Roll No · 202cd005

Assignment 1

GOODLUCK | Page No. 1
Date 9, 11, 20

Que 1: Explain differences between a process and thread.

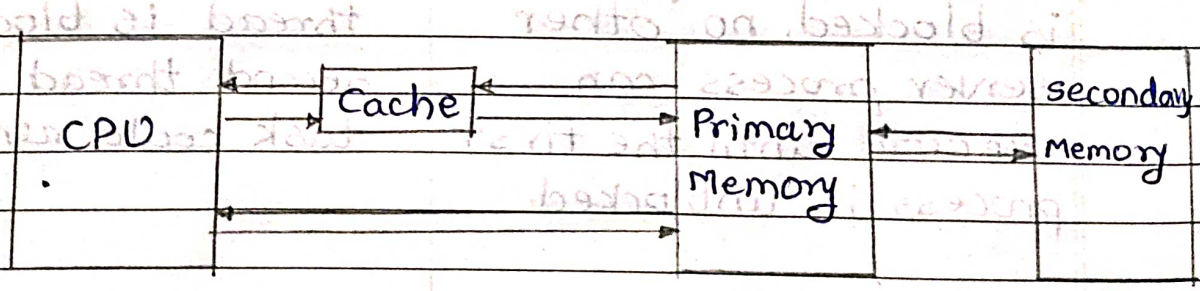
Process	Thread.
1) Process means any program in execution.	1) Thread means segment of a process.
2) Process takes more time to terminate.	2) Threads take less time to terminate.
3) Takes more time for creation	3) Takes less time for creation.
4) Also takes more time for context switching.	4) Takes less time for switching context.
5) Less efficient in terms of communication.	5) More efficient in terms of communication.
6) Consumes more resources.	6) Consumes less resources
7) Process is isolated.	7) Threads share memory.
8) Process is called heavy weight process.	8) Thread is called light weight process.
9) Process switching uses interface in operating system.	9) Thread switching does not require to call a operating system & cause an interrupt to kernel.
10) If one server process is blocked, no other server process can execute until the first process is unblocked.	10) While one server thread is blocked, second thread in same task could run.

	Process	thread
i)	Process has its own process control block stack & address space.	ii) Thread has parents' PCB, its own thread control & stack & common address space.

Que 2: What is cache memory? Explain different cache mapping techniques with an example.

→ Cache memory :-

i) It is a very high speed memory.
ii) It is used to speed up & synchronizing with high-speed CPU.
iii) It is costlier than main memory or disk memory but economical than CPU registers.
iv) It is a extremely fast memory type & acts as buffer betn RAM and CPU.
v) It holds frequently requested data & instructions so that they are immediately available to CPU when needed.
vi)



