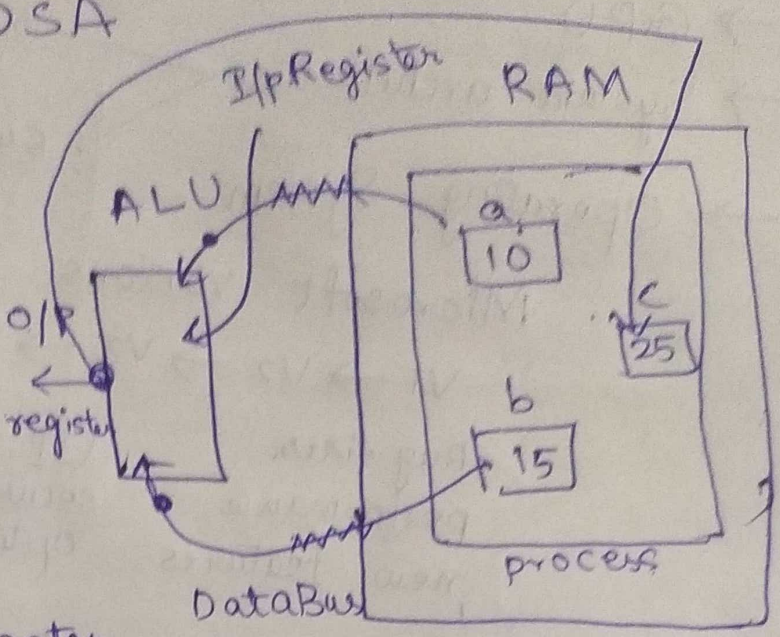
Session - 1 5/11/2022

DSA

```
int a = 10
int b = 15
int c = a+b
print c //25
```

10 : 1 0 1 0

15 : 1 1 1 1



* ALU made of gates ;

AND, OR, XOR gates.

System A ✗

1 GHtg

1×10^9 cc/sec.

System B. ✓

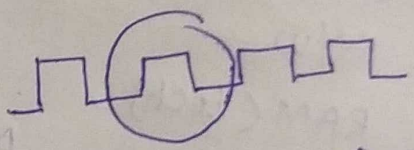
2 GHtg.

2×10^9 cc/sec, 1 cc

$K = 10^3$

$M = 10^6$

$G = 10^9$

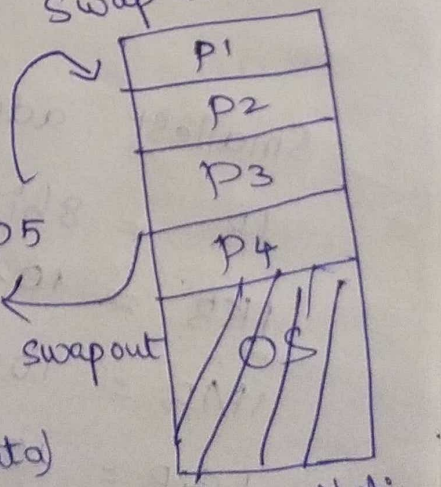


1 CC = 1 instruction

* processing power, speed

→ memory → RAM < size. (active data)
technology.

- If the RAM size is very less → CS ↑
  CPU Time ↑ wasted
  ↓
  Thrashing.



context switching.

→ HD < size. (Inactive data)
technology

magnetic mechanism ← HDD, SSD → digital mechanism.

- Cache memory.

memory pyramid (TO DO !)

→ GPU

→ System architecture — 32 bit / 64 bit

→ Operating System

Microsoft versions

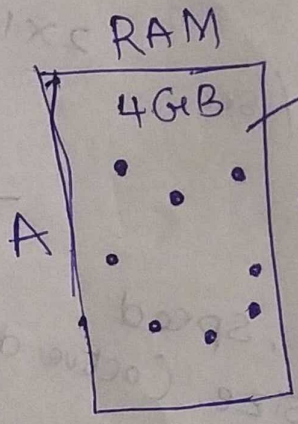
V1 → V2 → V3 → V4

Bug fixes. UI
performance security
new features optimizations.

* processing power, RAM size.

