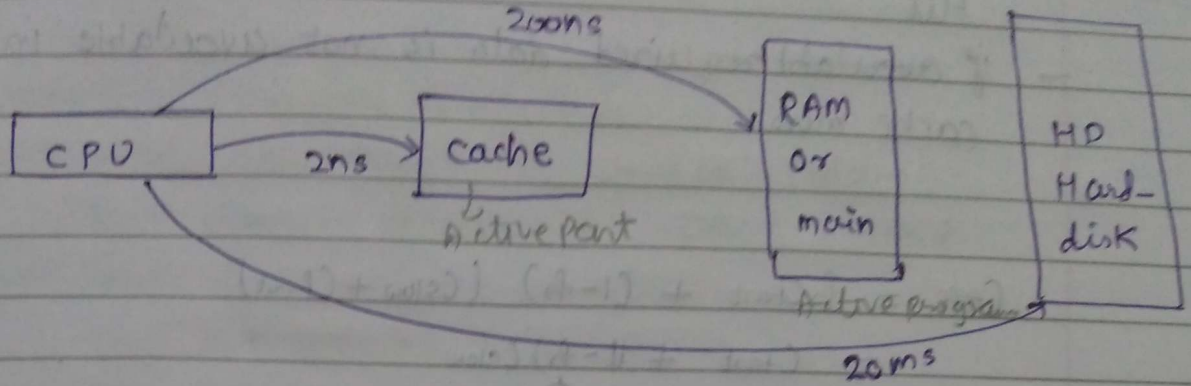
CACHE MEMORY:

Chapter-18

- its very nearer to processor but access time is very less



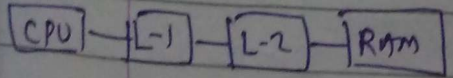
cache memory ^ used Static Ram ^ memory concept

- RAM runs the active program.
- repeatative data is stored in cache memory. So we can say that cache hold the active part of program or which is frequently used.

- level 1, level 2, level 3 caches are used. and we don't used level 4 cache because access time with RAM becomes same as cache memory.
- Cache size would be limited
- L1 < L2 < L3

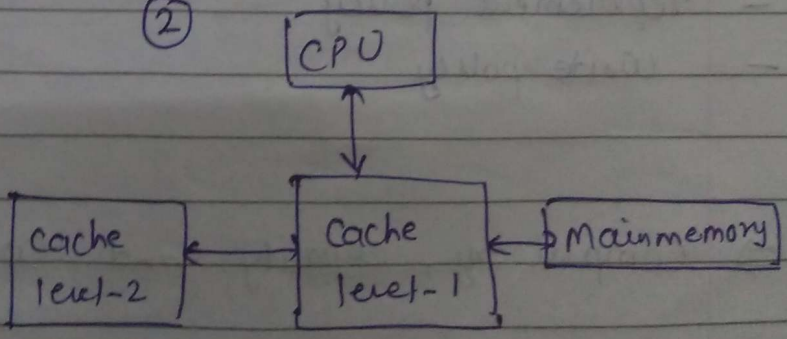
Multiple cache levels:

①



level-2 connected between L-1 and RAM

②



level 2 connected to "backside" Bus

- as level memory size is less than distance or time is also less.

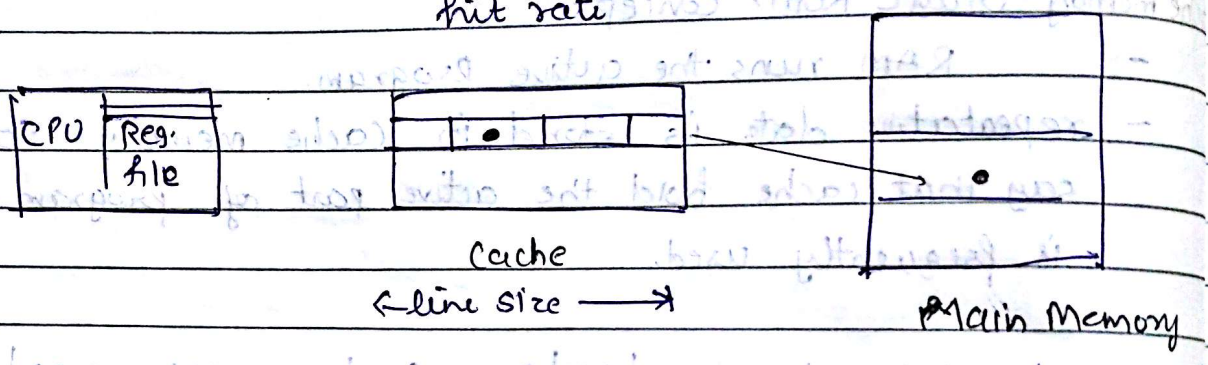
Cache Hit/Miss :

