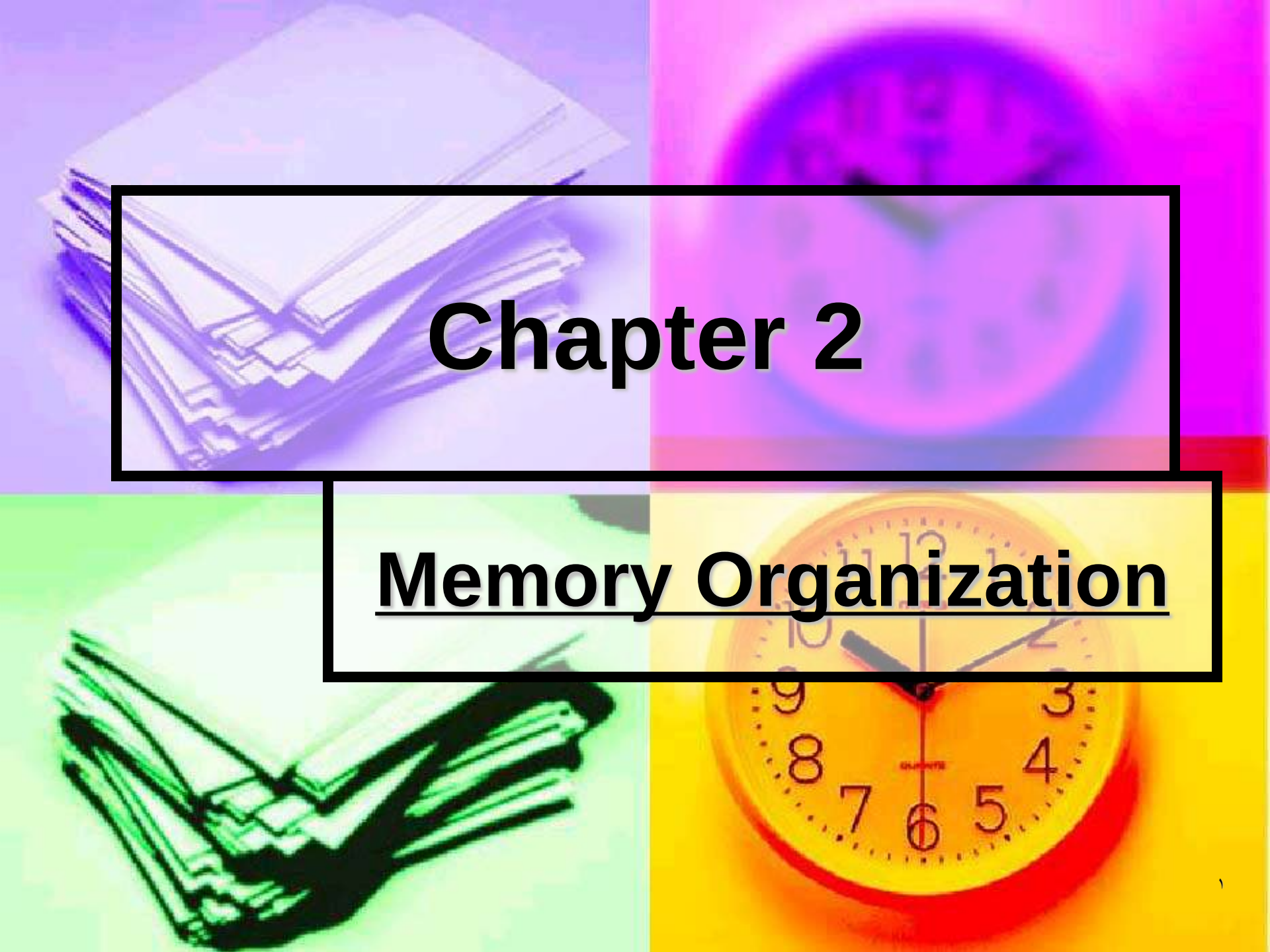
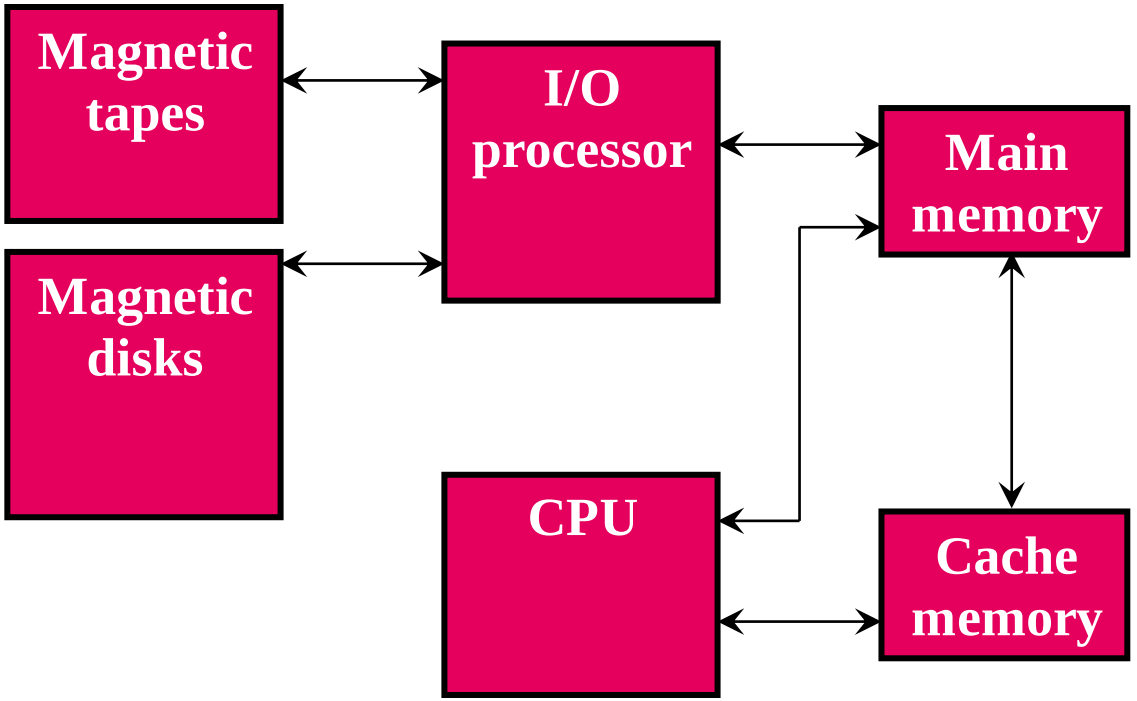
Chapter 2

Memory Organization



Memory Hierarchy

- Every computer contain many levels of memory, which is called memory hierarchy.
- Main memory, Cache, Auxiliary storage.

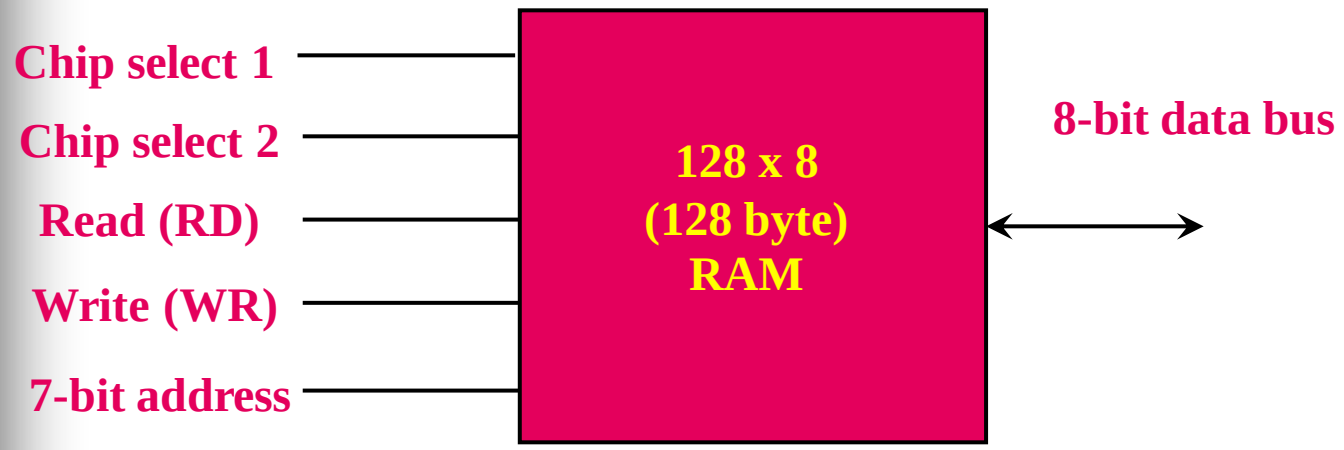


Main Memory

- The main memory is the central storage unit in a computer system.
- It is a relatively large and fast memory used to store programs and data during the computer operation.
- Most of the main memory in a general-purpose computer is made up of RAM integrated circuit chips, but a portion of the memory may be constructed with ROM chips.
- RAM and ROM chips are available in a variety of sizes.

RAM Chips

- A RAM chip is faster when communicating with CPU than dealing with Auxiliary devices directly.
- If the memory needed for the computer is larger than the capacity of one RAM chip, we combine a number of chips to form required memory size.
- If we have many RAM chips, we must choose to access one of them
- We use control inputs to select the chip only when needed.



$128 = 2^7$ Address = 7 bits

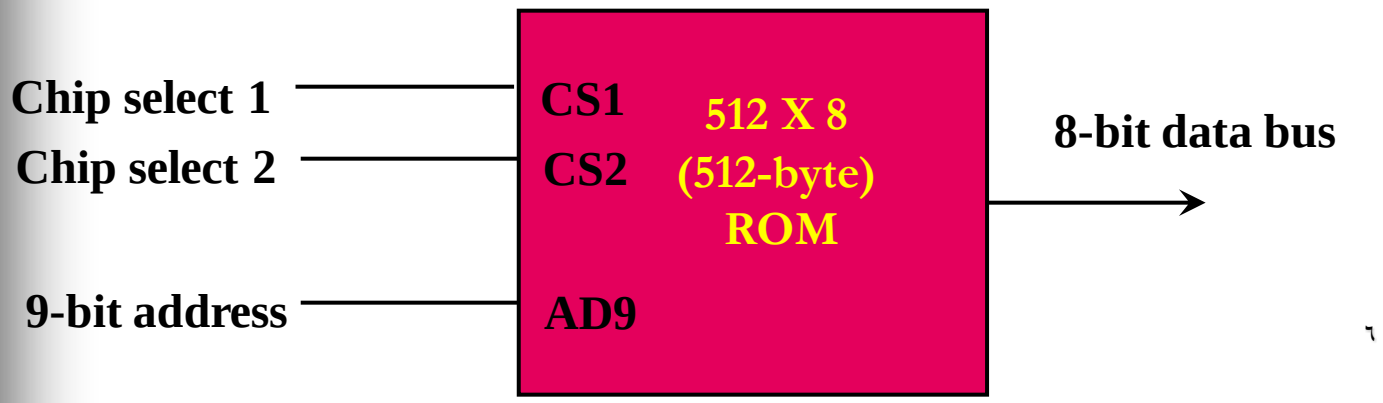
8 = 1 byte of data

RAM chip function table

CS1	CS2	RD	WR	Memory function	State of data bus
0	0	X	X	Inhibit	High-impedance
0	1	X	X	Inhibit	High-impedance
1	0	0	0	Inhibit	High-impedance
1	0	0	1	Write	Input data to RAM
1	0	1	X	Read	Output data from RAM
1	1	X	X	Inhibit	High-impedance

ROM Chips

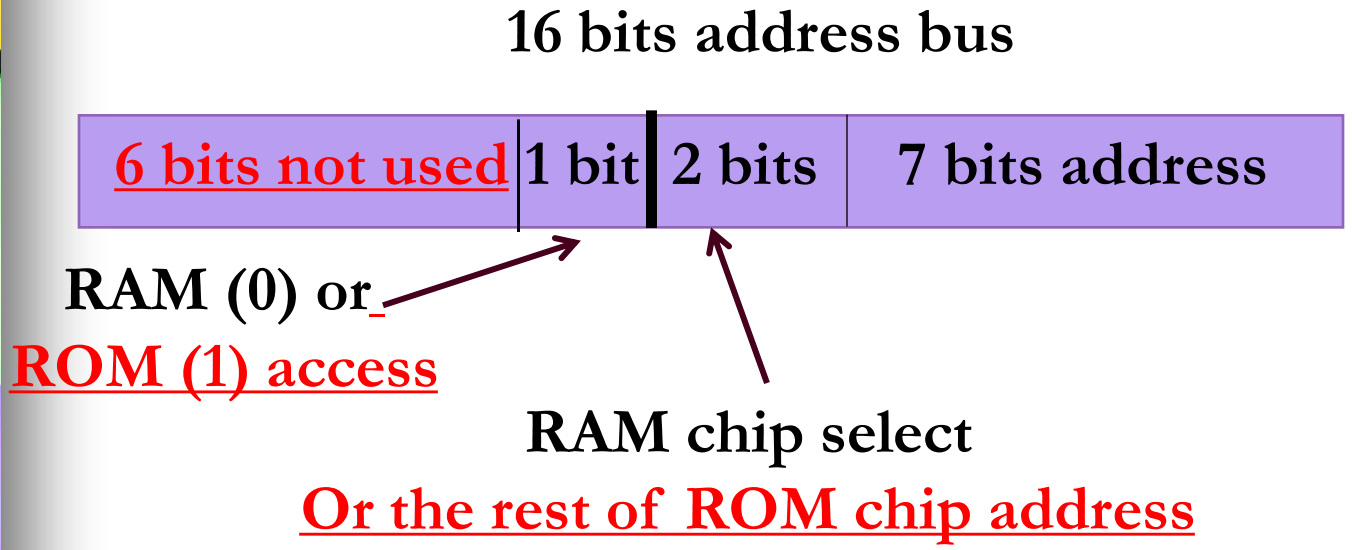
- A ROM chip is organized externally in a similar manner. ROM can only read, the data bus can only be in an output mode.
- For the same–size chip, it is possible to have more bits of ROM than of RAM, because the internal binary cells in ROM occupy less space than in RAM.
- For this reason, the diagram specifies a 512-byte ROM, while the RAM has only 128 bytes.
- address lines = 9 bits ($512 = 2^9$)
- The two chip select inputs must be CS1=1 and CS2= 0 for the unit to operate.
- Otherwise, the data bus is in a high-impedance state.