System A
HD
GPU
Architecture
N/w
RAM (tech)
4 GB

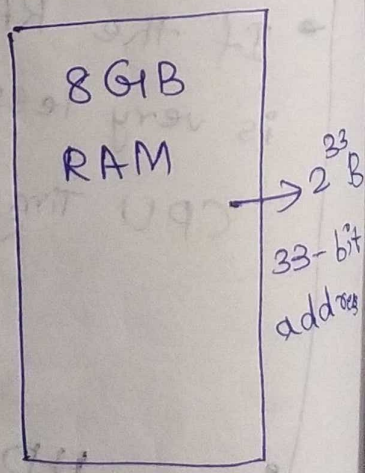
System B.
8 GB

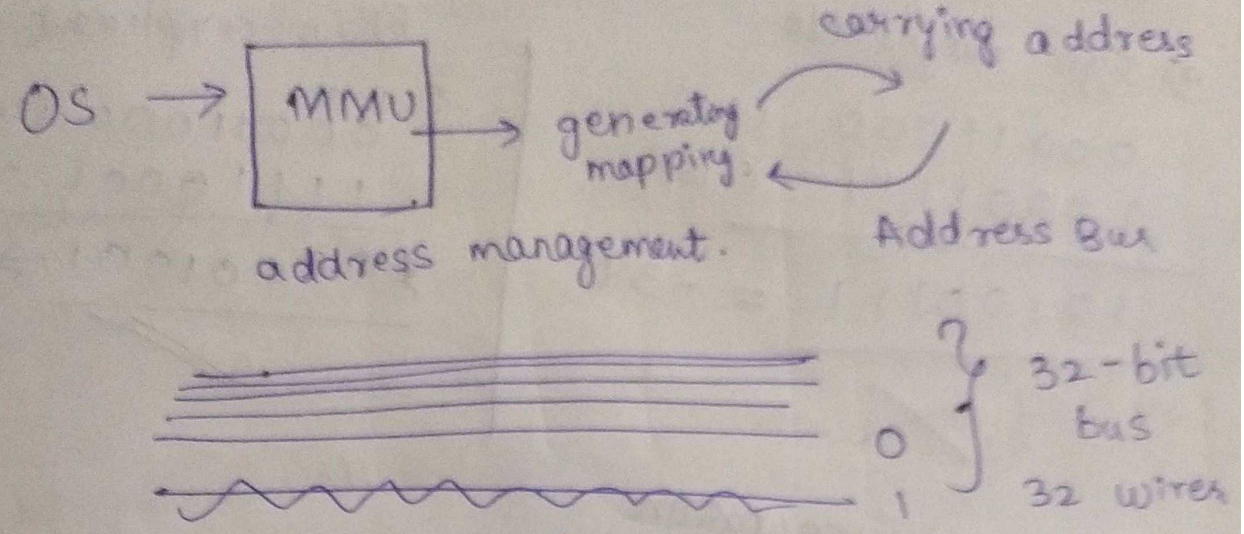


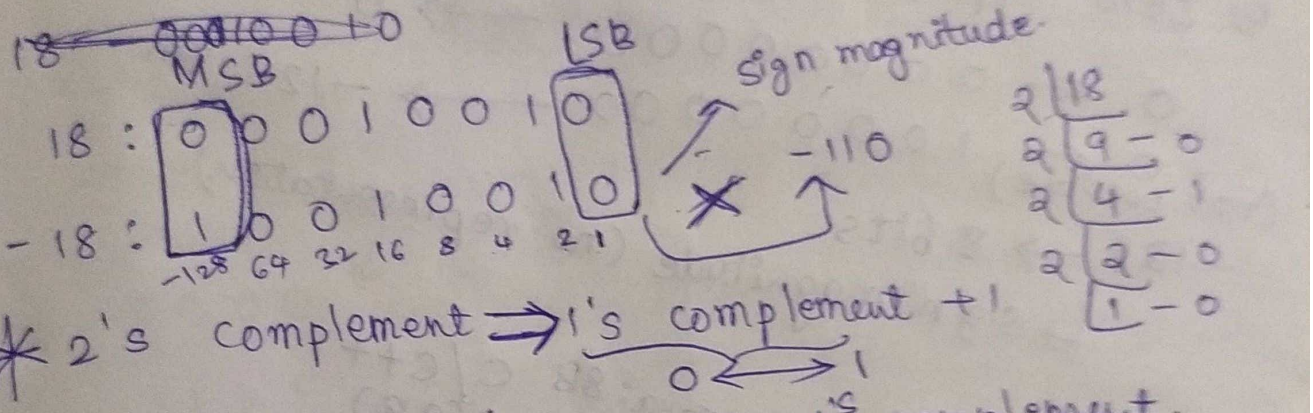
2^{32} B ≈ 2^{32} unique addresses

Smallest addressable unit is - 1 Byte.
B

1B = 8 bits
1KB = 1024B = 2^{10} bits
1MB = 1024KB = 2^{20} Bytes
1GB = 1024MB = 2^{30} B.







