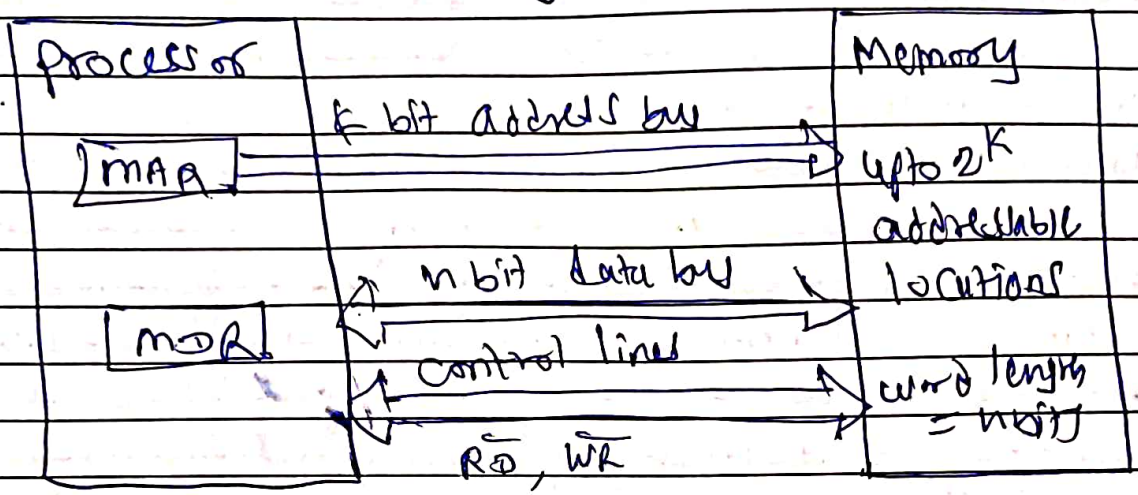
UNIT IV Memory organization



connection between memory & processor.

Characteristics of Memory Systems

1 Location — CPU — CPU registers; Cache memory
internal — main memory
External — Secondary memory - magnetic disk
I/O controller.

2 Capacity —
Word size — 8, 16 & 32 bits
No. of Word — memory capacity is
4K×8 word size is 8. & No. of words
are 4K = 4096

3 Unit of Transfer — It is maximum number of bits that can be read or written into the memory at a time.
(1) word
(2) block.

4. Access Method —
The order or sequence in which information can be accessed.
① Random Access — If storage locations can be accessed in any order and access time is independent of the location being accessed. the access method is known as random access. (RAM)
② Serial Access — If storage locations can be accessed only in a certain predetermined sequence, the access method is known as serial access.
eg. magnetic disk or tapes, CD-ROMs.

V Performance —
① Access Time — it is time taken by memory to complete read/write operation from the instant that an address is sent to the memory.
② Memory cycle time —
③ Transfer rate — It is defined as the rate at which data can be transferred into or out of a memory unit.

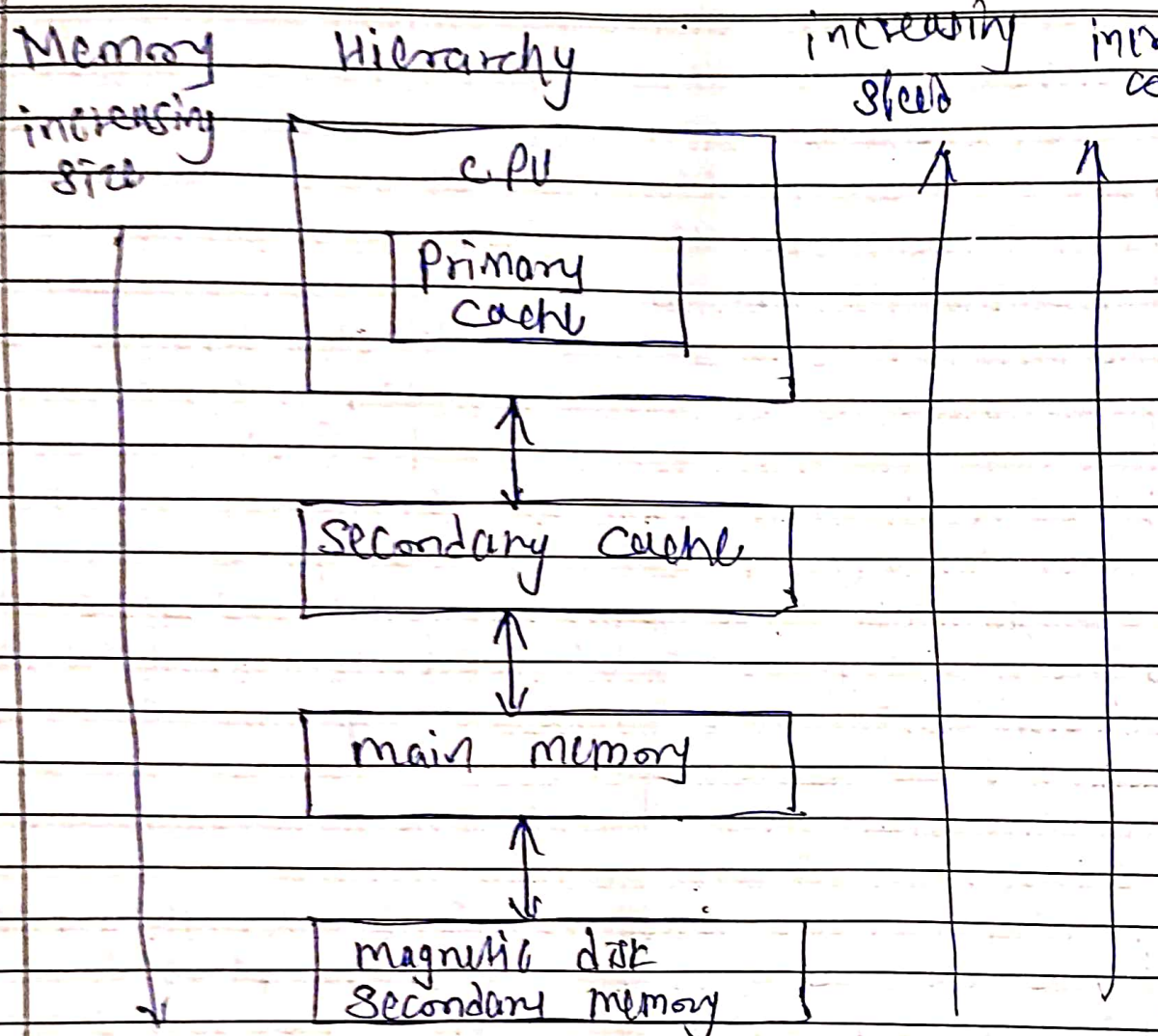
VI physical characteristics —
① Volatile / non volatile —
If memory can hold data even if power is turned off, it is called nonvolitile memory. otherwise it is called volatile memory.
② Erasable / Non erasable — The memories in which data is once programmed cannot be erased are called as non erasable memories. if data is the memory is erasable then memory is called erasable memory.

