
FORMULA SERIES
ONE SHOT FOR COMPLETE CS
TOC &CD
OS&ALGO
DE
CN
DMGT
Math
C&DSA
DBMS
&COA

START

# CoA

1. Introduction of COA
2. M/c Instn & Addressing Mode
3. Floating Point Representation
4. ALU Data Path & Control Unit
5. Pipelining
6. Cache Memory
7. Secondary Memory & IO Org

① Introduction of COA

(i) PC Value → Word } Addressable
→ Byte }
⇒ Interrupt occur During the execution

$I_i$ then what Return address Push into the Stack ?

② Memory Concept

$2^n \times m \Rightarrow$ n is # A·L
m is # D·L

$2^{10} = 1K$

$2^{20} = 1M$

$2^{30} = 1G$

$2^{40} = 1T$

$2^{50} = 1P$

$2^{60} = 1E$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{10} = 1024$

1 Byte = 8 bit

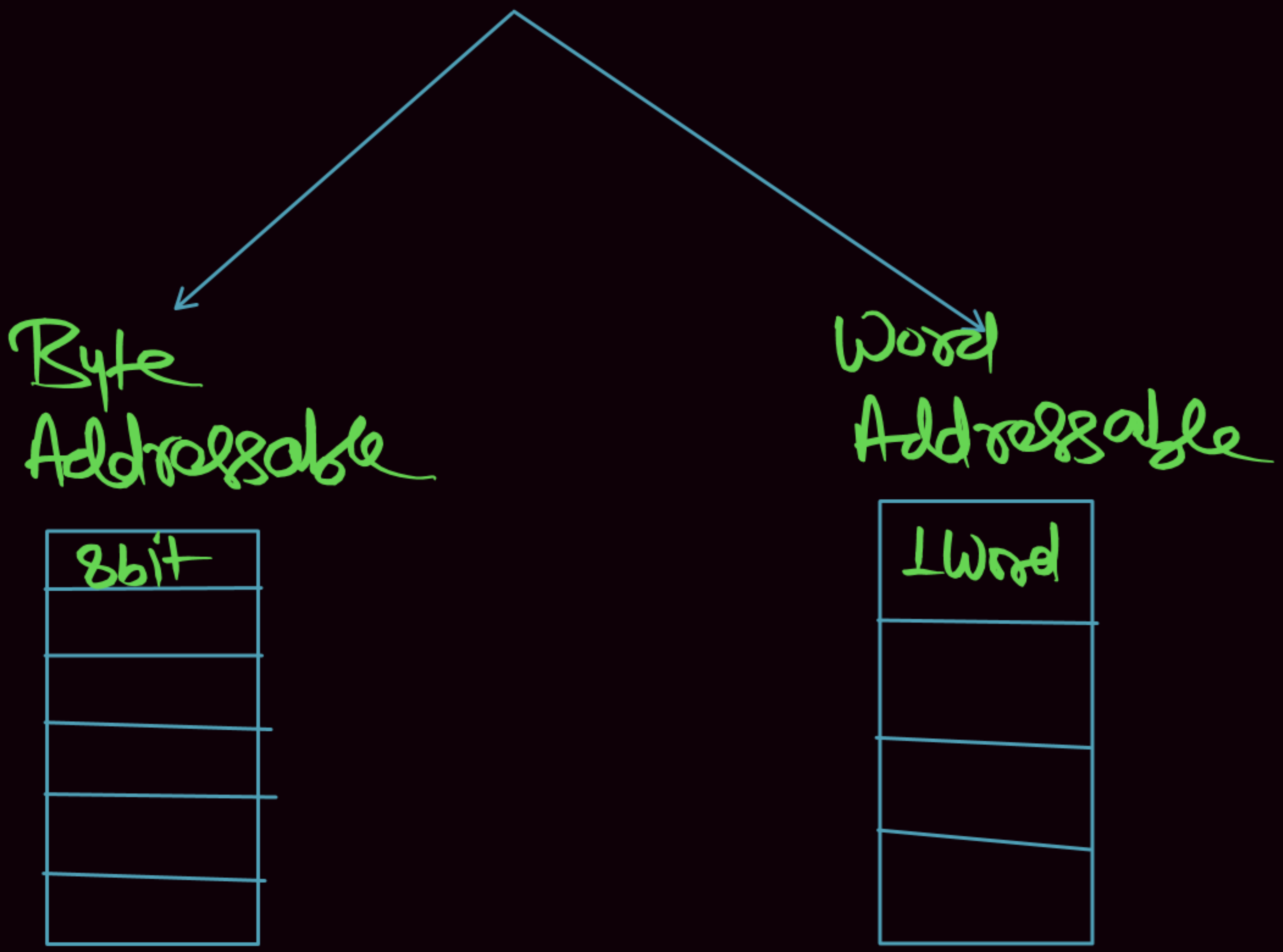
1 Nibble = 4 bit

1. 4 mByte $\Rightarrow 2^2 \cdot 2^{20}$ Byte

$\Rightarrow 2^{22}$ Byte

Address = 22 bit

$$8GB \Rightarrow 2^3 \cdot 2^{30} \Rightarrow 2^{33} \Rightarrow 33\text{bit Address}$$