Cache Mapping Techniques:-

@ Direct Mapping:-

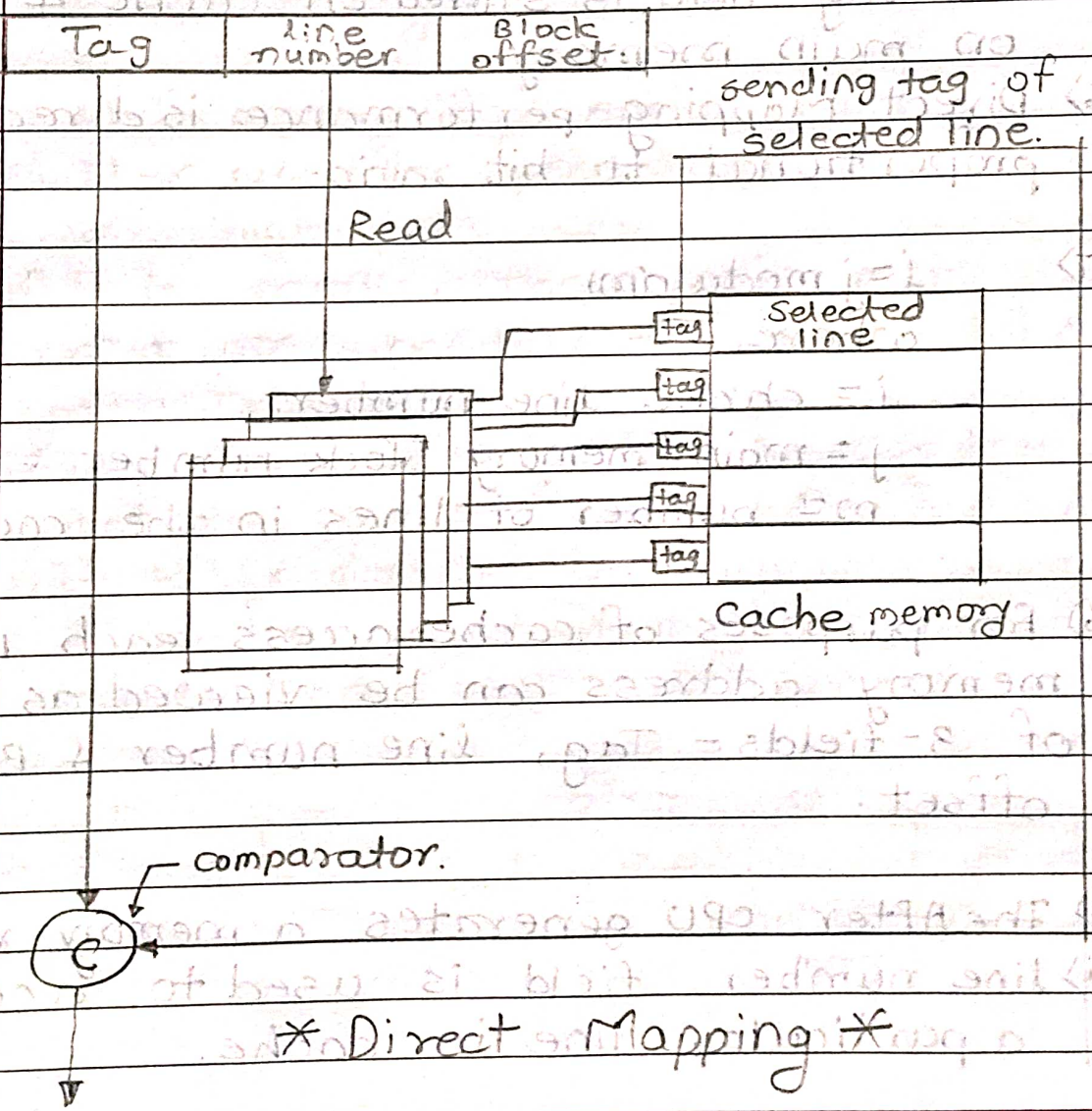
1) Simplest technique.
2) Assigne each memory block to specific line in cache.
3) If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed.
4) An address is split into 2-parts - index field & tag field.
5) The tag field is stored on cache & rest on main memory.
6) Direct Mapping performance is directly proportional to hit ratio.

7) $i = j \bmod (m)$
where,
i = cache line number.
j = main memory block number.
m = number of lines in the cache.

8) For purposes of cache access, each main memory address can be viewed as consisting of 3-fields:- Tag, line number & Block/line offset.

9) ~~The~~ After CPU generates a memory request,
i) line number field is used to access a particular line in cache.

ii) The tag field of CPU address is then compared with the tag of line.
iii) If the 2-tags match, a cache hit occurs & the desired word is found in the cache.
iv) If 2-tags donot match then a cache miss occurs.
v) In such case, the required word has to be brought from main memory.
vi) It is then stored in cache together with new tag replacing the previous one.



* Direct Mapping *

⑥ Associative / Fully associative mapping :-

1) In this type of mapping, the associative memory is used to store content & addresses of the memory word.
2) Any block can go into any line of cache.
3) This means that word id bits are used to indentify which word in the block is needed.
4) The tag becomes all of the remaining bits.
5) This enables placement of any word at any place in the cache memory.
6) It is considered as fastest & most flexible mapping form.

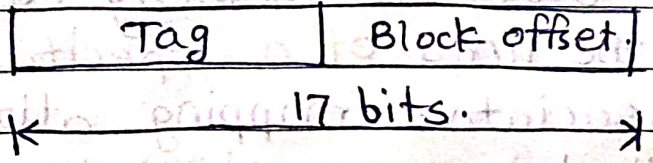
Example :-

Consider fully associative mapped cache of size 16KB with block size of 256 ~~bytes~~ bytes. The size of main memory is 128KB then.

① No. of bits in physical address ⇒

size of main memory = 128KB.
= 2^{17} ~~bits~~ bytes.

⇒ No. of bits in physical address = 17.



② Number of ~~blo~~ bits in block offset:-

Block size = 256 bytes.

