Welcome to ACE Engineering Academy - online live class

Subject: **Computer Organization and Architecture**

Faculty: **Y.V. Ramaiah** 9866339106

Subject

Computer organization & Architecture

Chapters (Topics)

I. Computer Arithmetic

II. Memory Organization

III. Secondary Memories

IV. Basic processor organization and Design

V. Pipeline organization

VI. Control unit Design

VII. IO Organization

Memory Organization

- → Introduction
- → Memory Basics
- → Memory Classification
- → Memory Size Expansion
- → Primary Memory
- → Secondary Memory
- → ROM and its design
- → RAM and its design
- → Memory Hierarchy
- → Cache Memory
- → Mapping Technics
- → Different misses occurred in cache
- → Different block replacement techniques
- → Tag directory design
- → Associative Memory

Q. A processor can support a maximum memory of 4GB, where the memory is word-addressable (a word consists of two bytes). The size of the address bus of the processor is at least ____31____ bits. 2016

Max size of Mem. to be Supported = 4G x 8 bits.
$= 2^{32} \times 8$ bits.

But Memory is word Addressable. word size = 2 Bytes

No. of words in memory = $\frac{4GB}{2B} = \frac{2G}{2}$ words.
$= 2G$ words $= 2^{31}$

Q. The main disadvantage of DRAM over SRAM is _____.

(a) High package density

(b) Costly

(c) External memory refresh logic is required

(d) High power consumption

Q. A function table is required in very large numbers. The memory most suitable for this purpose would be

(a) ROM

(b) RAM

(c) EPROM

(d) EAPROM

Q. When the power supply of a ROM is switched off, its contents

(a) Become all zero's ✗

(b) Become all one's ✗

(c) Remain same ✓

(d) Are unpredictable

Byte Addressable Memory:-

In this, each memory location is used to store only one Byte

let word size = 32 bit (4 Bytes)

then Byte Addressable Memory requires 4 memory locations to store one word

Word Addressable Memory:-

Each memory location is used to store one word (whatever the word size)

let word size = 4 Bytes, only one memory location (Register) is used to store 4 Bytes

→ Electronic memories are designed with electronic components like diode, BJT/FET and IC, to operate electronic memory, electromechanical devices (like Head and motor) are not required

Electronic Memory is faster than Magnetic and optical memory

CPU directly communicates with Electronic Memory, hence all electronic Memories are known as primary Memory

Generally Electronic Memory Speed almost equal to the processor speed.

Secondary Memories:- All Magnetic and optical Memories are operated with Head and Motor, these devices are operated in m.s. time only.
DMA Controller / Io processor is used to transfer the Data b/w secondary memories and main Mem.

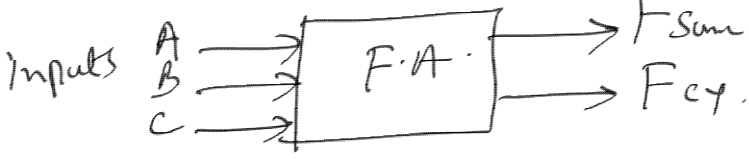
Electronic ROM design:- Rom is the the combination of Decoder and OR gates.
One OR gate is used to deliver only one data bit i.e. No. of OR gates required = no. of Data bits

To design $2^n \times m$ bit size Rom,
size of Decoder needed $(n * 2^n)$ size Decoder
and no. of OR gates required = m

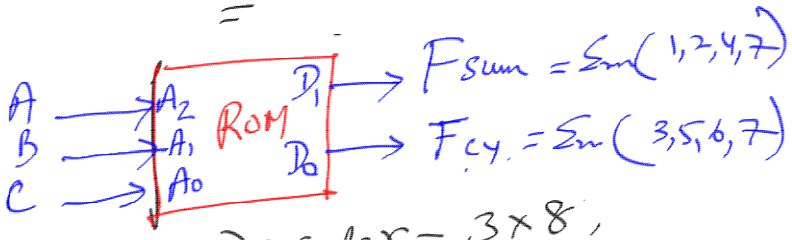
→ Generally ROMs are used to implement function tables.

→ In this process, functional inputs are applied to the ROM address and functional outputs are taken from Data lines.

Implement Full Adder with ROM.