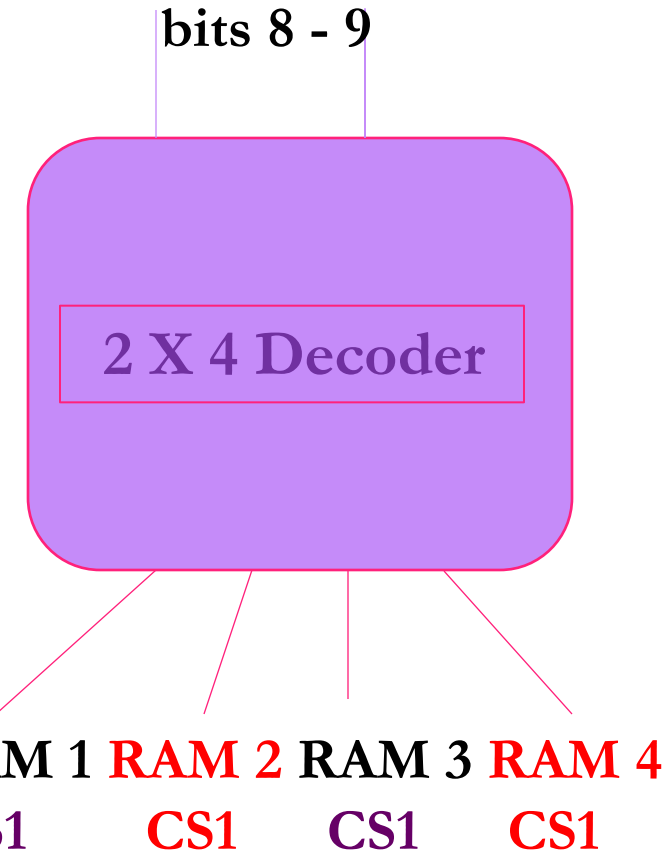


Practical Example : How CPU deal with RAM and ROM

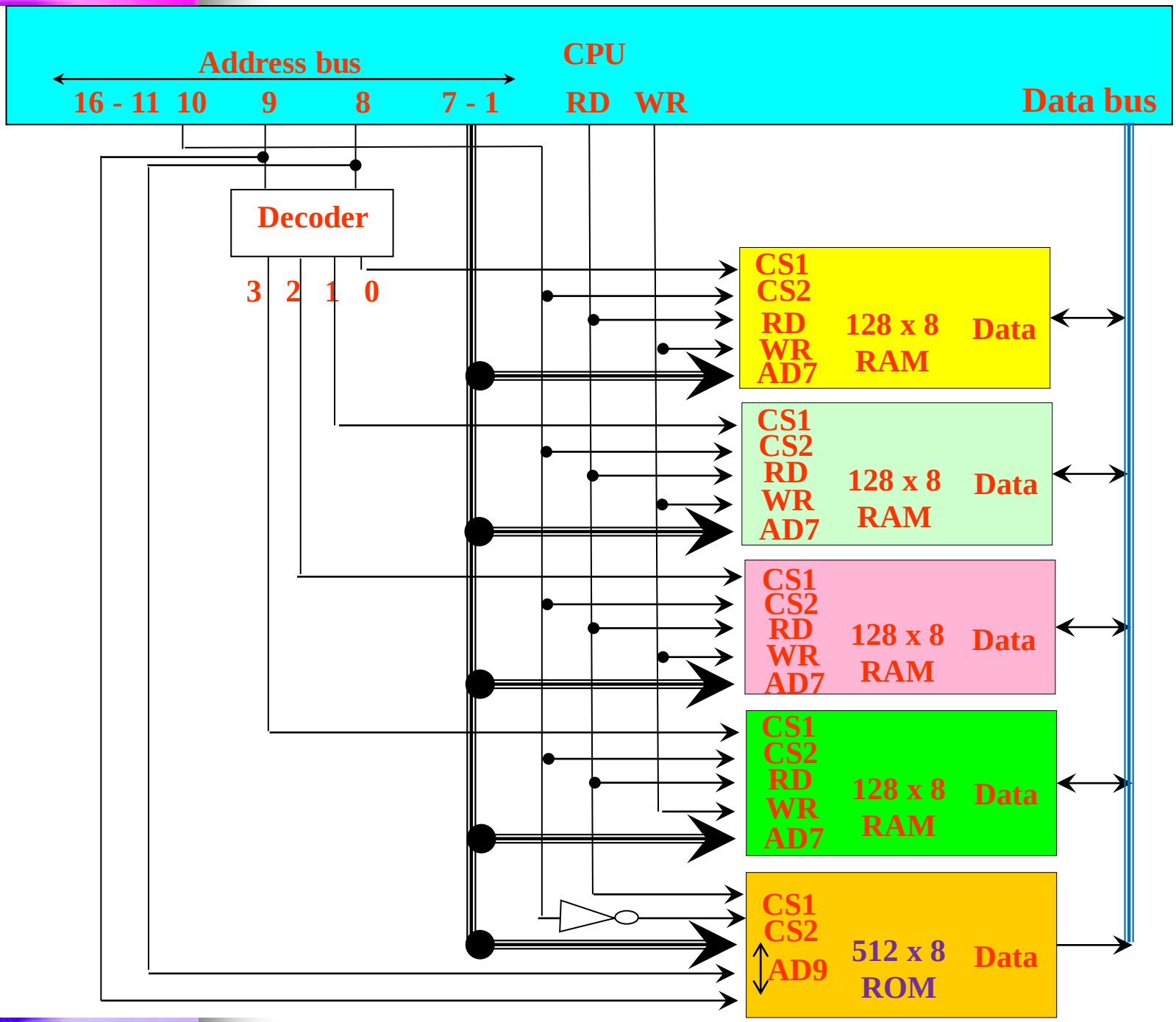
Assume that a computer system has:

- 512 bytes of RAM (we use 4 blocks of RAM of the same type 128X8)
- 512 bytes of ROM (single block).





We use a decoder to select one of the 4 RAM chips using bits 8 – 9 by connecting the output to CS1 of the RAM Chip


Address bus
16 - 11
10
9
8
7 - 1
CPU
RD
WR
Data bus
Decoder
3
2
1
0
CS1
CS2
RD
WR
AD7
128 x 8
RAM
Data
CS1
CS2
RD
WR
AD7
128 x 8
RAM
Data
CS1
CS2
RD
WR
AD7
128 x 8
RAM
Data
CS1
CS2
RD
WR
AD7
128 x 8
RAM
Data
CS1
CS2
AD9
512 x 8
ROM
Data




Component	Hexadecimal address	Address bus									
		10	9	8	7	6	5	4	3	2	1
RAM 1	0000-007F	0	0	0	X	X	X	X	X	X	X
RAM 2	0080-00FF	0	0	1	X	X	X	X	X	X	X
RAM 3	0100-017F	0	1	0	X	X	X	X	X	X	X
RAM 4	0180-01FF	0	1	1	X	X	X	X	X	X	X
ROM	0200-03FF	1	X	X	X	X	X	X	X	X	X

$(0000\ 0000\ 0000\ 0000)_2 = (0000)_{16}$
$(0000\ 0000\ 0111\ 1111)_2 = (007F)_{16}$

$(0000\ 0000\ 1000\ 0000)_2 = (0080)_{16}$
$(0000\ 0000\ 1111\ 1111)_2 = (00FF)_{16}$

$(0000\ 0001\ 0000\ 0000)_2 = (0100)_{16}$
$(0000\ 0001\ 0111\ 1111)_2 = (017F)_{16}$

Auxiliary Memory

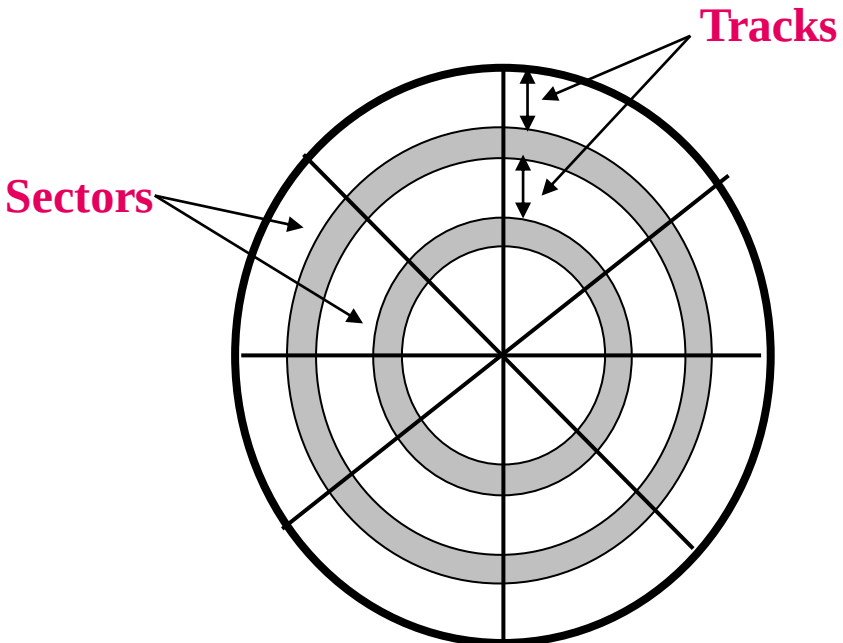
- The most common auxiliary memory devices used in computer system are magnetic disks, tapes, CDs and DVDs.
- The important characteristic of any device are its *access mode*, *access time*, *transfer rate*, *capacity*, and *cost*.
- The access time consists of a seek time required to position the read-write head to location and a transfer time required to transfer data to or from the device.
- Because the seek time is usually much longer than the transfer time, auxiliary storage is organized in records or blocks.

Auxiliary Memory

- A record is a specified number of characters or words.
- Reading or writing is always done on entire records.
- The transfer rate is the number of characters or words that the device can transfer per second, after it has been positioned at the beginning of the record.

1- Magnetic Disks

- A magnetic desk is a circular plate constructed of metal or plastic coated with magnetized material.
- Bits are stored in the magnetized surface in spots along concentric circles called tracks.
- The tracks are commonly divided into sections called sectors, usually has fixed size (512 byte is nearly universal sector size) .



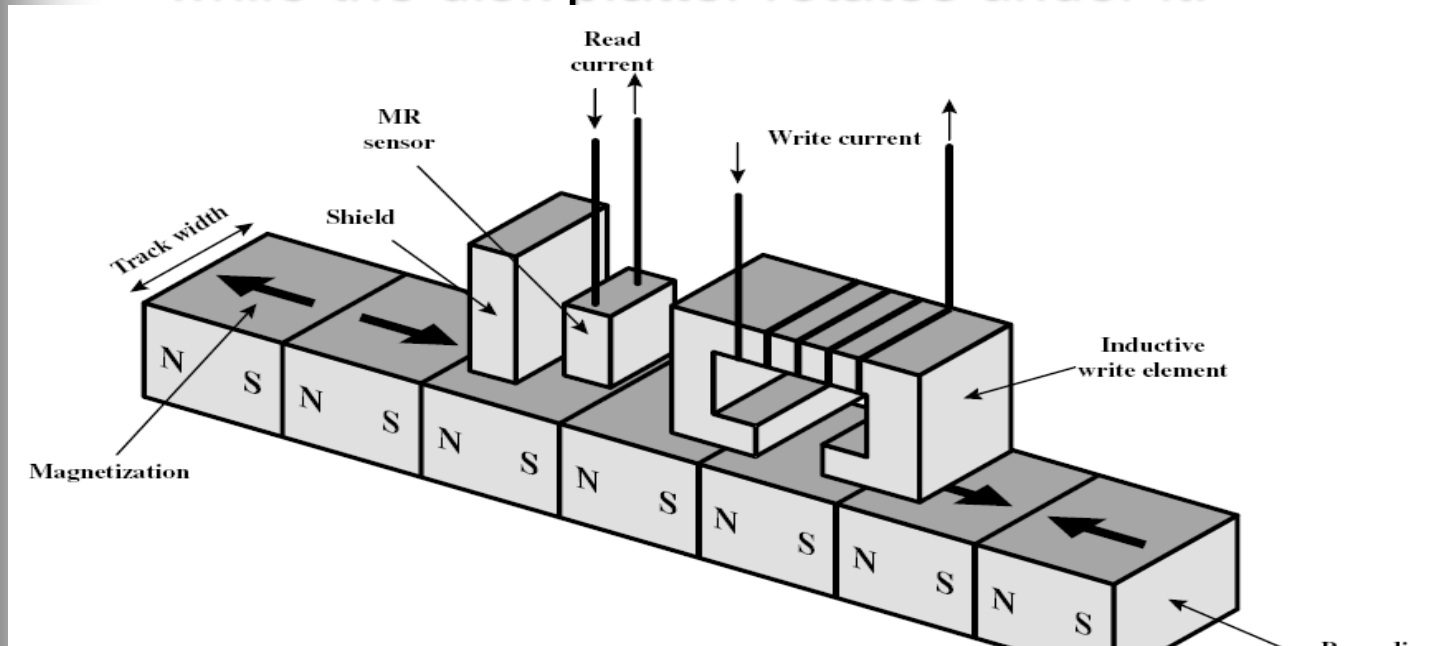
Magnetic Disks

- Heads: Some units use a single read/write head for each disk surface.
- In other disk systems, separate read/write heads are provided for each track in each surface.
- Disks that are permanently attached to the unit assembly and cannot be removed by the occasional user are called *hard disks.*
- A disk driver with removable disks is called a *floppy disk.*
- The disks used with a floppy disk driver are small removable disks made of plastic coated with magnetic recording material.

Magnetic Disks

Read/ write mechanisms:

- Data are recorded then later retrieved via conducting coil named head.
- The disk surface is coated with magnetizable material.
- Some systems have two heads: read and write.
- During a read or write the head is stationary while the disk platter rotates under it.



The write mechanism:

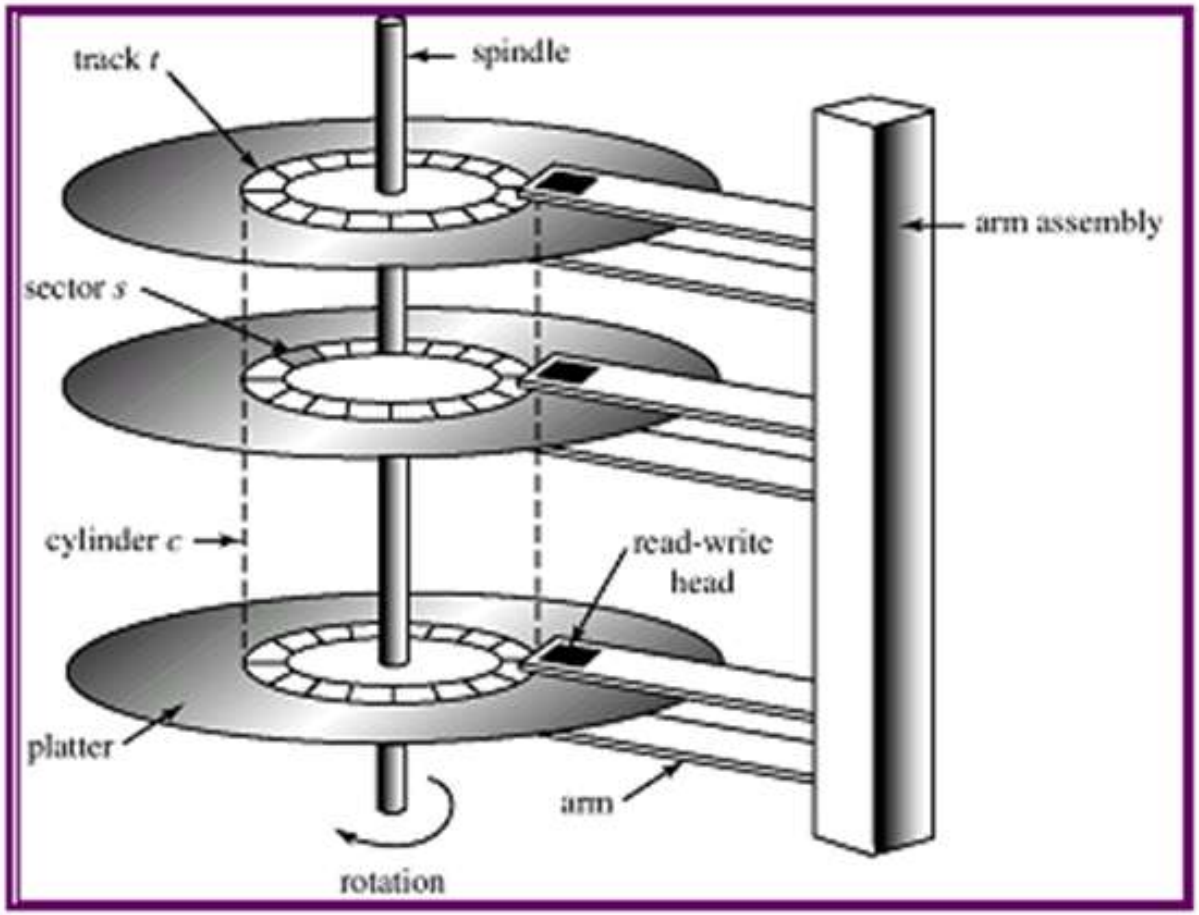
- Uses the idea that electricity flowing through a coil produces a magnetic field.
- Data pulses are sent to the write head, and magnetic patterns are recorded on the surface below, with different patterns for positive and negative currents.
- The 1 and 0 are stored as two different directions of magnetization, through sending currents in two different directions to the writing head.