Comparision between serial & Random access memories.

Serial access memories	Random access memories
① use serial access method	① use random access method
② memory is organized into units of data, called records.	② each storage location in the memory has an unique address. It can be accessed independently of the other locations.
③ memory access time is dependent on the position of storage location.	③ memory access time is independent of storage location being accessed.
④ The time required to bring the desired location into correspondance with a read-write head increases effective access time. slower.	④ memory access time is less.
⑤ cheaper than random access memories	⑤ Random access memories are comparatively costly.
⑥ Non volatile memories	⑥ may be volatile or non volatile depending on physical char.
⑦ magnetic tape is an example of serial access memories.	⑦ Semiconductor memories are random access memories.

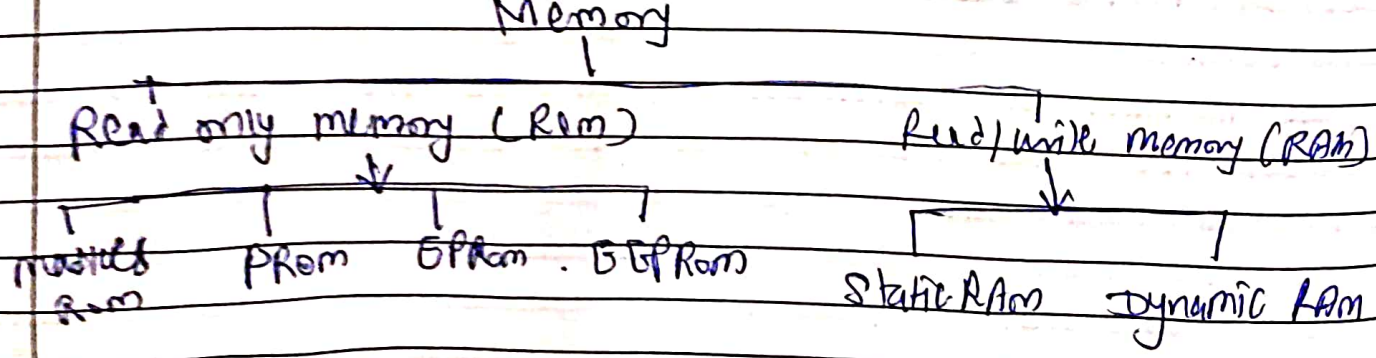
Memory Hierarchy

Memory Hierarchy



Main Memory organization

main memory consists of DRAMs supported with SRAM cache.



1. ROM – Read only memory

It is a read only memory. It is non volatile memory i.e. it can hold data even if power is turned off. Rom is used to store the binary codes for the sequence of instructions you want the computer to carry out and data such as look up tables. There are 4 types of Rom.

1 masked Rom
2 PRom
3 EPRom
4 EEPROM.

II PROM – Programmable Read only memory.

III EPROM – Erasable programmable Read only memory

IV EEPROM – Electrical Erasable programmable Read only memory.

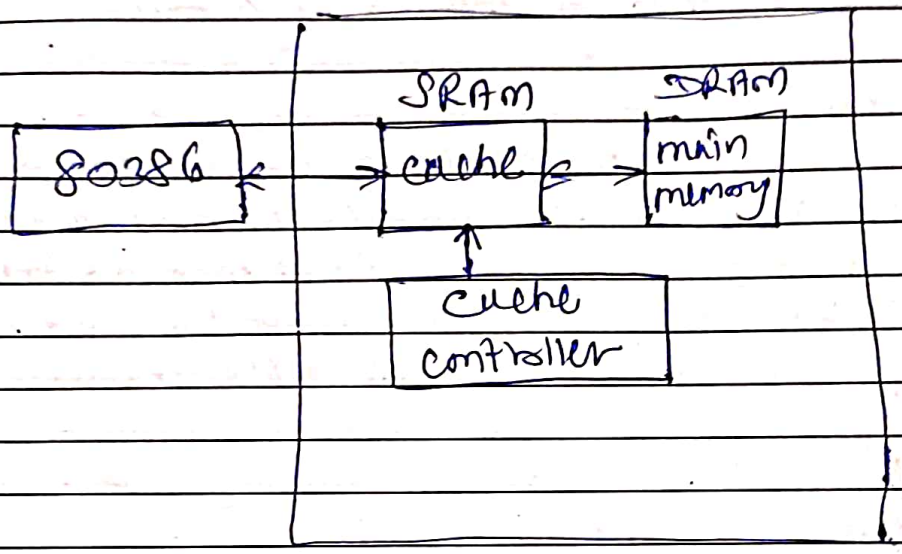
2 RAM – It is called read/write memory. It is volatile memory. i.e. It cannot hold data when power is turned off. There are 2 types of RAM

1 Static RAM
2 Dynamic RAM

Comparision between SRAM & DRAM

Static RAM	Dynamic RAM
① Static RAM contains less memory cells for per unit area	① Dynamic RAM contains more memory cells as compared to static RAM per unit area.
② it less access time hence faster memories	② its access time is greater than static RAM
③ Static RAM consists of number of FF .. each ff. Stores one bit	③ Dynamic RAM stories the data as a charge on the capacitor it consists of MOSFET & capacitor for each cell.
④ Refreshing circuitry is not required	④ Refreshing circuitry is required to maintain the charge on the capacitor.
⑤ cost is more	⑤ cost is less.