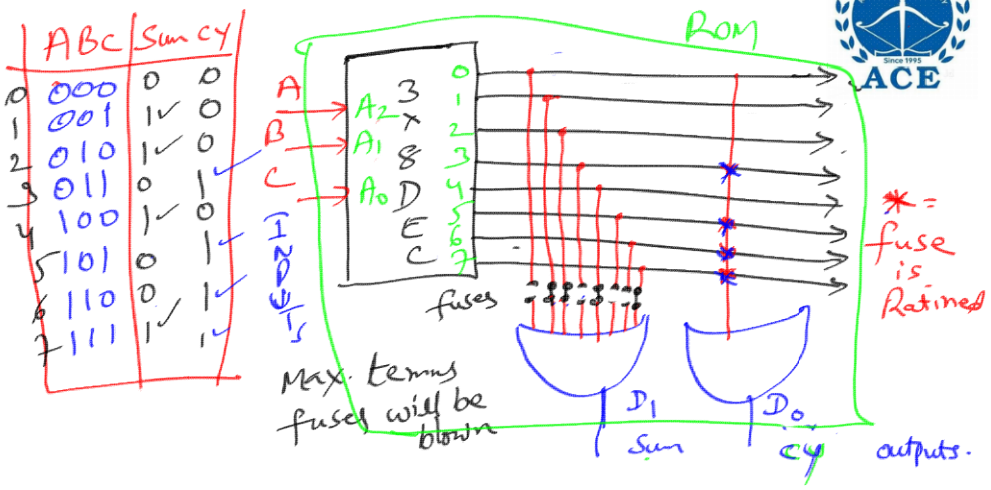


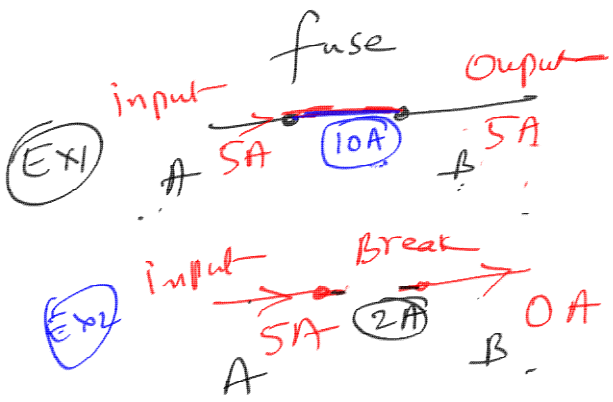
=



Size of the Decoder = 3 × 8,

No. of OR gates = 2





ROM's programming according to the Table.

Programming is the process of blowing the fuses by sending Higher Current.

Special Applications of Rom

→ Roms are used for Storing the Result of Arithmetical operations.

→ In this process; input operands are applied to Address and their Result is taken from Data

Ex: Calculator

Inputs Result

Ex: $4 \times 5 = 20$

$100101 : 101.00_2$

Address stored at Data

Calculator brings pre-defined values.

Design a ROM for 2 bit multiplication

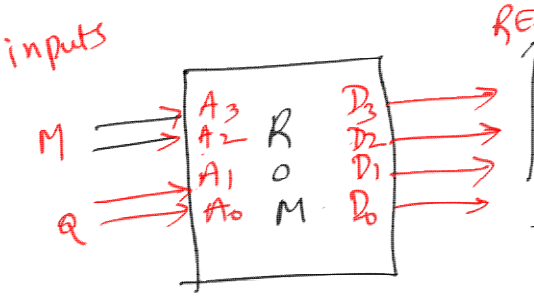
Each input operand size = 2 bits.

M = multiplicand
Q = multiplier

M * Q = Product
(2) (2) (4) bit

Address

After multiplying 2 no. of n bit data; the longest Result size = 2n bits.



Address size = 4bit
Decoder size = 4 x 2^4
No. of OR gates = 4

$F_{D_3}(A_3A_2A_1A_0) = \Sigma m(15)$

$F_{D_2}(A_3A_2A_1A_0) = \Sigma m(10,11,14)$

$F_{D_1}(A_3A_2A_1A_0) = \Sigma m(6,7,9,11,13,14)$

$F_{D_0}(A_3A_2A_1A_0) = \Sigma m(5,7,13,15)$

Size of Decoder needed $= 4 \times 2^4$

ROM design for (n) bit multiplication

(M) Multiplicand = 'n' bit

(Q) Multiplier = n bit

One operand size = n bit

Address Size = 2n bit

Max. no. of Addressable Regs = 2^{2n}

Longest Result size = 2n bits

No. of OR gates requires = No. of Data lines = 2n

Size of Decoder = $2n * 2^{2n}$.

ROM size Required for 'n' bit multiplication :-

Address size = 2n bit

Longest Result size = 2n bit (Data)

$2^{2n} * 2n$ bits.

NAT

Q) The size of ROM required for 8 bit multiplication is 128 KiloBytes.

$1K = 2^{10}$

$2^{16} * 16$ bit

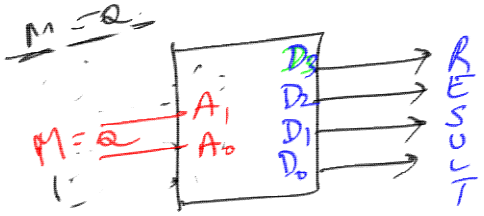
$64K * 2B = 128K$ Byte

$x \times x$ ROM design for Squarer operation

Both Multiplicand and Multiplier are Same

let Multiplicand size is 'n' bits, the longest Result size = 2n bits

ROM design for 2 bit squarer



A_3A_2	A_1A_0	$P_3D_2D_1D_0$	Result
0 0	0 0	0 0 0 0	MR_0
0 1	0 1	0 0 0 1	MR_1
1 0	1 0	0 1 0 0	MR_2
1 1	1 1	1 0 0 1	MR_3

ROM Size $2^2 \times 4$ bits

Address size = 2 bit, Result size = 4 bit

Decoder size = 2×4

No. of OR gates = 4

For 'n' bit Squarer operation

Rom size Required

Operand size = n bit = Address size.

