UNIT-3

- The instructor thanks the authors for sharing the slides

Memory System

Introduction

- Goal: To provide the programmer with a large storage capacity for programs and data
- Effects the speed with which the CPU fetches the data from the memory
- Two types of memory
  - Read/write random access memory (RAM)
  - Read only memory (ROM)

Introduction contd.,

- For both RAM and ROM the time required to access a specific word is independent of its location

- Sequential memory – Magnetic Tape – access time to a particular data item depends on its relative location. Reading the n^{th} word requires that the previous n-1 words be read first

Memory Capacity

- Number of bytes that can be stored

Term	Normal Usage	Usage as Power of 2
K (Kilo)	10^3	$2^{10} = 1,024$
M (Mega)	10^6	$2^{20} = 1,048,576$
G (Giga)	10^9	$2^{30} = 1,073,741,824$
T (Tera)	10^{12}	$2^{40} = 1,099,511,627,776$

Key Characteristics

- Location
  - CPU
  - Internal (main)
  - External (secondary)
- Capacity
  - Word size
  - Number of words
- Unit of transfer
  - Word
  - Block
- Access methods
  - Sequential access
  - Direct access
  - Random access
  - Associative access
- Performance
  - Access time
  - Cycle time
  - Transfer rate

Key Characteristics contd.,

- Physical Type
  - Semiconductor
  - Magnetic surface
  - Optical
- Physical Characteristics
  - Volatile / Non-Volatile
  - Erasable / Non-erasable
- Organization

Location

- Three locations of memories
  - CPU
    - Registers – used by CPU as its local memory
  - Internal memory
    - Main memory
    - Cache memory
  - External memory
    - Peripheral devices – disk, tape – accessible to CPU via I/O controllers

Capacity

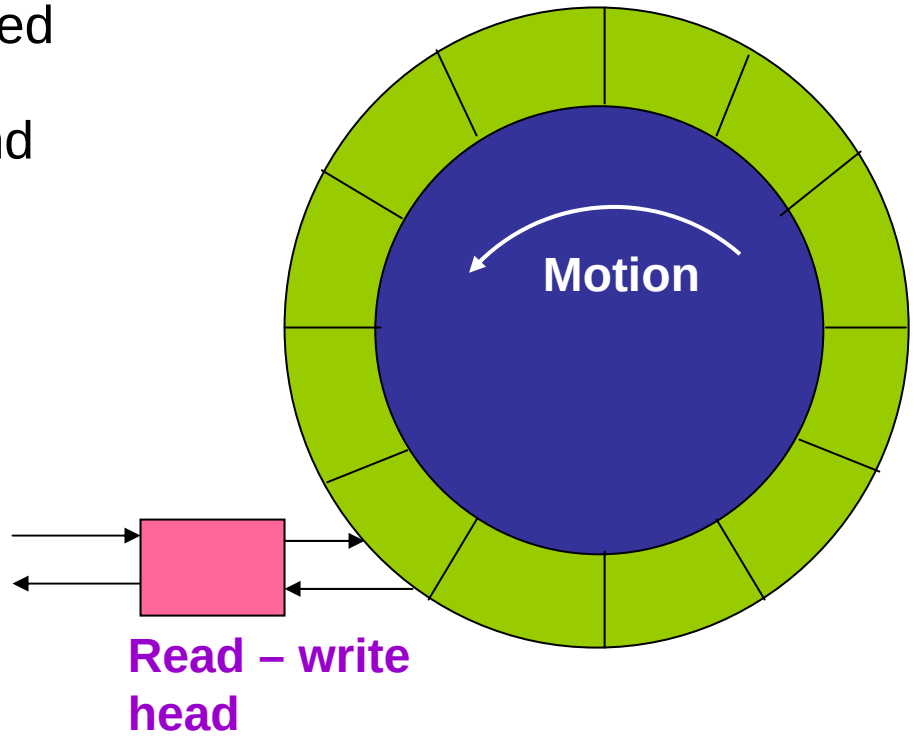
- Internal memory capacity is expressed in terms of bytes or words.
- External memory capacity is expressed in terms of bytes (depends on words in memory)
- Total memory = number of words × word length
- Number of words = $2^{\text{address bus width}}$
- Word length = Data bus width

Unit of transfer

- Internal memory – number of data lines into and out of the main memory module
- External memory – blocks – longer units than a word

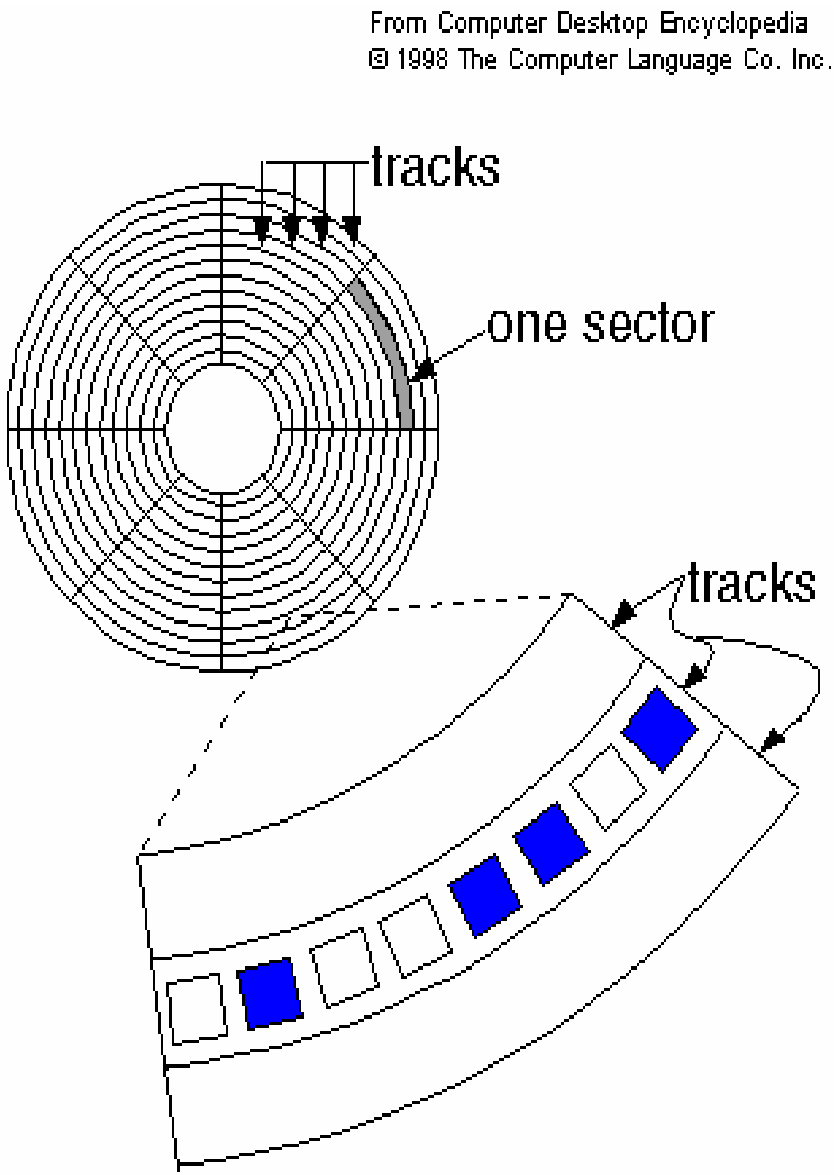
Access Methods

- Four types
  - Sequential Access
    - Shared read/write head is used, and this must be moved its current location to the desired location, passing and rejecting each intermediate record.
    - So, the time to access an arbitrary record is highly variable
    - Accesses the memory in predetermined sequence
    - Slower than random access memory
    - Ex: Magnetic Tapes