= 2^8 bytes.

⇒ **No. of bits = 8 bits.**

⇒ ③ Number of bits in Tag:-

= 17 − 8 = 9 bits.

∴ **No. of bits in tag = 9**

© Set-associative mapping:-

1) This form of mapping is an enhanced form of mapping where the drawbacks of direct mapping are removed.
2) Set associative mapping addresses the problem of possible thrashing in direct mapping.
3) It does this by saying that instead of having exactly one line that a block can map ~~into~~ in cache, we will group few lines together creating a set.
4) Then a block in memory can map to any one of the lines of a specific set.
5) Set-associative mapping allows that each word that is present in the cache have

2-or-more words in the main memory for same index address.

6) Set associative mapping combines best of direct & associative mapping techniques.

7) The relationships are:

$$m = v * k$$
$$i = j \bmod v.$$

where,
i = cache set number.
j = main memory block number.
v = number of sets.
m = number of lines in the cache number of sets.
k = no. of lines in each set.

8) Example :-

Consider a 2-way set associative mapped cache of size 16KB with block size 256 bytes. The size of main memory is 128 KB. Find -
1. Number of bits in tag
2. Tag directory size.

⇒ Given, set size = 2,
cache memory = 16KB.
Block size = 256 bytes.
Main memory = 128 KB.
No. of bits in phy. address = 128 KB = 2^{17} bytes.

⇒ No. of bits in physical address = 17 bits.

No. of bits in Block of offset = 256 bytes.
= 28 bytes.

⇒ No. of bits in block offset = 8 bits.

No. of lines in cache = cache size / line size.
= 16 KB / 256 bytes.
= $2^{14}/2^{8}$
= 64 lines.

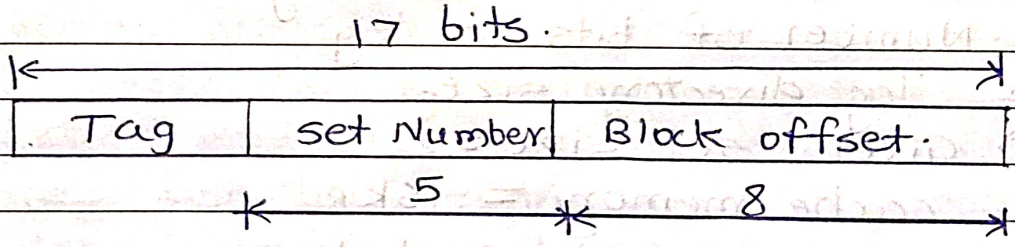
∴ No. of sets in cache = $\frac{64}{2}$ = 32 sets.

= 2^5 sets.

⇒ No. of bits in set number = 5 bits.

∴ No. of bits in tag = 17 − (5 + 8)
= 4 bits.

∴ Tag directory size = No. of tags × Tag size
= 64 × 4
= 256 bits.
= 32 bytes.



* Example on direct Mapping:-

Consider a direct mapped cache of size 16KB with block size 256 bytes. Size of main memory is 128KB. Find 1) No. of bits in tag 2) Tag directory size.

⇒ Given, Cache memory = 16KB.
Block size = 256 bytes.
Main memory = 128KB.
Main memory = 2^{17} bytes.
No. of bits in physical address = 17 bits.

Block size = 256 bytes = 2^8 bytes
⇒ No. of bits in block offset = 8 bits.

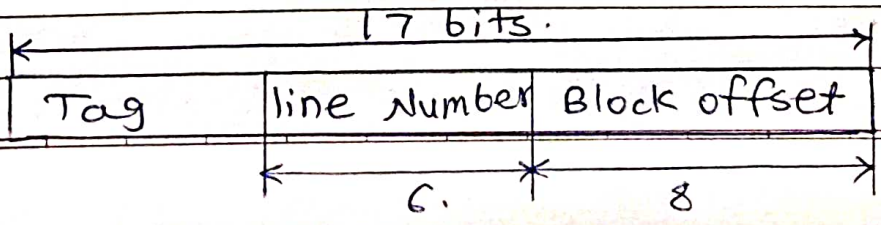
Total No. of lines in cache = $\frac{\text{Cache size}}{\text{line size.}}$
= 16KB / 256 bytes
= $2^{14} / 2^8$ bytes.
= 2^6 lines.

⇒ No. of bits in line number = 6.