Cache Memory System.



Cache memory system.

A cache memory system includes a small amount of fast memory (SRAM) and large amount of slow memory (DRAM). This system is configured to simulate a large amount of fast memory.

1. Cache — This block consists of static RAM.
2. Main memory — This block consists of dynamic RAM
3. cache controller — This block implement the cache logic.

Elements of Cache design

1. Cache size
2. mapping function
3. replacement algorithm write policy
4. block size
5. No. of caches.

1. Cache size — The size of the cache should be small enough so that the overall average cost per bit is close to that of main memory alone.

large enough so that the overall average access time is close to that of the cache alone.

II Mapping Function —
The cache memory can store a reasonable number of blocks at any given time but this number is small compared to the total number of blocks in the main memory. There are two mapping functions commonly used.
1 Direct mapping
2 associative mapping

III Replacement Algorithm —
When new block is brought into the cache one of the existing blocks must be replaced by a new block.
There are 4 replacement Algorithms.
1 Least - Recently used (LRU)
2 First in First out (FIFO)
3 Least frequently used (LFU)
4 Random.

IV Write Policy — It is also known as cache updating policy. In cache system, two copies of the same data can exist at a time one in cache and one in main memory, if one copy is altered and other is not, two different sets of data become associated with the same address. To prevent this the cache system has updating systems such as: write through system, buffered write through system and write back system. The choice of cache write policy affect the design of cache.

V. Block Size –
It should be optimum for cache memory ~~size~~ system. ~~its performance is already~~

VI. Number of caches – when on chip cache is insufficient the secondary cache is used. The cache design changes as number of caches used in the system changes.

Mapping functions

The cache memory can store a reasonable number of blocks at any given time but this no. is small compared to the total number of blocks in the main memory. There are two main mapping techniques

1. direct mapping technique
2. Associative mapping technique.

I Direct mapping –

In this technique, each block from the main memory has only one possible location in the cache organisation. The block i of main memory maps to block i modulo 128 of the cache, whenever one of main memory blocks 0, 128, 256 ... is loaded in the cache, it is stored in cache block 0. blocks 1, 129, 257, are stored in cache block 1, & soon.

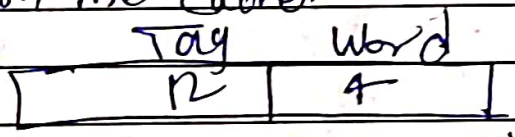
To implement such cache system, the address is divided into 3 fields.

Tag	Block	word
5	7	4

main memory address

II Associative Mapping - (Fully Associative mapping)

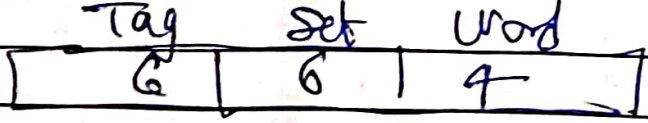
In this technique, a main memory block can be placed into any cache block position. As there is no fix block the memory address has only 2 fields word & tag. The 12 tag bits are required to identify a memory block when it is resident in the cache. The high order 12 bits of an address received from the CPU are compared to the tag bits of each block of the cache to see if the desired block is present. once the desired block is present the 4 bit word is used to identify the necessary word from the cache.



main memory address

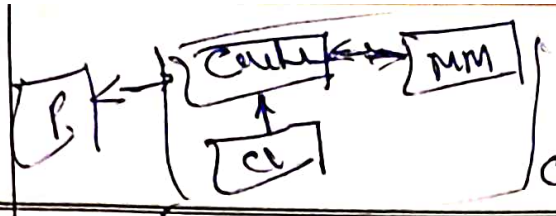
II Set associative mapping

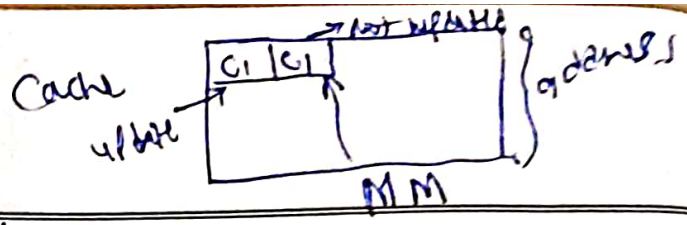
The set associative mapping is a combination of both direct and associative mapping. A block of data from any page in the main memory can go into particular block location of any direct mapping cache.



Comparision between mapping techniques

Direct mapping	Associative mapping	Set associative mapping
(1) Each block from the main memory has only one possible location in the cache.	(1) A block of data from main memory can be placed into any cache block position	(1) A block of data from main memory can go into a particular block location of any direct mapped cache
(2) Needs only one comparision.	(2) Needs comparision with all tag bits.	(2) Needs no. of comparisions equal to no. of blocks per set.
(3) Cache hit ratio decreased if Processor needs to access same memory location from two different pages of the main memory frequently.	(3) Cache hit ratio has no effect if Processor needs to access same memory location from two different Pages of main memory frequently.	(3) The effect of reduction in cache hit ratio in case of frequent access to the two different pages of the main memory is reduced.
(4) main memory address is divided into three fields - Tag, block, word.	(4) 2 Fields Tag & Word	(4) 3 Fields Tag, Set, word.
(5) Searching time is less	(5) Searching time is more	(5) Searching time increases with number of blocks per set.