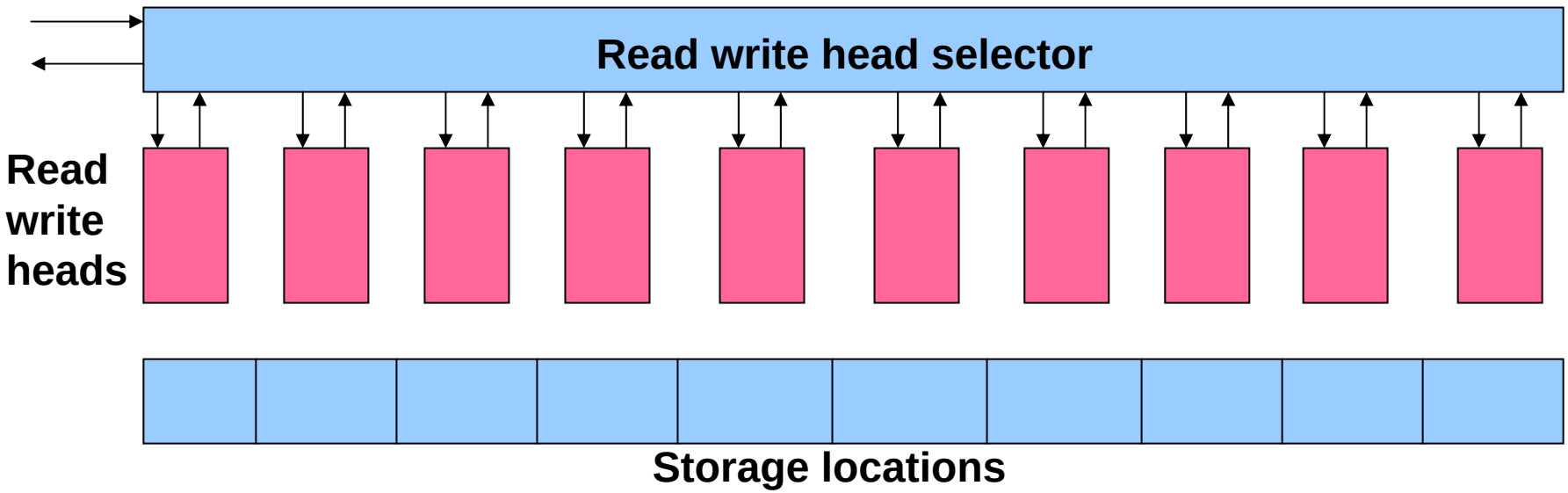
Access Methods contd.,

- Direct access
  - Also referred as semi random access memory
  - Shared read/write head is involved.
  - Access time is variable
  - The track is accessed randomly but access within each track is serial
  - Ex: Magnetic Disk



Access methods contd.,

- Random Access
  - Each addressable location in memory has unique, physically wired – in addressing mechanism
  - Time to access a location is independent of the sequences of prior access and is constant
  - Main memory systems are a random access
  - Storage locations can be accessed in any order
  - Semi conductor memories



Access Methods contd.,

- Associate Access
  - Word is retrieved based on portion of its contents rather than its address
  - Has own addressing mechanism
  - Retrieval time is constant
  - Access time is independent of location or prior access patterns
  - Cache memories

Performance

- Access time
  - The time required to read / write the data from / into desired record
  - Depends on the amount of data to be read / write
  - Random access memory
    - Time from the instant that an address is presented to the memory to the instant that data have been stored or made available for use.
  - Non-random access memory
    - Time it takes to position read-write head at the desired track (seek time) + transfer time + to position read-write head at the desired sector (rotational latency)
- Memory Cycle time
  - Access time + time required before a second access can commence
  - Access + latency

Performance contd.,

- Transfer rate
  - Rate at which the data can be transferred into or out of a memory unit
  - Random access memory
    - 1/cycle time
  - Non-Random access memory
    - $T_n = T_a + (N/R)$, where
      - T_n – average time to read or write N bits
      - T_a – average access time
      - N – Number of bits
      - R – Transfer rate, in bits per second (BPS)

Physical characteristics

- Volatile memory
  - Information decays naturally or lost when electrical power is switched off
- Non-volatile memory
  - Once recorded is retained until deliberately changed
  - No electrical power is needed to retain information
  - Magnetic surface memories
- Semiconductor memories may be either volatile or non-volatile
- Non-erasable memory
  - Cannot be altered, except by destroying the storage unit (ROM)
  - A practical non-erasable memory must also be non-volatile

Organization

- Physical arrangement of bits to form words
- 2 types
  - 1 dimensional
  - 2 dimensional

Byte Storage Methods

- Big-Endian
  - Assigns MSB to least address and LSB to highest address
  - Ex: 0 × DEADBEEF

Memory Location	**Value**
Base Address + 0	DE
Base Address + 1	AD
Base Address + 2	BE
Base Address + 3	EF

Byte Storage Methods contd.,

- Little Endian
  - Assigns MSB to highest address and LSB to least address
  - Ex: 0 × DEADBEEF

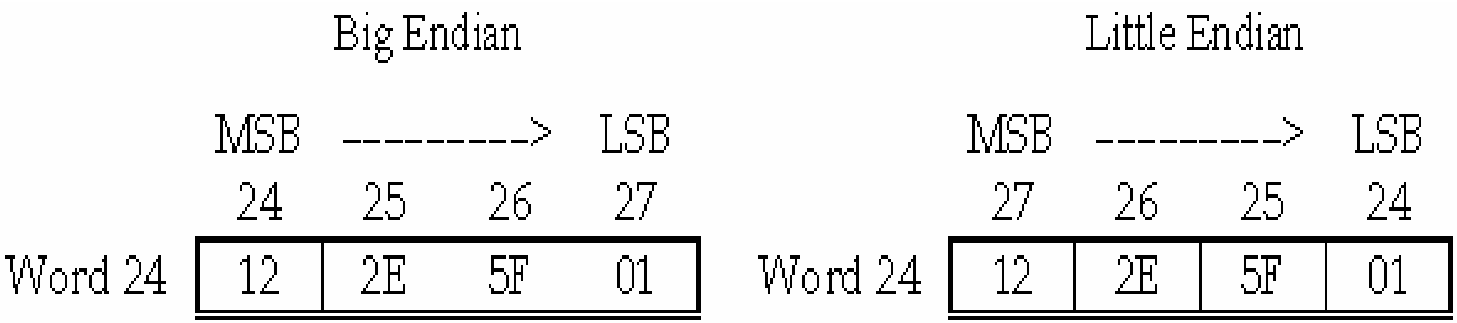
Memory Location	**Value**
Base Address + 0	EF
Base Address + 1	BE
Base Address + 2	AD
Base Address + 3	DE