1 Word = 32 bit [4 Byte]

Byte $\rightarrow$ Word

Word $\rightarrow$ Byte

2W $\Rightarrow$ 2 x 4 Byte = 8 Byte

1 km = 1000 meter

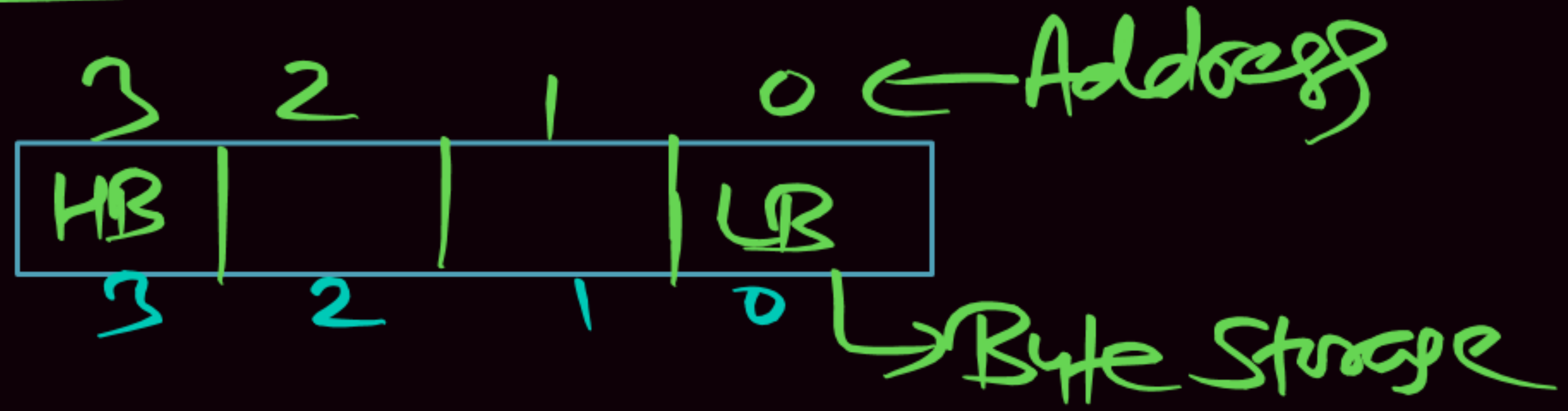
km $\rightarrow$ meter

meter $\rightarrow$ km

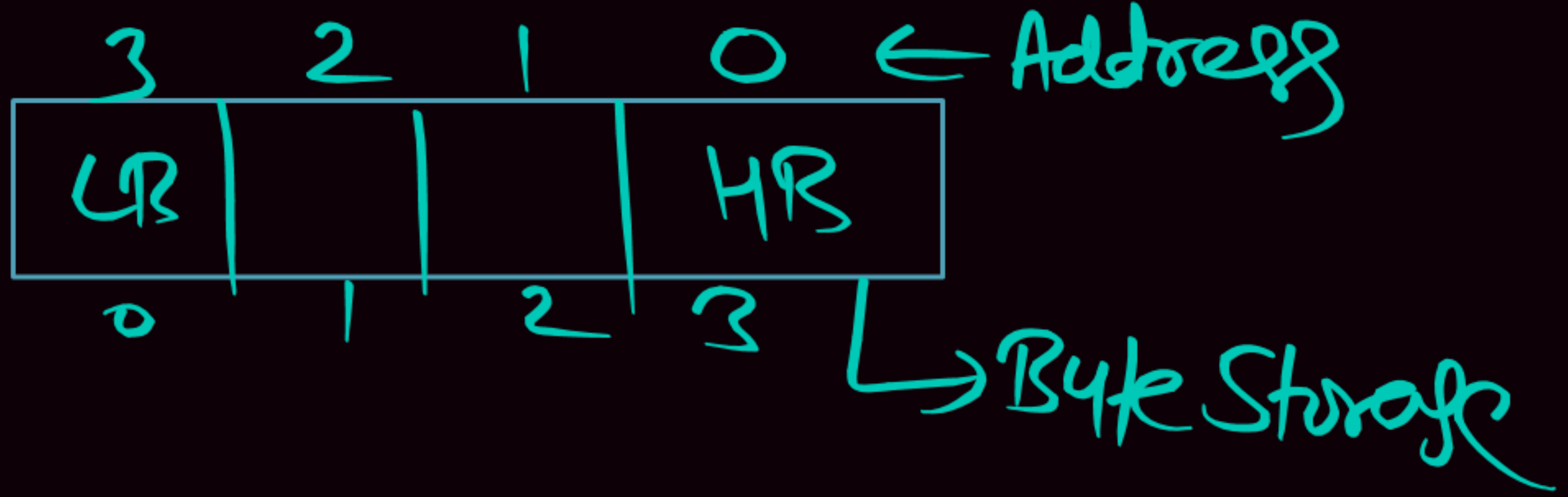
16bit Processor
32bit

Byte Ordering Endian m/c ism.

1. Little Endian




2. Big Endian



32bit Processor $\Rightarrow$ 4 Byte

① Little Endian : Right to Left

② Big Endion : Left to Right

32bit Processor Starting address : 100

0001 0001

$(15, 25, 29, 11)_H$

## Little Endian

| 103 | 102 | 101 | 100 |
|---|---|---|---|
| 15 | 25 | 29 | 11 |

## Big Endian

| 103 | 102 | 101 | 100 |
|---|---|---|---|
| 11 | 29 | 25 | 15 |

## ② M/c & AM

n bit opcode ⇒ $2^n$ operation

← Instruction →

| OPCODE | OPERAND Reference (AF) or OPERAND |
|---|---|

Instn size = 16 bit

1KB Memory ⇒ AF = 10 bit

Opcode = 16 − 10 = 6 bit

Total # operation = $2^6$ = 64

50 operation

⇓

opcode = 6 bit

$\longleftarrow$ Instruction size $\longrightarrow$

| OPCODE | $AF_1$ | $AF_2$ . . . | Immediate field |
|---|---|---|---|

AF → Reg AF

AF → Mem AF

Asking ?

Instruction Size = ?

Immediate field = bit ?

Total Memory Size by the Program = ?

Range [unsigned] = 0 to $2^n - 1$

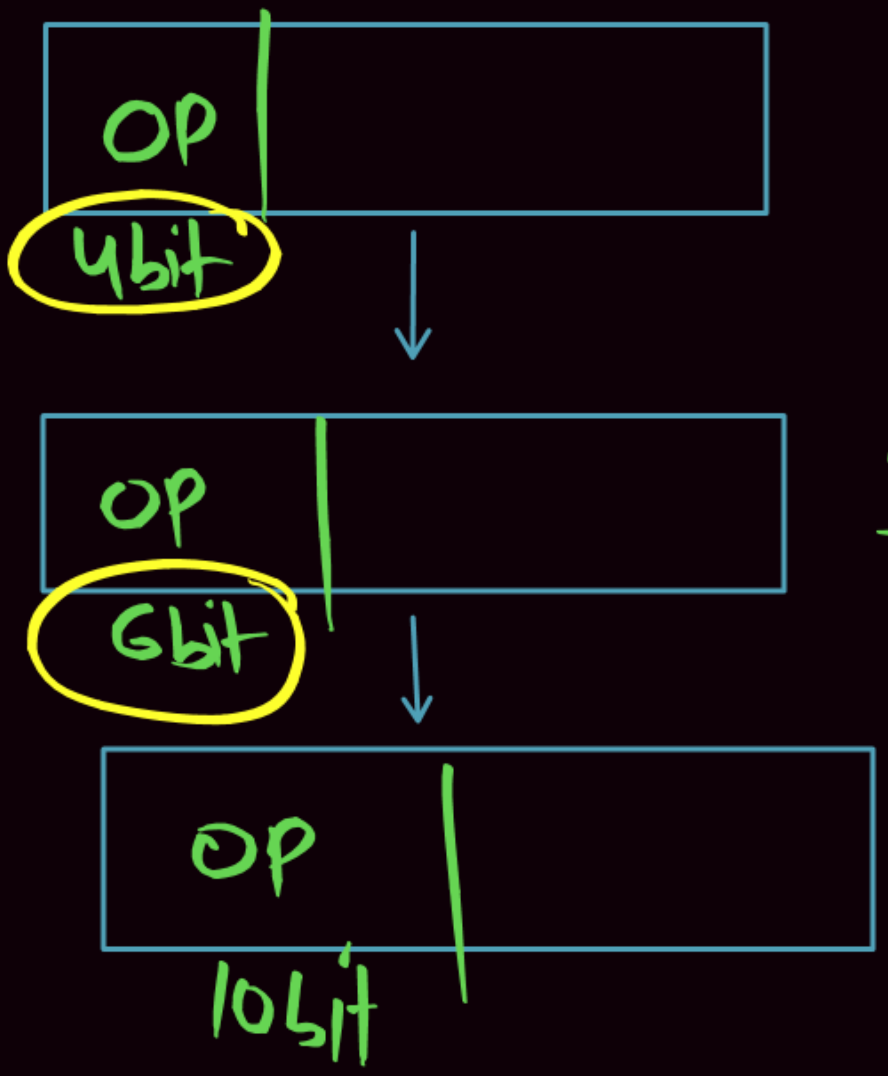
32 Register

Reg AF = 5bit

1MByte

Mem AF = 20bit

# Expand opcode Technique




1 Primitive $\Rightarrow$ Free $x$

2 Derived $\Rightarrow x \times 2^{6-4}$ Free $y$

3 (Next) Further Derived

$y \times 2^{10-6}$

# Expand Opcode Technique:

Start from Primitive Instⁿ ⇒ [Lowest | Smallest Opcode bit)