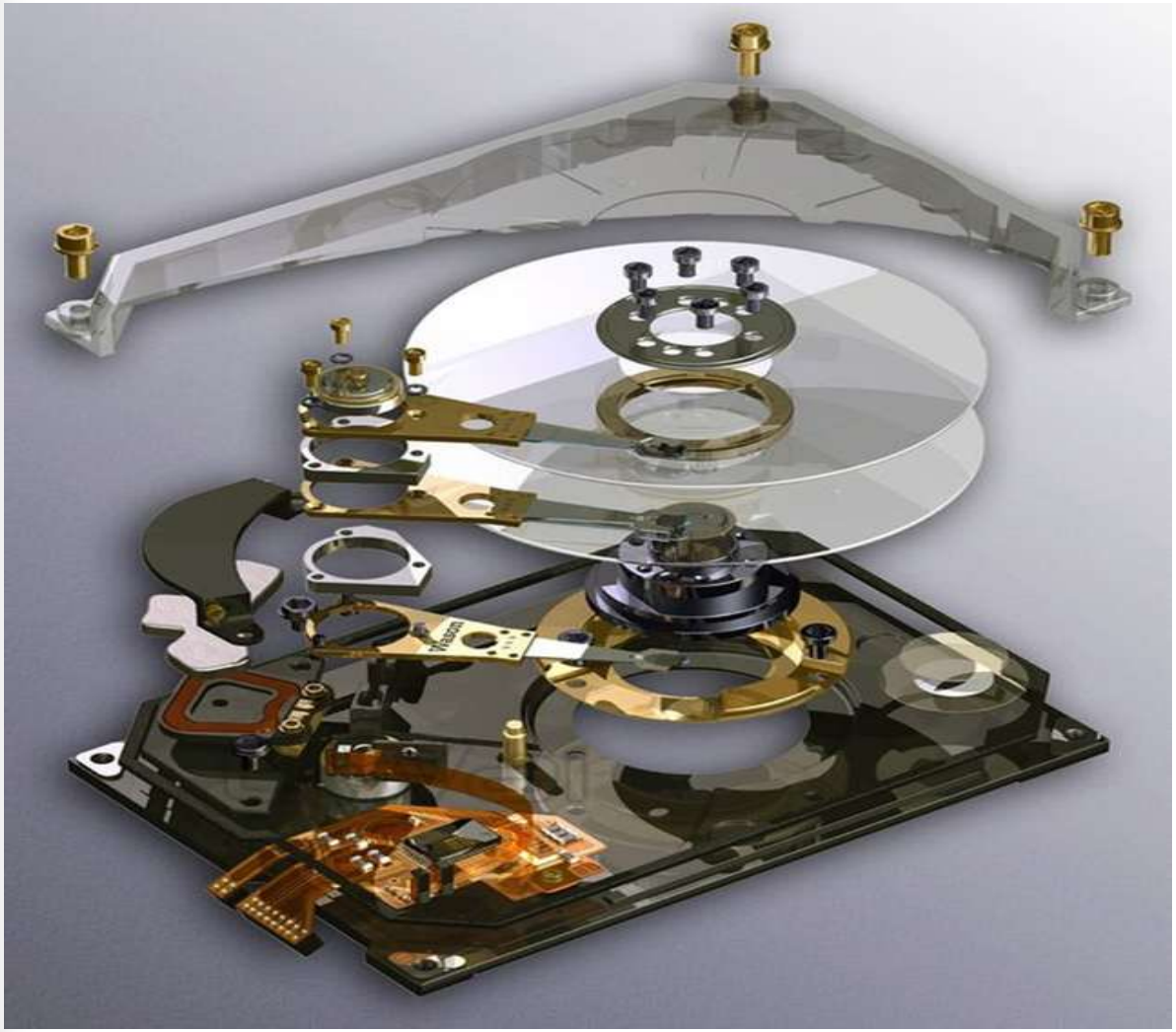
Magnetic Disks

The Read mechanism:

- The idea of reading is based on the fact that a magnetic field moving relative to a coil produces an electrical current in the coil.
- When a surface of a disk passes under the reading head, it generates a current of some polarity (direction) as the one already recorded.
- As seen the idea of reading is similar to that or writing, that is why some systems have single read/write head (floppy disk system and old hard disk systems).
- In new systems, they use MR head for reading.

Magnetic Disks Internal Structure





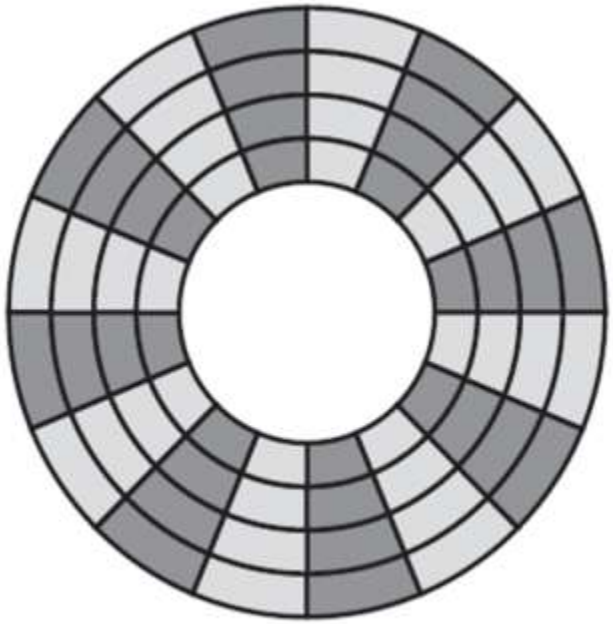
http://www.reuk.co.uk/Hard-Disk-Drive-Magnets-For-Wind-Turbines.htm

Magnetic Disks

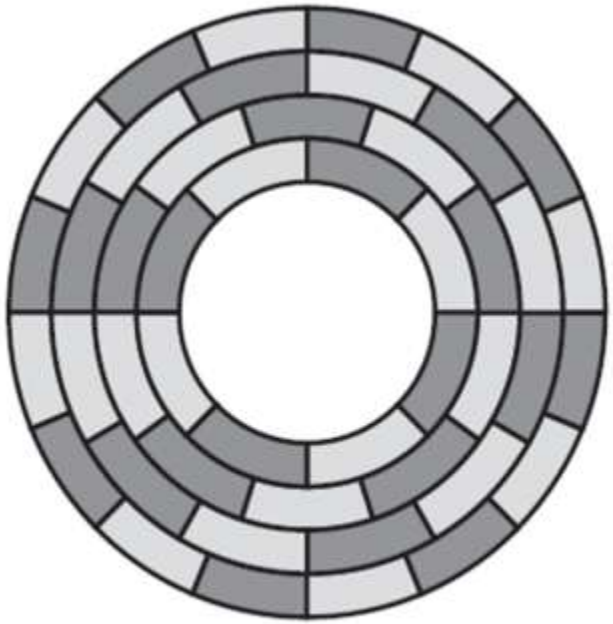
- Data organization on the disk has two methods:

1- constant angular velocity

- The disk is divided into a number of pie-shaped sectors.
- Each track are divided into equal number of sectors.
- Thus, the internal track sector's size are very small compared to the outer track sector.
- Advantage: ease of reaching the data
- Disadvantage: amount of data stored on the outer tracks is the same ass what can be stored on the short inner tracks.
- Density of data on track is limited by the inner sector size.



(a) Constant angular velocity



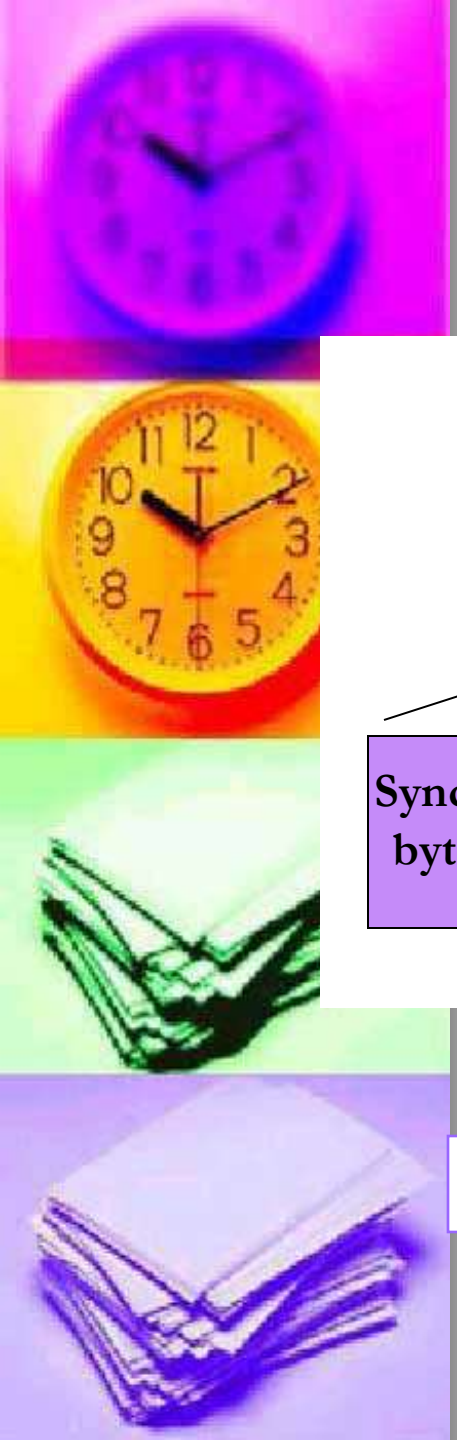
(b) Multiple zoned recording

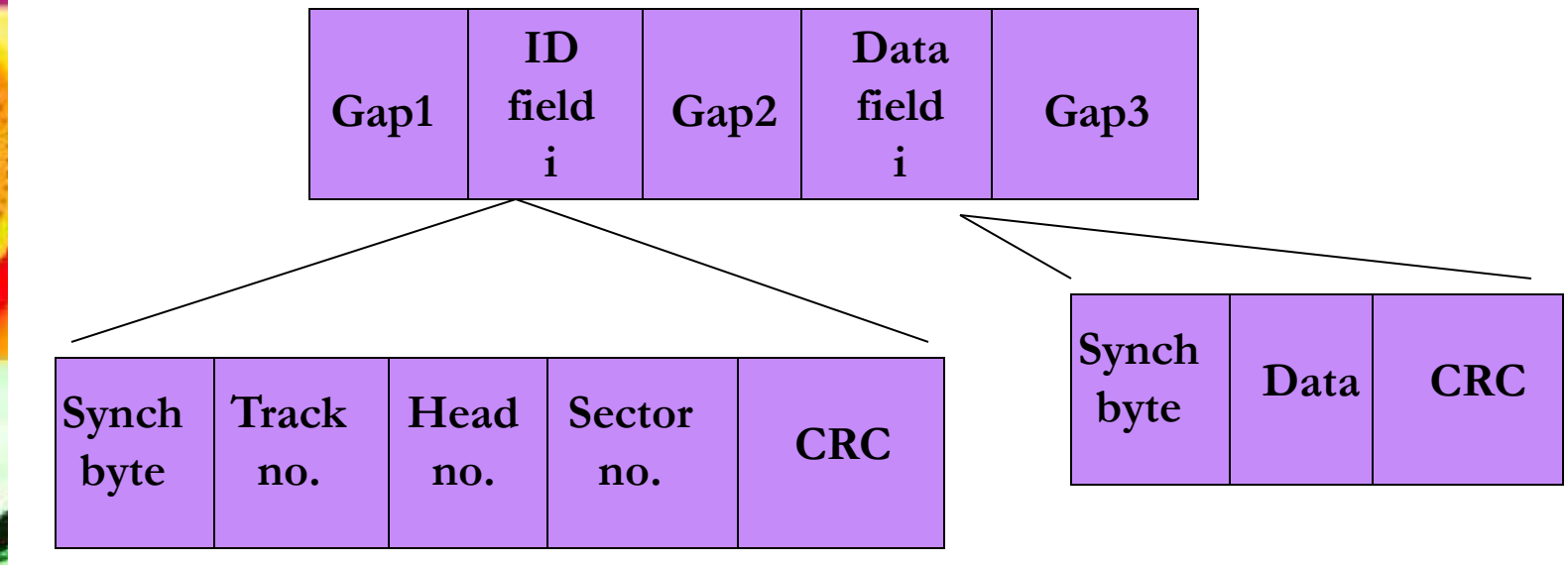
Magnetic Disks

2- multiple zoned recording

- Used to increase the track storage density.
- The disk surface is divided into zones (typically 16).
- Within each zone, the number of bits per track is constant.
- Zones near the center have less bits (less sectors) than the outer zones.
- Advantage: The overall capacity is higher.
- Disadvantage: the circuit used for reading and writing is more complex.
- We need a mean to locate sector positions within the track, so we need to identify the start and end of each sector.

Disk track format





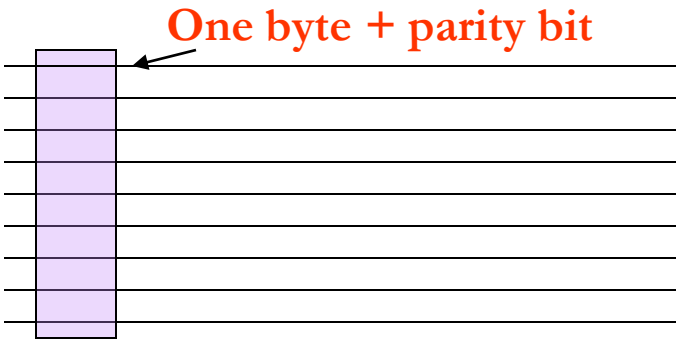
Hard Disk + multiple heads (upper and lower)

2- Magnetic Tape

- The tape is a strip of plastic coated with a magnetic recording medium.
- Bits are recorded as magnetic spots on the tape along several parallel tracks.
- A tape unit is addressed by specifying the record number and the number of characters in the record. (sequential access)
- Records may be of fixed or variable length.