Longest Result size = 2n bits = Data Bus size

No. of OR gates = 2^n

Size of Decoder

$n \times 2^n$

Rom size

$2^n * 2n$ bits

NAT

Q) The size of the Rom required for 6 bit Squarer is 768 bits

$2^6 * 12$ bits

$= 64 * 12$ bits

$= 768$

2013 GATE

9) Size of Rom required for 4 bit Squarer is ____ Bytes.

$2^4 \times 8$ bit = 16 Bytes.

Design of Electronic RAM

→ All Electronic RAMS are Volatile Memories

→ Electronic RAM can be designed in 2 ways. (i) Dynamic RAM

(ii) Static RAM.

These can be generally designed with BJT / FETs

BJT = Bipolar Junction Transistor

FET: Field Effect Transistor

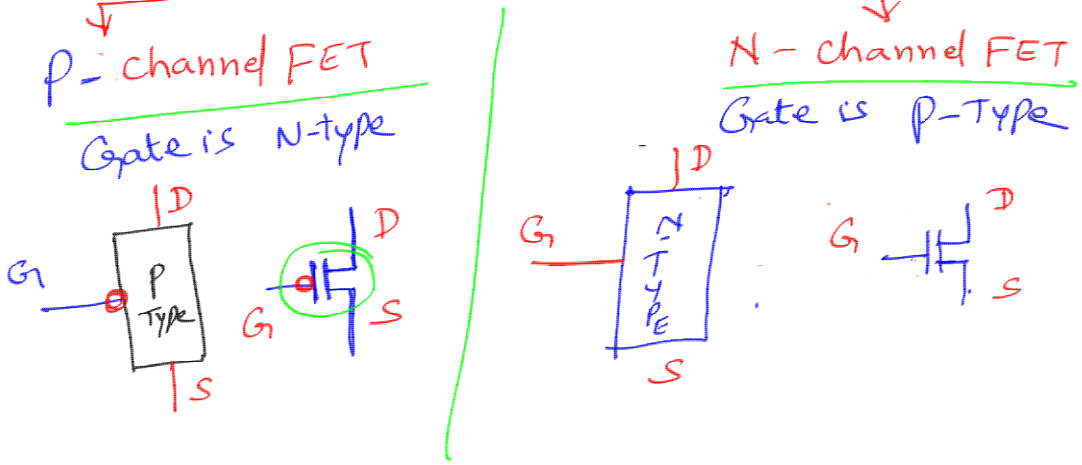
→ BJT is faster and costlier than FET.

→ The memories in Super fast Computers are designed with BJT and the memories in General purpose Systems are designed with FETs.

SNo	parameter	'D' RAM	'S' RAM
1	No of FETs needed to store one bit	One	Six
2	Cost/bit	Cheaper	Costliest
3	Capacitor	is needed	is not Needed
4	Refresh logic circuit	is needed	is not Needed
5	Power Consumption	Smaller	Very High
6	Bit Density	more	less
7	Speed	Moderate	7 to 10 times to 'D' RAM
8	Application in Digital Computer	Main Mem Ex: 4GB, 8GB	Cache mem

Bit density = No. of bits in one unit Area

FET Operation

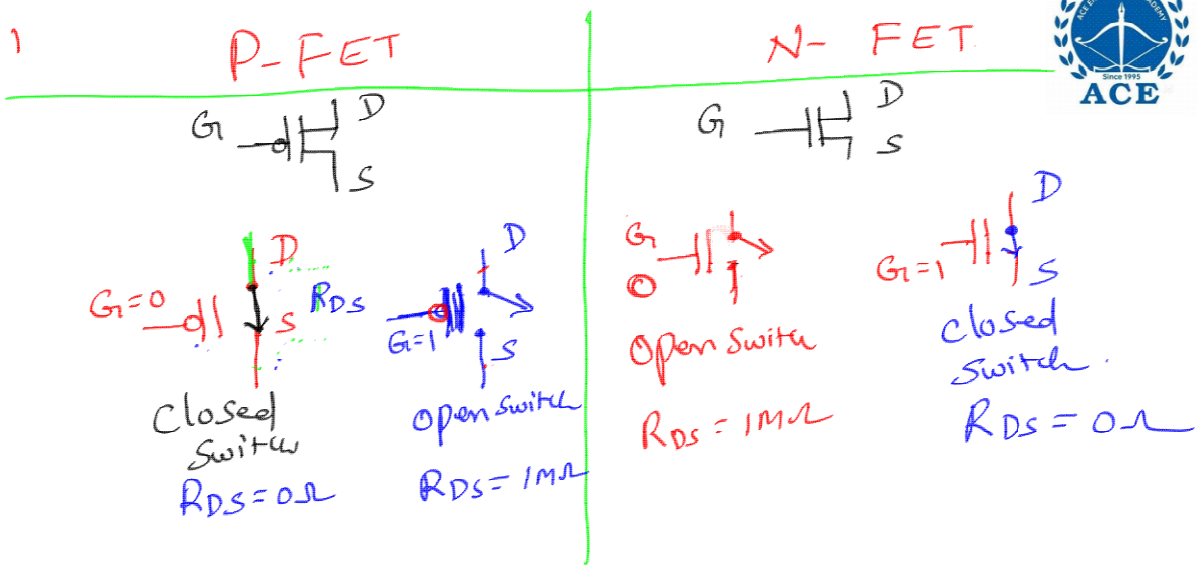


FET is havin 3 terminals

GATE is Control inpu

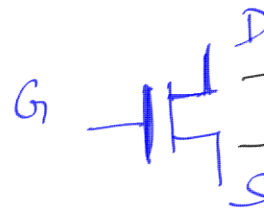
Drain

Source.



