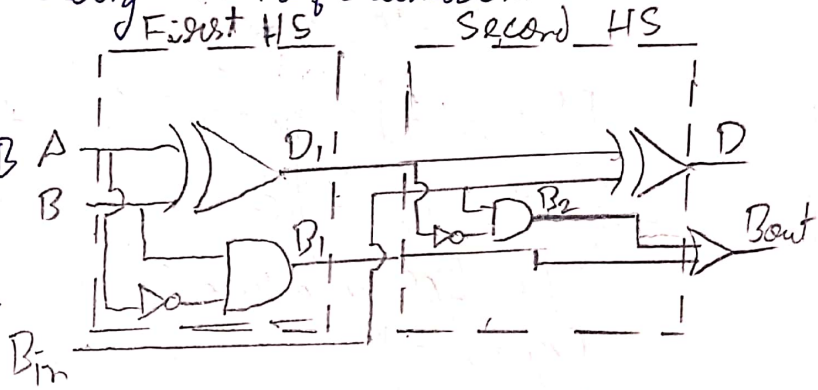
Output:

$D = D_1 \oplus B_{in}$

$= (A \oplus B) \oplus B_{in}$

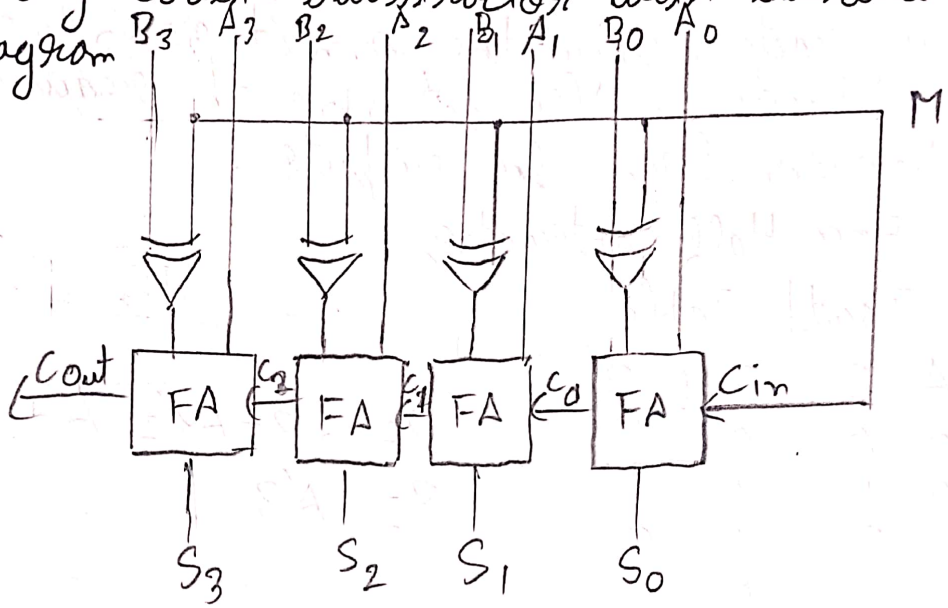
$B_2 = \bar{D}_1 \cdot B_{in} = (\overline{A \oplus B}) B_{in}$

$B_{final} = B_1 + B_2 = (\bar{A} \cdot B) + ((\overline{A \oplus B}) \cdot B_{in})$



Q8) Explain how control input M results a 4-bit binary adder / substractor with suitable logic diagram

A8)



Let Consider, two inputs

$A : A_0 A_1 A_2 A_3$

$B : B_0 B_1 B_2 B_3$

M is the control line that has value either 0 or 1

If $M = 1$, then $B_0 (XOR) M = B_0'$

The formula would be $A + B_0' + C_{in}$ which now equals the 2's complement substraction of two numbers A & B

This shows, on M=1 2's complement substraction occurs

If $M = 0$, then $B_0 (XOR) M = B_0$

This is A+B which is simple addition procedure

This shows, on M=0 binary addition occurs

Q9) Implement the following Boolean function with a suitable multiplexer

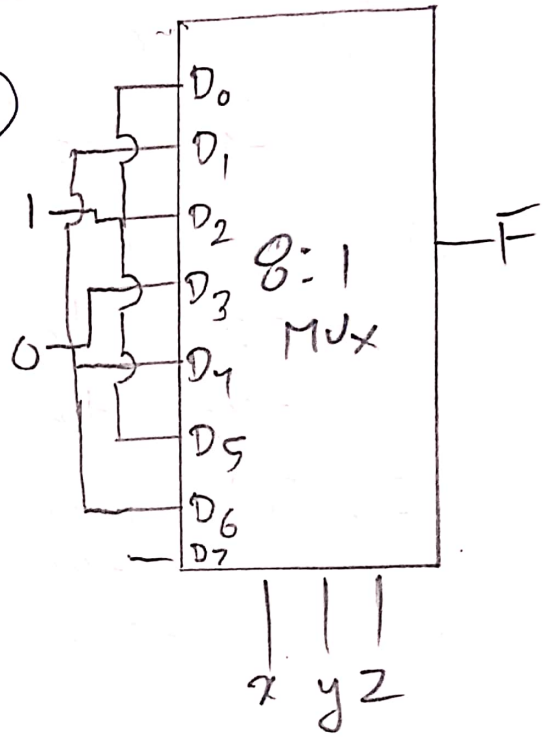
$F(x,y,z) = \Sigma(0,2,5,7)$

A9) $F(x,y,z)$ has 3 variables

∴ 8:1 MUX will be used

$F(x,y,z) = \Sigma(0,2,5,7)$

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



Select lines will be x, y, z

$D_0=1, D_1=0, D_2=1, D_3=0, D_4=0, D_5=1, D_6=0, D_7=1$

$$F(x,y,z) = x'y'z' + x'yz' + xy'z + xyz$$
$$= x'(y'z' + yz') + xy'z + xyz$$
$$= x'$$

Q10) Determine the minimum number of bits needed to represent -32 in 2's complement representation

A10) The n-bit two's complement, range is given by as

$$-(2^{n-1}) \text{ to } (2^{n-1}-1)$$

for -32

$n = 6$

-2^5 to $(2^5 - 1)$

-32 to 31

∴ The minimum number of bits to represent -32

$\Rightarrow$ 6 bits