Cache write/ updating

In a cache system, two copies of same data can exist at a time, one in cache and one in main memory. If one copy is altered and other is not, two different sets of data become associated with same address. To prevent this the cache system has updating systems such as write through system, buffered write through system and write back system.

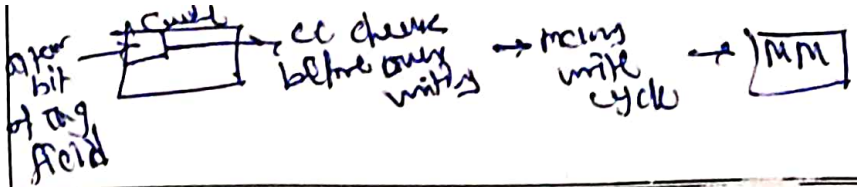
1 Write through system —

In write through updating system, the cache controller copies data to the main memory immediately after it is written to the cache. due to this main memory always contains a valid data and block in the cache can be overwritten immediately without data loss.

The write through is a simple approach. This approach requires time to write data in main memory with increase in bus traffic. This in effect reduces the system performance.

2 Buffered write through system

In buffered write through system the processor can start a new cycle before the write cycle to the main memory is completed. This means that the write accesses to the main memory are buffered. In such systems, read access which is a cache hit can be performed simultaneously when main memory is updated. However two consecutive write operations to the main memory or read operation with cache "miss" require the processor to wait.



3 Write back System. —

In a write back system, the alter bit in tag field is used to keep information of the new data. Cache controller checks this bit before overwriting any block in the cache. if it is set the controller copied the blocks to main memory before loading new data into the cache.

due to one time write operation number of write cycles are reduced in write back system. but this system has following disadvantages.

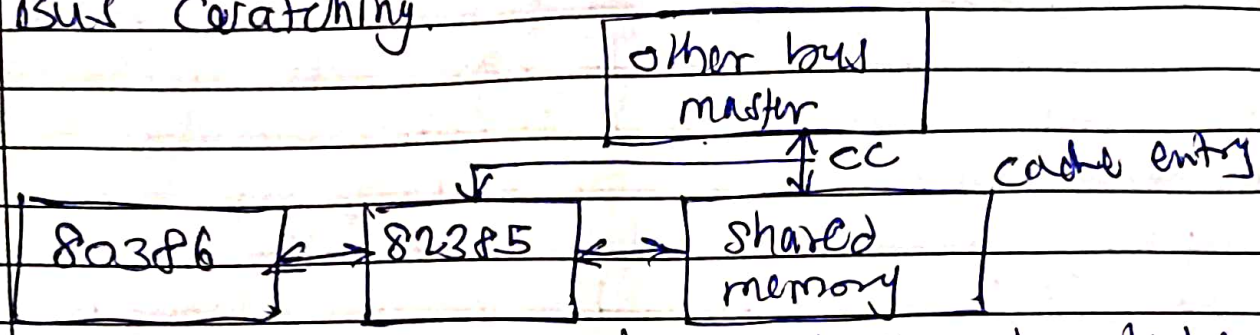
1 write back cache controller logic is more complex
2 It is necessary that all altered blocks must be written to the main memory before another device can access these blocks in main memory.
3 In case of power failure, the data in the cache memory is lost.

Cache coherency

Cache updating systems eliminates data inconsistency in the main memory caused by cache write operations. in multiprocessor systems another bus master can take over control of the system bus. This bus master could write data into a main memory blocks which are already held in the cache of another processor. when this happens the data in the cache no longer match those held in main memory creating inconsistency. The 80386 supports 4 different approaches to prevent data inconsistency that is to protect cache coherency

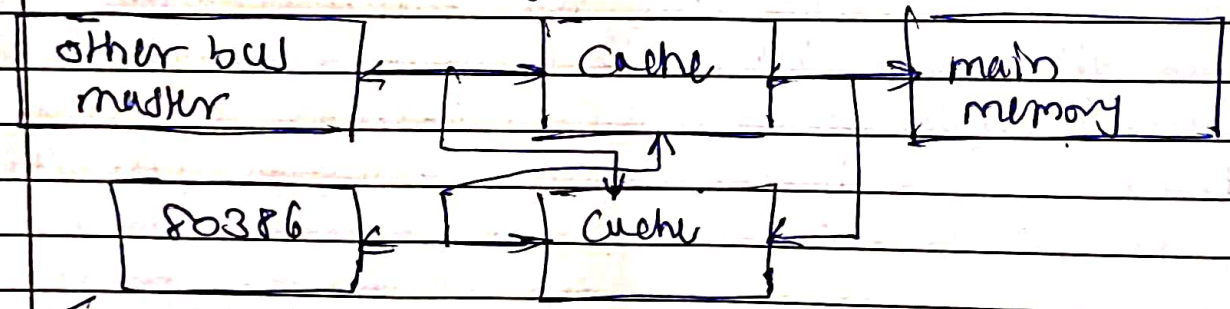
1 bus watching (snooping) 2 Hw transparency
3. Non cacheable memory 4 Cache flushing

I Bus Coratching.



