Module - V

Memory & I/O organisation

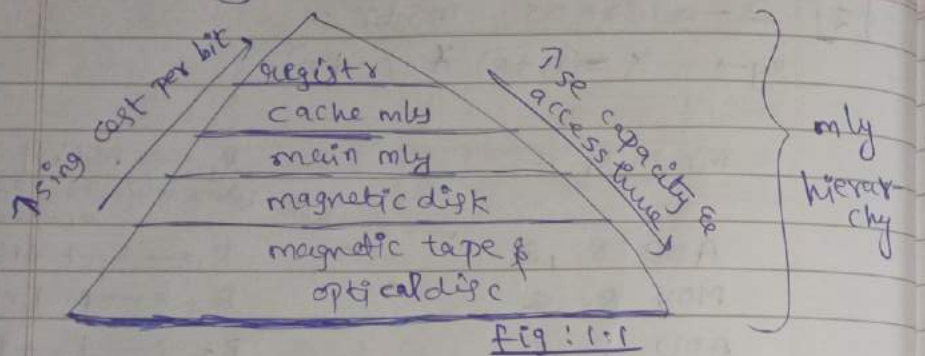


fig : 1.1

[1] **Memory System** =

Every comp contains several types of devices to store the instrn & data required for its opr.

→ **characteristics of mly system** :

1) **mly type** : Storage components of a comp system can be placed in 4 grps- cpu registrs, cache mly, main mly & secondary mly.

2) **mly size** :
Size of storage component of a comp system depends on its type.
Size of cpu (R) is b/w 8 & 32 bits

3) **mly locations** :

cpu (R)
cache mly
main mly
} internal mly.

peripheral storage (disk & tape) } external mly.

4) **Reliability** :
It is measured by the mean time befre failure.

→ **Mly Hierarchy** :
(refer fig : 1.1).

* In comp system, mly is used to store data temporarily / permanently.
* There are 4 major storage levels in mly. H —
1) Internal - processor (R) & cache
2) Main — system RAM & controller card
3) on-line mass storage - sec storage
4) off-line bulk " - Tertiary & off-line storage.

I **CPU registers** : (has 8 (R) inside CPU)
↳ fastest, smallest, & most expensive type of mly.
↳ Almost all comps load data frm

a larger mly into (R).
↳ User-accessible (R) can be either data-(R) / address (R)
↳ eg: ① GPRs can store both data & address.
② SPRs (Sple purpose (R)) hold pgm state.

II Cache mly :
↳ Not visible to prgmer.
↳ Smaller, faster mly, close to a processor core
↳ Acts as buffer b/w CPU & main mly.
↳ Used to hold those parts of data & prgm which are most frequently used by CPU.
↳ parts of data & prgms are transfrd frm disk to cache mly by OS.

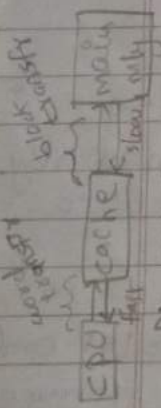


fig: use of cache mly

III Main mly / primary mly :
↳ mly directly accessible by CPU.
↳ holds only those data & instrn on which the comp is crntly working.
↳ Has limited capacity & data is lost whn the power is switched off.

main mly working { whn the processor needs to read data frm mly, it snds a request to the mly controller. The mly controller then locates the data in mly & snds it to the processor.
↳ Generally made upof semicond devices.
↳ It is ÷ into 2 categories: RAM & ROM.

RAM	ROM
* Temporary Storage	permanent Storage
* Store data as MBs	" as GBs.
* Volatile [data lost whn comp is switched off]	non-volatile
* writing data is faster	is slower

IV Secondary / external mly :
↳ not directly accessible by CPU.
↳ data frm Sec Storage needs to ~~The~~ be brought into the primary Storage bfre CPU can use it.
↳ eg: mag disk, mag tape, optical disk.

[2] Techniques / modes of I/O Transfer =
* An I/O opr refers to a data transfr b/w an I/O device & memory / b/w an I/O device & the CPU.
* 3 modes —
1) Programmed I/O :

* prgm ഓരോന്നായി തീർന്നിട്ട് comp-ൽ നടക്കുന്നതാണ്.
* ഒരു i/p & o/p cpu-യോട് ഇത് ആവശ്യം ഉണ്ട്, ഇതെല്ലാം ചെയ്തിരിക്കും. → p·I/o.
* Adv → execute easily

* Here, I/o opr are completely controlled by cpu.
* with p·I/o, data are exchanged b/w the cpu & the I/o interface module.
* I/o device does not have direct access to main mly.