Byte Storage Methods contd.,

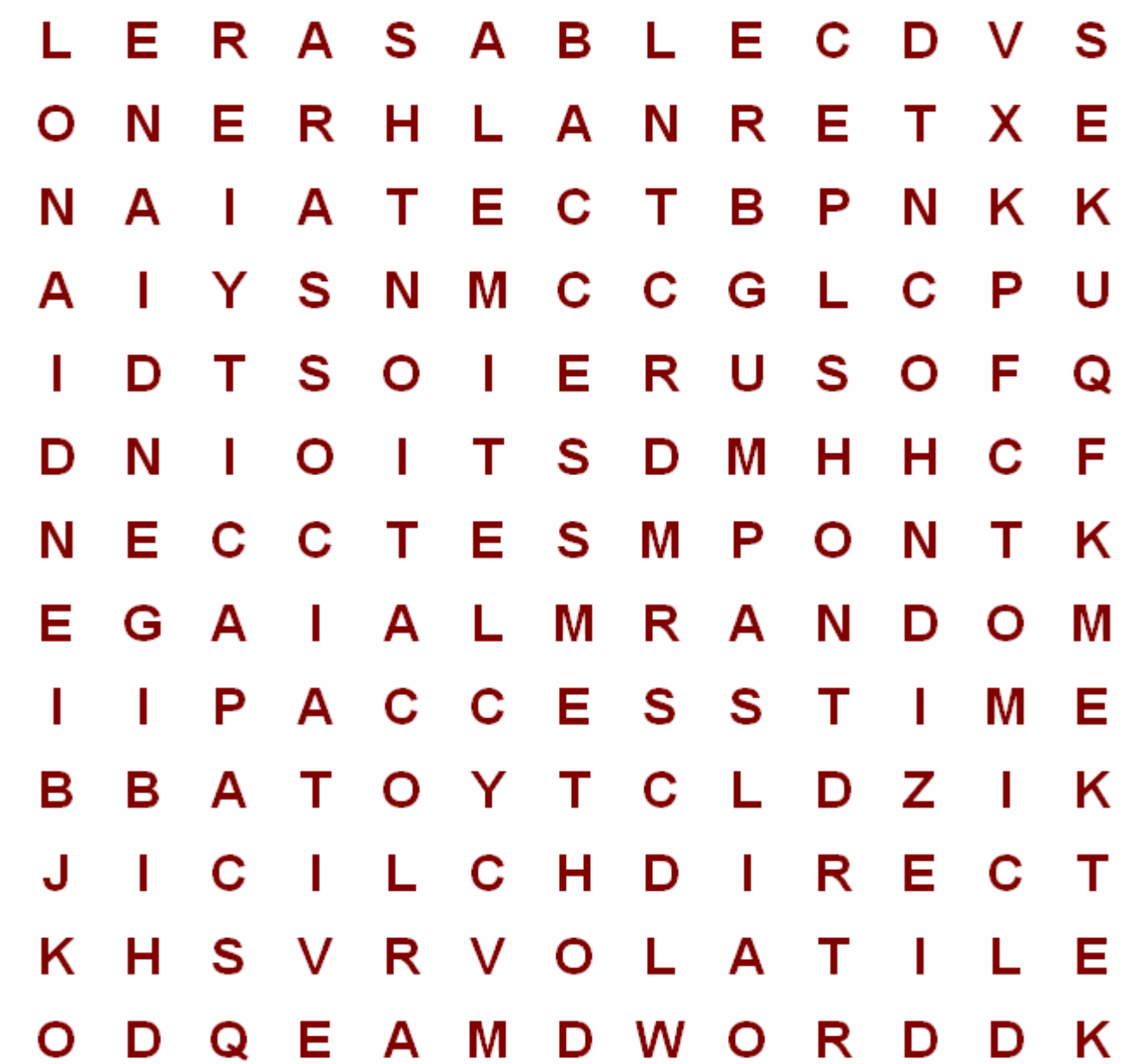
- Little Endian
  - Intel × 86 family
  - Digital equipment corporation architectures (PDP – 11, VAX, Alpha)
- Big Endian
  - Sun SPARC
  - IBM 360 / 370
  - Motorola 68000
  - Motorola 88000
- Bi-Endian
  - Power PC
  - MIPS
  - Intel’s 64 IA - 64

Example

- **Example**: Show the contents of memory at word address 24 if that word holds the number given by 122E 5F01H in both the big-endian and the little-endian schemes?



Recap: Find the Memory characteristics that are hidden



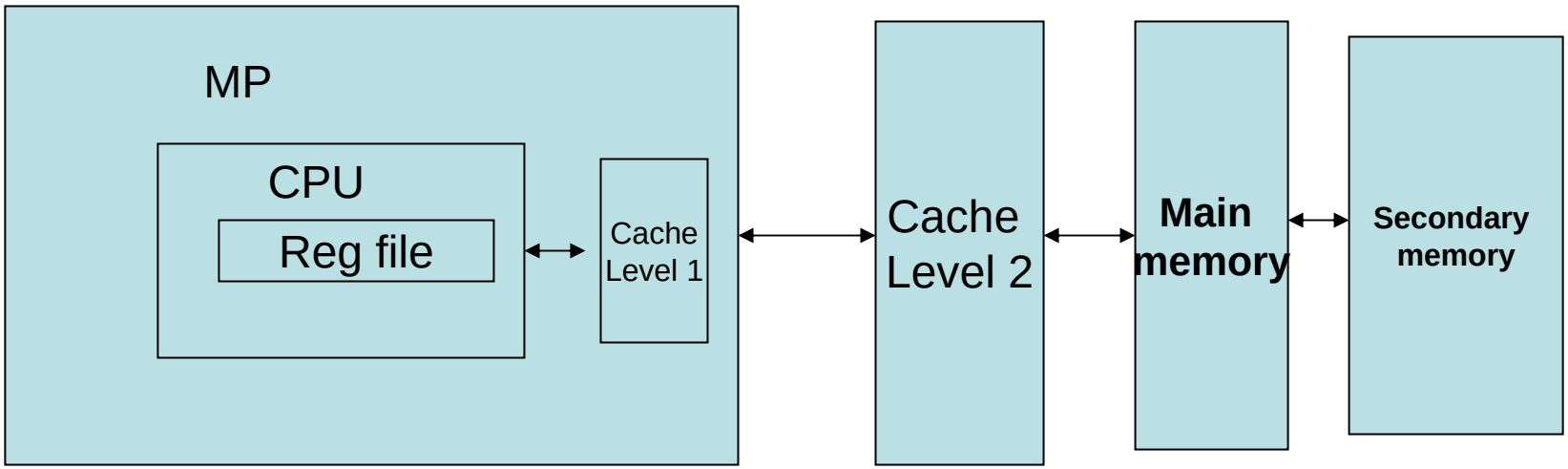
The hidden Memory Characteristics are:

ACCESS METHOD
ACCESSTIME
ASSOCIATIVE
BIENDIAN
BIGENDIAN
BLOCK
CAPACITY
CPU
CYCLETIME
DIRECT
ERASABLE
EXTERNAL
LOCATION
RANDOM
VOLATILE
WORD

Solved puzzle

Memory hierarchy

- Multilevel memory system



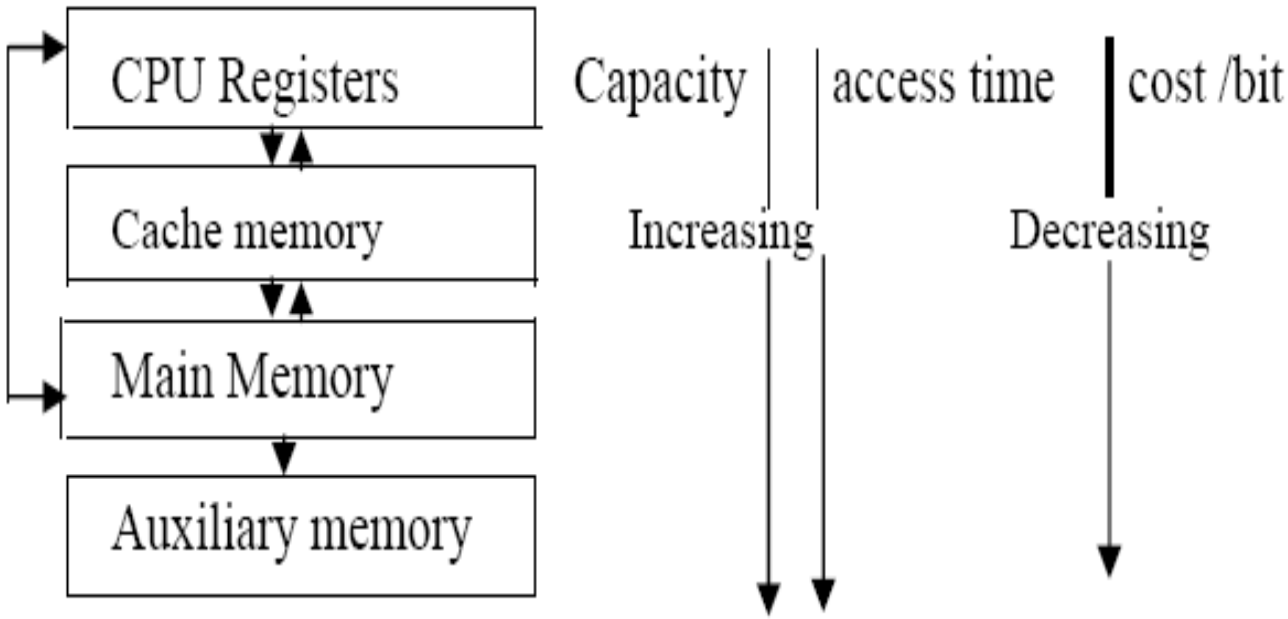
CPU REGISTER

- Temporary storage of inst & data.
- Usually form as general purpose register file for storage data as it is processed.
- Each register in the register file can be individually addressed.