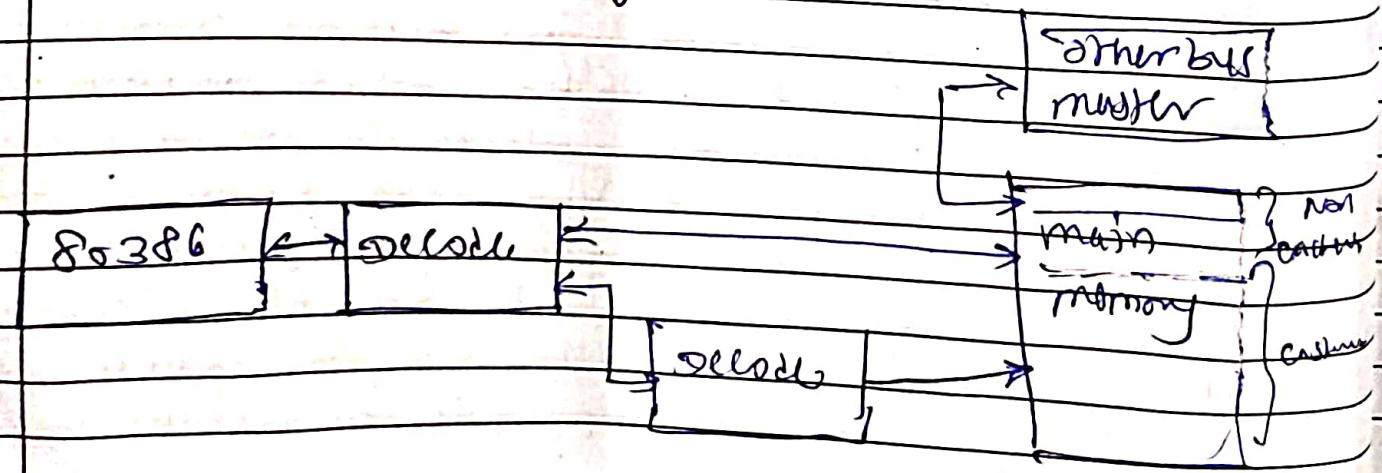
In bus watching, cache controller invalidates the cache entry, if another master writes to a location in shared memory which also resides in the cache memory.

II HW transparency —



In HW transparency accesses of all devices to the main memory are routed through the same cache or by copying all cache writes both to the main memory & to all other caches that share the same memory.

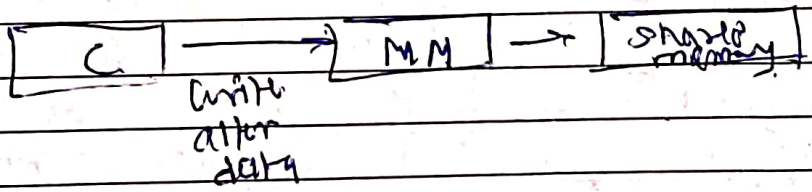
III Non cacheable memory —



The 80386 DX can partition its main memory into a cacheable and non cacheable memory. By designing shared memory as non cacheable memory cache coherency can be maintained. Since shared memory is never copied into cache.

iv) Cache flushing —

To avoid data inconsistency, a cache flush writes any altered data to the main memory and caches in the system are flushed before a device writes to shared memory.



Replacement Algorithms-

When a new block is brought into the cache, one of the existing blocks must be replaced by a new block. in case of of direct mapping cache, we know that each block from the main memory has only one possible location in the cache Hence there is no choice. The previous data is replaced by the data from the same memory location from new page of the main memory. but for associative and set associative technique there is a choice of replacing existing block. The choice of replacement of the existing block should be such that the probability of accessing same block must be very less. The replacement algorithms do the task of selecting the existing block which must be replaced.

There are 4 replacement algorithm.

1. Least Recently used (LRU)
2. First - in first - out (FIFO)
3. Least Frequently - used (LFU)
4. Random.

1 Least Recently Used - LRU

In this technique the block in the set which has been in the cache longest with no reference to it is selected for the replacement. we assume that most recently used memory locations are more likely to be referenced again. This technique can be easily implemented in the two way set associative cache organization.

II First - in - first - out - FIFO

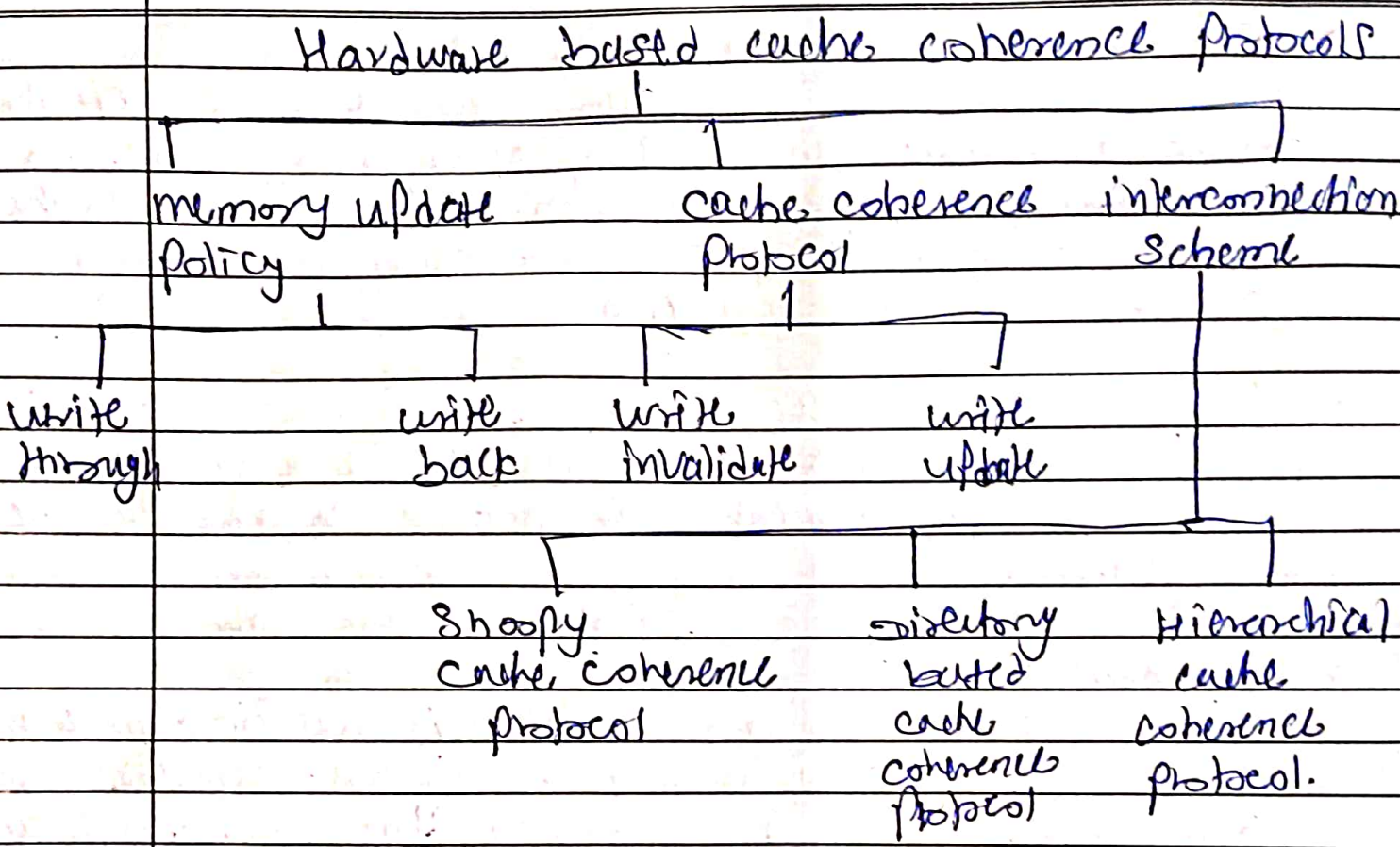
This technique uses same concept that stack implementation used in the microprocessors. In this technique, the block which is first loaded in the cache amongst the present blocks in the cache is selected for the replacement.