Steps

① Total # operation

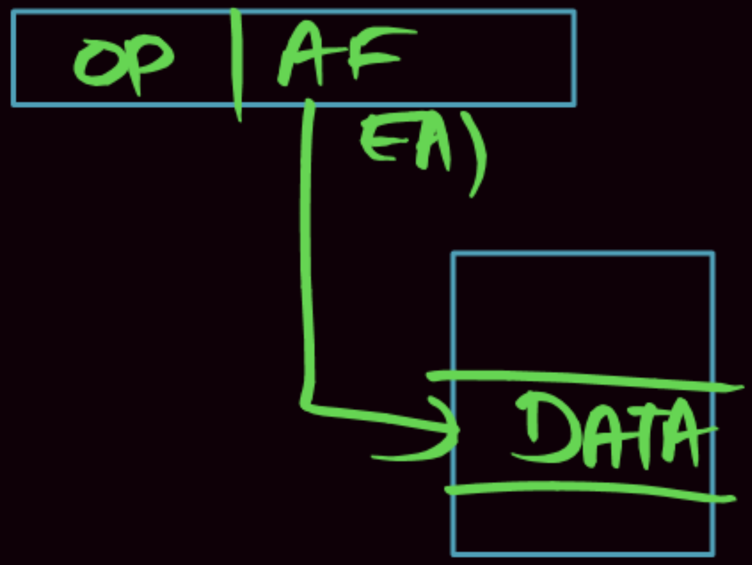
② Free OPCODE ⇒ Total − Given (in the Question)

$$\text{Derived Inst}^n\left(\begin{matrix}\text{Total}\\\text{operation}\end{matrix}\right) = \text{Free OPCODE} \times 2^{\text{Increment bit in opcode}}$$

# ADDRESSING MODE ⇒ EA [Effective Address]

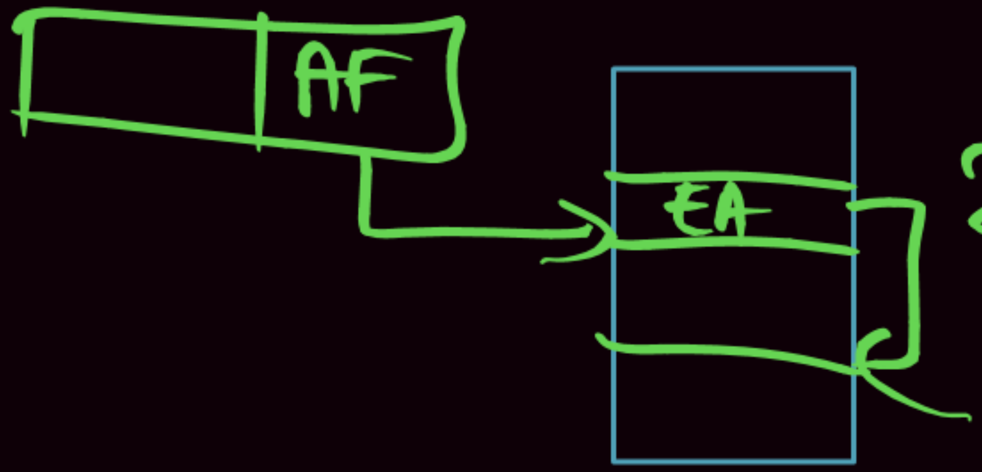
① Immediate AM


Direct | Absolute AM :




1 Mem Ref.

Indirect AM




2 mem Ref

Immediate AM → I / #

Direct AM : [ ]

Indirect AM → @ / [( )]

Register AM → Reg Name

1. Immediate AM — Constant
2. Direct AM — Variable
3. Indirect AM — Pointer
4. Index AM — Array
5. Base Reg AM — Reallocation
6. Auto Decrement/Increment — Loop.

# Disblacement AM

## PC Relative AM

$$\underset{\text{(Target Address)}}{EA} = \text{Current PC Value} + \underset{\text{(Relative Value)}}{\overset{\text{(AF)}}{\text{(OFFSET)}}}$$

R.V ⇒ +ve : Forward Jumping

R.V = −ve Backword Jumping.

2 marks GATE

Assume

i 1000

i+4 1012 — i

1016

# 3 Floating Point Representation

1 bit

| S | E | M |
|---|---|---|

S → 0 (+ve)
S → 1 (−ve)

E/BE

E or Bias Exponent

$$\boxed{E = e + bias}$$

OR

$$\boxed{BE = AE + bias}$$

M: Mantissa

Exponent = k bit

$$\boxed{bias = 2^{k-1}}$$

OR

Excess − 8

bias = 8

k : # exponent bit

$2^{k-1} = 8 \Rightarrow 2^{k-1} = 2^3$   $k-1 = 3$

$k = 4$

$bias = 2^{4-1} = 8$

E = 4 bit

$bias = 2^{4-1}$

bias = 8

$$\boxed{E = e + 8}$$

1bit

$\boxed{S \mid E \mid M}$

Explicit Repr.

$$\boxed{0 \cdot 1 \ldots\ldots \times 2^{e}}$$

$$\boxed{(-1)^{S}\ 0 \cdot M \times 2^{e}}$$

$E = e + \text{bias}$

$e = E - \text{bias}$

$$(-1)^{S}\ 0 \cdot M \times 2^{E-\text{bias}}$$

Implicit Rep.

$$\boxed{1 \cdot \text{Something} \times 2^{e}}$$

$$\boxed{(-1)^{S}\ 1 \cdot M \times 2^{e}}$$

$$(-1)^{S}\ 1 \cdot M \times 2^{E-\text{bias}}$$

# IEEE 754 Floating Point

**Single Precision**

$\leftarrow$ 32bit $\rightarrow$

| S | E | M |
|---|---|---|
| 1bit | 8bit | 23bit |

$$bias = 2^{8-1} - 1$$

$$\boxed{bias = 127}$$

**Double Precision**

$\leftarrow$ 64 bit $\rightarrow$

| S | E | m |
|---|---|---|
| 1bit | 11bit | 52bit |

$$bias = 2^{11-1} - 1$$

$$\boxed{bias = 1023}$$

# Single Precision

00000000

$E = 0$

- $M = 0$ · $+/-\ 0$
- $M \neq 0$ [Denormalized Number]