bit 0 = 0 to 0.6 V

bit 1 = 2.4 V to 5 V.

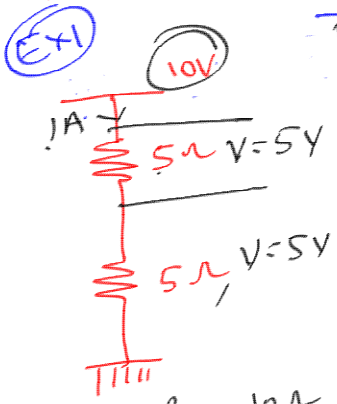


R_{DS} = Resistance b/w Drain and Source

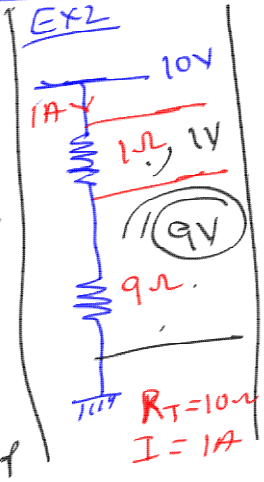
$V = IR$

VV VIMP

Ex1

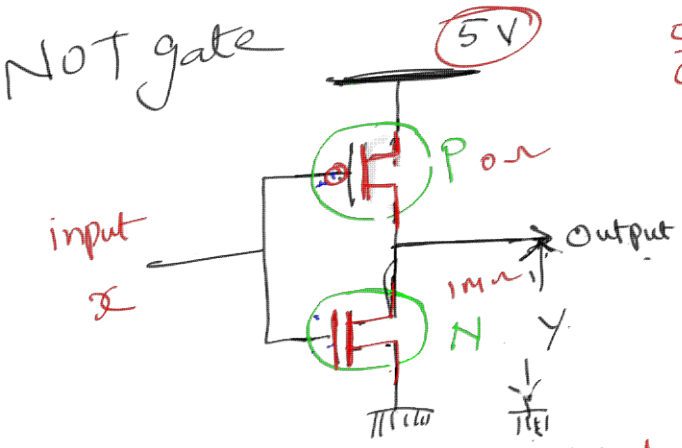


Ex2



When (2) Resistors are connected in series then max Voltage is available across High Resistor Value.

NOT gate



5V = bit 1
0V = bit 0

x	P	N	Y
0	ON	OFF	1
1	OFF	ON	0

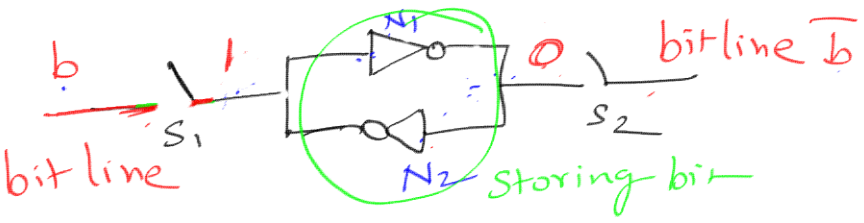
Two FETs are required to design a NOT gate.

x	Y
0	1
1	0

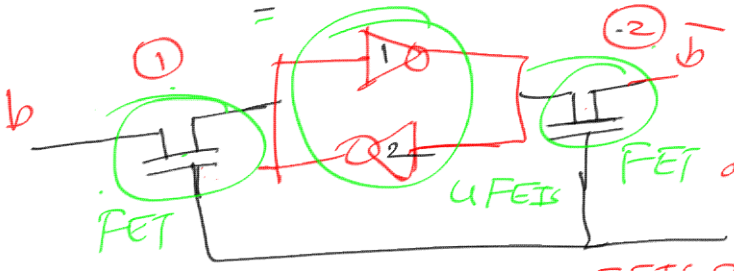
NOT gate

'S' RAM design for storing one bit

→ It requires 6 FETs; in these 6 FETs 2 FETs are used for switching and 4 FETs are used for bit voltage storing



'S' RAM is also known as Flipflop



① and ② FETs are used for switching

4 FETs are needed to design Two NOT gates

FET and BJT are used as High Speed Electronic Switch.