⇒ Number of bits in tag = 17 − (6 + 8)
= 3 bits. ⇐

Tag directory size = No. of lines in cache × No. of bits in tag
= $2^6 \times 3$ bits = 24 bytes.

∴ Tag directory size = 24 bytes.



Que 3: What is instruction level Parallelism? How does it improve the performance?

→ 1) Instruction level parallelism is a measure of how many of the instructions in a computer program can be executed simultaneously.

2) There are 2 approaches to instruction level parallelism. — Hardware & software.

3) Hardware level works upon dynamic parallelism, whereas the software level works on static parallelism.

4) Dynamic level parallelism means the processor decides at run time which instructions to execute in parallel, whereas static ILP means the compiler decides which instruction to execute in parallel.

5) E.g. pentium processor works on dynamic sequence of parallel execution, but the Itanium processor works on static level parallelism.

6) Example :-

Consider the following program,

1 e = a + b
2 f = c + d
3 m = e * f

Here operation 3 depends on result of operation 1 & 2, so it cannot be calculated until both of them are completed. However operations 1 & 2 do not depend on any other operation so they can be calculated simultaneously. If we assume that each operation takes one unit of time &to

complete then the above program can be executed within a total of 2-units of time. with instruction level ||elism of 3/2.

7) Hence instruction level parallelism improves system performance by executing independent processes or operations simultaneously.

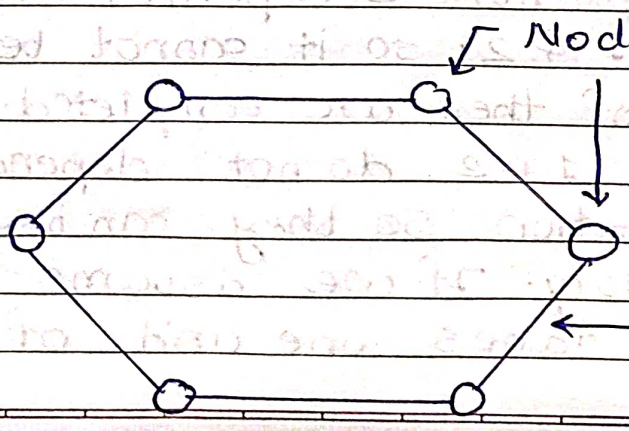
Que 4: Explain following intercommunication networks. comment on their bisection width

a] Ring topology

→ 1) A ring network is a network topology in which each node connects to exactly 2-other nodes, forming single continuous pathway for signals through each node.

2) Data travels from node to node, with each node along the way handling every packet.

3) Rings can be unidirectional travelling clockwise or anti-clockwise around the ring or bidirectional (as in synchronous optical networking / synchronous digital hierarchy (SONET/SDH).



(a) Ring topology

b) Toroidal Mesh :-

* Bisection width :-

1) Bisection width is a measure of number of simultaneous communications or connectivity.

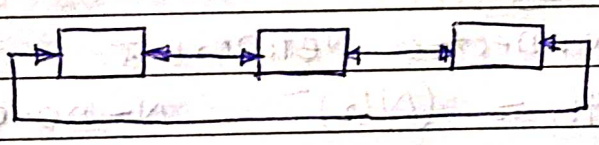
2) It gives, number of possible communications that can take place across the divide betn 2 halve of network.

4) Hence Bisection with of Ring topology is 2.

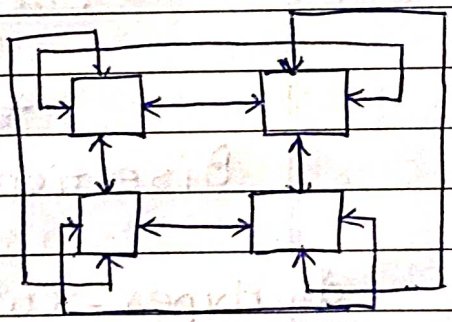
6) Toroidal Mesh :-

1) It is a switch-less network topology for connecting processing nodes in a parallel computer system.

2) It can be 1D or multi-dimensional.



a) 1D Torus



b) 2D Torus.

Bisection width :-

If there are n nodes, then $\sqrt{n}$ links should be broken to bisect the network, hence bisection width is $\sqrt{n}$.