11111111

$E = 255$

- $M = 0$ Infinite
- $M \neq 0$ Nan (Not a Number)

# Double Precision

00000000

$$E = 0 \begin{cases} M = 0 & \cdot\ +/-\ 0 \\ M \neq 0 & \text{[Denormalized Number]} \end{cases}$$

---

$$E = 2047 \begin{cases} M = 0 & \text{Infinite} \\ M \neq 0 & \text{Nan (Not a Number)} \end{cases}$$

Right Alignment $= 2^{+ve}$

Left Alignm $= 2^{-ve}$

$110.1$

1 Exblicit

$0.1101 \times 2^{+3}$

2 Implict

$1.101 \times 2^{+2}$

# 4 ALU DATA Path & control Unit

## Micro operation

## Micro Program

Fetch

Direct AM

Indirect AM

Interrupt

## Working of Combuter

## MUX Role

## Control Unit

1. Hardwired CU

CS ⇒ S.O.P [Sum of Product]

$$S_5 = T_1(I_1 + I_3)$$

2. MicroProgrammed CU.

Micro Instn | CW | CDR

| Flag | Branch Condition | Control Field | (CM Address / AF / CAR / NIA) |
|---|---|---|---|

Control Signal

Decoded Format
(Horizontal μprog)

$\begin{bmatrix} 1\text{ bit} \Rightarrow 1\text{ CS} \\ N\text{ CS} \Rightarrow N\text{ bit} \end{bmatrix}$

Encoded Format
[Vertical μprog]

$n\text{ bit} = 2^n\text{ CS}$

$N\text{ CS} \Rightarrow \log_2 N\text{ bit}$

Horizontal

$8\text{ bit} \Rightarrow 8\text{ CS}$

$16\text{ CS} \Rightarrow 16\text{ bit}$

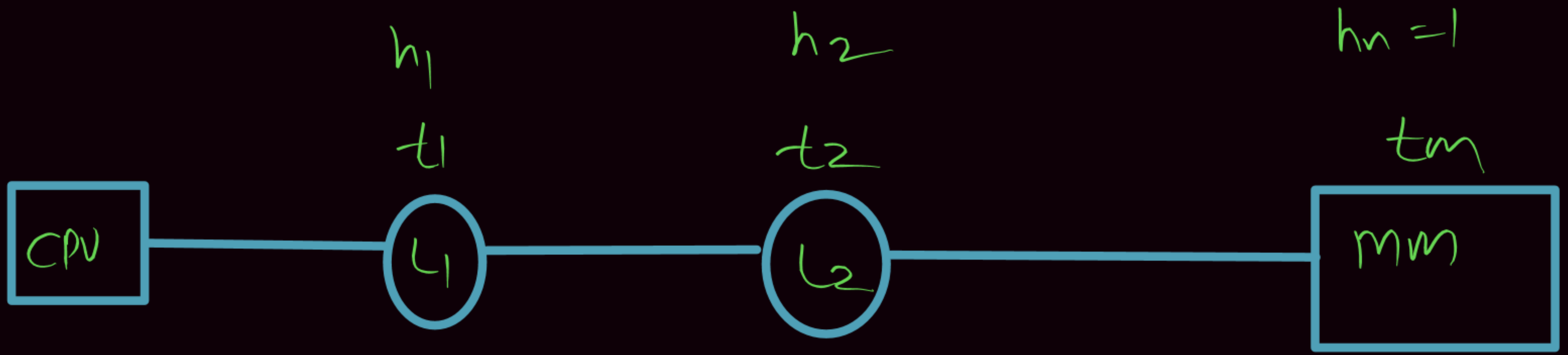
Vertical

$8\text{ bit} \Rightarrow 2^8 = 256\text{ CS}$

$16\text{ CS} \Rightarrow \lceil \log_2 16 \rceil = 4\text{ bit}$

# 5 Cache Memory

$$T_{avg} = \text{HIT Ratio} \times \text{Time taken when Cache hit} + \text{Miss Ratio} \left( \text{Time taken when Cache Miss} \right)$$



h1
h2
hn =1
t1
t2
tm
CPU
L1
L2
MM

Simultaneous

$$T_{avg} = h_1 t_1 + (1-h_1) h_2 t_2 + (1-h_1)(1-h_2) h_3 t_3$$

Hierarchical

$$T_{avg} = h_1 t_1 + (1-h_1) h_2 (t_2 + t_1) + (1-h_1)(1-h_2)(t_m + t_2 + t_1)$$

2 level CM & MM

$$T_{avg} = h t_c + (1-h)(t_m + t_c)$$

OR

$$T_{avg} = t_c + (1-h) t_m$$


Locality of Reference

1. Mapping Technique
2. Replacement Algo
3. Cache Updating Tech.
4. Multi Level Cache

## Direct Mapping

← Physical address →

| TAG | Line OFFSET | WORD OFFSET |
|---|---|---|

Word OFFSET = $\log_2$ Block size

$$\#\text{Lines} = \frac{\text{CM size}}{\text{Block Size}}$$

L.O = $\log_2 \#\text{Lines}$

$$\text{TAG} = \text{P.A} - (\text{L.O} + \text{W.O})$$

$$K \bmod N = i$$

K: MM Request
N: #CM Lines
i: Cache Address
CM Line Number

## Set Associative Mapping

← P.A →

| TAG | S.O | W.O |
|---|---|---|

$$\#\text{SET} = \frac{\#\text{Lines}}{\text{N-way}}$$

$\text{S.O} = \lceil \log_2 \#\text{SET} \rceil$

$$\text{TAG} = \text{P.A} - (\text{S.O} + \text{W.O})$$

$$K \bmod S = i$$

S: #SETS.

## Fully Associative Mapping

← P.A →

| TAG | W.O |
|---|---|

No Mapping function.

# TAG bit in Set Associative

$$\boxed{\text{TAG bits} = \text{Tag bits in Direct Mapping} + \lceil \log_2 N\text{way} \rceil}$$

## Direct Mapping

$$\#\text{Tag} = \frac{\text{MM Size}}{\text{CM Size}}$$

$$\text{Tag bits} = \lceil \log_2 \#\text{Tags} \rceil$$

# Replacement Algo

1. FIFO
2. LRU
3. Direct Mapping
4. N Way Set Associative with LRU.

(A)

① Compulsory | First Reference | Cold Start Miss

(B)