* 2's complement ⇒ 1's complement +1

0 ⟷ 1

← 1's complement.

18 1 1 1 0 1 1 0 1

+ 1

-18 1 1 1 0 1 1 1 0

-128 64 32 16 8 4 2 1

↳ -18.

carry = S/2

value = S%2

18 → 0 0 0 1 0 0 1 0

-128 64 32 16 8 4 2 1

98 0 1 1 0 0 0 1 0

-98 1 0 0 1 1 1 1 0 → -98

0 1

+ 1

-41 1 1 0 1 0 1 1 1 → -41

0+1

1

0 1 1 0 0 0 1 0

1 0 0 1 1 1 1 0

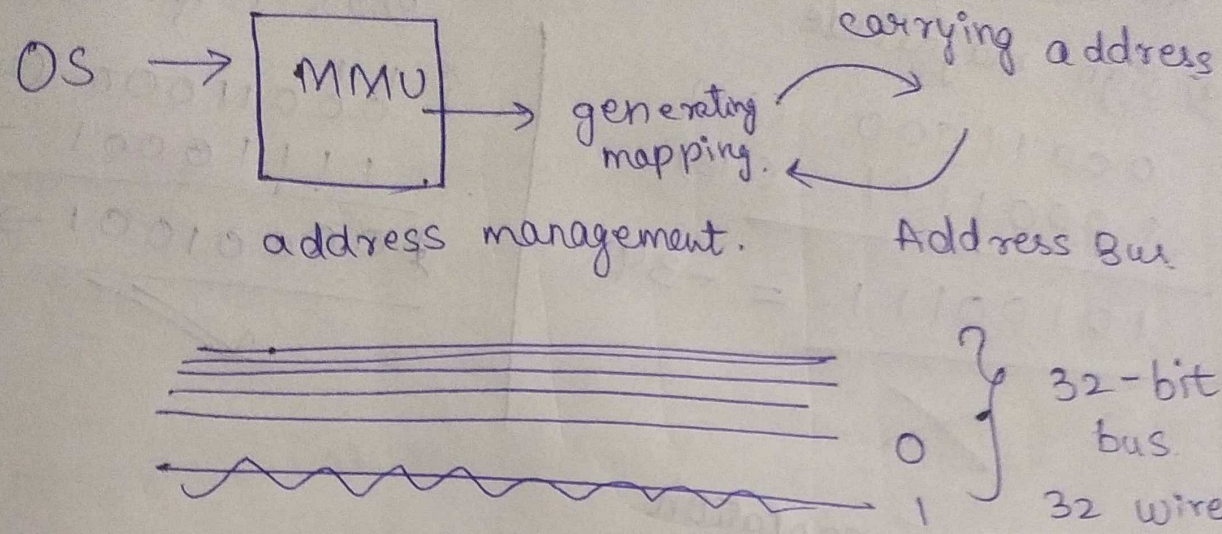
1

1 0 0 1 1 1 1 0

1 1 0 1 0 1 1 0

1

1 1 0 1 0 1 1 1



18 → 00010010

MSB LSB

18 : 0 0 0 1 0 0 1 0

-18 : 1 0 0 1 0 0 1 0

-128 64 32 16 8 4 2 1

sign magnitude.

-110

2 | 18
2 | 9 - 0
2 | 4 - 1
2 | 2 - 0
1 - 0

* 2's complement ⇒ 1's complement +1

0 ⇄ 1

18 1 1 1 0 1 1 0 1 ← 1's complement.