Welcome to ACE Engineering Academy - online live class

Subject: **Computer Organization and Architecture**

Faculty: **Y.V. Ramaiah** 9866339106

Subject

Computer organization & Architecture

Chapters (Topics)

I. Computer Arithmetic

II. Memory Organization

III. Secondary Memories

IV. Basic processor organization and Design

V. Pipeline organization

VI. Control unit Design

VII. IO Organization

Memory Organization

- → Introduction
- → Memory Basics
- → Memory Classification
- → Memory Size Expansion
- → Primary Memory
- → Secondary Memory
- → ROM and its design
- → RAM and its design
- → Memory Hierarchy
- → Cache Memory
- → Mapping Technics
- → Different misses occurred in cache
- → Different block replacement techniques
- → Tag directory design
- → Associative Memory

Q. A 32-bit wide main memory unit with a capacity of 1 GB is built using 256M × 4 - bit DRAM chips. The number of rows of memory cells in the DRAM chip is 2^{14}. The time taken to perform one refresh operation is 50 nanoseconds. The refresh period is 2 milliseconds. The percentage (rounded to the closest integer) of the time available for performing the memory read/ write operations in the main memory unit is (59 to 60).

Chip refresh time = 16K = 16384 * 50ns = 0.8192 msec

RD/WR time = (2 − 0.8192) ms

% time RD/WR = 1.1808 ms / 2 ms = 59.04 = (59)

≈ 0.8 use, ≈ 60%

Q. Assume that for a certain processor, a read request takes 50 nanoseconds on a cache miss and 5 nanoseconds on a cache hit. Suppose while running a program, it was observed that 80% of the processor's read requests result in a cache hit. The average read access time in nanoseconds is __14__.

$T_{cm} = 5ns$

$T_{mm} = 50ns$

$H = 0.8$

$n = 50$, Hit time $= (40 \times 5) = 200ns$

Miss time $= 500ns$ $= 700ns$

AMAT $= (0.8 \times 5ns) + (0.2 * 50)ns$

$= 14ns$

50 instructions $= 700ns$.

$AMAT = \frac{700ns}{50}$

$= 14$

2022 GATE

Q. A cache memory that has a hit rate of 0.8 has an access latency 10 ns and miss penalty 100 ns. An optimization is done on the cache to reduce the miss rate. However, the optimization results in an increase of cache access latency to 15 ns, whereas the miss penalty is not affected. The minimum hit rate (rounded off to two decimal places) needed after the optimization such that it should not increase the average memory access time is _____.

$H = 0.8, T_{CM} = 10ns$

$T_{MM} = 100$

$AMAT = (0.8 \times 10ns) + (0.2 * 100)ns = 28ns$

After optimization

$T_{CM} = 15ns$

$T_{MM} = 100ns$

$AMAT \leq 28ns$

$H = ?$

$AMAT \leq H_{CM} * 15ns + (1 - H_{CM}) * 100ns$

$28 \leq$

$28 \leq 15 H_{CM} + 100 - 100 H_{CM}$

$+72 \leq +85 H_{CM}$

$85 H_{CM} \geq 72$

$H_{CM} \geq \frac{72}{85} = 0.847$

$= 0.85$ ✓

MSQ 8) In the give below, with reference to Static RAM for storing one bit, the correct statement(s) is/are

~~X~~ a) Bit density is more than 'D' RAM

~~b)~~ Total 6 FETs / BJTs are used

~~c)~~ Power Consumption is High Compared to DRAM

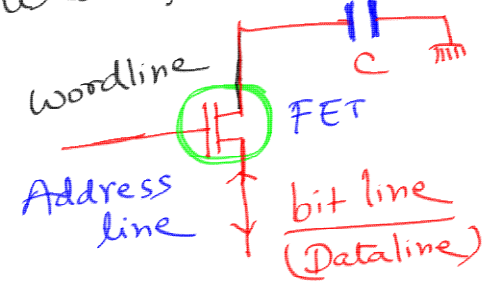
~~d)~~ Memory Refresh logic circuit is not needed

b, c and d

One bit storage in 'DRAM

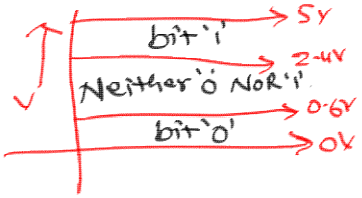
→ It requires one FET and one capacitor

→ FET is used for switching and Capacitor is used for bit storing



bit 1 = 2.4 to 5V

bit 0 = 0 to 0.6V



→ Due to the capacitor nature, the bit voltage across the capacitor gets discharge gradually, when it falls down to 2.4 Volts then bit '1' becomes bit '0' which may cause lot of disturbance in the program.