Magnetic Tape(Extra)

- It is analogous to tape recorder system.
- Old tapes had 9 Tracks.
- This enabled storing one byte (8 bits) at parallel with the ninth bit as parity bit.
- Later, the tap had 18 or 36 tracks, corresponding to digital word size.
- This type of recoding is called parallel recording.



Magnetic Tape

- Most modern systems use serial recording.
- Data are laid out as a sequence of bits along each track, as done with magnetic disks.
- Information is recorded and read in contiguous blocks referred to as records.
- Blocks are separated by gaps called interrecord gaps.
- The tap is formatted to assist in locating records.
- The recoding technique used in serial tapes is referred to as serpentine recording.

Magnetic Tape

- The data is recoded along the whole length of tape.
- When the end of the tape is reached, the heads are repositioned to record a new track, but this time the recording is one the opposite direction.
- To increase speed, the read-write head is capable of reading and writing a no. of adjacent tracks simultaneously (2-8 tracks).
- Data are still recorded serially along individual tracks, but blocks in sequence are stored on a adjacent tracks.

Optical memory

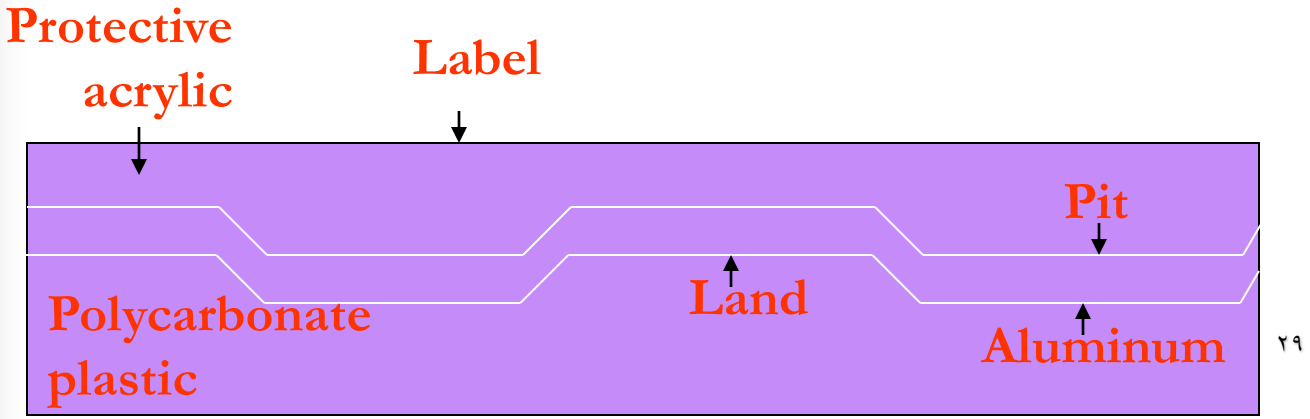
- Compact disk (CD) in 1983.
- Nonerasable disk, store more than 60 mins. of audio information on one side.

CD-ROM (Read only):

- CD and CD-ROM has the similar technology.
- But CD-ROM players are more rugged and have error correction devices.
- The disk is made from resin, such as polycarbonate.
- The digital data is imprinted as a series of microscopic pits on the surface of the polycarbonate.
- The master disk is imprinted with a finely focused, high-intensity laser.

3- CD-ROM

- The master used to make a die to stamp out copies onto polycarbonate.
- The pitted surface is then coated with a highly reflective surface, usually aluminum. This shiny surface is protected against dust and scratches by a top coat of clear acrylic.
- Finally, a label can be silkscreened onto the acrylic.



Reading data from CD

- Read using low-powered laser housed in an optical disk player.
- The laser shines through clear polycarbonate while a motor spins the disk.
- The intensity of the reflected light of laser changes as it encounters a pit.
- The beginning or end of a pit represents a 1, when no change in elevation occurs between intervals, a 0 is recorded.
- The disk has a single spiral track, beginning near the center and spiraling out to the outer edge.
- Sectors near the outside are the same size as those near the inside.

CD Recordable

- Called write-once read-many CD.
- Used in applications in which only one or small copies of a set of data is needed.
- The disk is prepared to be written with a modest intensity laser beam.
- The disk medium includes a dye layer.
- The die is used to change reflectivity by making pits with different reflectivity on the surface of the medium.

CD Rewirtable

- Can be written and overwritten repeatedly.
- The approach used in this type is called Phase Change.
- It uses a material with two significantly reflectivities in two different states, called amorphous state and crystalline state.
- A beam of laser can change the material from one phase to the other.
- Disadvantage:
- The material looses its desirable properties over usage time. (500,000 to 1,000,000 erase cycle)

Digital Versatile Disk (DVD)

- It will replace video cassette, video tapes and CD-ROMs.
- It has huge storage due to three differences from CD:

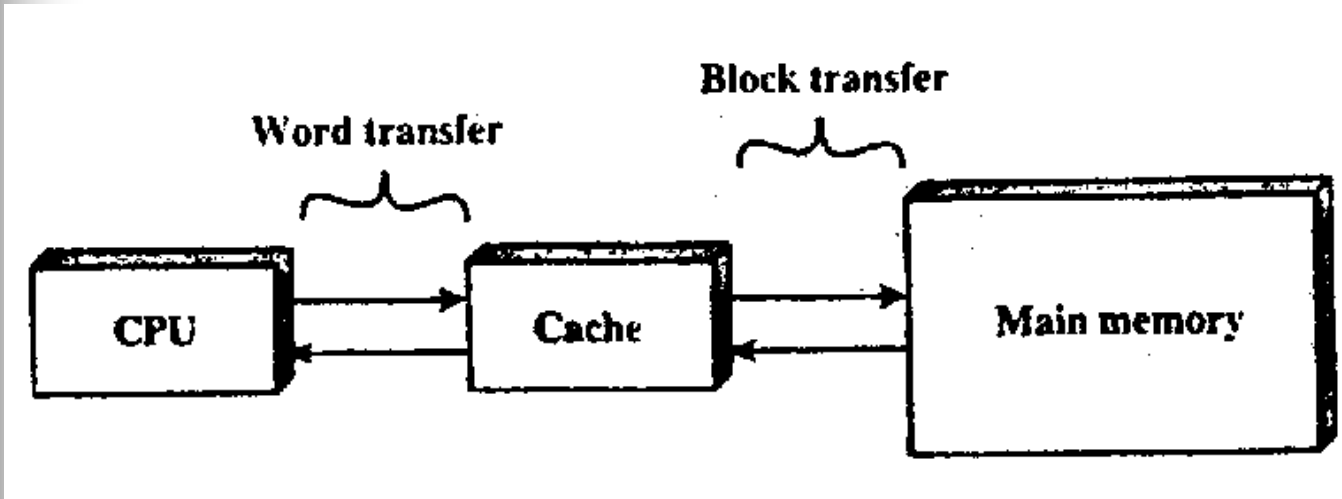
1- Bits are packed more closely on a DVD. (1/2 CD)

2- Has second layer of pits and lands on the top of the first layer (dual-layer). It has a semireflective layer on top of the reflective layer. By adjusting focus, the lasers in DVD drivers can read each layer separately (2 CD).

3- Has two sides whereas CD has one side.

Cache Memory

- Cache memory is intended to give memory speed approaching that of the fastest memories available.
- It is made of SRAM which is faster than DRAM.
- The cache contains a copy of portions of main memory.
- When the processor attempts to read a word of memory, a check is made to determine if the word is in the cache.



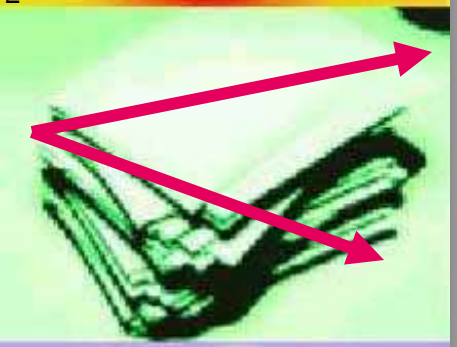
Cache Memory

- If so, the word is delivered to the processor.
- If not, a block of main memory, consisting of some fixed number of words, is read into cache and then the word is delivered to the processor.
- Why do we get blocks of data?

Because of the phenomenon of locality of reference.

- When a block of data is fetched into the cache to satisfy a single memory reference, it is likely that there will be future reference to that same memory location or to other words in the block.

2^n

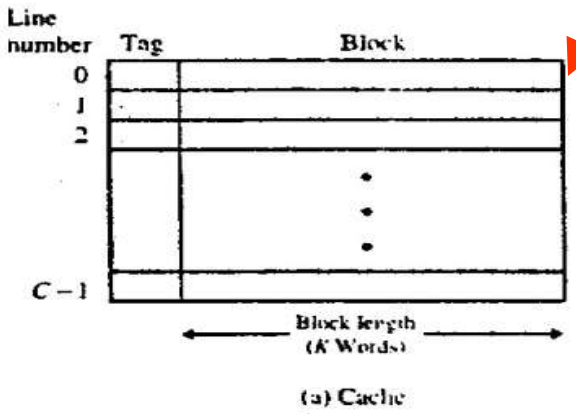


• Main memory consists of up to addressable words, with each word having a unique n-bit address.

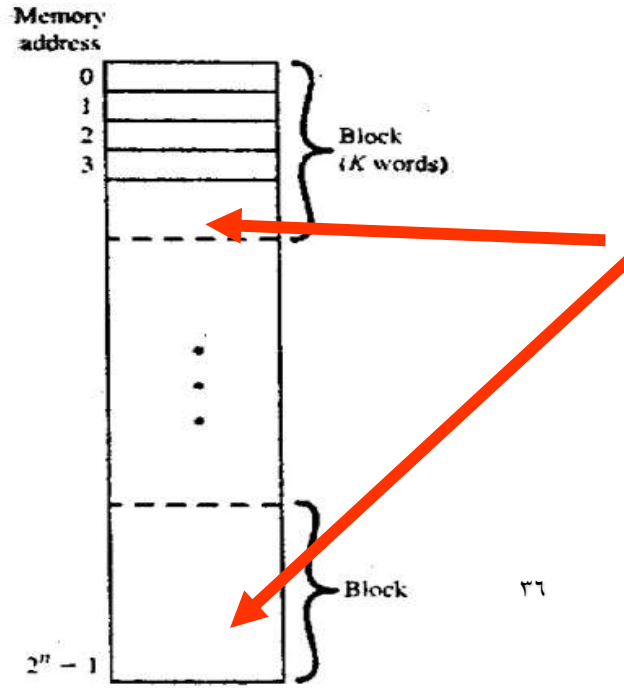
For mapping purpose, this memory is considered to consist of a number of fixed-length blocks of K words each.

That is, there are $M = 2^n / K$ blocks.

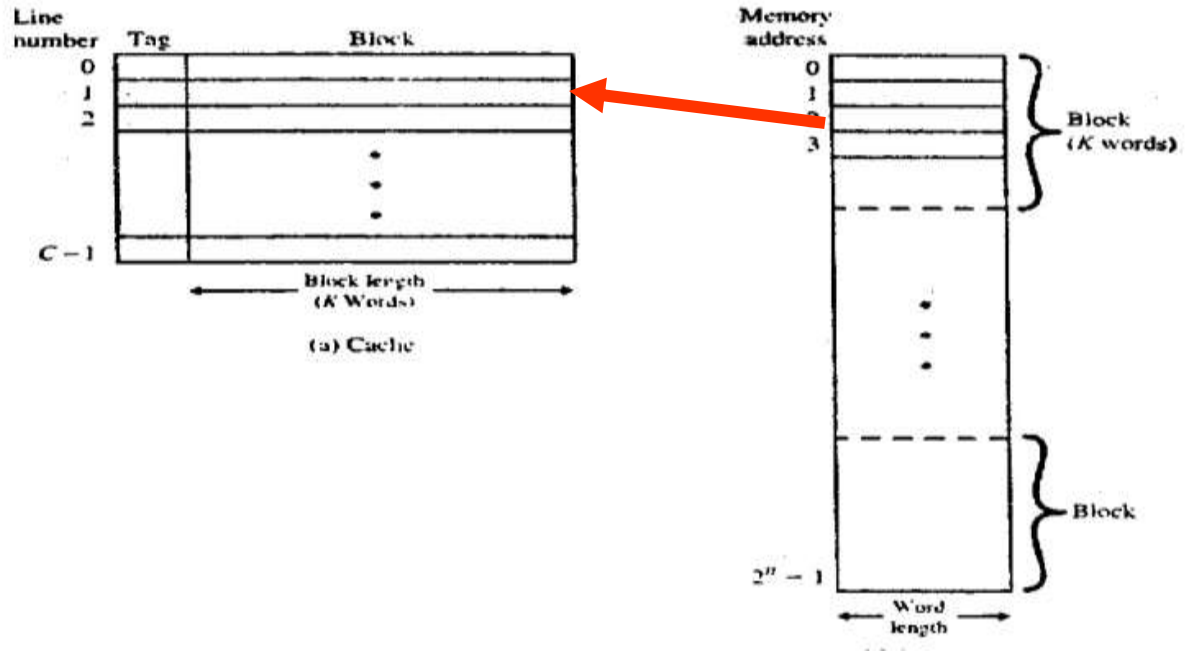
Cache consists of C lines of K words each, and the number of lines is considerably less than the number of main memory blocks ($C << M$).