+ 1

-18 1 1 1 0 1 1 1 0

-128 64 32 16 8 4 2 1

↳ -18.

carry = S/2

value = S%2

18 → 0 0 0 1 0 0 1 0

-128 64 32 16 8 4 2 1

98 0 1 1 0 0 0 1 0 → 0'

+1

-98 1 0 0 1 1 1 1 0 → -98

41 0 0 1 0 1 0 0 1 → 0+1

-41 1 1 0 1 0 1 1 1 → 1

↳ -41

0 1 1 0 0 0 1 0

1 0 0 1 1 1 0 1

1 1

1 0 0 1 1 1 1 0

1 1 0 1 0 1 1 0

1

1 1 0 1 0 1 1 1

24 + 15
24 + (-15) } 9

a) sign magnitude

00011000 (24)
10001111 (-15)

10100111 = -39

✗

b) 2's complement

1 1 1
00011000 (24)
11110001 -15

00001001 → 9

✓

b) Zero representation

0 ← 00000000 → 0
(not possible) 10000000 → -128

-128 → 8 bits
↳ sign magnitude (no representation)

* Data Types: Java: 8B C/C++ (8B)

4B int → short / long / long long / signed / unsigned
2B ↳ C/C++ → 4B/8B (Architecture)
1B char
~~4B~~ float
8B double / long double
1B × boolean → 0 → F, 1 → T
1B void
Byte (Java)

bool ar[N/8]: → ceil[N/8]

bool ar[N]; → N Bytes

bool b0, b1, b2, b3, b4 → b.

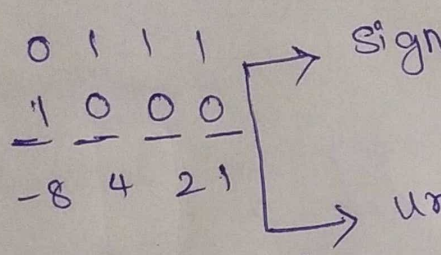
b: _ _ _ 1 0 0 1 1 } 8 boolean values

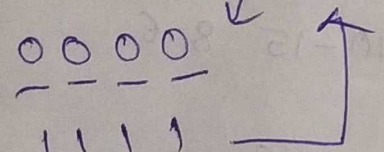
int ar[N/32] → 4 × N/32 ⇒ N/8

* In a single boolean variable, we have 8 boolean values.
* to store 8 boolean values, we will use.

N/8.

* variable by default → signed.
* msB -ve → signed
  +ve → unsigned.
* Negative values can not store in unsigned.





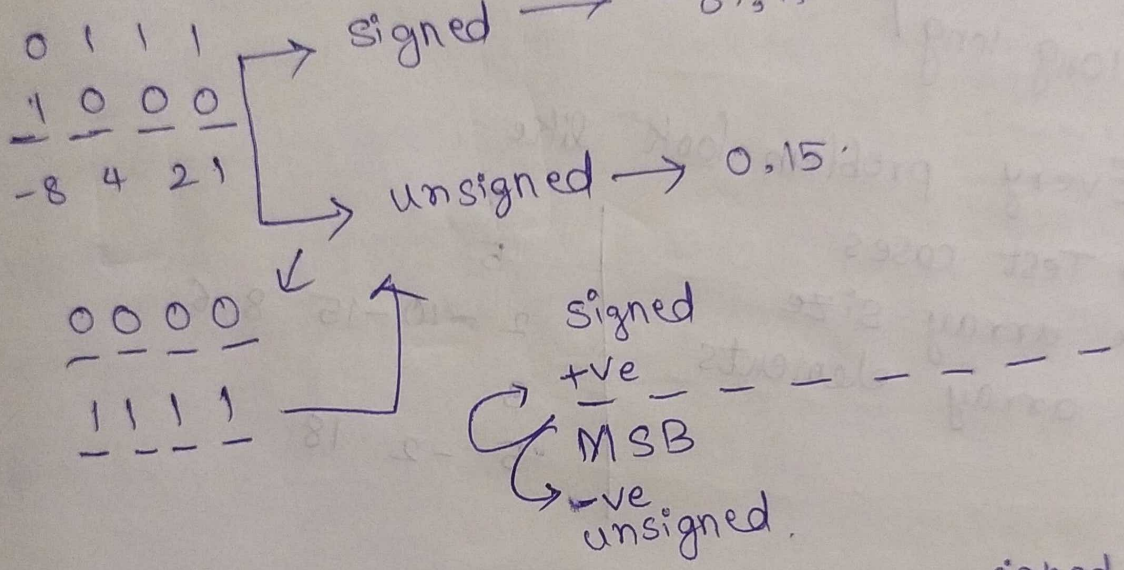
signed
+ve
MSB
-ve
unsigned.