→ To prevent this discharge, one separate Memory refresh logic circuit is used for each 'D' RAM chip, this circuit refreshes all the capacitors which are used to store bit '1' will be recharged

→ one Refresh operation refreshes all the capacitors available in one Row

→ Since each 'D' RAM chip is having separate Refresh circuit, Refreshing operation is performed in parallel, one chip refresh time is sufficient to Refresh all the chips available in the System

→ Total no of Refresh operations needed to Refresh one chip = No of Rows

→ chip refresh time = No of Rows * one Refresh time

(EX) In a 1K x 16 bit 'D' RAM chip, one Refresh operation takes 10 n.s, total amount of time required to Refresh the entire chip is 10.24 micro second

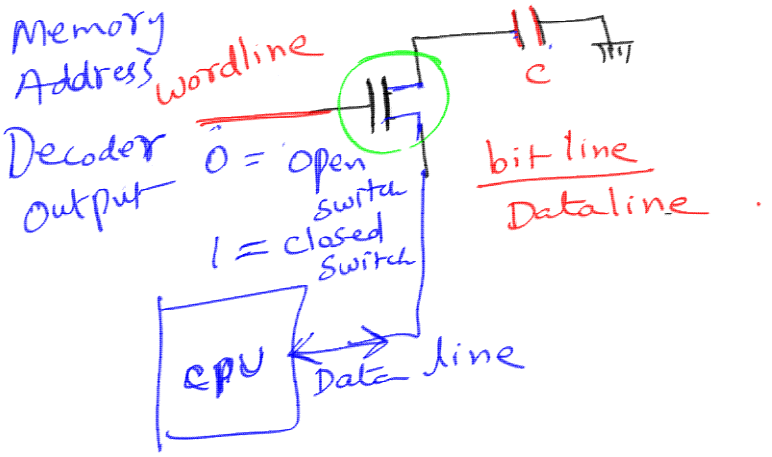
No of Rows = 1K = $2^{10} = 10^3$

Chip refresh time $\simeq 10^3 \times 10 \times 10^{-9}$ sec

$\simeq$ 10 μsec

$= 1024 * 10 \times 10^{-9}$ sec

$= 10.24$ μsec



→ while performing READ/WRITE operation for certain Row, all the Capacitors will be automatically refreshed

→ Refresh time = Amount of time needed for only Refresh operation
Row/chip

Refresh period of Row :— Refresh time + RD/WR time of a Row

Refresh period of a chip :— Refresh time of the given chip + RD/WR time of the chip

(EX) one Refresh operation takes 100ns in 4K×8 bit 'D' RAM chip, Chip refresh period is 1200 micro second, the % of time used to perform RD/WR chip is ——

No. of Rows = 4K ≅ 4×10^3 = 4096

Chip refresh time = 4096 × 100 ns
= 409.6 μsec

Chip RD/WR = 1200 − 409.6 μsec
= 790.4 μsec ✓

% of time = $\frac{790.4\ \mu sec}{1200\ \mu sec} \times 10 = 65.8666 = 65.87$

Approximate Value

Chip refresh time = 4000 × 100ns ≈ 400μsec

Chip RD/WR time ≈ 800μsec

% of time = $\frac{800\mu sec}{1200\mu sec} \times 100$

≈ 66.666 ≈ 66.67

Mark will be awarded, for the value 65 to 67

Memory Hierarchy

→ It is the combination of all different Memories used in the system from costliest to cheapest i.e (fastest to slowest)

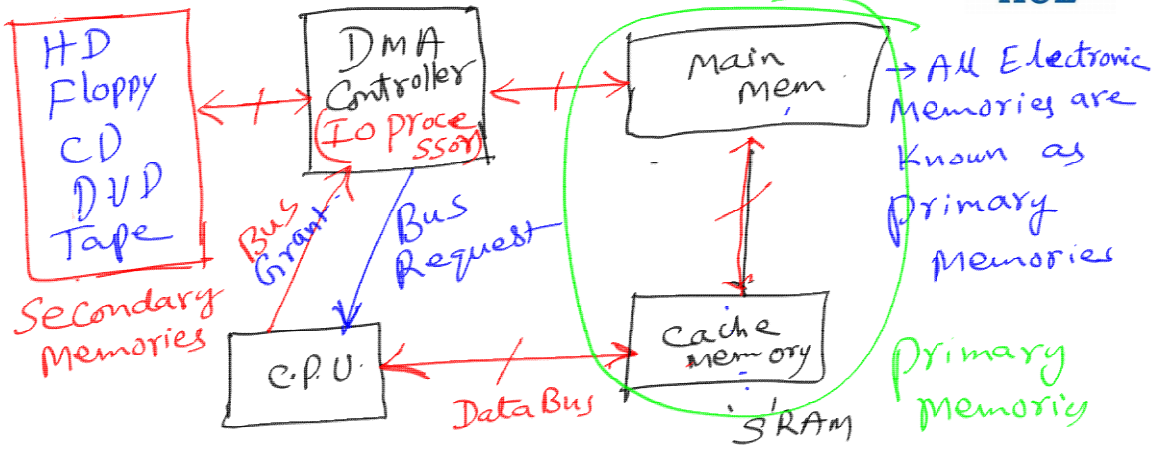
→ Faster Memories are costlier and slower Memories are cheaper.

→ In a system, fastest memory (cache) is used to store current executing file and slower Memories are used for storing back-up files.

VVV Imp

In order to Control the overall System design Cost; we use Smaller size Cache Memory and larger size Secondary Memories

Memory Hierarchy Basic diagram



→ During program execution, the CPU first visits cache memory, let target word is available in C.M, the CPU directly Reads that word and that operation is known as Cache Hit

→ When the target word is not available in Cache the operation is known as Cache Miss, then CPU visits main memory

→ Let target word is available in M.M then CPU transfers (Maps) the Associative block from M.M to C.M then only CPU reads the target word

→ Let the Target word is not available in M.M, then O.S takes responsibility to initialize the DMA Controller for transferring the target words from Secondary mem. to M.M.