(a) Cache

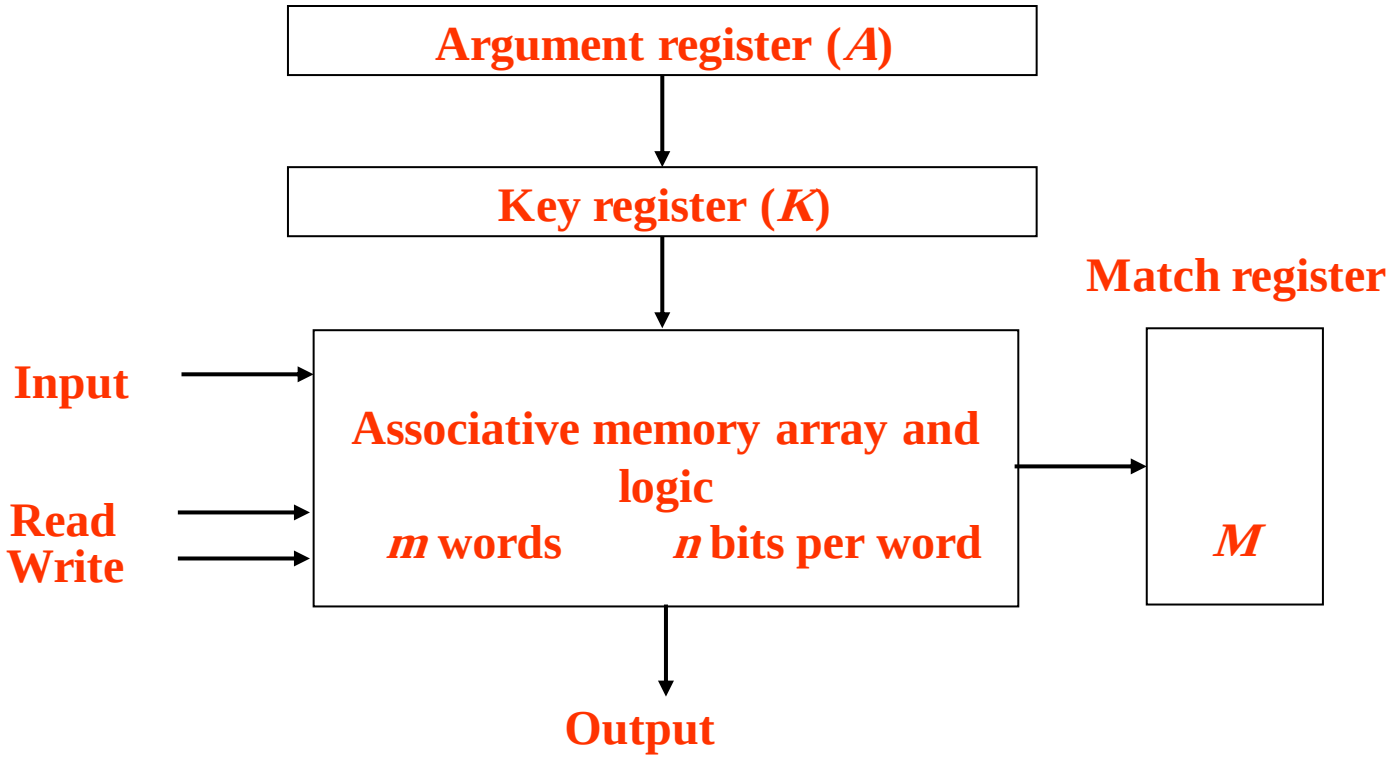


- Each line in cache contains a block and a tag.
- At any time, some subset of the blocks of memory resides in lines in the cache.
- If a word in a block of memory is read, that block is transferred to one of the lines of the cache.
- Because there are more blocks than lines, an individual line cannot be uniquely and permanently dedicated to a particular block.
- Thus, each line includes a tag that identifies which particular block is currently being stored.
- The tag is usually a portion of the main memory address.



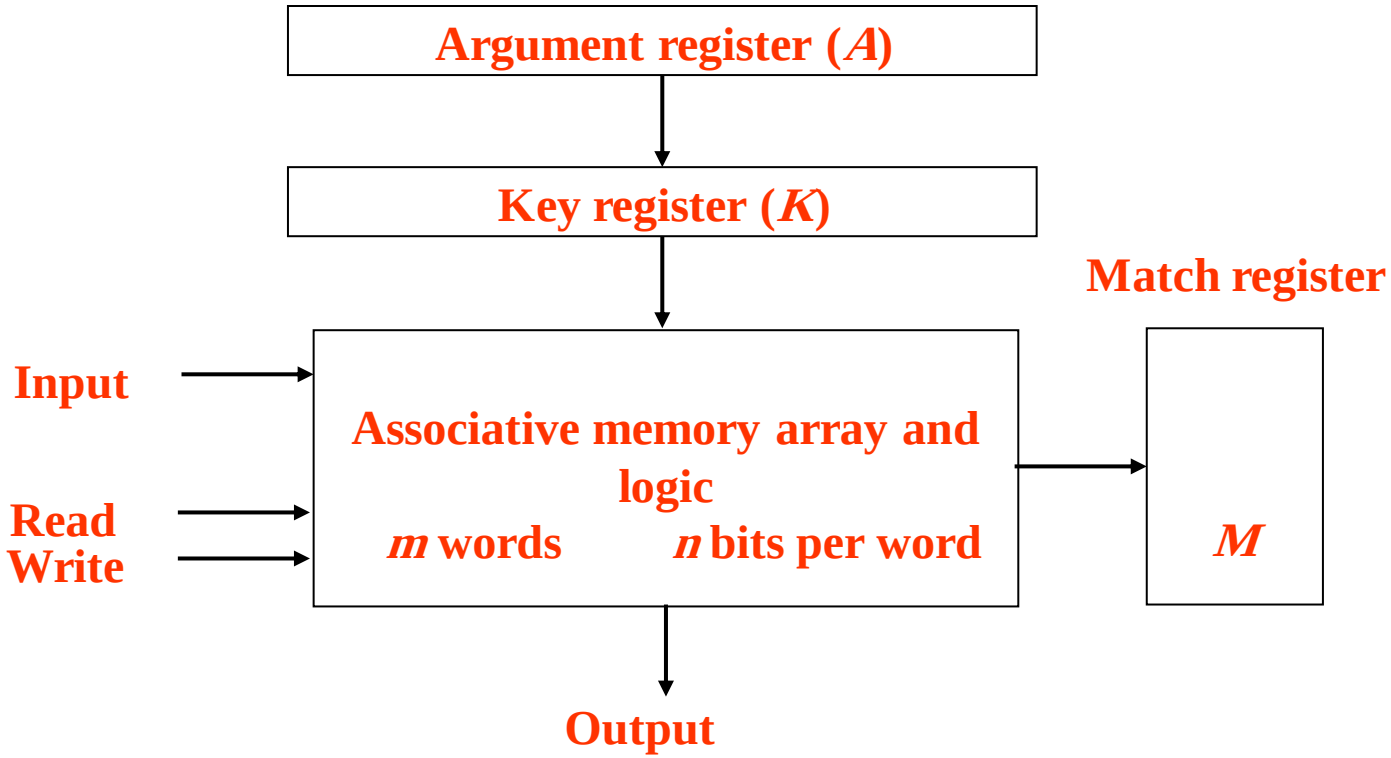
Associative Memory

- An associative memory is more expensive than a random access memory because each cell must have storage capability as well as logic circuits for matching its contents with an external argument.
- For this reason, associative memories are used in applications where the search time is very critical and must be very short.



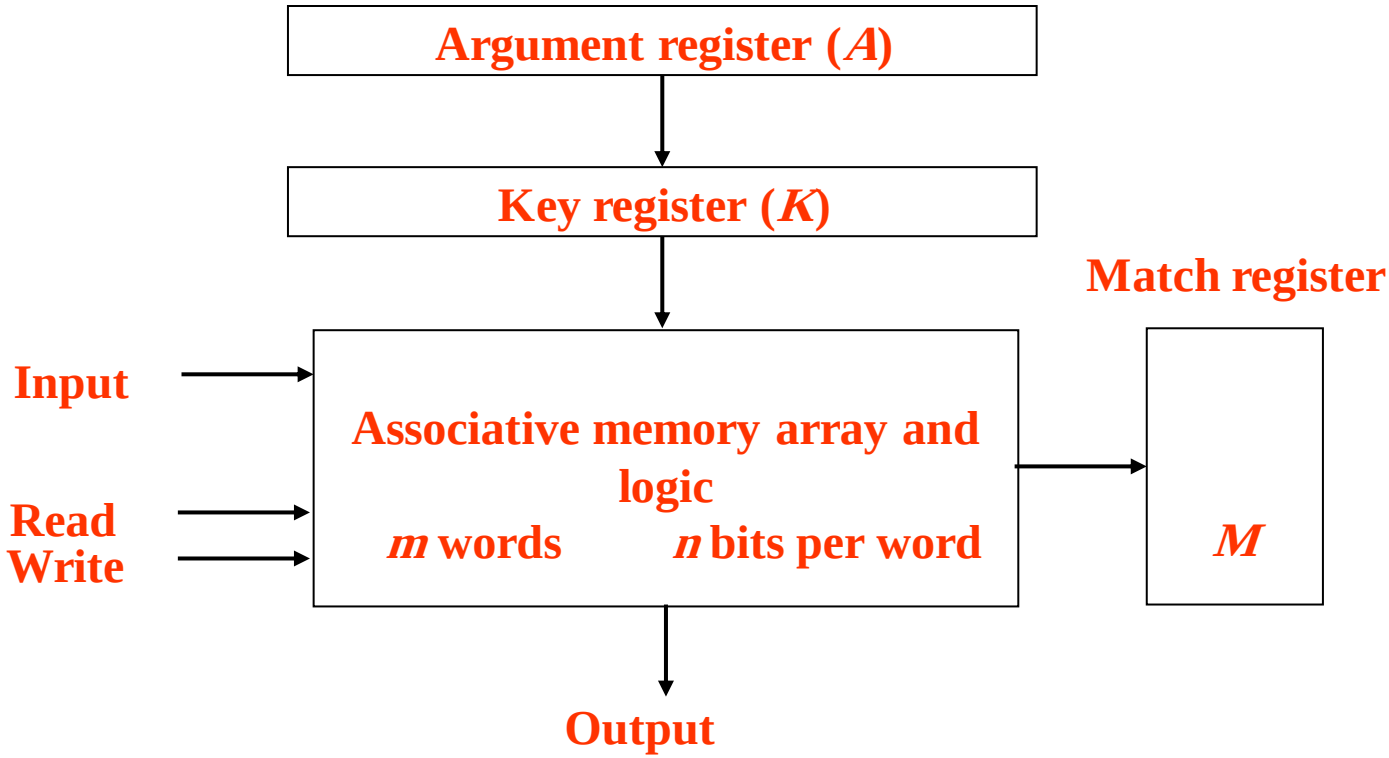
Associative Memory

- It consists of a memory array and logic for *m* words with *n* bits per word.
- The argument register *A* and key register *K* each have *n* bits, one for each bit of a word.
- The match register *M* has *m* bits, one for each memory word.



Associative Memory

- Each word in memory is compared in parallel with the contents of the argument register.
- The words that match the bits of the argument register set a corresponding bit in the match register.



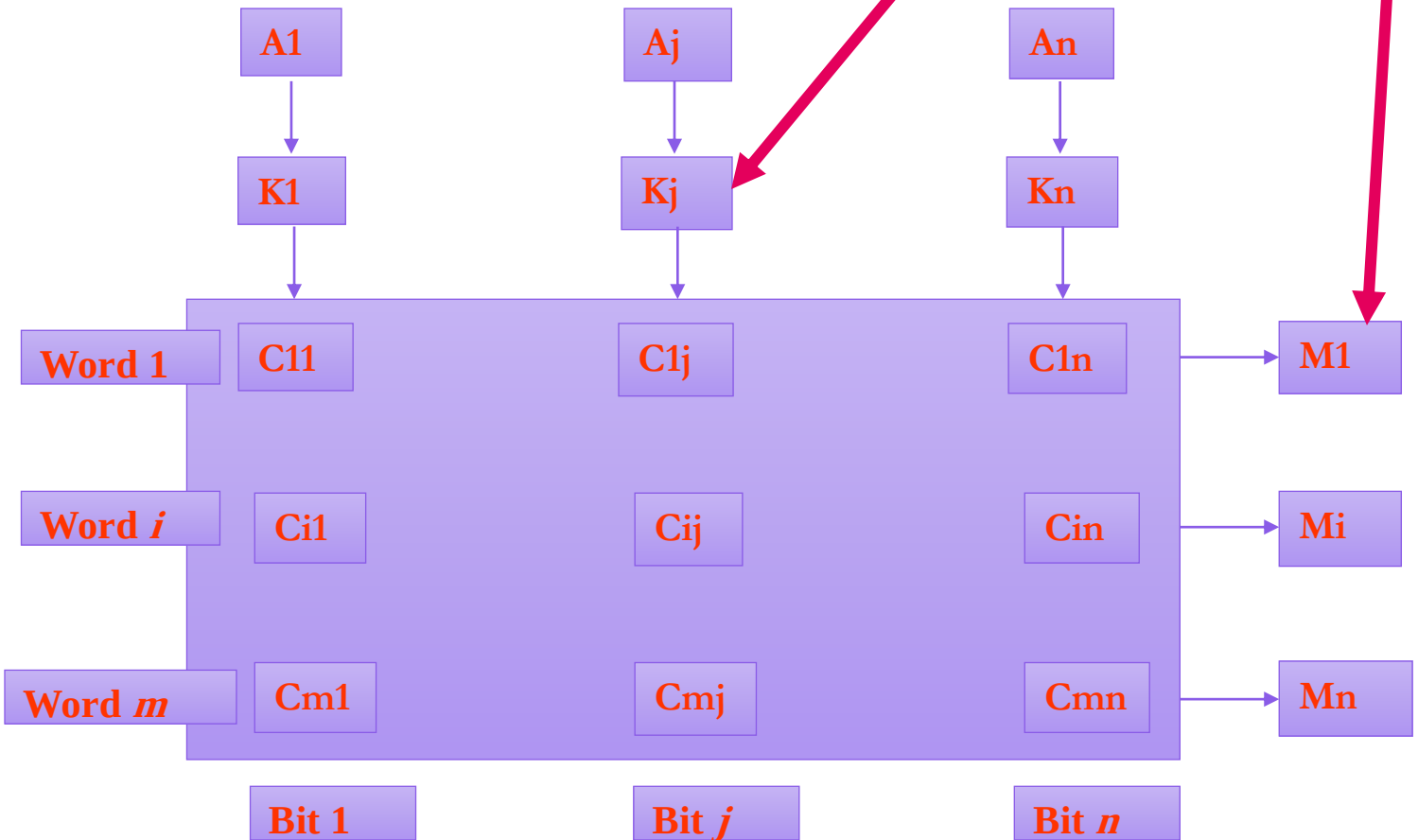
Example

- Suppose that the argument register *A* and the key register *K* have the bit configuration shown below.
- Only the three leftmost bits of *A* are compared with memory words because *K* has 1's in these positions.