- **MAIN MEMORY**
- Program and data that are in active use.
- Storage locations in the main memory are directly addressed by CPU.
- Access time is more than CPU register and the capacity of main memory is large and it is physically separable from CPU.

- **SECONDARY MEMORY**
- Large in capacity but slower than main memory.
- Stores programs, data files are not required by the CPU.
- Access time is large when compared with main memory.
- **CACHE**
- Positioned b/w CPU register and main memory
- Several levels of cache memory is also available.
- Storage capacity is less but its access time is less than main memory.
- The goal of every memory system is to provide adequate storage capacity with an acceptable level of performance and cost.

Memory hierarchy contd.,



Random-access memories

- Storage unit composed of a large number (2m) of addressable locations.each of which stores a w-bit word.
- The address of the target location to be accessed is transferred via the address bus to the RAM’s address buffer.
- The address is then processed by the address Decoder,which selects the required location in the storage cell unit.
- A control line indicate the type of access to be performed. If a read operation (load) is requested , the contents of the addressed location are transferred from the storage cell unit to the data buffer and from there to the data bus.
- If a write (store) is requested, the word to be stored is transferred from the data bus to the selected location in the storage unit.

- The storage unit is made up of many identical 1-bit memory cells and their interconnections.
- The actual no of line connected to the cell and their functions depend on the memory tech and addressing scheme .
- Each cell is connected to a set of data , address, and control signals.
- One physical line often has several logical function. Ex both address and data.
- The drivers, decoders, and control circuits form the access circuitry of RAM an can have significant impact on the total size and cost of the memory.

2-D RAM addressing scheme

- The m-bit address word is divided into two parts x any y consist Mx and My bits resp.
- Total no of cells N=NxNy.
- A cell is selected by the coincidence of signal applied to its X and Y address lines.
- The 2-D org requires much less access circuitry than 1-D org for same storage capacity.

Semiconductor RAMs

- Two categories:
  - SRAMs
  - DRAMs

SRAM

- SRAM consists of memory cells that resemble the flip-flop used in processor design.
- SRAM cells differ from flip-flop primarily int the methods use to address the cell an transfer data to and from them.
- Multifunction lines minimize storage-cell complexity and the number of cell connections thereby facilitating the manf of very large 2-D array of storage cells.

DRAM

- The DRAM cell the 1 and 0 states correspond to the presence or absence of stored charge in a capacitor controlled by a transistor switching circuit
- Single transistor whereas a static cell requires up to six transistors higher storage density is achieved with DRAM
- The charge stored in a DRAM cell tends to decay with time and cell must be periodically refreshed.
- SRAM and DRAM are volatile, the stored infor is lost when the power source is removed.

- A) The six-transistor SRAM cell superficially resembles a flipflop. A signal applied to the address line (word line) by the address ecoder selects the cell for either the read or write operation.
- The two data line also bit line are used in a complex way to transfer the stored data and its complement between the cell and the data drivers.

- B) one transistor DRAM cell comprises an MOS transistor T,which acts switch and capacitor C, which stores a data bit .
- Apart form power an ground the cell has only tow external connections data(bit) line is place on the data line.
- A signal is applied to the address line to switch on T. This action transfers a charge to C if the data line 1 no charge is transferred otherwise.
- To read the cell, the address line is again activated transferring any charge stored in C to the data line where it is detected. Since the readout process is destructive the data being read out is amplified and subsequently written back to the cell this process may be combined with the periodic refreshing operation required by dynamic memories

Advantage

- Small size which means that ICs with very high cell density can be manf and low power consumption.

Cache memory

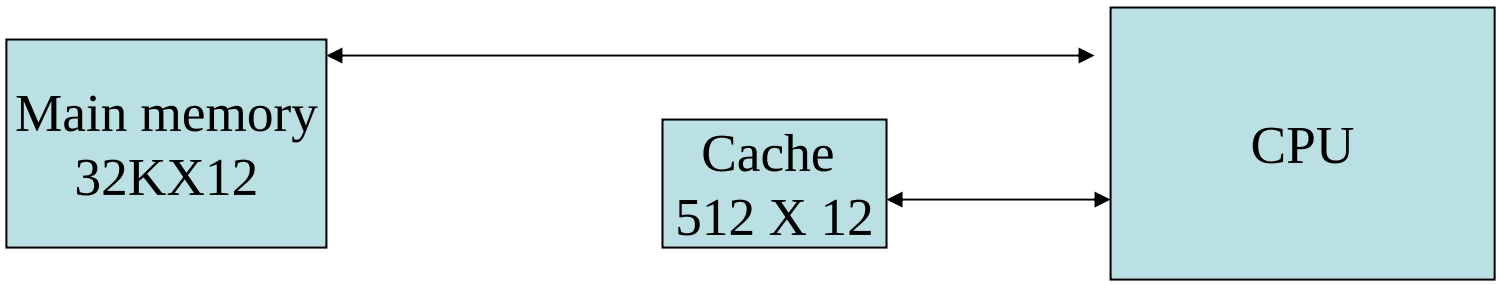
- If the active portions of the program and data are placed in a fast small memory the average memory access time can be reduced ,thus reducing the total execution time of the program.
- Such fast small memory is referred to cache memory.
- It is placed between the CPU and main memory .
- CM access time is less than access time of MM hierarchy by a factor 5 to 10.
- The cache is the fastest component in the memory hierarchy and approaches the speed of components.

- **operation of the cache:** if the word is found in the cache ,it read from the fast memory.
- if the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word.
- A block of words containing the one just accessed is then transferred from main memory to cache memory.

- **Hit ratio:**
- The performance of cache memory is frequently measured in terms of a quantity called **hit ratio**.
- When the CPU refers to memory and finds the word in cache, is said **hit.**
- If the word is not found in cache, it is in main memory and it count as **miss.**
- The ratio of the number of hits divided by the total CPU references to memory (hits plus misses) is the **hit ratio.**
- The basic characteristic of cache memory is fast access time. Therefore very little or no time must be wasted when searching for words in the cache .
- The transformation of data from main memory to cache memory is referred as **mapping process**