# bits	signed.	unsigned
1 →	-1, 0	0, 1
2 →	-2, 1	0, 3
4 →	-8, 7	0, 15
8 →	-128, 127	$0, 2^8 - 1$ 0 - 255.
N →	~~$-2^{N-1}/2$, $2^{N-1}/2-1$~~ -2^{n-1}, $2^{n-1}-1$	$0, 2^N - 1$ $0, 2^{32} - 1$
int 32 → size.	-2^{31}, $2^{31}-1$ -2×10^9, $2\times10^9-1$ Range	~~$0, 4\times10^9-1$~~ $0,4\times10^9-1$ Range.

* In a single boolean variable, we have 8 boolean values.
* to store 8 boolean values, we will use.

N/8.

* variable by default → signed.
* msB -ve → signed
  +ve → unsigned.
* Negative values can not store in unsigned.



# bits	signed.	unsigned
1 →	-1, 0	0, 1
2 →	-2, 1	0, 3
4 →	-8, 7	0, 15
8 →	-128, 127	$0, 2^8 - 1$ 0 - 255.
N →	$-2^{N-1}/2, 2^{N-1}/2 - 1$ $-2^{n-1}, 2^{n-1} - 1$	$0, 2^N - 1$
int 32 → size.	$-2^{31}, 2^{31} - 1$ $-2 \times 10^9, 2 \times 10^9 - 1$ (Range)	$0, 2^{32} - 1$ $0, 4 \times 10^9 - 1$ (Range.)

$2^x \rightarrow 10^y$.

$2^{10} = 1024 \simeq 10^3$

$2^{20} \simeq 10^6$.

$2^{30} \simeq 10^9$

long : 8×8 → 64 → long long

$-2^{63}, 2^{63}-1$ $\quad$ $0, 2^{64}-1$

$-8 \times 10^{18}, 8 \times 10^{18}-1$ $\quad$ $0.16 \times 10^{19}-1$

Every problem look like.

* Test cases
* array size
* array elements

```
2
5
2 -10 -15 8 6.
3
5 -2 18.
```

main:

```
Read T
loop(T)
  Read I/P
  process
  print O/P
```

problem:

```
int findSum(int arr[], int N)
{ int i, sum=0;
  for(i=0;i<N;i++)
    sum = sum+arr[i];
  return sum;
}
```

Sum of elements in the array

\t (tab) → indentation

constraints: (Important)

$1 \le T \le 100$

$1 \le N \le 10^3$

$-10^5 \le ar[i] \le 10^5$ → $10^3 \times 10^5$

$-10^8 \le sum \le 10^8$ → 10^8

\t → tab

\n → new line.

* correct data Types

Operators

1) Arithmatic operators +, −, *, /, %

2) logical operators &&, ||, !

3) Relational operators <, >, <=, >=, !=, ==

4) Bitwise operators &, |, ^, ~, <<, >>

5) ++, --, ?:, ,, sizeof(—)

a	b	a|b	a&b	a^b	~a
0	0	0	0	0	1
0	1	1	0	1	1
1	0	1	0	1	0
1	1	1	1	0	0

if there are odd one's → 1 } In XOR.

even one's → 0 }

int a=12, b=25 0000 1100 → 12 (32 bits)
0000 0000 0001 1001 → 25 (32 bits)

int c = a|b : 0000 1101 → 29

a&b: 0000 1000 → 8

a^b: 0001 0101 → 21

~a: 1111 0011 →

Shift operators: • Right Shift

a: 0001 1001

a>>1: 0 0001 100 /2

a>>2: 0 0000 110 /2

a>>3: 0 0000 011 /2

a>>i = a/2^i

a: 0000 0100 (5)

a<<1 0000 1010 (10) *2

a<<2 0001 0100 (20) *2

a<<3 0010 1000 (40)

a<<i = a*2^i

a<<100 → not possible

int 32 long long int 64

<<< Java TO DO

int a= ___ , b= ___ , c= ___ ;

OR operator.

a|b = b|a.

a|b|c = c|a|b

} associative commutative

} AND XOR

a

odd even

a|a = a

a&a = a

a^a = 0

a|0 = a

a&0 = 0

a^0 = a.

a|1 = a a+1

a&1 = a 0

a^1 = a-1 a+1