→ Generally buses are under the control of CPU; for asking the Bus control, DMAC sends Bus request signal to the CPU.

⇒ After Receiving Bus request signal, the CPU first finishes the present operation and then handovers all the Buses to DMAC by sending Bus Grant signal

→ word access time in Secondary Memories is in millisecond (because of R/W Head and motor); hence CPU can't Communicate with these memories.

To transfer data b/w Main Memory and Secondary Memory; DMA Controller (Io processor) is used

→ In Memory Hierachy, the fastest memory is electrically closer to the C.P.U. and Cache occupies top position in the memory Hierarchy.

→ Main Memory occupies Central position and Secondary memory occupies least Positim in Memory Hierarchy

→ After Receiving Bus Grant Signal, the DMAC starts data transfer operation.

→ During DMA operation, the CPU Neither READ nor writes.

→ After Completion of DMA operation, all the buses will be automatically relinquished to C.P.U.

block = Group of words

Mapping:- It is the process of transferring the block of words from main memory to Cache mem, when miss has occured in Cache

Hit:- When the Searched word is available in Searched memory, the operation is known as Hit

Other wise miss.

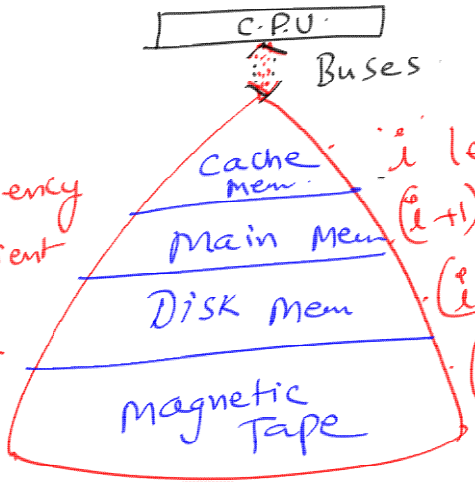
Hit Rate is directly Proportional to the size of the memory space.

→ Always Highest level memory provides 100% Hit Rate.

Hit Rate of a memory = $\frac{\text{No. of Hits}}{\text{Total no. of CPU references.}}$

Generally Secondary Memory is Highest level memory. It gives 100% Hit Rate

Memory Hierarchy in Pyramid Structure



$f_{clk} = \frac{1}{T}$

High frequency CPU is efficient than Low frequency CPU

CPU speed is directly proportional to it's f_{clk}

f = Speed

S = Size in no. of words

H = Hit Rate

C = Cost/bit

$C_i > C_{i+1} > C_{i+2} > C_{i+3}$

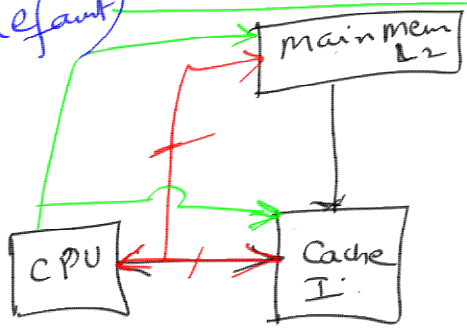
$f_i > f_{i+1} > f_{i+2} > f_{i+3}$

$S_i < S_{i+1} < S_{i+2} < S_{i+3}$

$H_i < H_{i+1} < H_{i+2} < H_{i+3}$

Note :— Highest level memory produces 100% Hit Rate.

(In default) Two-level memory Hierarchy



→ Always L_2 memory provides 100% Hit Rate

Both L_2 and L_1 memory Data buses are connected to the CPU in parallel.

T_{word} = word Access time

H_{CM} = Hit Rate in C.M

T_{CM} = word access time in C.M

T_{MM} = word access time in M.M

Let $T_{CM} = 1ns$

$T_{MM} = 10ns$

$H_{CM} = 90\%$

Average Memory Access time = Average word Access time

let $n = 100$

n - length of the program

$H = 90\%$, $M = 10$

Total Cache access time $= 90 \times 1ns = 90ns$

Total M.M. access time $= 10 \times 10ns = 100ns$

$T_{100} = (90 + 100)ns = 190ns$

$A.M.A.T = \frac{190ns}{100} = 1.9ns$

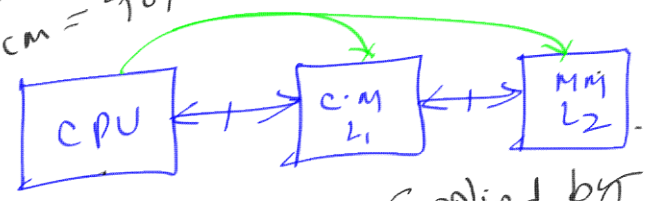
$$AMAT = (H_{CM} * T_{CM}) + (1 - H_{CM}) * T_{MM}$$

$$(0.9 * 1ns) + (0.1 \times 10ns)$$

$$= 1.9ns$$

$T_{CM} = 1ns$
$T_{MM} = 10ns$
$H_{CM} = 90\%$

Two level Memory Hierarchy (Serial)



words are Supplied by L_1 mem only.