② Conflict | Collision Miss

③ Capacity Miss

15 = 5 ⇒ A

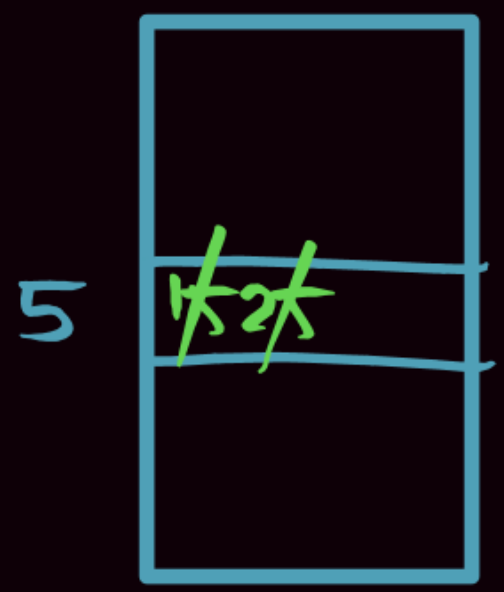
25 = 5 = (A)

15 MOD10 = 5 = M - (B)

25 M → B

15 M → (B)

25 M → B



## Cache Updating Technique

- Write Through ..
- Write Back (Extra bits Dirty bit)

## Multilevel Cache

Local Miss Rate

Global Miss Rate

6 Pipelining

$$\boxed{ET_{PIPE} = [k+(n-1)]\,tp}$$

k: # Stage | # segment
n: # Task | # Instn
tp: Each Stage Delay in Pipeline

$$\boxed{ET_{NP} = n * tn}$$

tn: Each Instn ET in Non Pipeline

$$\boxed{\underset{\text{Performance GAIN}}{S} = \frac{\text{Performance of Pipe}}{\text{Performance of Non Pipe}}}$$

$$\Rightarrow \frac{1/ET_{PIPE}}{1/ET_{NonPi}} \Rightarrow \frac{ET_{NP}}{ET_{Pipe}}$$

$$\boxed{S = \frac{n\,tn}{[k+(n-1)]\,tp}}$$

$$\boxed{S = \frac{tn}{tp}}$$

n is very large
or
n is Not given
or
Ideal case

Uniform Delay

$\boxed{tp = \text{Stage Delay}}$

if Buffer Delay

$$\boxed{tp = S.D + B.D}$$

$$\boxed{\eta = \frac{S}{K}}$$

efficiency

S : Speed up Factor

k : #Stage

Non Uniform

$tp = \max(\text{Stage Delay})$

If Buffer Delay

$$\boxed{tp = \max(\underbrace{S.D}_{\text{Stage Delay}} + \underbrace{B.D}_{\text{Buffer Delay}})}$$

**Throughput** : Rate of O/P.

n Instrn $\Rightarrow$ $[k+(n-1)]tp$ time

$$\boxed{\text{Throughput} = \frac{n}{[k+(n-1)]tp}}$$

$$\boxed{\text{Throughput} = \frac{1}{tp}}$$

n is very Large
or
Not given
or
Ideal Case

**In Pipeline**

$$\boxed{CPI = 1}$$

| | Cause | Solution |
| --- | --- | --- |
| ① Structural Hazards/Dep. → | Resource Conflict → | Renaming & Resource Replication |
| ② Data Hazards/Dep → | Result use as operand → | H/w Interlock, operand forwarding |
| ③ Control Hazards/Dep. → | Branch Instn → | Delay Branch<br>Branch Predication<br>bit Implementation<br>B.L. Buffer. |

$CPI = 1$

① Structural Dep.

② Data Dep.

Operand Forwarding

ANTI Dep : WAR

O/P Dep : WAW

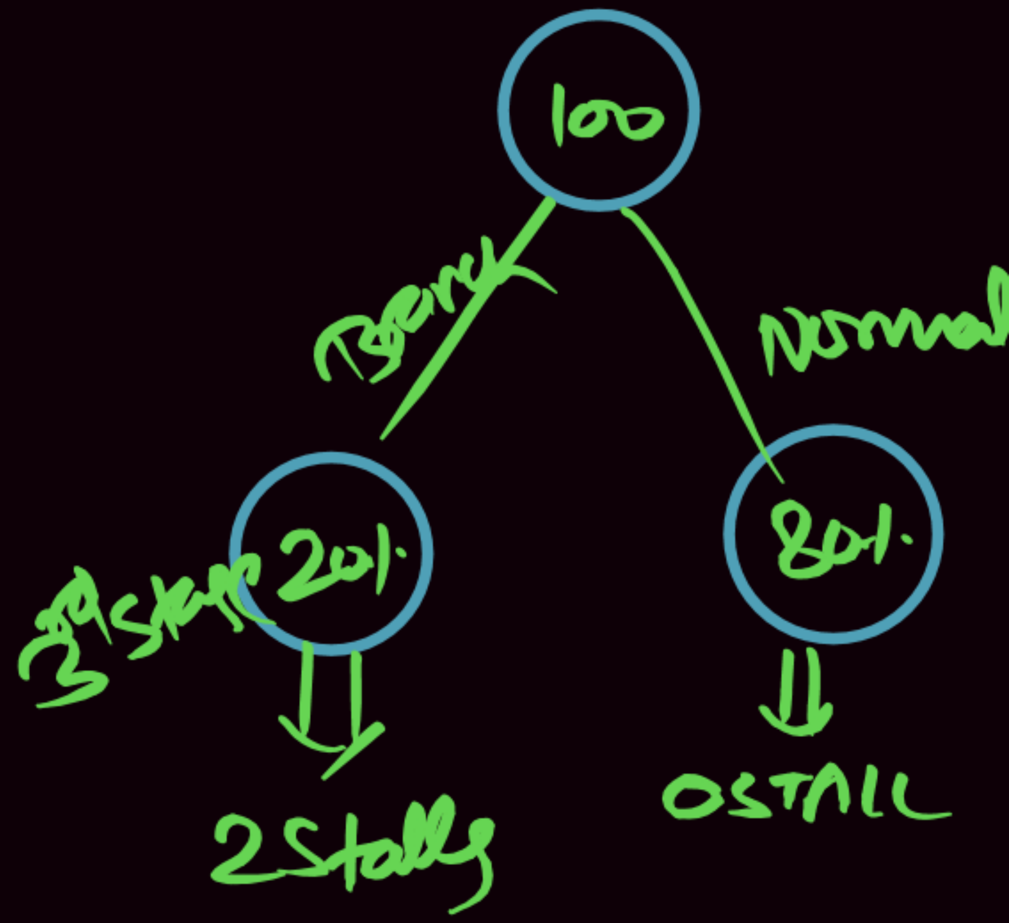
## Control Dep.

$$\boxed{\text{Avg Instr}^n\ ET = (1 + \#\text{Stalls/Instr}^n) \times \text{Cycle time}}$$

$$\boxed{S = \frac{ET_{NP}}{ET_{pipe}}}$$

When Perfectly Balance

$$S = \frac{\text{Pipeline Depth}(\#\text{Stage}) \times \cancel{\text{Cycle time}}}{(1 + \#\text{Stall/Instr}^n)\ \cancel{\text{Cycle}}}$$




$$\#\text{Stall/Instr}^n = .20 \times 2 + .80 \times 0$$

$$= 0.4$$

# 7 secondary Memory & I/O

## Disk

Platter

↳ Surface

↳ Track

↳ Sector

## Disk Capacity

$$\boxed{\text{Disk Access time} = \text{Seek time} + \text{Avg Rotational Latency} + \text{Data transfer time} + \text{Overhead (if given)}}$$

Data transfer Rate

360 RPM

360 Rotation 1 min (60 sec)

$$1 \text{ Rotation} = \frac{60}{360} = \frac{1}{6}$$

$$R.L = \frac{1}{2} \times \text{Rotation time}$$

$$\Rightarrow \frac{1}{2} \times \frac{1}{6} = \frac{1}{12} \text{ sec}$$

# sector/per track = 256 Each Capacity = 8 Byte.

360 RPM

1 Rotation = $\frac{1}{6}$ sec

1 Track Capacity = $256 \times 8B \Rightarrow 2^8 \times 2^3 = 2^{11}B = 2kB$

## Data Transfer time

In 1 Rotation ⇒ 1 Track.