2) <u>Interrupt Driven I/o</u>:

I/O-ക്കിന് ആവശ്യം ഉള്ള സമയത്ത് permission നൽകുന്നു. & is more efficient

* The prbm with p·I/o is that the cpu has to wait a long time for the I/o interface module ~~#~~ whereas in I·D I/o, let I/o interface module inform the cpu whn it is ready to transfer data. This mode of transfr uses interrupt facility.
* while the cpu is running a prgm, at the end of each instrn cycle, the cpu checks for interrupts.

* I·D I/o is more efficient than p·I/o bcz it eliminates needless waiting.

3) <u>Direct Memory Access (DMA)</u>:

* Direct cpu-യുടെ involvement ഇല്ല. it directly goes to mly & access data.

* In both p·I/o & I·D I/o, ~~I/o transfer~~ data is transfered to/from I/o devices to the mly via a cpu (R).
The DMA transfers are controlled by a control circuit called DMA controller, associated with the I/o module.
* DMA controller allows direct data transfer b/w peripheral device & the main mly without involving cpu.
* DMA controller needs to take control of the bus in order to transfer data to & and from mly.

[3] <u>Modes of Data Transfer</u> =
* Transmission of digital data through a transmission medium can be performed either in serial/parallel mode

* Parallel data Transmission	Serial data transmission
- each bit of the msg has its own path & the total msg is transmitted at same time.	Involves sending 1 data bit at a time & uses a pair of wire for commun of data in bit-serial form
- adv → higher speed disadv → higher cost of cabling.	

* Synchronous trans.	Asynchronous trans.
- Synchronized by an ext clk.	Synchronized by sple signals along the trans. medium
[illegible] data transfer [illegible]	[illegible] multiple lines [illegible] transfr [illegible]
- data transfer is possible b/w 2 units when each of them knows the behaviour of the other	This approach is widely used in most comp systm.

* Strobe control

- method of asy. data transfer employs a single control line to time each transfer
- The strobe may be activated by either the source / destination unit.

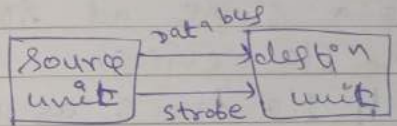


fig: source initialed strobe control for data transfr.

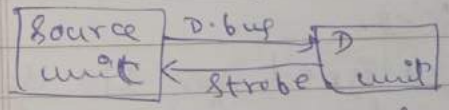


Fig: destinan i·s·c·for data transfr

Handshaking.

Solves Strobe's demerit by introducing a 2nd control signal.

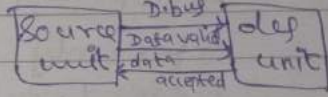
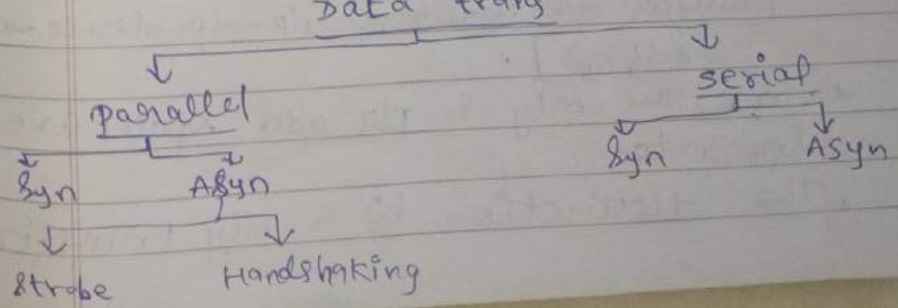


fig: block diagrm



[4] I/O Addressing =

[i/p & o/p device-ಅನ್ನು address ಮಾಡುವುದು].

* Each I/O device is given a unique address
* When the CPU issues an I/O ~~cmd~~ cmnd, the cmnd contains the address of the desired I/O device.
* 2 modes of I/O addressing —

1) **Memory Mapped I/O:**

[mly-ಗೆ ಮೀಸಲಾದ ಕೆಲವು range ಅನ್ನು ಮಾಡುವುದು i/p - o/p device-ಅನ್ನು ad ಮಾಡುವುದು]

* Here, there is a single ad space for mly loc & I/O devices.
* CPU treats each I/O device as mly loc & uses the same machine instr. & control signals to access both mly & I/O devices.
* The addresses assigned for I/O devices cannot be used for mly words.

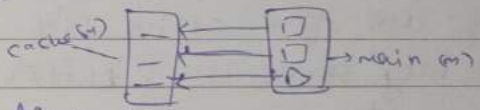
2) **Isolated I/O:**

[unique ad ಮಾಡುವುದು i/p-o/p device-ಅನ್ನು ಮಾಡುವುದು].