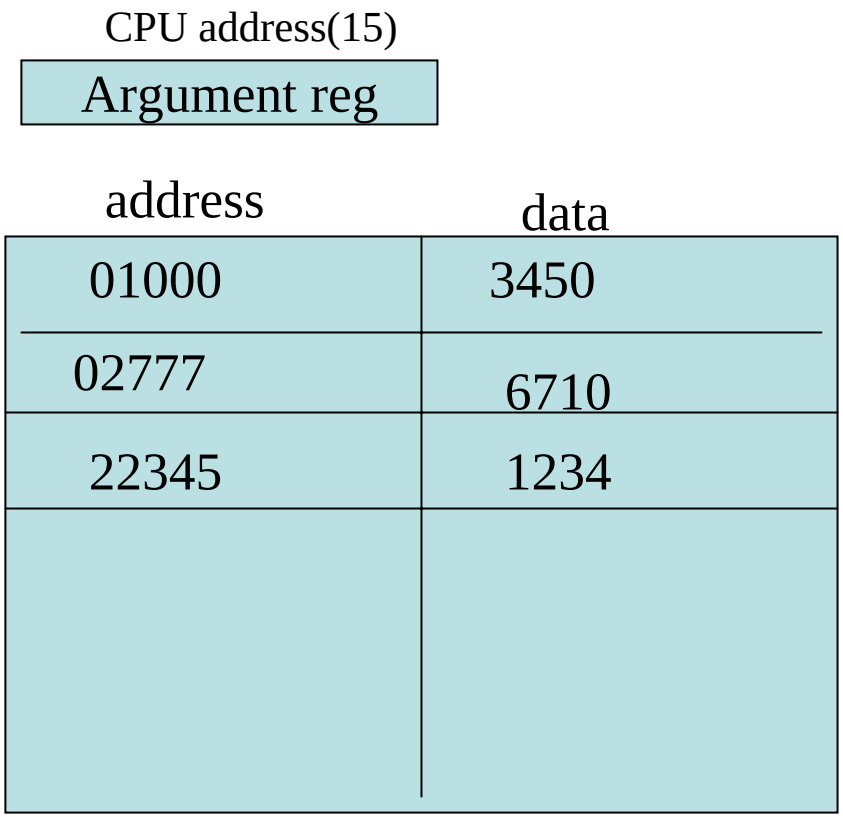
mapping

- Associative mapping
- Direct mapping
- Set-associative mapping
- Ex



Associative mapping

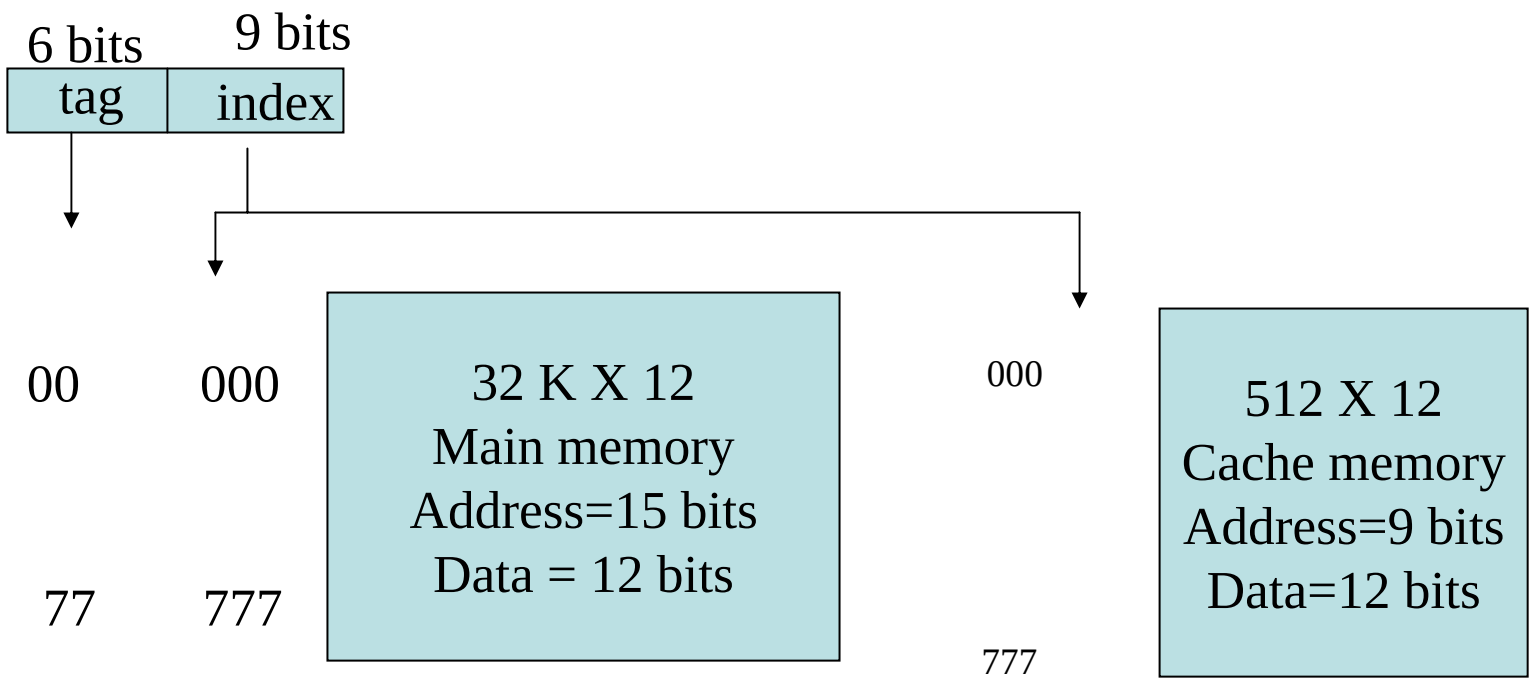
Associative mapping cache



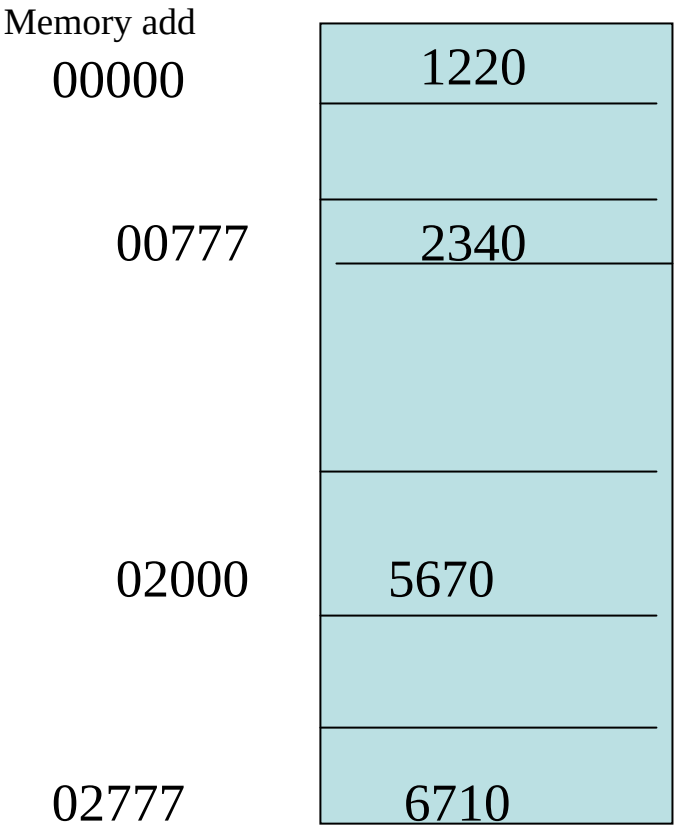
- The associative memory stores both address and content (data) of the memory word. This permits any location cache to store any word from main memory .
- Address-15 and data-12.
- A CPU address of 15 bits placed in the argument register and associative memory is searched for a matching address.
- If address is found, the corresponding 12 bit data is read and sent to the CPU.
- If no match occurs the main memory is accessed for the word. the address data pair is then transferred to the associative cache memory. If the cache is full an address data pair must be displaced to make room for a pair that is needed and not presently in the cache.