III Least frequently used - LFU

In this technique the block in the set which has the fewest references is selected for the replacement.

IV Random -

There is no specific criteria for replacement of any block. The existing blocks are replaced randomly. Simulation studies have proved that random replacement algorithm provides only slightly inferior performance to algorithms.



In a coherent microprocessor system, the caches provide both migration and replication of shared data items. replication allows that multiple copies of the same data item reside simultaneously in different local caches, in order to reduce both latency of access and contention for a read shared data item. migration implies a single copy of a data item that must be moved to the accessing site for exclusive use, that reduces the latency to access a shared data item that is allocated remotely.

In multiprocessor systems supporting both migration and replication of shared data items, coherence is maintained by introducing HW based or SW based protocols.

The protocols used to maintain coherence for multiple processors are called cache coherence protocols. Hardwired based protocols can be classified according to their.

1 memory update policy
2 cache coherence protocol.
3 interconnection scheme.

I Memory update policy — In cache system two copies of same data may exist at a time, one in cache and one in main memory, if one copy is altered and other is not, a different sets of data become associated with same address. To prevent this memory system uses two types of memory update policies!

1 write through policy
2 write back policy.

In write through policy the cache controller copies data to the main memory immediately after it is written to the cache. due to the main memory always contains a valid data. and any block in the cache can be overwritten immediately without data loss. The write through policy maintains consistency between the main memory and cached.

In write back policy the alter bit in the tag field is used to keep information of the new data memory contents are not updated immediately. memory contents are updated eventually when the modified block in the cache is replaced or invalidated.

II Cache coherence protocol –

Cache coherence protocols are used to update copies in the multiprocessor system. Two protocol are used for updating the cache copies of a data.

1. write update protocol
2. write invalidate protocol.

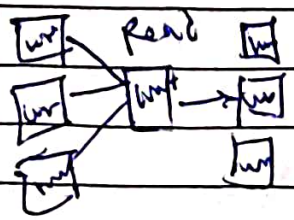
write invalidate protocol

1. multiple writes to the same word with no intervening reads require only one initial invalidation.
2. multiword cache blocks Only the first write to any word in the block needs to generate an invalidate. This protocol works on cache blocks.
3. Any read to invalidated data forces to fetch a new copy of data. This takes comparatively longer time.

1. write update protocol

multiple writes to the same word with no intervening read require multiple write broadcasts.

2. multiword cache blocks, each word written in a cache block require a write broadcast. This protocol works on individual word.
3. The written data is immediately updated in other caches. Thus any read to the data in the cache is valid. Thus less time is required to read the data.



III Interconnection Scheme —

Depending on the nature of the inter-connection network HW based protocols are classified 3 basic classes.

I Snoopy cache coherence Protocol —

Snoopy cache coherence protocols are very popular in shared bus multiprocessors due to their relative simplicity. every cache that has a copy of the data from a block of physical memory also has copy of a sharing status of the block & no centralized state is kept. This protocol is typically used in single bus based shared memory system where consistency commands are broadcast via the bus, & all cache controller monitor or snoop on the bus to determine whether or not they have a copy of a block that is requested on the bus.

II Directory based cache coherence Protocol —

Large interconnection nws cannot support broadcasting efficiently and therefore a mechanism is needed that can directly forward consistency command to those caches that contain a copy of the updated data. For this purpose the sharing status of a block of physical memory is kept in just one location called the directory.

III Hierarchical cache coherence Protocol.