* Here, the mly & I/O add spaces are separate.
* The distinction b/w a mly transfer ~~ad~~ & I/O transfer is made through separate READ & WRITE lines.
* It is data mly & I/O add so that mly ad values are not affected by interface ad.
* Here, CPU has distinct i/p & o/p instr.

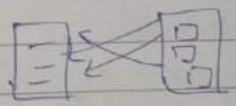
Direct mapping:

main mly-ನ blk ಅನ್ನು mly cache mly-ಯ ಒಂದು single line ಅನ್ನು connect ಮಾಡುವುದು.



Associative M:

cache-ನ ಯಾವ line-ಅನ್ನಾದರೂ blk of mly ಆಗಿರಬಹುದು.



Set Associative M:

ಯಾವ blk of mly-ಅನ್ನು we can use. here, we are can map ~~connecting~~ set of lines.

miss ratio : the % of times that a reqst for data is not found in the cache.

virtual memory : * physically exist [illegible] mly.

* sec storage -[illegible] space of mly virtual [illegible] store [illegible].

⇒ cache Memory :

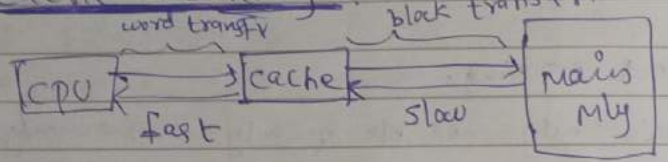


fig : use of a cache mly.

* cache mly serve as a buffer b/w CPU & main mly.
* Hit ratio : performance of cache mly is frequently measured in terms of a quantity → H. ratio.

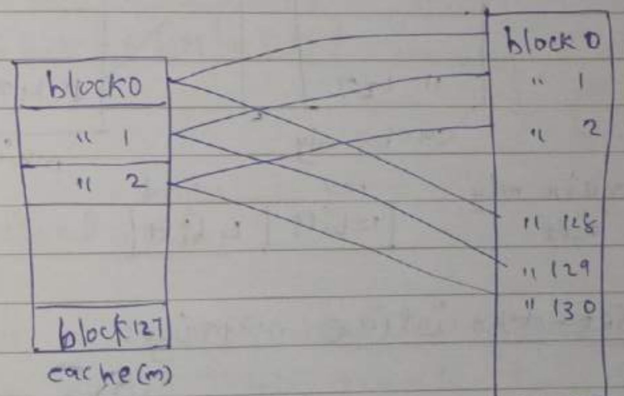
$$\text{Hit ratio} = \frac{\text{no. of hits}}{\text{no. of hits} + \text{no. of misses}}$$

→ Mly mapping in cache :

* cache mly & main mly are ÷ into small fixed sized blocks.
* Each cache block includes a tag that identifies which particular mly block is currently being stored.
* The choice of mly mapping determines how the cache is organized.
* 3 techniques used for mly mapping —

a) Direct mapping :

* Simplest way to determine cache loc in which to store mly blocks is the direct mapping technique.



main mly addr

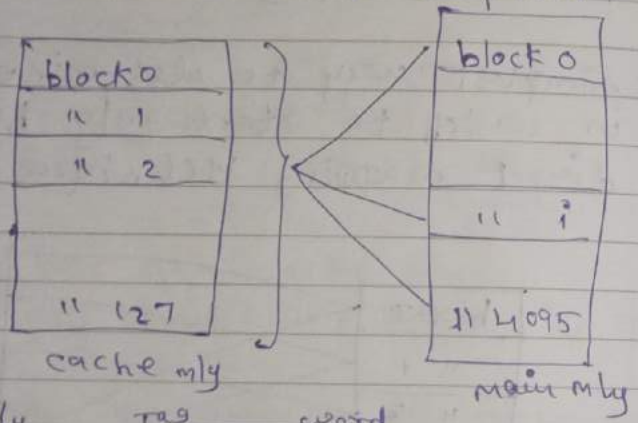
Tag	block	word
5 bit	7 bits	4 bits

b) Associative mapping :