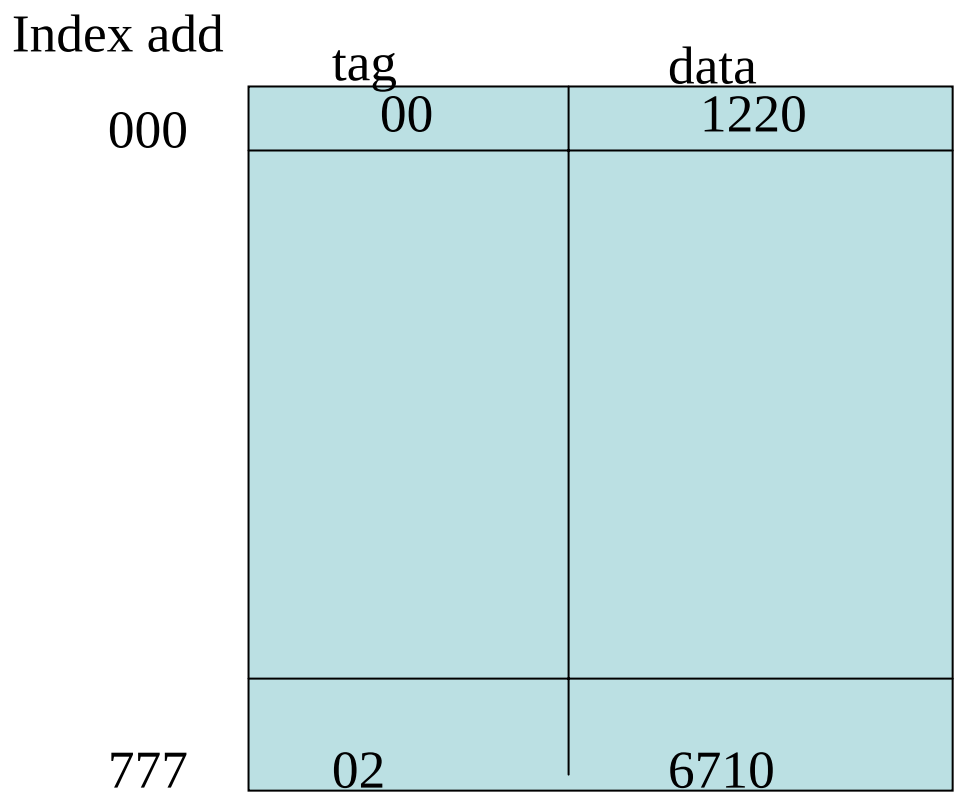
Direct mapping

- Associative memories are expensive compared to random-access memories because of the added logic associated with each cell.



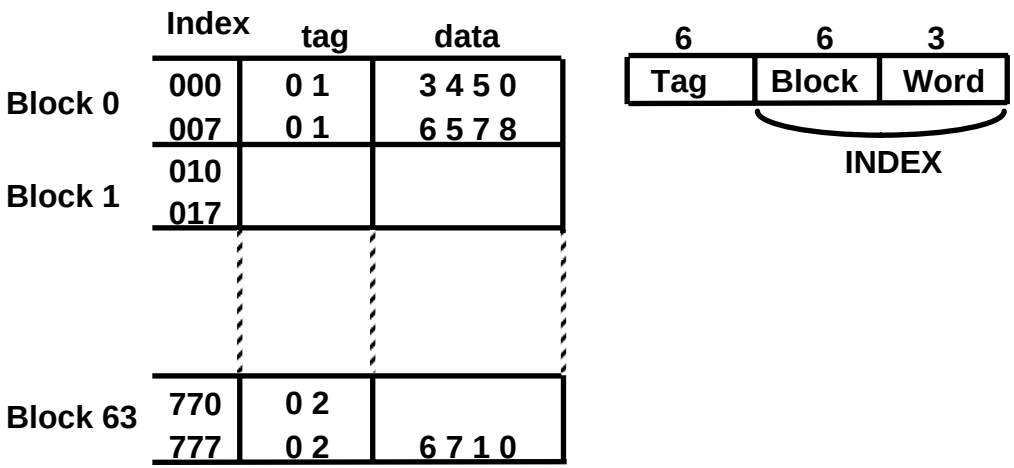
- 2(k) words in cache memory 2(n) MM
- N bit divided into two fields k bits index and n-k bits for tag field.



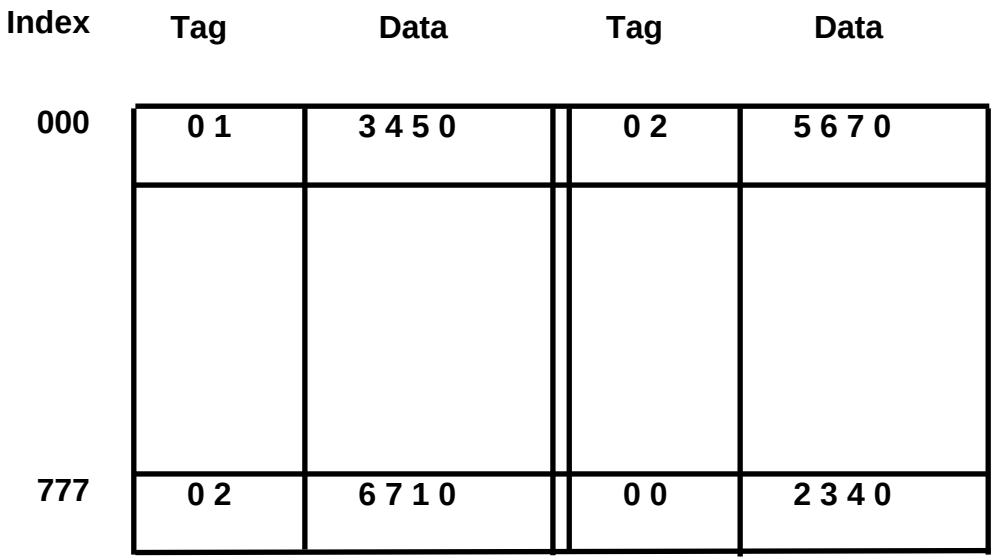


DIRECT MAPPING

Direct Mapping with block size of 8 words



Set Associative Mapping Cache with set size of two



Set associative mapping

- **Replacement algorithm**
- When a miss occurs in a set-associative cache and set is full ,it is necessary to replace one of tag-data items with a new value. Most common replacement algorithm used (FIFO) and least recently used (LRU).
- With random replacement policy, control choose one tag-data item for replacement at random.
- FIFO procedure selects for replacement the item that has been in the set the longest.
- LRU algorithm selects for replacement the item that has been least recently used by CPU.

CACHE WRITE

Write Through

When writing into memory

If Hit, both Cache and memory is written in parallel
If Miss, Memory is written
For a read miss, missing block may be overloaded onto a cache block

Memory is always updated
-> Important when CPU and DMA I/O are both executing

Slow, due to the memory access time

Write-Back (Copy-Back)

When writing into memory

If Hit, only Cache is written
If Miss, missing block is brought to Cache and write into Cache
For a read miss, candidate block must be written back to the memory

Memory is not up-to-date, i.e., the same item in Cache and memory may have different value

- **Cache initialization.**
- Cache is initialized when power is applied to the computer on when main memory is loaded with a complete set of program from auxiliary memory .
- After initialization the cache is considered to be empty but in effect it contains some non valid data indicate each word in cache a valid but to indicate whether or not the word contains valid data cache it initialized by clearing all valid bits to 0 valid bit of a particular cache word is set to 1 the first time this word is loaded from main memory and stays set unless the cache has to be initialized again.

Problem -1

- A digital computer has a memory unit of 64 K X 16 and a cache memory of 1K words. The cache uses direct mapping with a block size of four words.
  A) How many bits are there in tag, index, block and word fields of the address format?
- b) How many bits are there in each word of cache and how are
- they divided into functions? Include a valid bit.
- c) How many blocks can the cache accommodate?

SOLUTION

Memory capacity--- 64KX16

Cache capacity------1KX16

Direct mapping with block size ---4 word

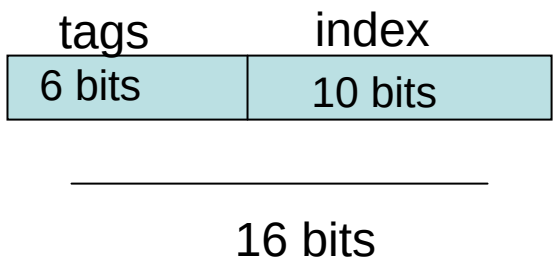
a) Memory space= $2^6 * 2^{10} \rightarrow 2^{16}$

Therefore 2^{16} words in memory

Therefore 16 bits are required to address the memory

Cache 2^{10} words

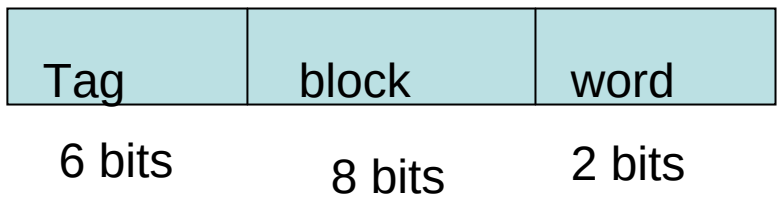
10 bits are register to address the cache



- Cache divided into blocks of 4 words

  Number of blocks ---- $\frac{2^{10}}{2^2}$ = 2^8 blocks

  256 blocks



- b) Tag bits + data word bits +valid bit

  6 + 16 + 1--→ 23 bits

C) Blocks 2^8 =256 blocks

---

2) An address space is specified by 24 bits and coresponding memory space by 16 bits.
   - How many words are there in the address space?
   - How many words are there in the memory space?
   - If a page consists of 2K words, how many pages and blocks are there in the system?