2kB ——— $\frac{1}{6}$ sec.

1 Byte ——— $\frac{1}{6 \times 2k}$ sec

~~x Byte~~ ⇒ $\boxed{\frac{x \text{ Byte}}{6 \times 2k\text{Byte}} \text{ Sec}}$

## Data transfer Rate

In 1 Sec How Much Data transfered.

2kB ——— $\frac{1}{6}$ sec

$\frac{1}{6}$ sec ——— 2kB

In 1 Sec ——— 2kB × 6

= 12 kBps

## I/O Org

1. **Program Driven** : No High Speed Interface is Used

   ↳ CPU time Depend Speed of I/O Device

2. **Interrupt Driven**

   ↳ High speed Interface Chip is Used.

   CPU time ⇒ Depends On High Speed Interface Chip

③ DMA (Direct Memory Access)

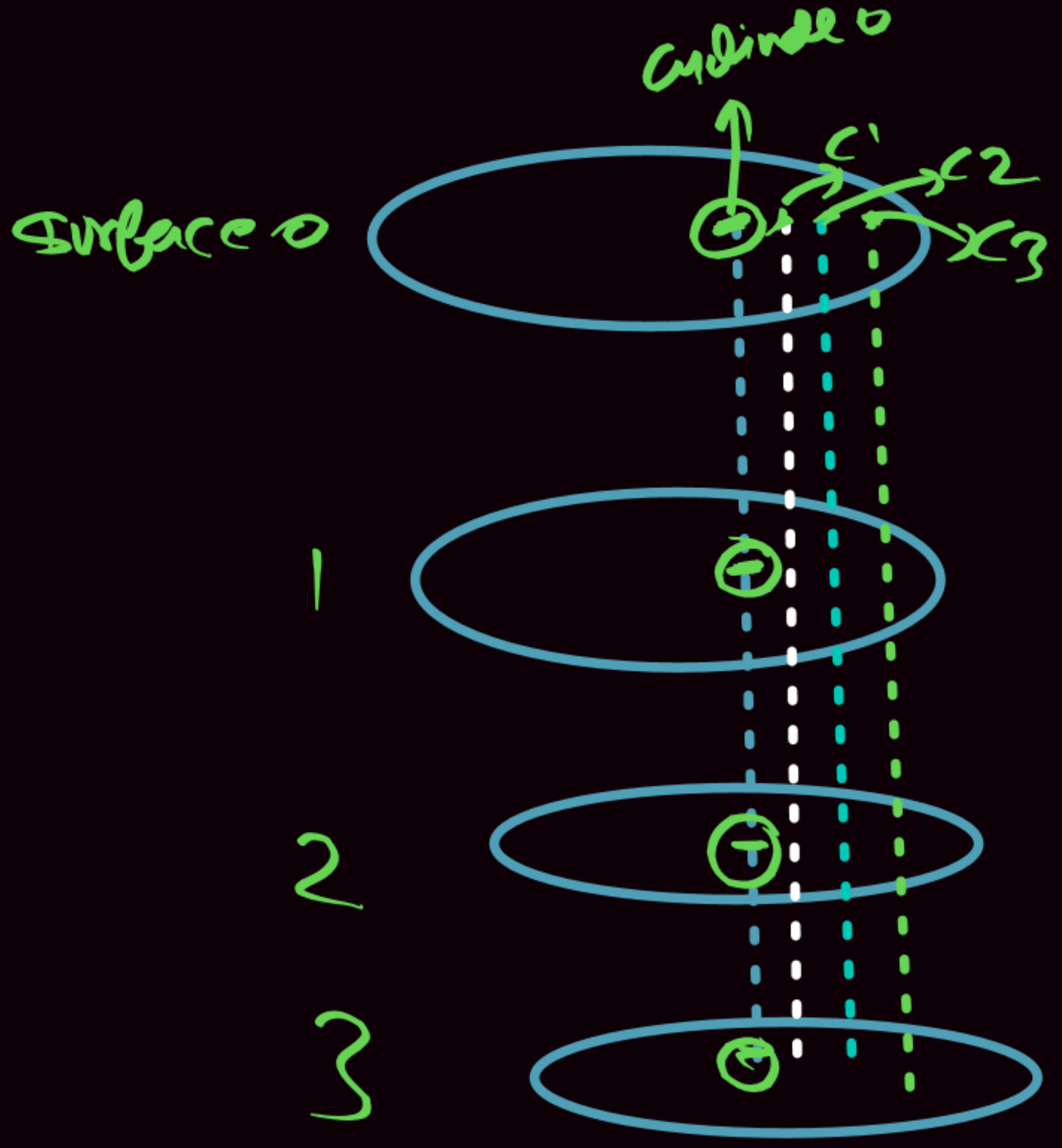
- → Highest Priority.
- → Bulk Amount of Data transfer without the involvement of CPU.
- → Virtual Memory Concept

DMA Controller

Count Register ⇒ # Byte/Words Transferred.

$\frac{Poe}{Tr}$ % Busy =

# Disk Addressing : <c, h, s>




c : cylinder No

h : Surface No

S : Sector.

C: cylinder
h: surface
S: sector

$\langle C, h, S \rangle$

$\langle 400, 16, 29 \rangle$

$$\boxed{\text{Sector Number} = S + ST * h + \underbrace{ST * TC}_{SC} * C}$$

ST: #Sector Per Track

TC: #Track Per Cylinder

SC: #Sector Per Cylinder

# DBMS

1. FD & Normal Form
2. Transaction
3. ER Model & Foreign key
4. Query lang.
5. File org & Indexing.

## ① FD & NF

(i) RDBMS → Schema, Arity(Degree), Cardinality, Instance

(ii) FD Concept & type

$$\boxed{\text{If } t_1.x = t_2.x \text{ then } t_1.y = t_2.y \text{ must be same}}$$

(iii) Attribute closure

(iv) Key Concept

(v) Candidate key → Prime/Key Attribute

Non Prime/Non key Attribute

(vi) Membership set

Equality b/w 2 FD.

## (6) Minimal Cover

- Step1 :: R.H.S : Single Attribute
- Step2 L.H.S ⇒ Redundant Attribute
- Step3 Redundant FD.