A	101	111100	
K	111	000000	
Word1	100	111100	no match
Word2	101	000001	match

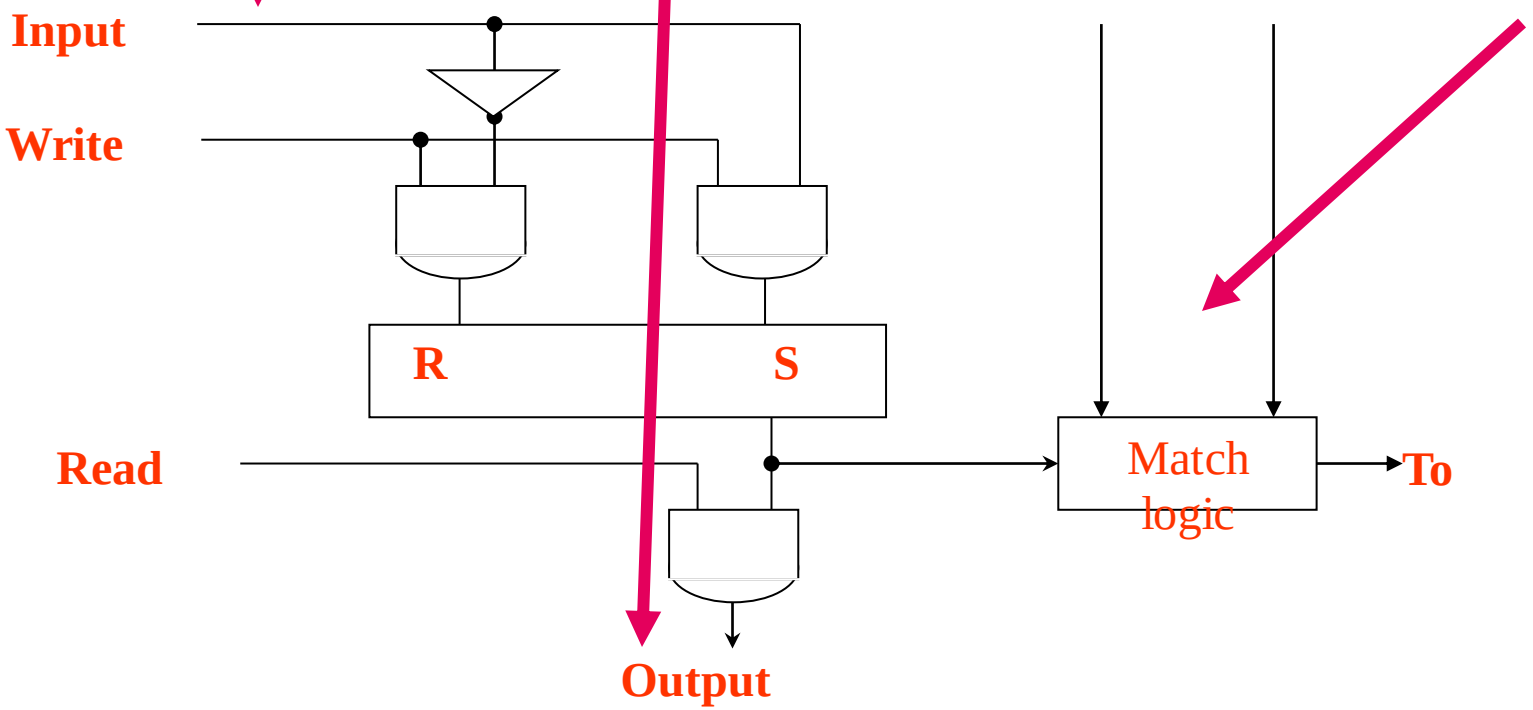
- Word2 matches the unmasked argument field because the three leftmost bits of the argument and the word are equal.

- A bit in the argument register (Aj) is compared with all the bits in column *j* of the array provided that Kj=1.
- This is done for all columns *j=1, 2,…, n* .
- If a match occurs between all the unmasked bits of the argument and the bits in word *i*, the corresponding bit in the match register is set to 1.
- If one or more unmasked bits of the argument and the word do not match, is cleared to 0.



One cell of associative memory

- It consists of a flip-flop storage element and the circuits for reading, writing, and matching the cell.
- The input bit transferred into the storage cell during a write operation.
- The bit stored is read out during a read operation.
- The match logic compares the content of the storage cell with the corresponding unmasked bit of the argument and provides an output for the decision logic that sets the bit in.



Read Operation

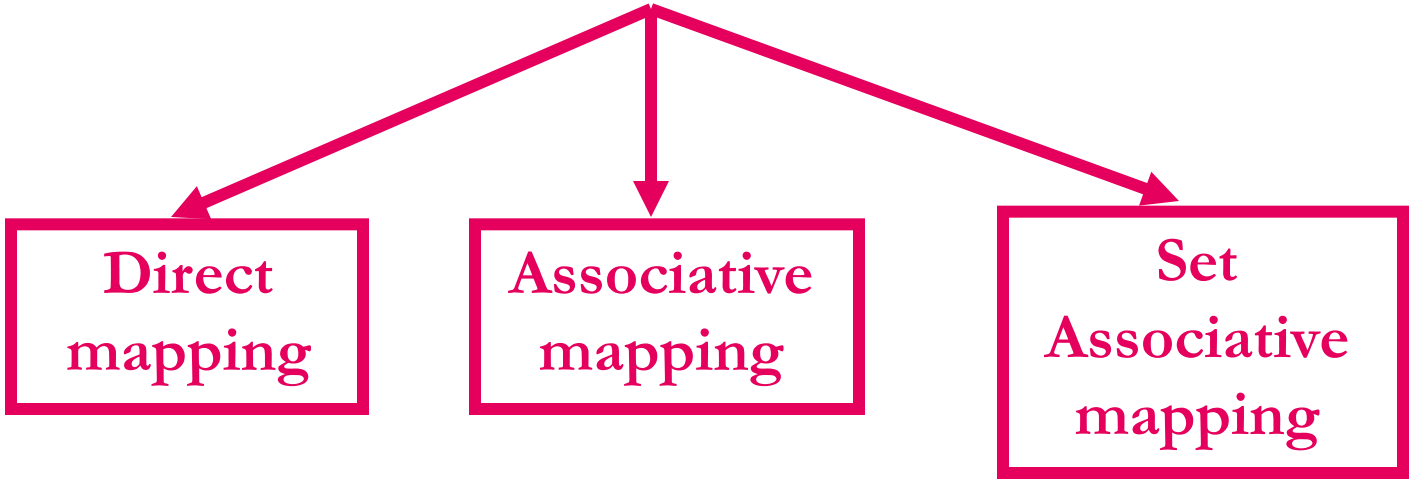
- Only one word may match the unmasked argument field.
- By connecting output directly to the read line in the same word position (instead of the *M* register), the content of the matched word will be presented automatically at the output lines and no special read command signal is needed.

Write Operation

- The writing can be done by addressing each location in sequence.
- This will make the device a random-access memory for writing and a content addressable memory for reading.
- The advantage here is that the address for input can be decoded as in a random-access memory.
- Thus instead of having *m* address lines, one for each word in memory, the number of address lines can be reduced by the decoder to *d* lines, where $m = 2^d$.

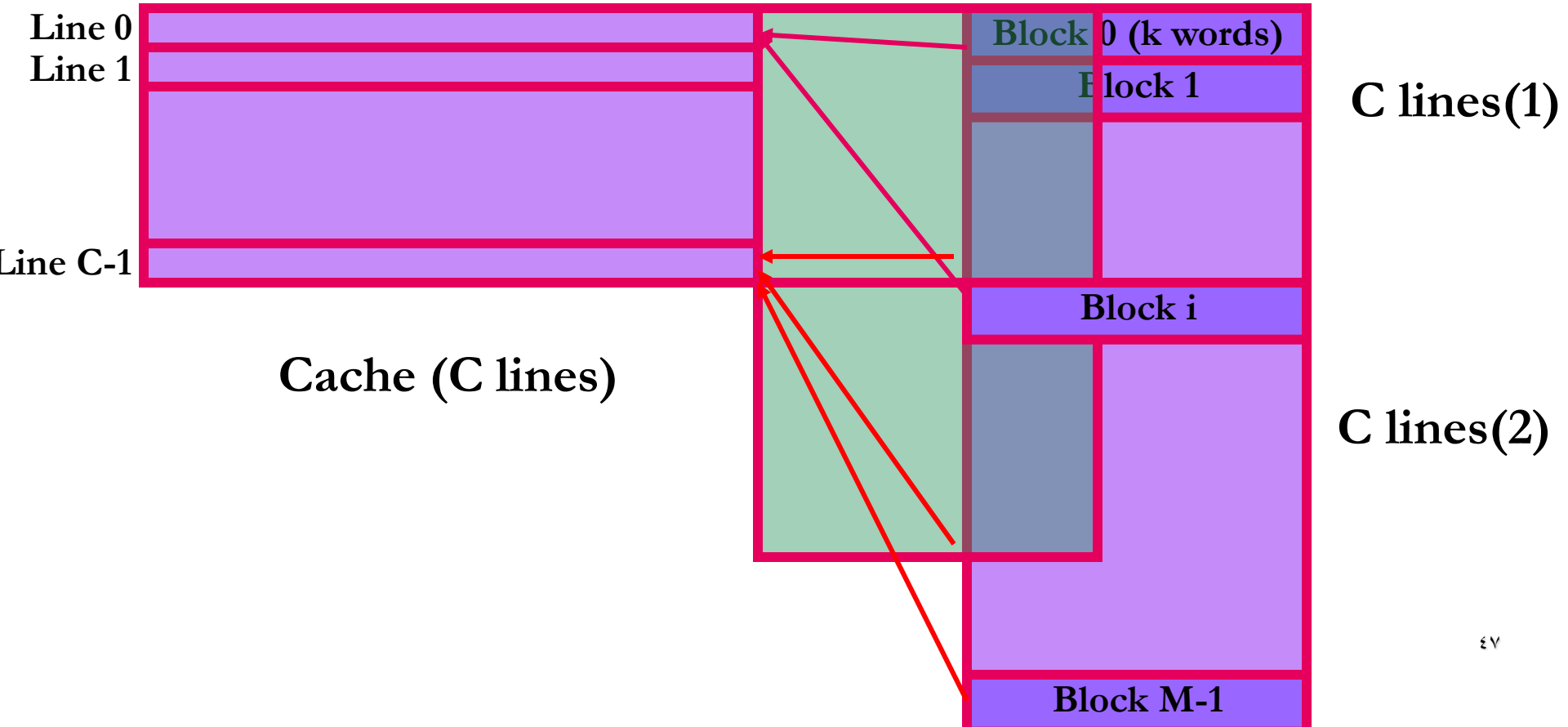
Mapping Function

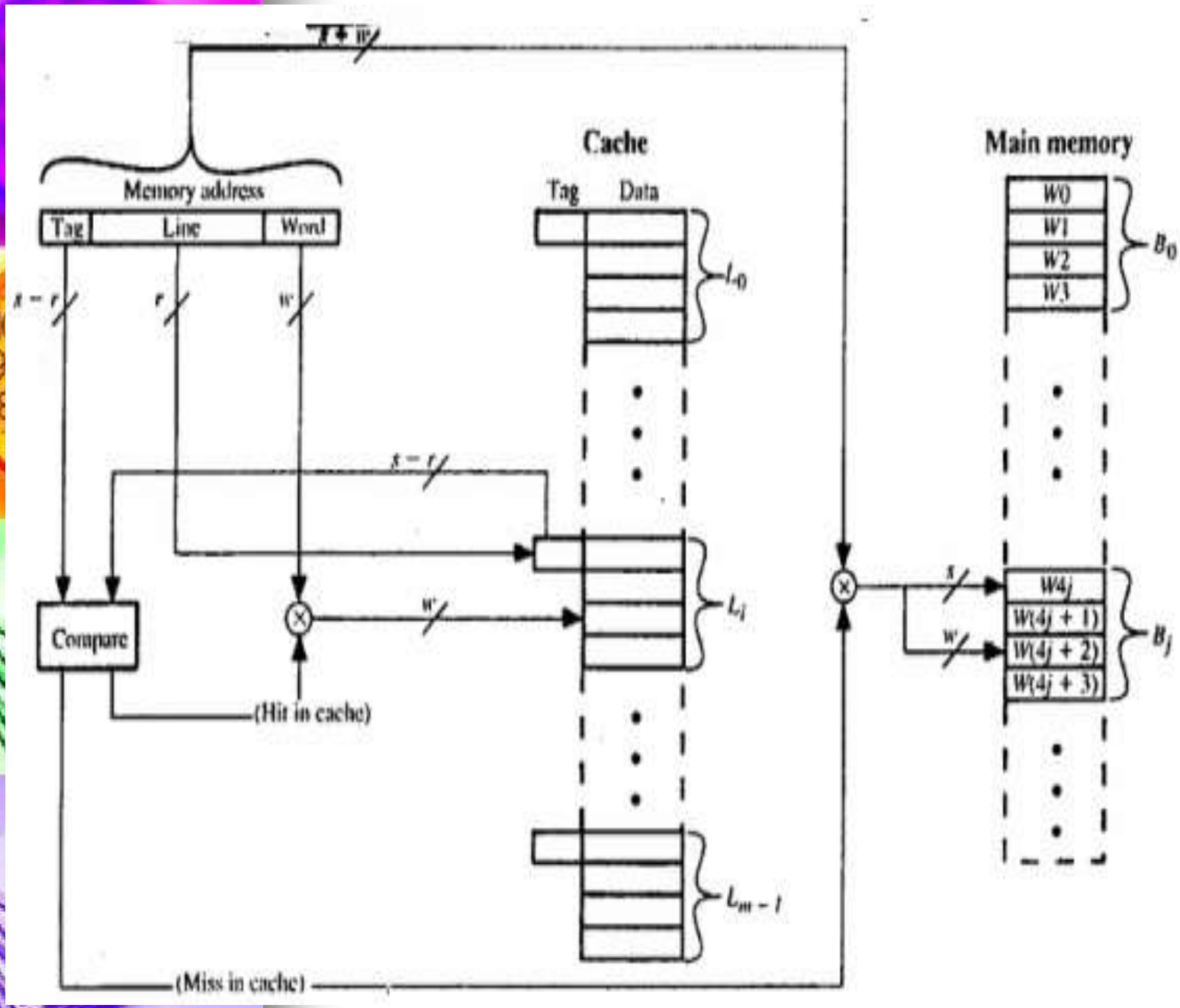
- Because there are fewer cache lines than main memory blocks, an algorithm is needed for mapping (allocating) main memory blocks into cache lines.
- Further, a means is needed for determining which main memory block currently occupies a cache line.
- The choice of the mapping function dictates how the cache is organized.



1- Direct mapping

- The easiest way.
- Each block of main memory is mapped into only one possible cache line.




Memory address
Tag
Line
Word
Cache
Tag
Data
Main memory
W0
W1
W2
W3
W4j
W(4j + 1)
W(4j + 2)
W(4j + 3)
Compare
(Hit in cache)
(Miss in cache)

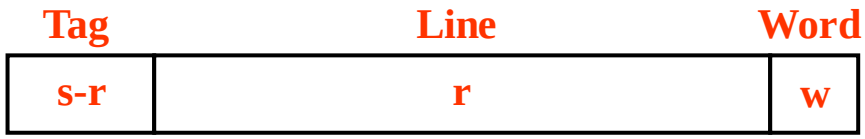
The mapping function

- The mapping function is easily implemented using the address.
- For purposes of cache access, each main memory address can be viewed as consisting of three fields.

1- The least significant *w* bits identify a unique word or byte within a block of main memory

2- The cache logic interprets these *s* bits as a tag of *s – r* bits

3- A line field of *r* bits that identifies one of the *m* = lines of the cache.



To summarize.