- if data is available in cache, thats known as Hit
- if ~~available~~ required data is not available in cache, cache miss.

$$C_{eff} = h C_{fast} + (1-h)(C_{slow} + C_{fast})$$

$$= C_{fast} + (1-h) C_{slow}$$

↑ hit rate



Cache Memory Design parameters :

- Cache size
- cache line size
- placement policy
- replacement policy
- write policy

⇒ Compulsory, capacity and conflict misses

Cache Mapping :

① Direct mapped

2^L → lines
2^w → words
L → line index
w → word index

- cache line size will be decided
- each word & line having the unique address.
- conflict miss is occur in Direct mapped.
- Add. trace:

1, 7, 6, 5, 32, 33, 1, 2

1 → miss, line 3, 2, 1, 0 fetched
7 → miss, line 7, 6, 5, 4 fetched
6 → hit
5 → hit

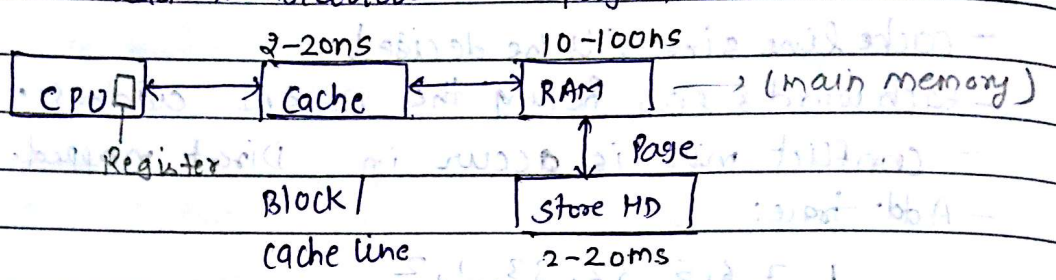
- for remove the conflict miss we are using Set Associative cache.

improving cache performance:

1. line width
2. Set size or associativity
3. line replacement policy (least recently used LRU)
4. write policy

Memory Management :

- Virtual memory
- add. translation

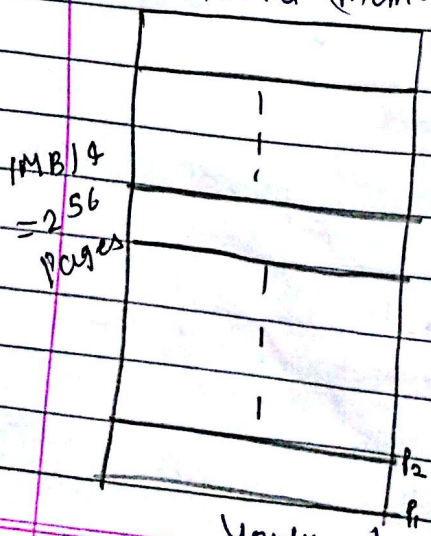


- page size (1KB, 2KB, 4K, 16MB)
- cache → only one single location
- RAM → fetch the complete cache line (means nearby location data is also fetched)

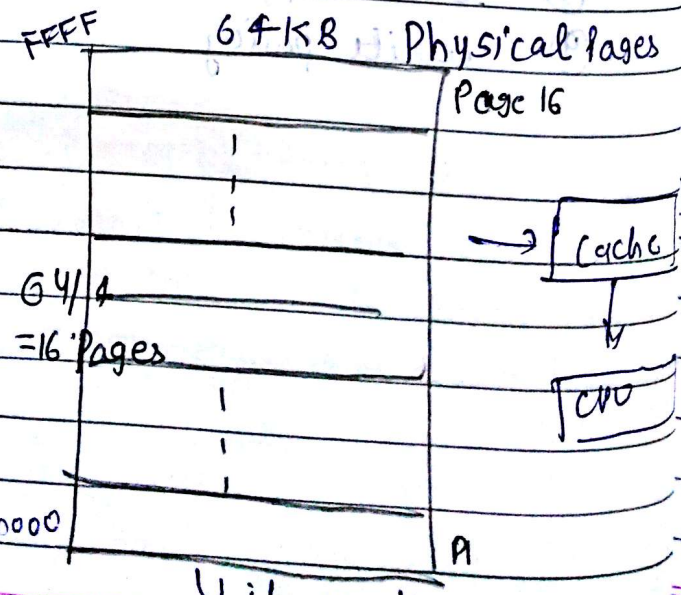
	Cache	memory
①	MISS	FAULT
②	Block Page fault	Page