Address space—24 bits
memory space—16 bits
a) Words in address space – 2^{24}

b) Words in memory space -- 2^{16}

c) Page size – 2K words
How many pages --- $\frac{2^{24}}{2X2^{10}}$ = 2^{24-11} = 2^{13} =8k

How many blocks------ $\frac{2^{16}}{2X2^{10}}$ = 2^{16-11} = 2^5 = 32 blocks

PROBLEM 3

- How many 128 X 8 RAM chips are needed to provide a memory capacity of 2048 bytes?
  - How many lines of the address bus must be used to access 2048 bytes of memory? How many of these lines will be common to all chips?
  - How many lines must be decoded for chip select? Specify the size of the decoders?

SOLN

(2) $\frac{2048}{128} = 16$ chips

(b) $2048 = 2^{11}$ 11 lines to address 2048 bytes

$128 = 2^{7}$ 7 lines to address each chip

4 lines to decoder for selecting 16 chips

(c) 4x16 decoder

Problem 4

- A 16MB main memory has 32 KB cache with 8 bytes per line

  i) How many lines are there in the cache?

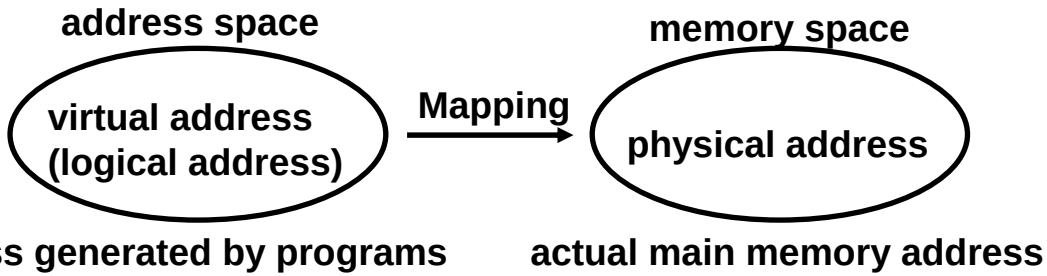
  ii) Show how the main memory and cache memory is organized when the cache is direct-mapped.

  iii) Show how the Main memory address is partitioned.

VIRTUAL MEMORY

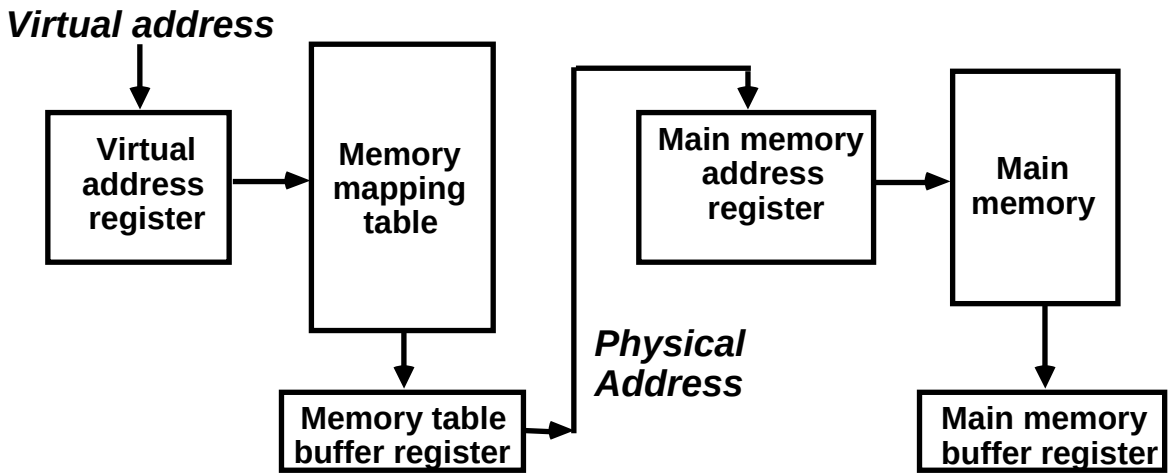
Give the programmer the illusion that the system has a very large memory, even though the computer actually has a relatively small main memory

Address Space(Logical) and Memory Space(Physical)



Address Mapping

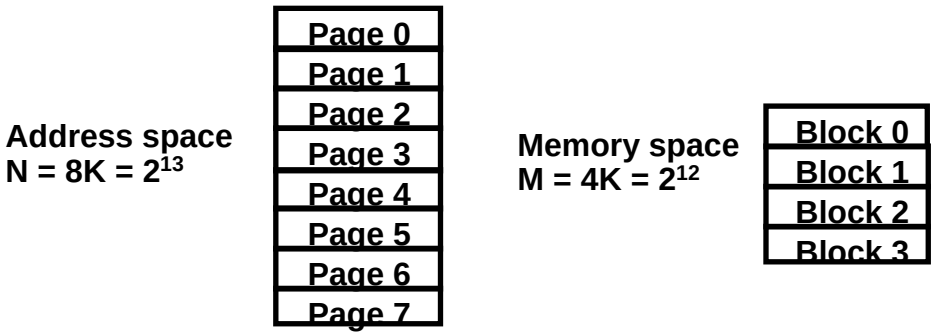
Memory *Mapping Table* for *Virtual Address -> Physical Address*



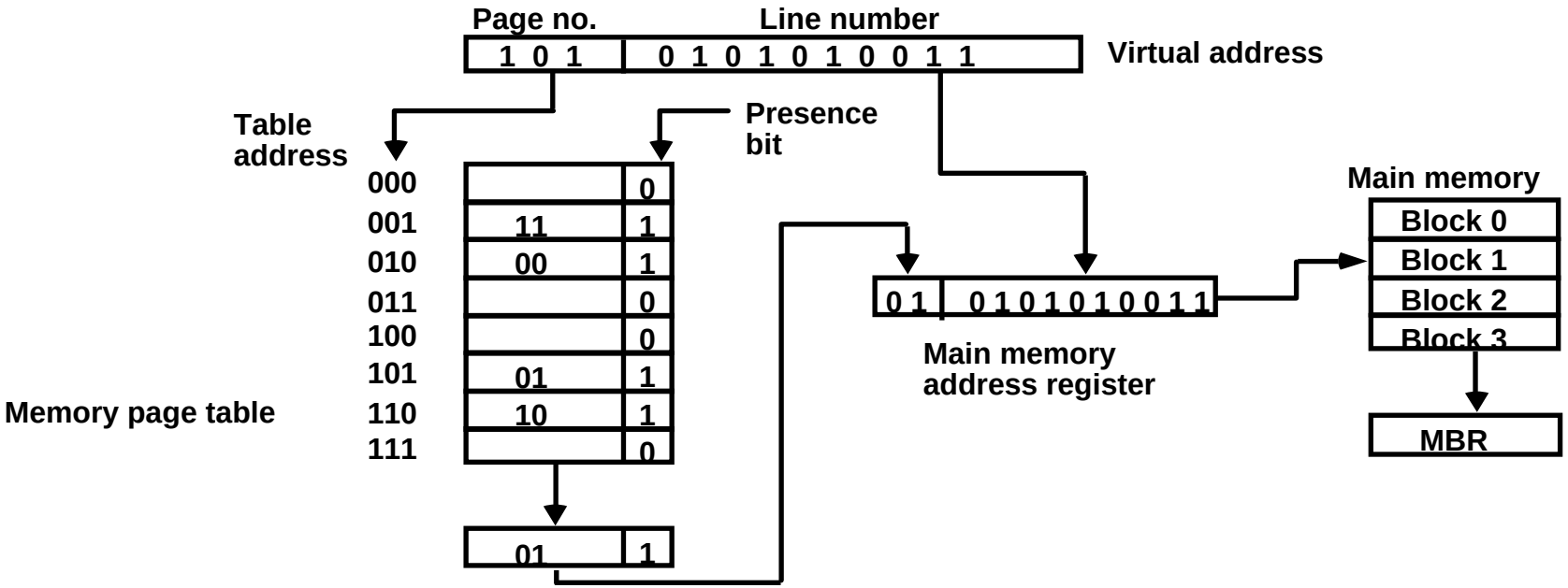
ADDRESS MAPPING

Address Space and Memory Space are each divided into fixed size group of words called *blocks* or *pages*