Total C.M. access time = 90 * 1ns = 90ns

Total M.M access time = 10 * (10+1) = 110ns

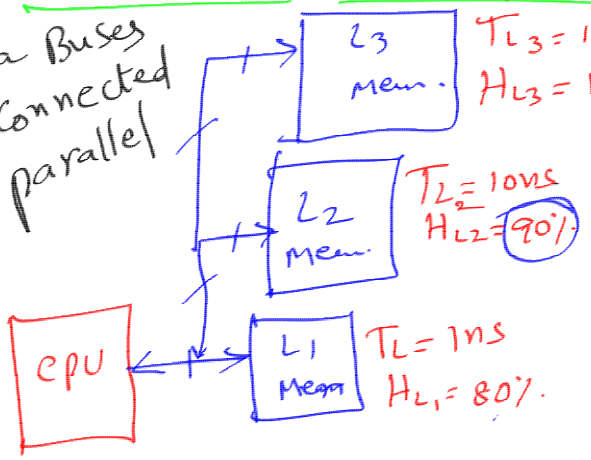
$T_{100} = 200ns$

$AMAT = (H_{CM} * T_{CM}) + (1 - H_{CM}) * (T_{MM} + T_{CM})$

$AMAT = 2ns$.

Three level Memory Hierarchy

All Data Buses are Connected in parallel



$AMAT = ?$

$n = 100$

Total L_1 time = 80 * 1ns = (80ns)

Total L_2 time = 18 * 10ns = 180ns

Total L_3 time = 2 * 100 ns

$T_{100} = (80 + 180 + 200)ns = 460ns$

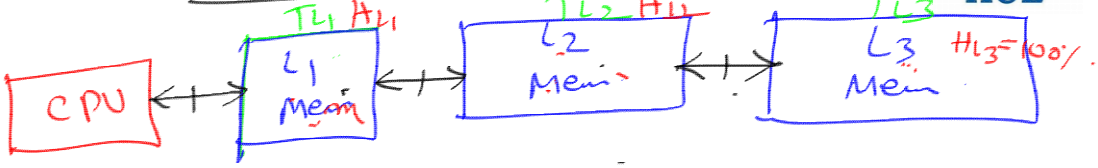
$AMAT = 4.6ns$ = 4.6 ns

$$AMAT = (H_{L_1} * T_{L_1}) + (1-H_{L_1}) * H_{L_2} * T_{L_2} + (1-H_{L_1}) * (1-H_{L_2}) * T_{L_3}$$

$$= (0.8 * 1ns) + (0.2 * (0.9) * 10ns) + 0.2 * 0.1 * 100ns$$

$$= 0.8ns + 1.8ns + 2ns = 4.6ns$$

Three level memory



only L_1 supplies the words to CPU.

$$AMAT = (H_{L_1} * T_{L_1}) + (1-H_{L_1}) * H_{L_2} * (T_{L_2} + T_{L_1}) + (1-H_{L_1}) * (1-H_{L_2}) * (T_{L_3} + T_{L_2} + T_{L_1})$$

Mapping

→ Mapping is the process of transferring the block of words from M.M to CM when Miss has occured in Cache only

→ Block is the group of words.

→ Block transfer is preferred for reducing the future Miss Penalities

Different mapping techniques

(i) Direct mapping (Cheapest)

(ii) Block set Associative mapping

(iii) Fully Associative mapping (Costliest)

Main memory = physical memory.

MMW = No. of words in Main Memory

CMW = No. of words in Cache Memory.

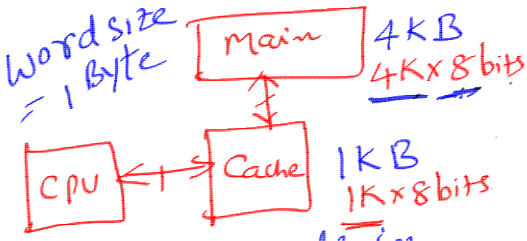
BW = No. of words in one Block.

MB = No. of blocks in main memory

CB = No. of blocks in Cache memory.

CL

Sometimes one Cache block is also known as one Cache LINE



No. of words in one Block = 64

$BW = 2^6$

MMW = 4K = 2^{12}

MA = PA = main memory Address size

12 bits = physical Address size.

CMW = 1K = 2^{10}

CA = 10 bits

Cache memory word Address size

MM Capacity = 4K x 8

CM Capacity = 1K x 8

$MMW = 4K = 2^{12}$

$CMW = 1K = 2^{10}$

PA = 12, CA = 10

TAG = (PA - CA)

2 bit

$BW = 64 = 2^6$

$CB = CL = \frac{CMW}{BW}$

$= \frac{2^{10}}{2^6} = 2^4$

CL_0 to CL_{15}

$CMW = CL * BW$

$MB = \frac{MMW}{BW}$

$= \frac{2^{12}}{2^6} = 64$

MB_0 to MB_{63}