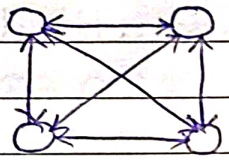
c) Fully connected Network:-

1) In a fully connected network, all nodes are interconnected.
(In graph theory, this is called complete graph).
2) The simplest fully connected network is a 2-node network.
3) A fully connected network doesn't need to use packet switching or broadcasting.
4) However since no. of connections grows quadratically with no. of nodes:

$$c = \frac{n(n-1)}{2}$$

c → connections
n → nodes.

This makes it impractical for large networks.



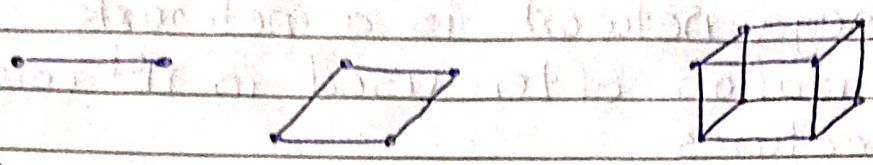
fully connected Network

Bisection width = $(N/2)^2$ N- no. of nodes.

d) Hyper-Cube:-

1) Used to connect multiple processors with memory modules & accurately route data.
2) Hypercube consist of 2^m nodes, which ~~from~~ form vertices of square to create an internetwork connection.
3) It is basically a multi-dimensional

mesh network.



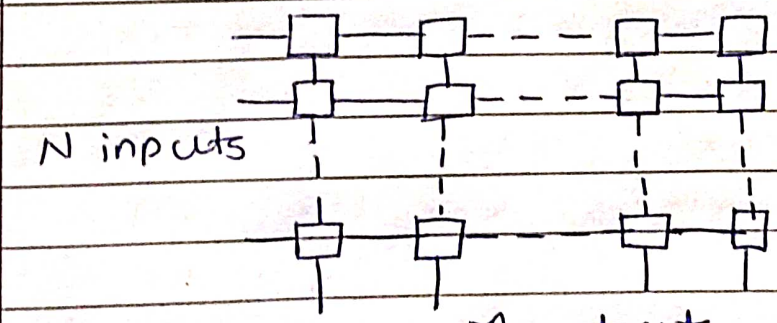
a) 1D b) 2D c) 3D – Hypercube

Good bisection width = $\frac{N}{2}$

Complexity :-
- Number of links : $N(\log_2 N)/2$.
- Node degree is $d = \log_2 N$

e) Crossbar Network :-

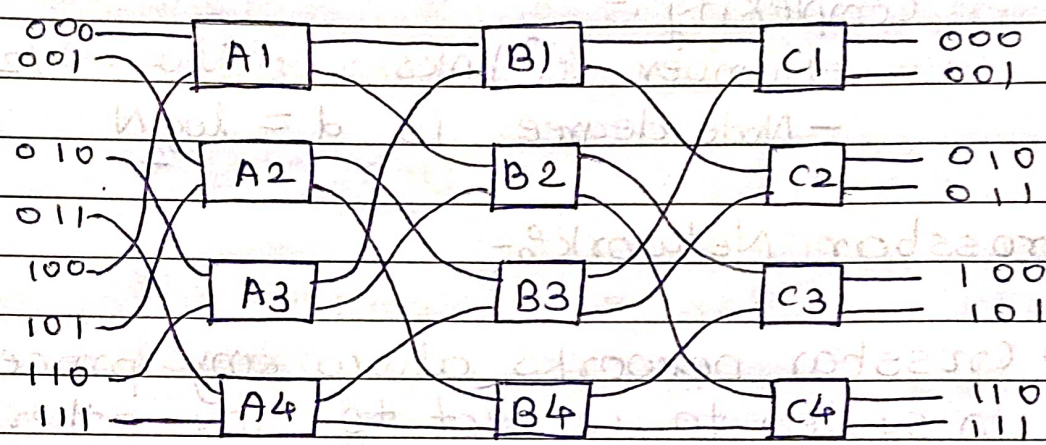
1) Crossbar networks allow any processor in system to connect to any other processor or memory unit so that many processors can communicate simultaneously without contention.
2) A crossbar can be defined as switching network with N inputs & M outputs which allows upto min {N, M} one-to-one interconnections without contention.



* NXM crossbar Network *

f) Omega Network:-

1) An omega network is a network configuration often used in 11el computing architectures.
2) It is an indirect topology that relies on the perfect shuffle interconnection algorithm.



* Omega network with 8 processing * elements.