- Address length = $(s + w)$ bits = n bits
- Number of addressable units = 2^{s+w} words or bytes
- Block size = line size = 2^{w} words or bytes
- Number of blocks in main memory = $2^{s+w} / 2^{w} = 2^{s}$
- Number of lines in cache = m = Size of tag = $(s – r)$ bits

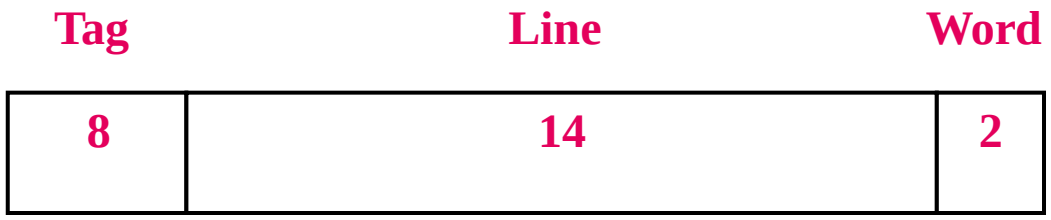
Example

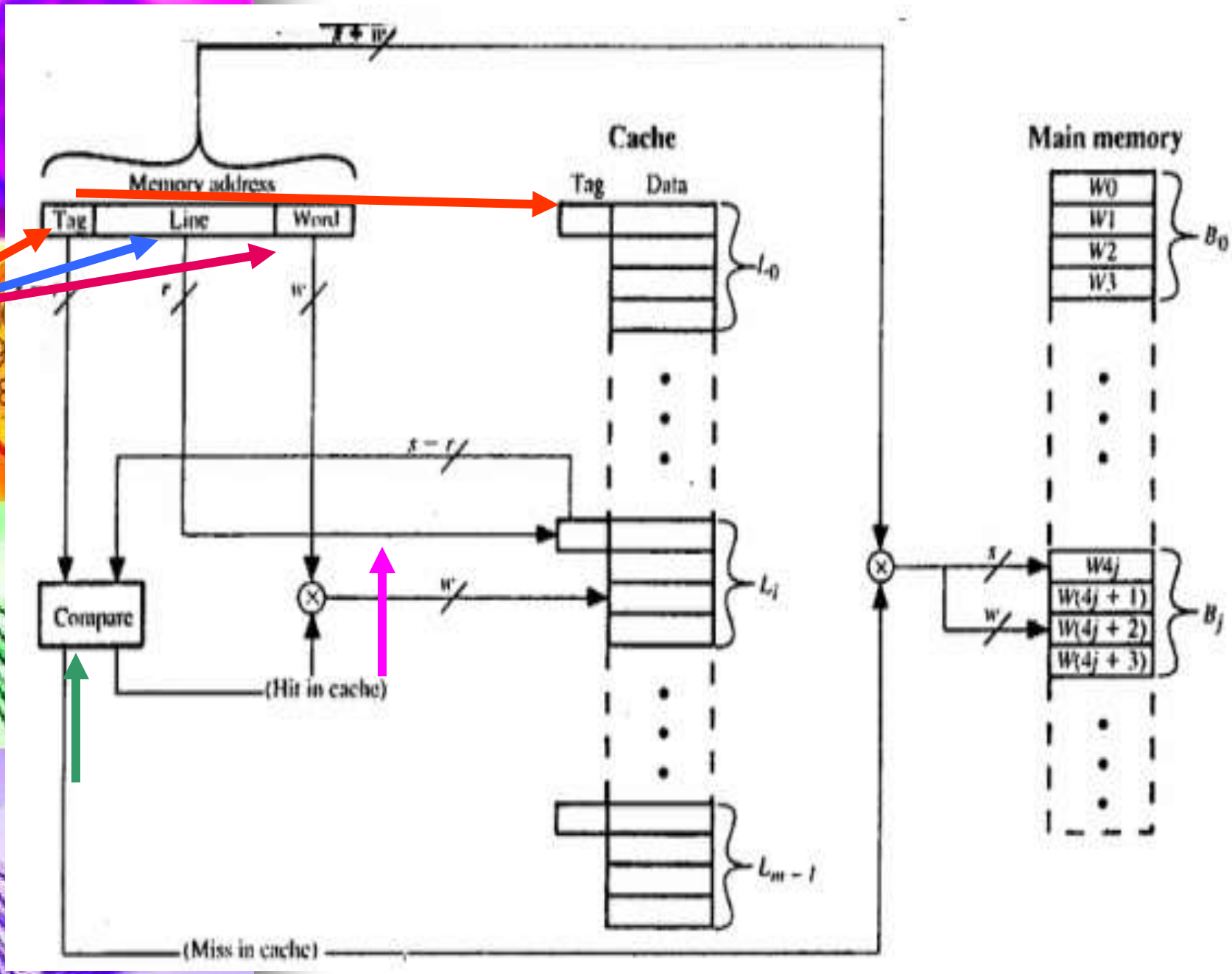
- The cache can hold 64 Kbytes.
- Data is transferred between main memory and the cache in blocks of 4 bytes each.
- This means that the cache is organized as 16K lines of 4 bytes each.
- The main memory consists of 16 Mbytes, with each byte directly addressable by a 24-bit address (= 16M).
- Thus, for mapping purposes, we can consider main memory to consist of 4M blocks of 4 bytes each.

Dividing the address and using it to identify the word in the cache

How to divide the 24-bit address?

- The 14-bit line number is used as an index into the cache (16 k lines) to access a particular line.
- We will need 2-bit word identifier to select one of the 4 bytes in that line.
- The rest of the address is a tag (8 bits)




Memory address
Tag
Line
Word
Cache
Tag
Data
Main memory
W0
W1
W2
W3
B0
L0
s − r
w
Compare
(Hit in cache)
Li
W4j
W(4j + 1)
W(4j + 2)
W(4j + 3)
Bj
Lm − 1
(Miss in cache)

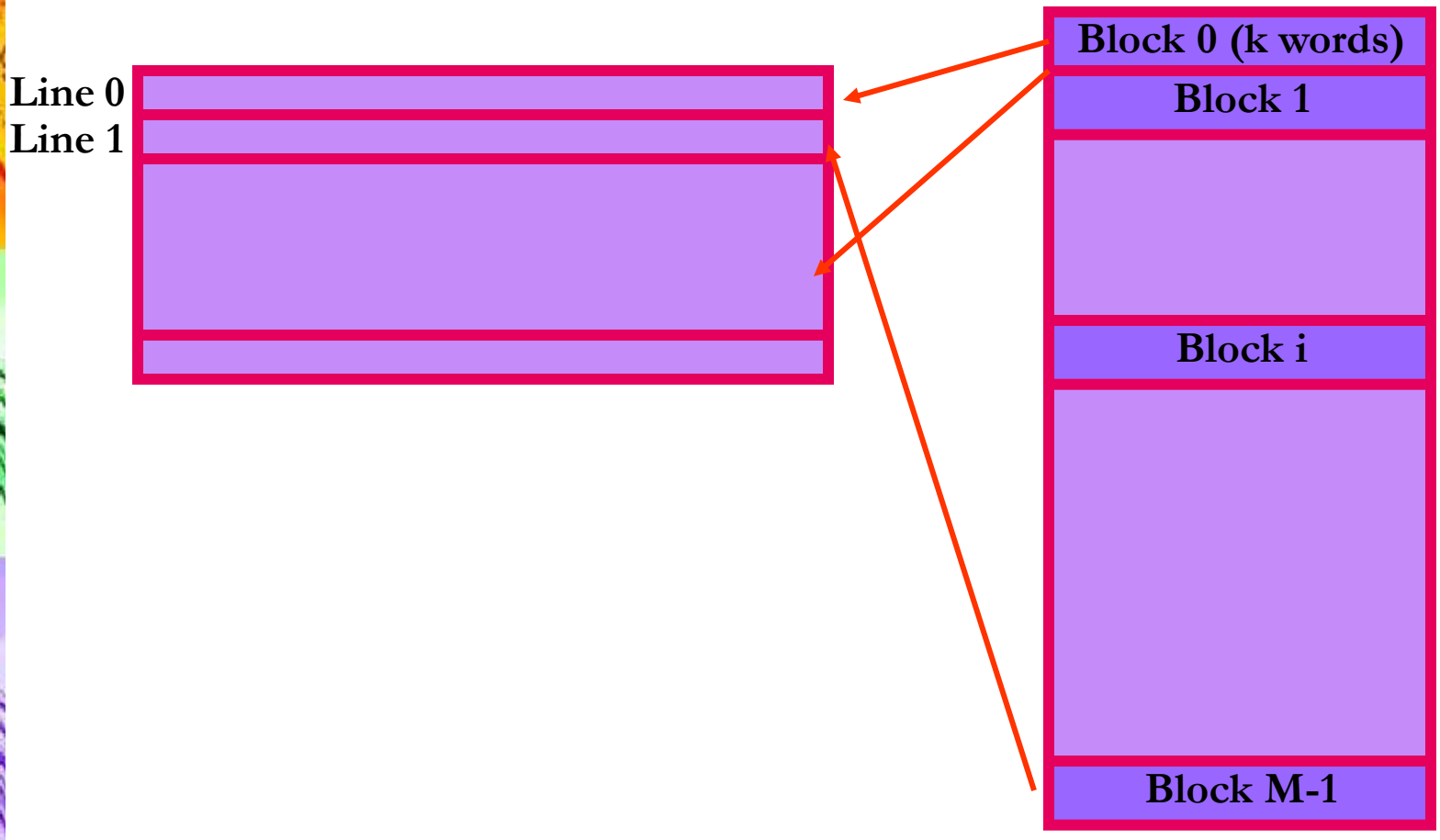
Dividing the address and using it to identify the word in the cache

How to find a word in the cache:

- If the 8-bit tag number matches the tag number currently stored in that line, then the 2-bit word number is used to select one of the four bytes in that line
- Otherwise, the 22-bit tag-plus-line field used to fetch a block from main memory.
- The actual address that is used for the fetch is the 22-bit tag-plus-line concatenated with two 0 bits, so that 4 bytes are fetched starting on a block boundary.

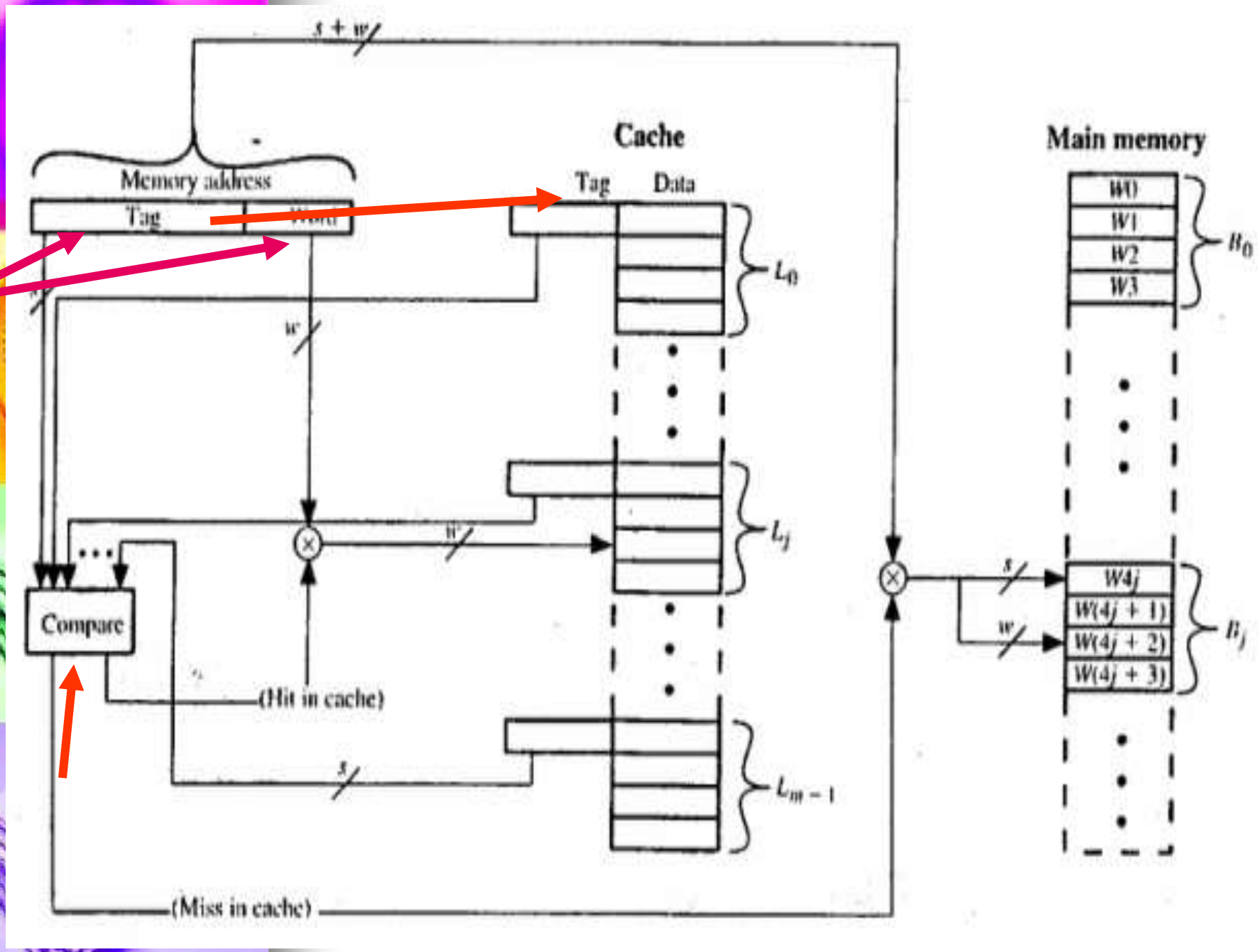
2- Associative Mapping

- Associative mapping overcome the disadvantages of direct mapping by permitting each main memory block to be loaded into any line of the cache.