## (7) Lossless Join

$R_1 \cup R_2 = R$

$R \to R_1$

$R \to R_2$

(1) Concept

(2) Binary Method

(3) CHASE TEST

$[R_1 \cap R_2]^+ \to R_1$

OR

$[R_1 \cap R_2]^+ \to R_2$

# Dependency Preserving

| $R_1$ | $R_2$ | $R_3$ |
|---|---|---|
| N.T FD | N.T FD | N.T FD |

$F_1 \cup F_2 \cup \ldots \cup F_n \equiv F$

Dep Preserving

Via

## 2NF

Proper Subset of CK → Non key Attribute

Violation of 2NF

Not in 2NF

② all CK are simple CK then R is in 2NF

## 3NF

Every Non Trivial FD

$X \rightarrow y$ is in 3NF

either

X: Super key

OR

y: key/Prime Attribute

② all attribute are Prime/key Attribute then R is in 3NF

## BCNF

$X \rightarrow y$ is in BCNF

X: Super key.

② If all keys are simple CK & R is in 3NF then R is in BCNF

③ Binary Relation (Relation with 2 Attribute) is in BCNF.

## 2 Transaction Management

A C I D

## Serializablity

### I. Conflict Serializable

- ↳ ① Conflict Equal to Any Serial schedule
- ↳ ② Convert into Serial Schedule by Swapping of Non Conflict Instn
- ↳ ③ Testing (Precedence Graph Method)

CNC [Cycle Not Conflict]

### View Serializable

① Initial Read

② Final Write

③ Write-Read Sequence (Updated Read)

Same on Each Data Item in Schedule S & S'.

## Problem bcz of Concurrent Execution

1. WR | Dirty | uncommit Read Problem
2. RW Problem
3. WW Problem | Lost Update Problem
4. Phantom Tuple Problem.

# Recoverability

- Recoverable
- Cascadeless
- Strict Recoverable

| $T_1$ | $T_2$ |
|---|---|
| W(A) | |
| | R(A) |
| C/R | |
| | •Commit |

Recoverable Schedule

Not free from:
- WR (Uncommitted Read)
- WW (Lost Update Problem)
- Cascading Rollback

RW Problem.

| $T_1$ | $T_2$ |
|---|---|
| W(A) | |
| C/R | |
| | R(A) |

Cascadeless Schedule

No Uncommitted (Dirty) Read
No Cascading Rollback
But WW (Lost update) Problem
RW Problem

| $T_1$ | $T_2$ |
|---|---|
| W(A) | |
| C/R | |
| | R(A) / W(A) |

Strict-@
Strict Recoverable

No Uncommitted Read
No Cascading Rollback
No Lost Update (No WW)
Only RW Problem.

Serializability

Recoverability.

Lock: $\frac{\text{Last Lock Position}}{\text{First Unlock}}$

Lock $\Rightarrow$ 2PL → Growing Phase (Acquire Lock)
→ Shrinking Phase (Release lock)

Strict 2PL (2PL + All X Lock Realease After Commit | Rollback)

Rigurious 2PL 2PL + All Locks [S & X] Relase After C/R.

**2PL** : Ensure Conflict Serializability

- Irrecoverable (Not Recoverability)
- Deadlock
- Starvation.

| $T_1$ | $T_2$ |
| --- | --- |
| X(A) | |
| C/R | |
| U(A) | |
| | S(A)/X(A) |

**Strict 2PL** :

- Ensure Conflict Serializability
- Ensure Recoverable, Cascadeless & Strict.
- Deadlock
- Starvation.

Conservative 2PL :

- Ensure Conflict Serializability
- Ensure Recoverable, Cascadeless & Strict.
- No Deadlock.
- Starvation.

# Time Stamp Protocal

$T_1 \rightarrow T_2$ Time Stamp order Same as ALL Conflict operation Pair order.

| 10 $T_1$ | 20 $T_2$ |
|---|---|
| W(A) | |
| | R(A) |
| R(B) | |
| | W(B) |

YES its TSP allowed

| 10 $T_1$ | 20 $T_2$ |
|---|---|
| R(A) | |
| | W(A) |
| | W(B) |
| R(B) | |

$T_1 \rightarrow T_2$

$W_2(B) - R_1(B)$

$T_2 \rightarrow T_1$

Not TSP

# Thomas Write Rule (View Serializability)

W → W (obselet write Ignored)

| 10 $T_1$ | 20 $T_2$ |
|---|---|
| R(A) | |
| | W(A) |
| | W(B) |
| W(B) | |

$T_1 \rightarrow T_2$

$W_2(B) - W_1(B)$ Not allowed under TSP.

But allowed under Thomas Write Rule.

## 3 ER Model & Foreign key :


## ER model

## ER to RDBMS Conversion

1 to Many ⇒ 2 Table $E_1(\underline{A}BC)$ & $RE_2(\underline{D}EF\ \overset{FK}{\underline{A}})$ & 1 FK

Many to 1 ⇒ 2 Table $E_1R(\underline{A}BC\ \overset{FK}{D})$ & $E_2(DEF)$ & 1 FK

1 to 1 ⇒ 2 Table $E_1R(\underline{A}BCD)$ & $E_2(DEF)$ (or) $E_1(ABC)$ & $E_2R(DEF\underline{A})$ & 1 FK

M to N. ⇒ 3 Table $E_1(\underline{A}BC)$ $R(\underline{AD})$ $E_2(\underline{D}EF)$