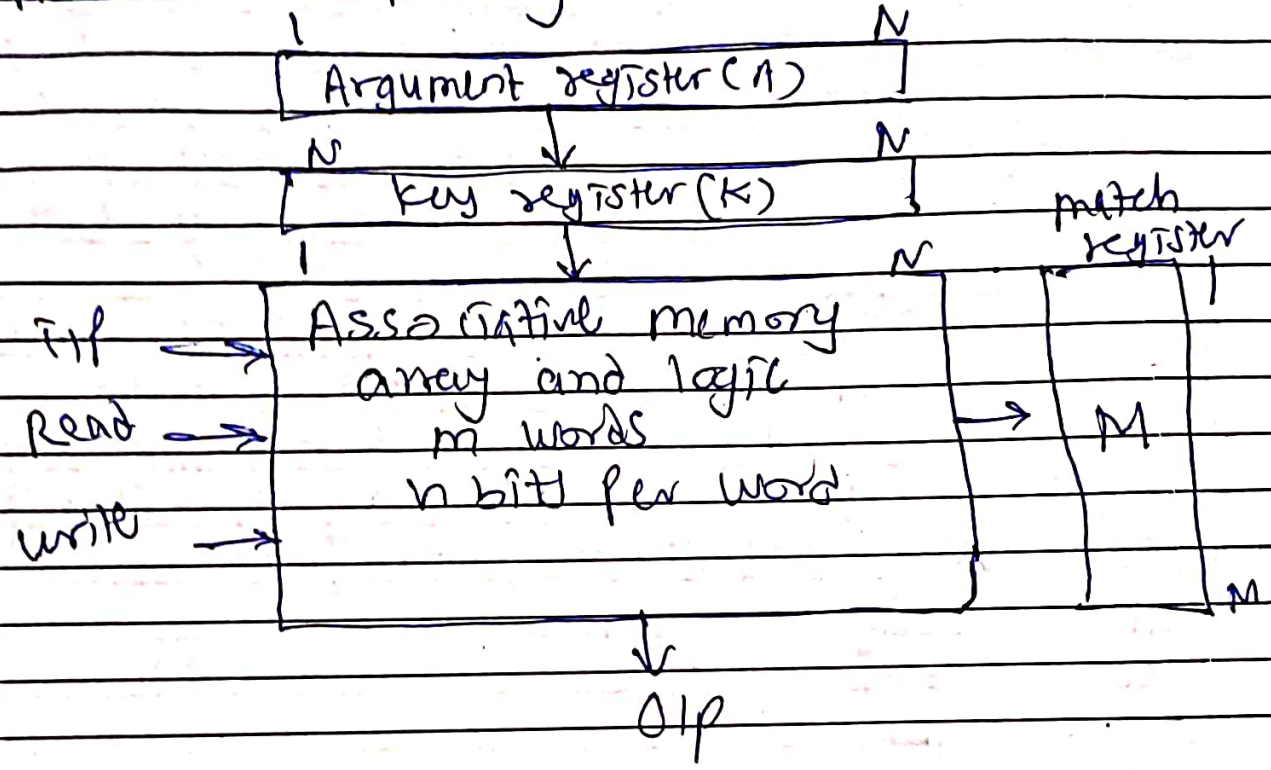
This protocol tries to avoid the application of the costly directory protocol but still provide high scalability. It proposes multiple bus nws with the application of hierarchical cache coherence protocols that are generalized or extended version of the single bus based snoopy cache protocol.

MESI: protocol

Illinois invalidate protocol is also called as MESI protocol.

1. Modified – The line in the cache, different from main memory is modified & this line is available only in this cache.
2. Exclusive – The line in the cache is same as that in main memory & it is not present in any other cache.
3. Shared – The line in the cache is same as that in main memory & the same line may be present in one or more other caches.
4. Invalid – The line in the cache does not contain valid data.

Associative memory



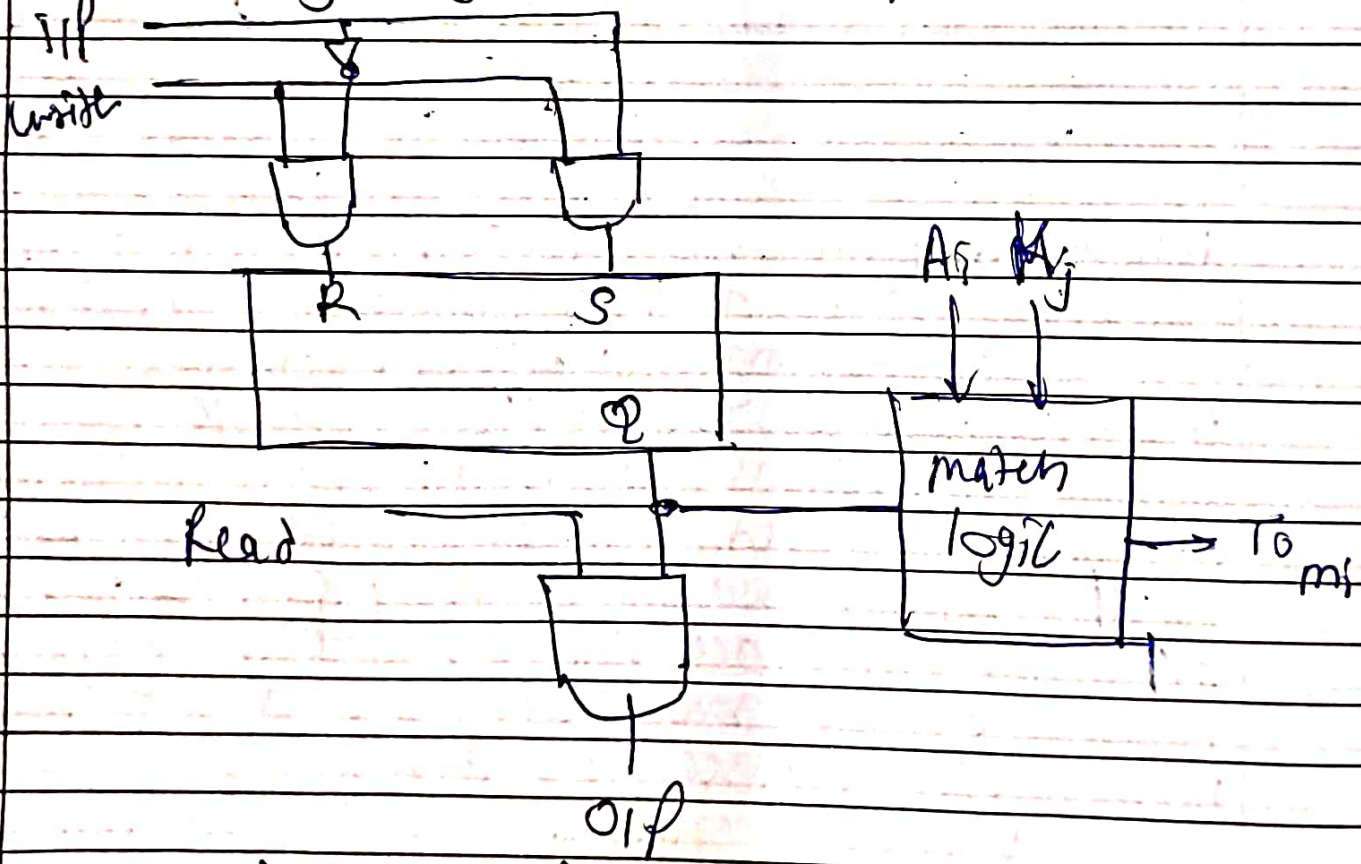
Block diagram of associative memory.