2- Associative Mapping

- In this case, the cache control logic interprets a memory address simply as a tag and a word field.
- The tag field uniquely identifies a block of main memory.
- To determine whether a block is in the cache, the *cache control logic must simultaneously examine every line's tag for a match.*
- Note that no field in the address corresponds to line number, so that the number of lines in the cache is not determined by the address format


s + w
Memory address
Tag
Word
Cache
Tag
Data
Main memory
W0
W1
W2
W3
B0
L0
w
s
w
Lj
Compare
(Hit in cache)
s
w
W4j
W(4j + 1)
W(4j + 2)
W(4j + 3)
Bj
Lm - 1
(Miss in cache)

Example

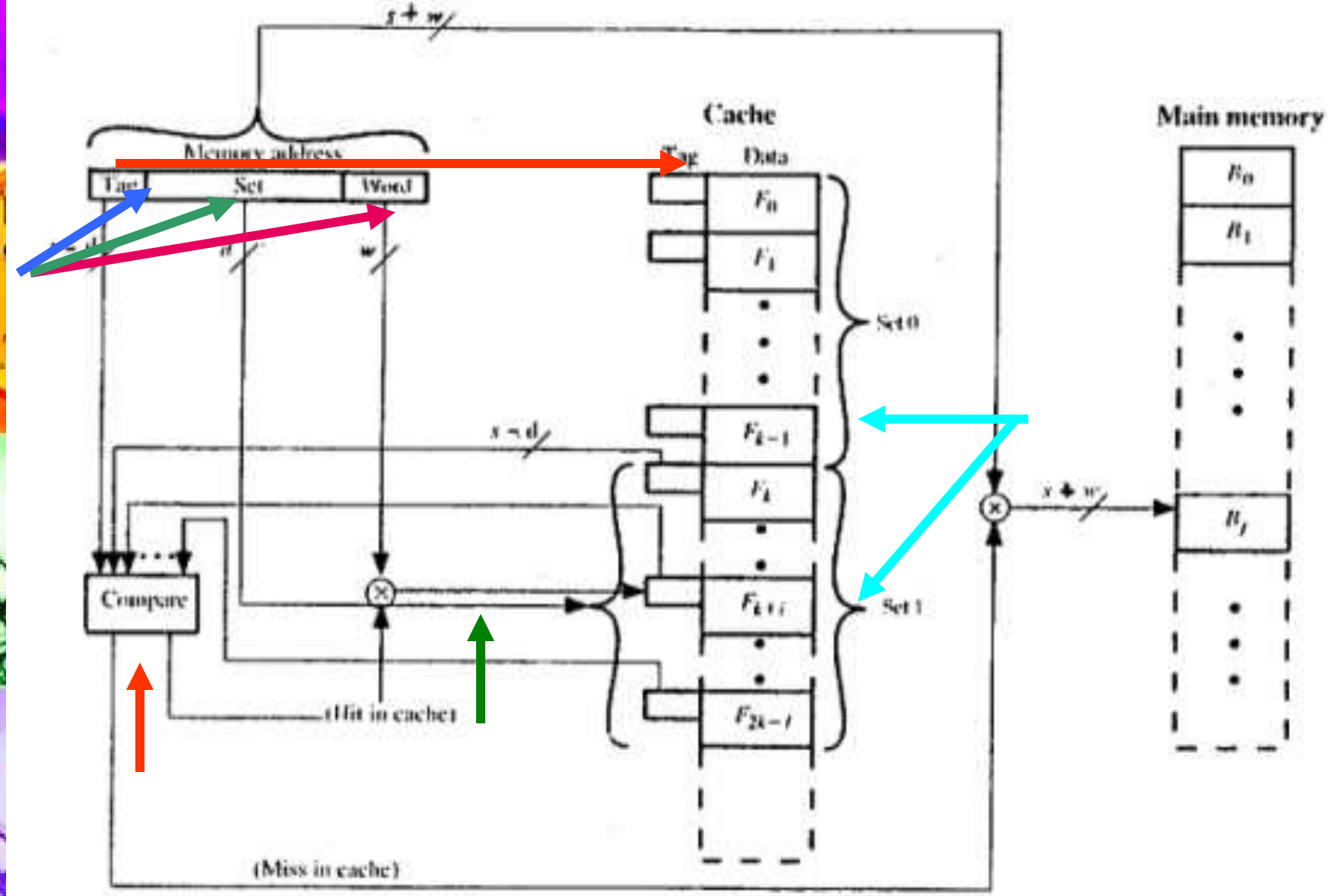
- A main memory address consists of a 22–bit tag and a 2-bit byte number.
- The 22-bit tag must be stored with the 32-bit block of data for each line in the cache.
- With associative mapping, there is flexibility as to which block to replace when a new block is read into the cache.
- The principal disadvantage of associative mapping is the complex circuitry required to examine the tags of all cache lines in parallel.

3- Set Associative Mapping

- Set associative mapping is compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages.
- In this case, the cache is divided into v sets, each of which consists of k lines.
- This is referred to as k-way set associative mapping.
- With set associative mapping, block , can be mapped into any of the lines of set i.
- In this case, the cache control logic interprets a memory address simply as three fields:

  tag, set, and word.
- The d set bits specify one of $v = 2^d$ sets.
- The s bits of the tag and set fields specify one of the blocks of main memory.

Tag	Set	Word
9	13	2

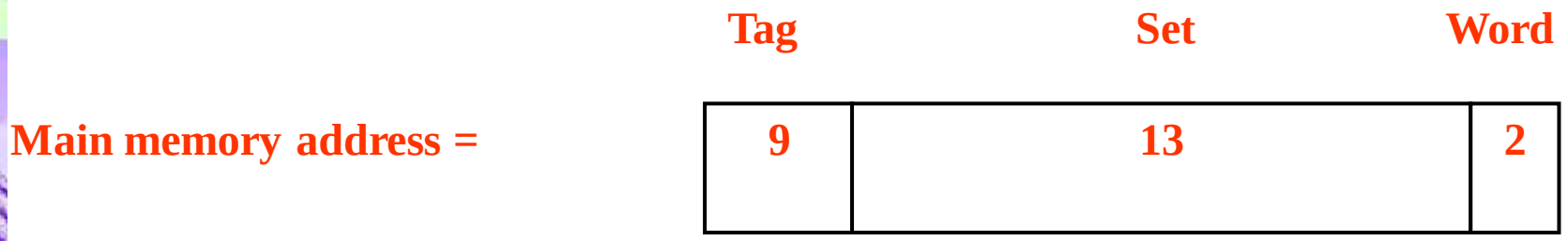

Memory address
Tag
Set
Word
Cache
Tag
Data
Main memory
Set 0
Set 1
Compare
(Hit in cache)
(Miss in cache)

Example

- Set associative mapping with two lines in each set, referred to as two-way set associative.
- Number of sets $v = 2^d$

  = total number of lines in cache/ 2 = 16k/2 = 8k

  $= 2^{13}$ → d = 13
- The 13-bit set number identifies a unique set of two lines within the cache.



Replacement Algorithms

- When a new block is brought into the cache . one of the existing blocks must be replaced.
- For direct mapping, there is only one possible line for any particular block, and no choice is possible.
- For the associative and set associative techniques, a replacement algorithm is needed.
- To achieve high speed, such an algorithm must be implemented in hardware.

1- First–In–First–Out (FIFO)

- Replace that block in the set that has been in the cache longest.
- FIFO is easily implemented as a round–robin or circular buffer technique.

2- Least recently used (LRU)

- Replace that block in the set that has been in the cache longest with no reference to it.
- For two-way set associative, this is easily implemented.
- Each line includes a USE bit.
- When a lines referenced, its USE =1 and the USE bit of the other line in that set = 0.
- When a block is to be read into the set, the line whose USE bit is 0 is used.
- Because we are assuming that more recently used memory locations are more likely to be referenced, LRU should give the best hit ratio.
- Hit ratio = probability that a word is found in the cache

3- Least Frequently Used (LFU)

- Replace that block in the set that has experienced the fewest references.
- LFU could be implemented by associating a counter with each line.

4- Random Replace

- Pick a line at random from among the candidate lines.
- Simulation studies have shown that random replacement provides only slightly inferior performance to an algorithm based on usage.

Write Policy

- An important aspect of cache organization is concerned with memory write requests.
- When the CPU finds a word in cache during a read operation, the main memory is not involved in the transfer.
- However, if the operation is a write, there are two ways that the system can proceed:

1. Write-through
2. Write-back

1. Write-through

- The simplest and most commonly used procedure.
- Update main memory with every memory write operation, with cache memory being updated in parallel if it contains the word at the specific address.
- This method has the advantages that main memory always contains the same data as cache.
- This characteristic is important in systems with direct memory access transfers.

1. Write-through

- It ensures that the data residing in main memory are valid at all times so that an I/O device communicating through DMA would receive the most recent updated data.
- However, it has the disadvantage of high memory traffic and may create a bottelneck.

2. Write-back

- Only the cache location is updated during a write operation.
- The location is then marked by a flag (called update bit or dirty bit) so that later when the word is removed (replaced) from the cache, it is copied into main memory.
- The reason for the write back method is that during the time a word resides in the cache, it may be updated several times;
- However, as long as the word remains in the cache, it does not matter whether the copy in main memory is out of date, since requests from the word are filled from the cache.

2. Write-back

- It is only when the word is displaced from the cache that an accurate copy need be rewritten into main memory.
- Analytical results indicate that the number of memory writes in a typical program ranges between 10 and 30% of the total references to memory.
- The problem of this technique is that portions of memory are invalid and thus access from I/O modules to this data can be through cache instead.
- This will require complex circuitry and potential bottleneck.

Multiple caches

- A computer now has multiple caches.
- A cache can be placed on the same chip with the processor: the on-chip cache (L1).
- It reduces the processor's external bus activity and therefore speed up the execution times and increase the overall performance.
- This is merged with the external cache (L2) giving two-level cache organization.
- This will complicate the circuit and will require replacement algorithms and write back polices.

Unified versus split caches

- We can also have two caches: one for holding the instructions and the other for holding the data.

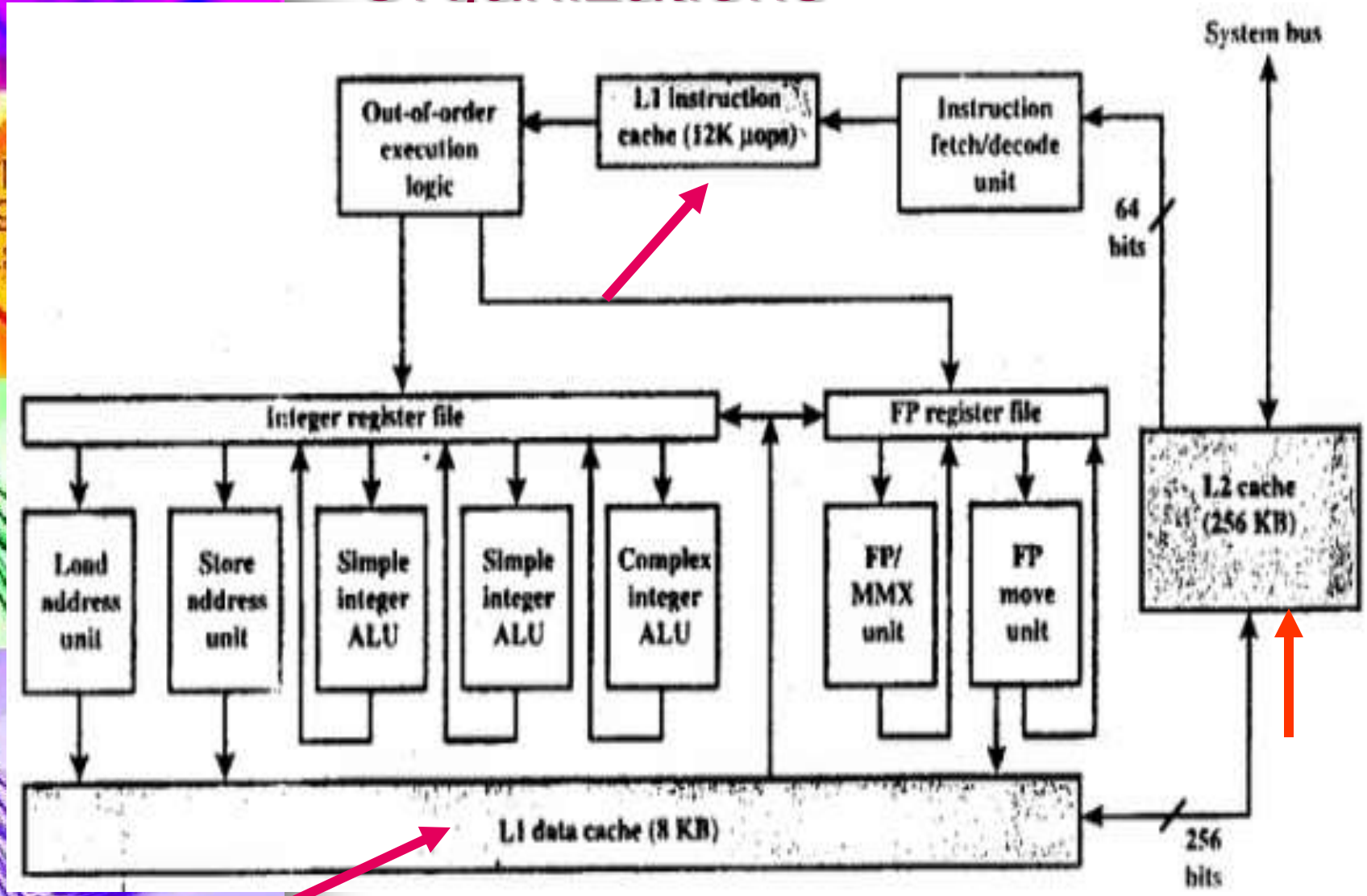
Advantage of unified cache:

- It has higher hit rate than split because it balances the load between instruction and data fetches. If the execution requires more instruction fetches, then it will use the cache size for instruction and if the execution requires more data fetches, then it will use the cache size for data. Unlike the split, which will end up filling the instruction size while the data is free or vice versa.
- Only one cache need to be designed and implemented.

Unified versus split caches

- Despite that, most modern system, specially the parallel systems, tend to have split cache.
- It has the advantage of eliminating the contention for the cache between the fetch/decode unit and the execution unit.
- Thus, the two operations can be made in parallel, on the same time, on the two caches.

Pentium 4 Cache Organizations



Wireless Cache Memory Invalidation

- With the increasing popularity of different mobile devices such as laptops, hand-helds, cellular phones, etc., we are expecting more services to be delivered in a wireless environment.
- In particular, people are interested in retrieving up-to-date information at any time and any place using their mobile devices in client-server wireless networks.
- These services include, but are not limited to, financial information, news and weather forecast, traffic schedules and conditions, online shopping catalogs, etc.
- We need an efficient on-demand data access mechanism to support such services.
- However, the wireless bandwidth is still severely limited and mobile devices are inevitably battery-operated.

- Caching frequently accessed data at the client side allows some queries to be served locally.
- Cached data will eventually become invalid due to asynchronous updates and removals at the data source.
- As a result, queries cannot be directly served by the local cache.
- To guarantee correctness, clients should instead ensure the cached data is consistent before using it to serve a query.
- This relies on the server to inform all its clients to discard any obsolete data in their caches.
- To efficiently utilize the precious wireless bandwidth, the server only delivers the invalidation information via the broadcast channel, but not the contents of the updated data items.
- This introduces extra overheads at the server and possibly longer query delay to the clients.

- *The basic cache invalidation strategy is the use of invalidation reports (IRs)*.
- The server periodically broadcasts IRs, each of which indicates those data items that are recently updated at the source.
- Clients use IRs to keep their caches consistent by discarding any obsolete data.
- If a query cannot be served locally, i.e., a cache miss, the client issues an uplink query request for the data item.
- The server aggregates the query requests from all its clients and broadcasts query replies only once every IR broadcast period.
- In this manner, a query reply can be shared by more than one client via the broadcast channel.
- The major advantages of this cache invalidation strategy are high scalability and energy efficiency.

- First, the size of IR is independent of the number of clients.
- Second, clients can exploit the periodicity of server broadcast to save power in that mobile devices can operate in doze mode most of the time and only activate during server broadcast.
- Disadvantage:
- If the disconnection time of a client is longer than a fixed period of time, the client should discard its entire cache, even if some of the cached data may still be valid.
- The large query delay involved since clients require an IR to ensure cache consistency.

- The addition of updated invalidation reports (UIRs) to the original IR-based approach (IR+UIR) aims to tackle this problem.
- In IR+UIR, the server broadcasts a number of UIRs between successive IRs.
- Each UIR only contains information about the most recently updated data since the last IR.
- As a result, the use of UIRs requires that the cache is consistent up to the last IR.
- Clients use UIRs to keep their cache consistent to the source.
- Thus, for cache hits, there is no need to wait for the next IR and query delay is further reduced.
- Previous research results on wireless cache invalidation are based on two simplifying assumptions:

1) the broadcast channel is error-free.

2) there is no other downlink traffic in the system.