& 2 F.K

## ER model


## ER to RDBMS Conversion

1 to Many ⇒ 2

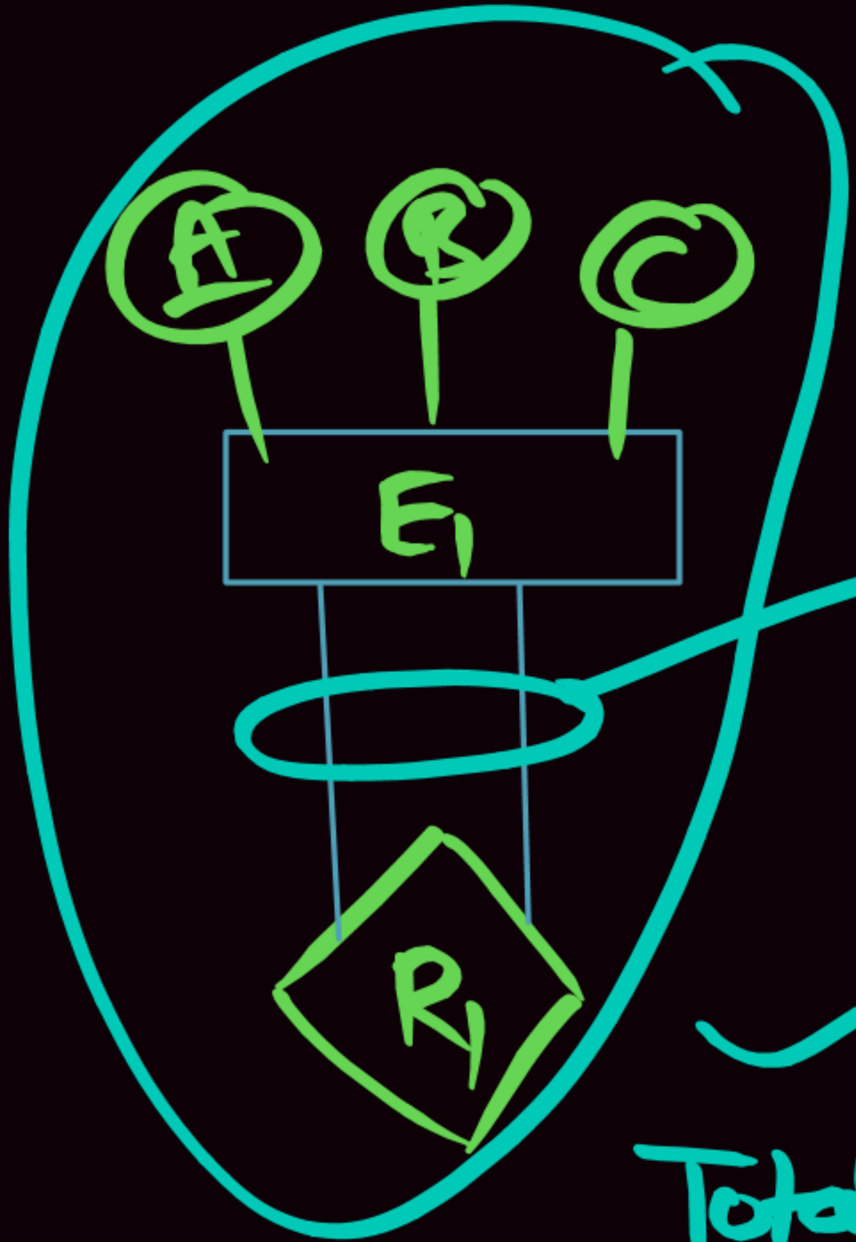
Many to 1 ⇒ ?

1 to 1 ⇒

M to N. ⇒

1 Table

4)



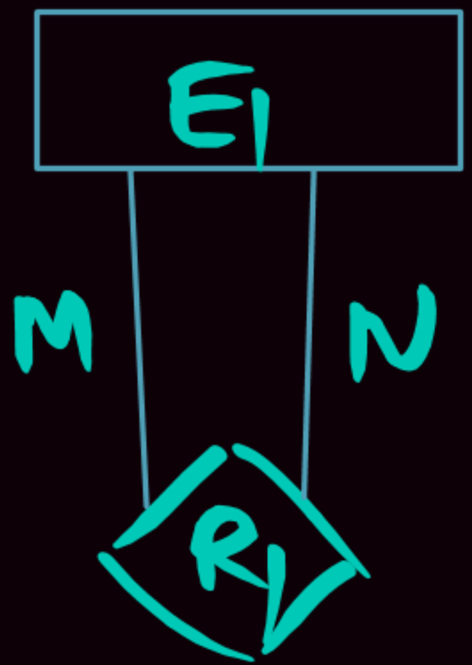

1 : M

M : 1

1 : 1

Total 1 table

$E_1R_1(\underline{A}\,B\,C\;A_1)$ →FK




2 Table

$E_1$

$R_1$

Foreign key :- (Child table)

→ Referencing Relation

Referenced (Parent Table)

Note F.K Contain Duplicates & NULL Value.

Note The Value Present in FK Must be Present in P.K of the Referenced Relation.

1. ON DELETE NO ACTION ⇒ Deletion Not allowed
2. ON DELETE CASCADE ⇒ Cascadelly
3. ON DELETE SET NULL.

## (4) Query Language

↳ R.A (Basic operations)
(Derived operator)

## Basic operator

1. Selection [$\sigma$]
2. Projection [$\pi$]
3. Union ($\cup$)
4. CROSS Product [$\times$]
5. Rename [$\rho$]
6. Minus ($-$)

## Derived operator

1. Intersection $\boxed{R \cap S = R - (R - S)}$
2. JOIN $\rightarrow$ Outer Join $\Downarrow$ Dangling Tuple.
3. Division

$$\frac{R(AB)}{S(B)} = \pi_A(R) - \pi_A\left[\pi_A(R) \times \pi_B(S) - R\right]$$

SQL : Multiset (Bag) {1, 1, 2, 2, 2, 3, 3}

## Aggregate operator

1. Count
2. MIN
3. MAX
4. SUM
5. AVG

1. Aggregate Operator first ~~Discard~~ Ignore the NULL Value.
2. Arithmatic operation with NULL give Result NULL

$$\boxed{NULL + 50 = NULL}$$

IN | NOT IN

ANY (MIN)

ALL (MAX)

EXISTS & NOT EXIST

⇓ True ⇓ Inner Query Non Empty

⇓ True ⇓ Inner Query Result ⇓ Empty

TRC

T | P(T)

DMGT

↳ First Logic

## 5 File org & Indexing.

Unspanned org. ⇒ Record belongs to Particular Block

$$\text{Block Factor} = \left\lfloor \frac{\text{Block Size}}{\text{Record Size}} \right\rfloor$$

B : Data Block

Ordered File

Avg Cost $\lceil \log_2 B \rceil$

Unordered File

Avg Cost $= B/2$

Worst $= B$

# Indexing

| Search key | Block Pointer. |
| --- | --- |

One Index Record Size = Size of key + Bp.

Index File Block Size $\underset{\text{same}}{=}$ Data File Block Size.

Index ordered File.

$B_i$ = # Index Block

Cost in Index Block : $\lceil \log_2 B_i \rceil$

To Fetch Record Using Index File = $\log_2 B_i + 1$

① Dense Index

\# Index entries = # DB Record

② Sparse Index

\# Index entries = # DB Block

1. P.I (P.K + ordered File)
2. C.I (Nonkey + ordered File)
3. S.I (~~Nonkey~~ / Cand key + Unordered File)

$$\boxed{\text{Avg \# Block Access (In MultiLevel Index)} = n+1}$$

n: # Level

$B_P$

## B Tree

ORDER : $P$

$$\boxed{P * B_P + (P-1)[\text{key} + R_P] \le \text{Block size}}$$

Min $B_P$ : $\lceil P/2 \rceil$

Max $B_P$ : $P$

Min key $= \lceil P/2 \rceil - 1$

Max keys $= P - 1$

## B+ Tree

### Internal Node

| $B_1$ | $k_1$ | $B_2$ | $k_2$ | . . . | $B_{P-1}$ | $k_{P-1}$ | $B_P$ |
|---|---|---|---|---|---|---|---|

$$\boxed{P * B_P + (P-1)\,\text{key} \le \text{Block size}}$$

### Leaf Node

$$\boxed{(P-1)[\text{key} + R_P] + 1 B_P \le \text{Block Size}}$$

OR

$$\boxed{P(\text{key} + R_P) + 1 B_P \le \text{Block Size}}$$

BEST Of Luck.