The time required to find an object stored in memory can be reduced considerably if objects are selected based on their contents, not on their locations. A memory unit accessed by the content is called an associative memory or content addressable memory (CAM). This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location.

block diagram of associative memory array with match logic for m n bit words & associated registers. The argument register (A) and key register (K) each have n-bits per word. Each word in memory is compared in parallel with the content of the argument register.

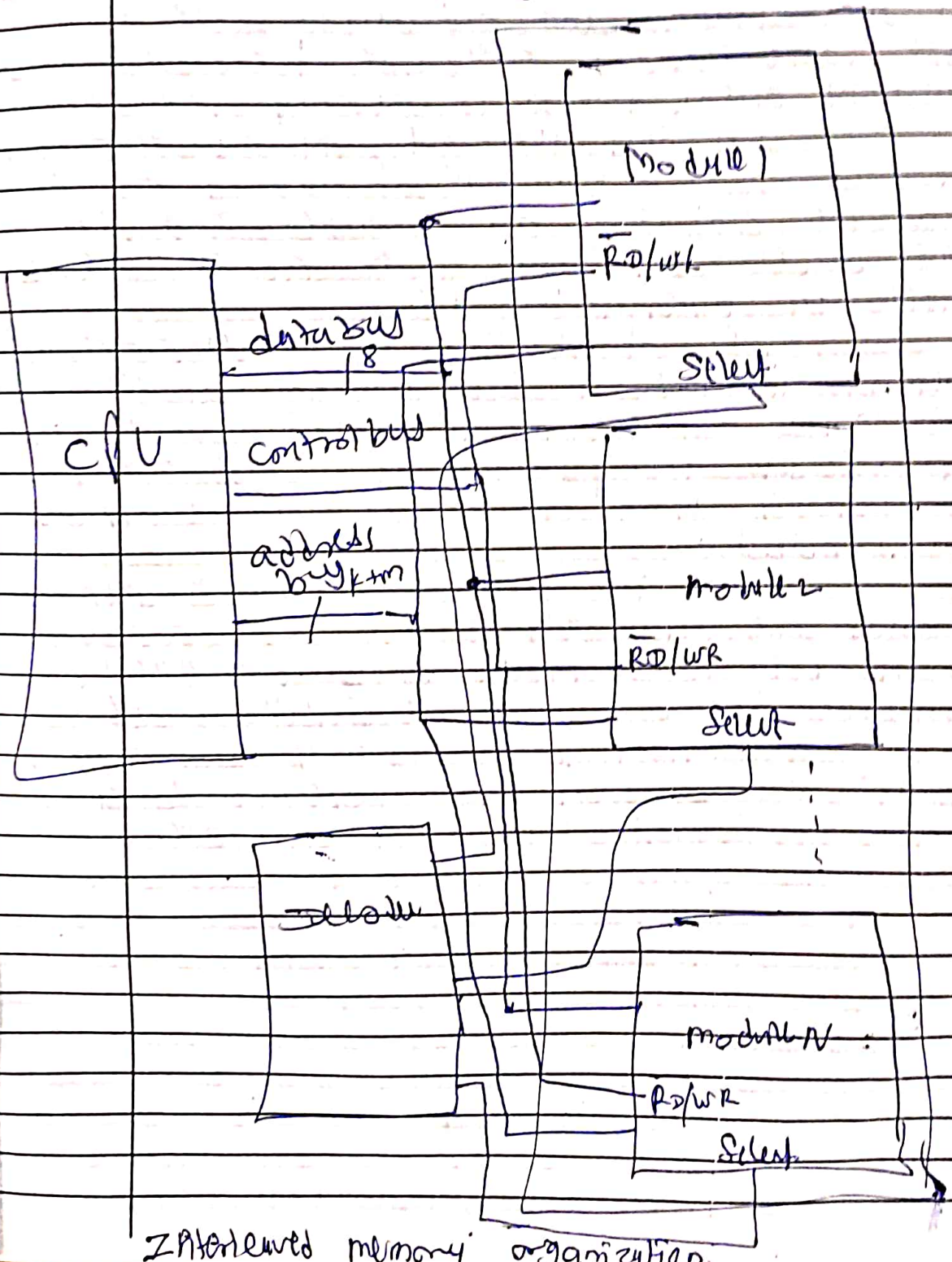
The words that match with the word stored in the argument register set a corresponding bits in the match register. Therefore reading can be accomplished by a sequential access to memory for those words whose corresponding bits in the match register have been set.

The key register provides a mask for choosing a particular field or bits in the argument word. only those bits in the argument register having 1's in their corresponding position of the key register are compared.



Internal organization of typical cell of associative memory

Interleaved memory



Interleaved memory organization.

The memory access time is the bottleneck in the system. And one way to reduce it is to use cache memory. Alternative technique to reduce memory access time is is memory interleaving.

In this technique, the main memory is divided into a number of memory modules and the addresses are arranged such that the successive words in the address space are placed in different module.

Most of the times cpu access consecutive memory locations. In such situation address will be to the different modules. These modules can be accessed in parallel, the average access time of fetching word from the main memory can be reduced.

The low order k bits of the memory address are generally used to select a module & the high order m bits are used to access a particular location within the selected module.

The effect of interleaving is substantial it does not speed up memory operations by a factor equal to the number of the module. It