Add. Bus → $2^{20} \cong 1$ MB

Virtual (Memory) Pages



Only stor the data not run the prog.



Yes run the program by RAM

Add.

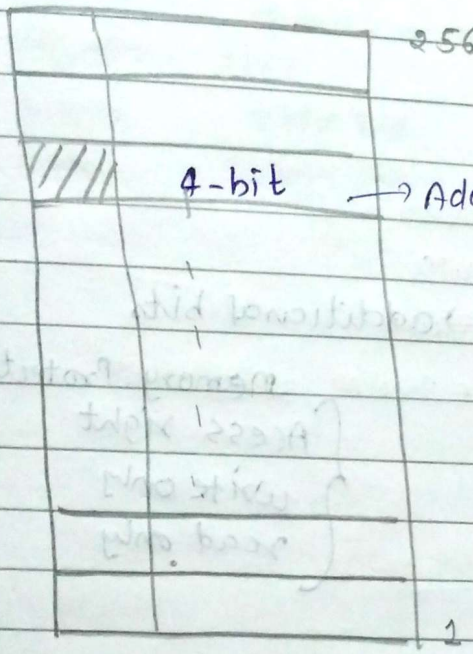
Phy

→ Hard disk
Physical → RAM

In this page size is 4K.
RAM = 64K
HD (Hard disk) = 1MB

- which 16 ?
- where call ?

page table :

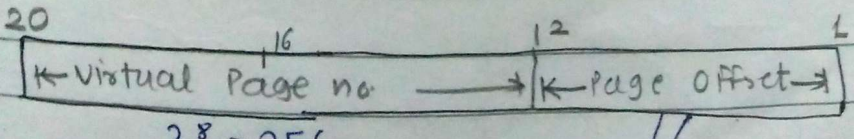


1 to 256 page entry based on virtual size.

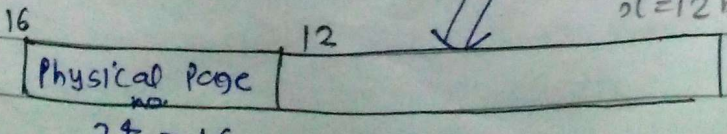
- page table we realised in only RAM so its present in RAM
- out of 256 page, which 16 pages are active → Page table → where these pages are located.

Add. translation :

Logical / virtual

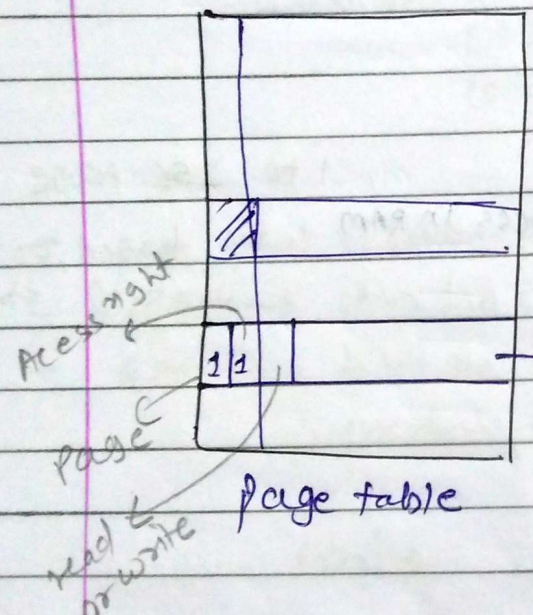


Page size = 4K
$2^x = 4K$
$x = 12$ bits



- page table origin register, stores the starting add. of the page table
- and that register stored in CPU register

⇒ Memory protection:



Memory Protection
- Acess right
  - write only
  - read only

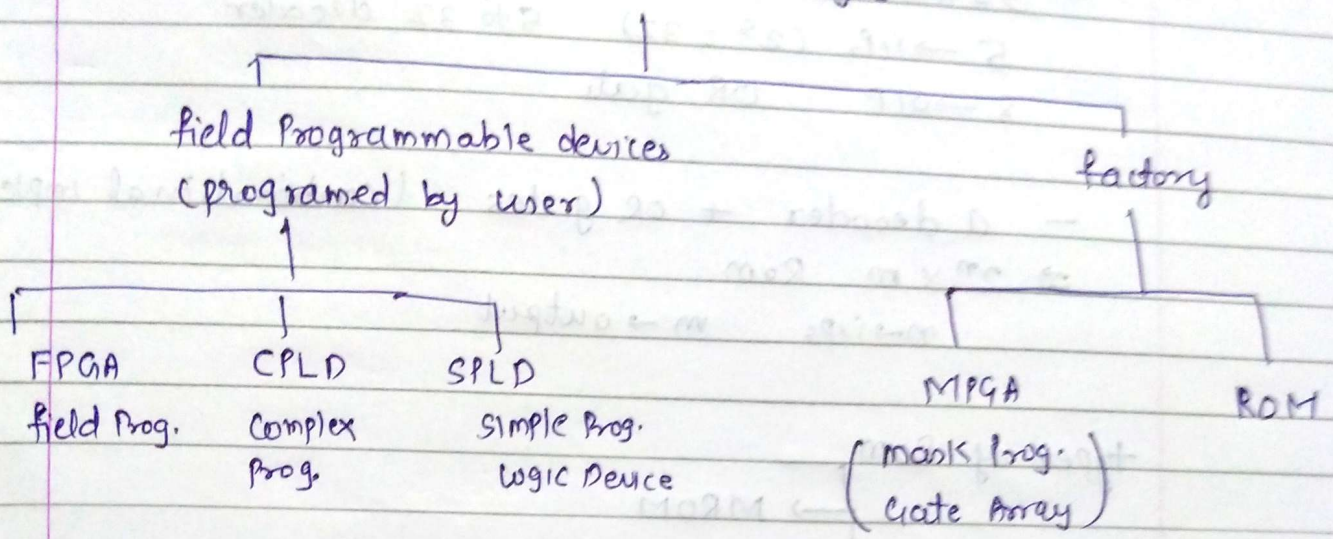
- memory management unit (MMU) control or protect our memory.

Ch - 17

Memory

19-Oct

Classification of memory :



Types of memory :

↳ Static (information stored in latch)

↳ Dynamic (" " in the form of charge)

Static	Dynamic
- information stored in latch.	- stored in the form of charge
- short read / write cycle	- larger memory capacity
- remains valid as long as power is applied	- reduced power consumption
- it's uses 4-5 transistor per bit	- need to be refreshed

Volatile	Non-volatile
- lose data while power is turned off	- ROM, EPROM, EEPROM
- SRAM, DRAM	- flash memory (Read/write access is fast in flash memory)

Read only memory :

- 32×8 ROM
  - 5 → i/Ps ($2^5 = 32$) 5 to 32 decoder
  - 8 → o/P OR gate

- a decoder + OR gates (combinational imple.)

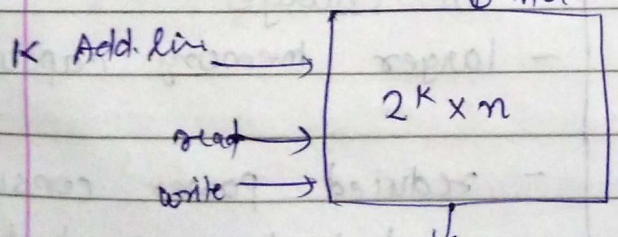
⇒ $2^n \times m$ ROM

n → i/Ps m → output

types of ROM
- MROM
- PROM
- EPROM
- EEPROM [electrical erasable prob. read m.

Random Access memory :

- stores binary information



- DRAm
- SRAm

Combinational PLD

- PROM
- PAL (Prog. And, fixed OR)
- PLA (Prog. OR, fixed And)

Seq. Prog. Devices:

Seq. Prog. logic devices:

- SPLD
- PLD + Flip-Flops