1K words group



Organization of memory Mapping Table in a paged system



ASSOCIATIVE MEMORY PAGE TABLE

Assume that

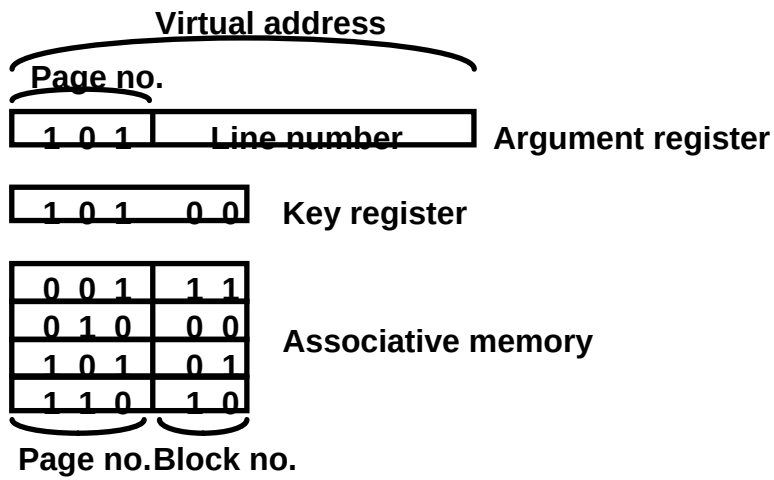
Number of Blocks in memory = m

Number of Pages in Virtual Address Space = n

Page Table

- Straight forward design -> n entry table in memory
Inefficient storage space utilization
<- n-m entries of the table is empty

- More efficient method is m-entry Page Table
Page Table made of an Associative Memory
m words; (Page Number:Block Number)



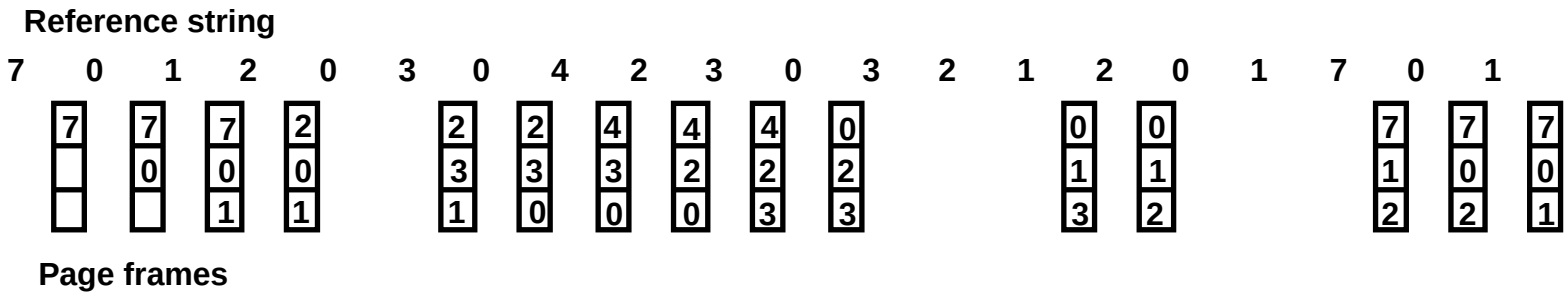
Page Fault

Page number cannot be found in the Page Table

- Mapping table may be stored in separate memory or in main memory .
- Case1:
- Additional memory unit is required as well as one extra memory access time.
- Case2:
- Table space from MM and two accesses to memory are required with pgm running at half speed.
- Case3:
- Use associative memory.

PAGE REPLACEMENT ALGORITHMS

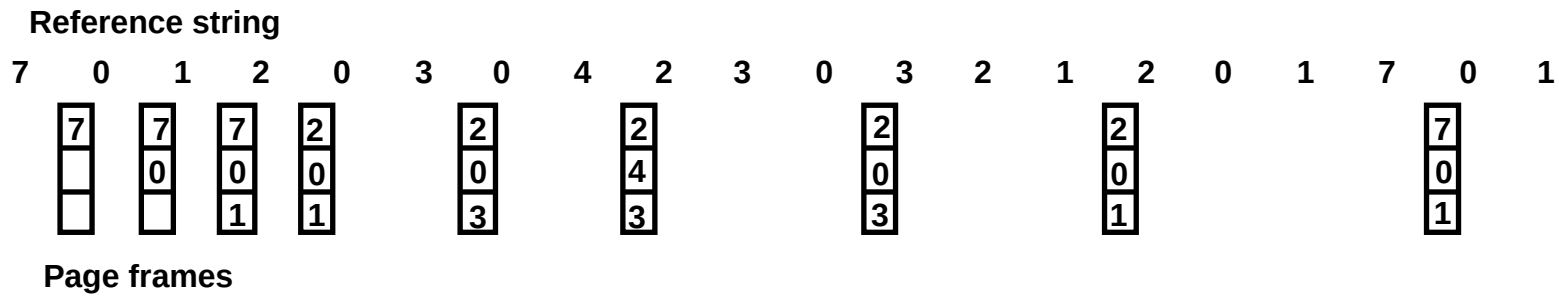
FIFO



FIFO algorithm selects the page that has been in memory the longest time
Using a queue - every time a page is loaded, its identification is inserted in the queue
-
Easy to implement
May result in a frequent page fault

Optimal Replacement (OPT) - Lowest page fault rate of all algorithms

Replace that page which will not be used for the longest period of time

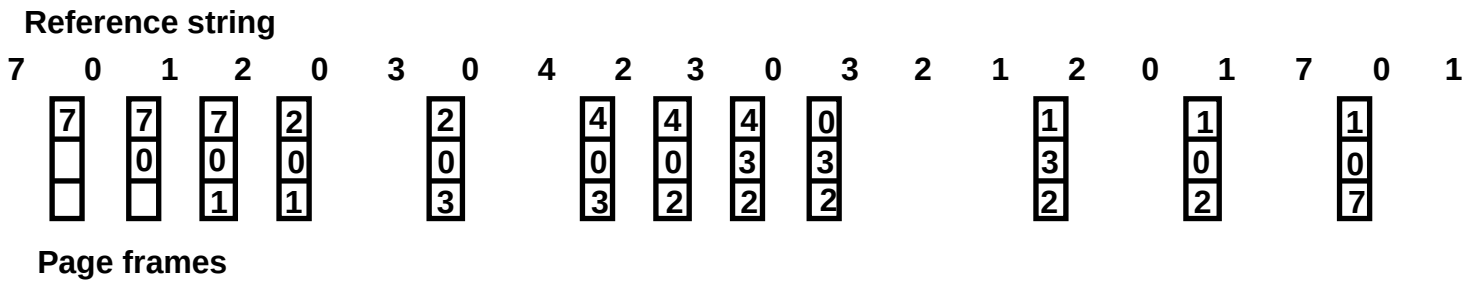


PAGE REPLACEMENT ALGORITHMS

LRU

- **OPT is difficult to implement since it requires future knowledge**
- **LRU uses the recent past as an approximation of near future.**

Replace that page which has not been used for the longest period of time



- **LRU may require substantial hardware assistance**
- **The problem is to determine an order for the frames defined by the time of last use**