* It is a flexible mapping that allows a mly block to reside in

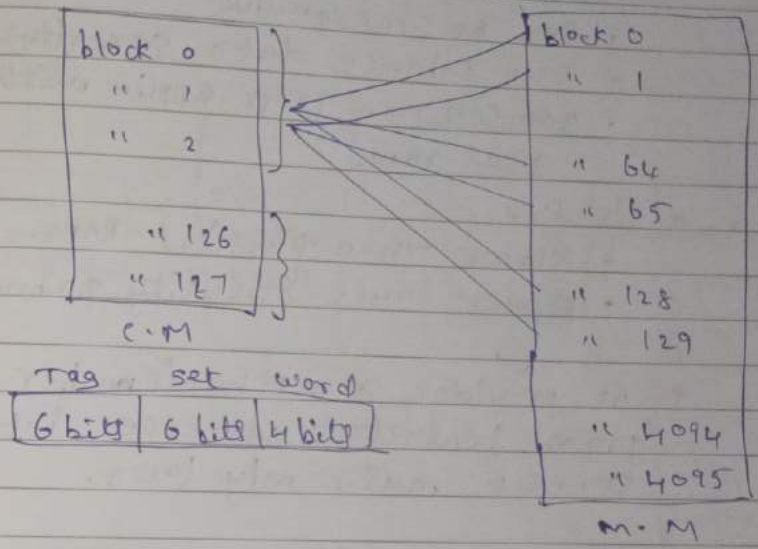
any cache block.

* tag bits of an address received frm the cpu are compared to the tag bits of each block of the cache to see if the desired block is prsnt.



c) Set-Associative mapping:

* combines direct mapping with associative mapping by arrangmnt lines of cache into sets.
* Has highest hit ratio compared to D. mapping & A. mapping.
  ∴ its performance is better.
* It is very expensive, bcz the set size uses the cost uses.



⇒ Virtual m/y :

* It is a concept used in some large comp systm that permit the user to construct prgms
* It is a partition of logical m/y frm physical m/y.
* Creates the C) of prgming easier bcz the prgmer no longer requires to worry about the multiple Physical m/y available.

* Adv:
  - can be inexpensive
  - can enhance data security.
  - you can run larger applin with less real ram.

* Disadv:
  - Slower than physical RAM.
  - may cause stability prbms.

* It provides a mechanism for translating prgm generated addresses into correct main m/y locs.

* Address Space (N) : virtual add (logical add) — add generated by prgms

  m/y space (M) : physical add (PA) — actual main m/y add.

* Add Space: An add used by a prgmr is known as virtual add (VA). & a set of such add → A. space

* Memory Space: An add in main m/y → physical add (PA) & a set of such loc → m/y space.

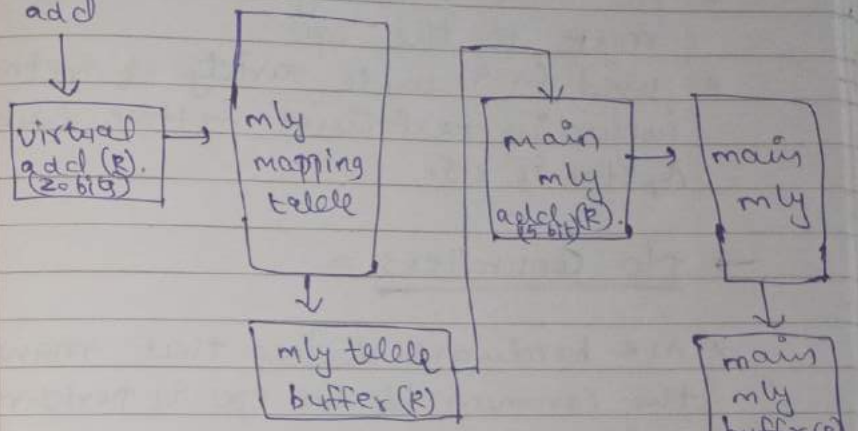


fig: m/y table for mapping a virtual add.

* add. refrnced by a running process → virtual add
* add available in primary m/y → real add
* converting virtual add to real add → dynamic add translation.

→ Priority Interrupt =

* Allows the CPU to prioritize interrupts & service the most imp ones 1st.

Interrupt → Signal that causes CPU to stop wht it is doing & deal with the interrupt. Generated by hardware devices like mouse, keyboard.

* Are implemented using prgmble interrupt controller (PIC).
* PIC sends the highest priority interrupt reqst to the CPU.
* used in a wide variety of systms including real-time systms, embedded systm & OS.

→ I/O Controllers =

* Are hardware devices that manage the comman b/w the CPU & peripheral devices.
* I/Os –
  - Handling errors & interrupts
  - controlling the timing of data transfr
* Are essential for the opr of any comp systm.
* Types → Disk controllers → CPU & hard drive
  - Network "
  - Graphics " → CPU & graphic card
  - Audio " → CPU & audio "
  - USB " → manage data trsfr b/w CPU & USB devices