# Computer Organization
## (Autonomous)

UNIT IV

Sections - A & D

Prepared by
Anil Kumar Prathipati, Asst. Prof., Dept. of CSE.

## SYLLABUS

- **The Memory System:** Memory Hierarchy, Main memory - RAM and ROM Chips, Memory Address Maps, Memory Connection to CPU, Auxiliary memory – Magnetic Disks, Magnetic Tape, Associative Memory – Hardware Organization, Match Logic, Cache Memory – Associative Mapping, Direct Mapping, Set- Associative Mapping, Writing into Cache, Virtual Memory – Address Space and Memory Space, Address Mapping using Pages, Associative Memory Page Table, Page Replacement.



## INDEX





## Memory Organization



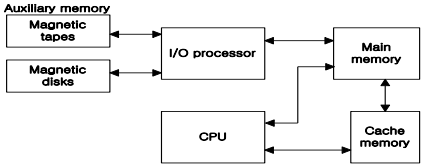
The memory system can be **characterized** with

- ***Location:*** Where it can be located, Processor, Internal, External .
- ***Capacity:*** size in terms of bytes, KB, MB, GB, etc…
- ***Unit of transfer:*** How many bits can be moved like bytes, words, Blocks, etc….
- ***Access method:*** How you pick of data Sequential, Direct, Random, etc…
- ***Performance:*** Transfer rate n terms of bps
- ***Physical type:*** Which material we are using like semiconductor, Magnetic, Optical, etc…
- ***Physical characteristics:*** like power consumption, information loss, volatile, etc…
- ***Organization:*** How it stored like continues, interleaved, etc….

## Memory Organization



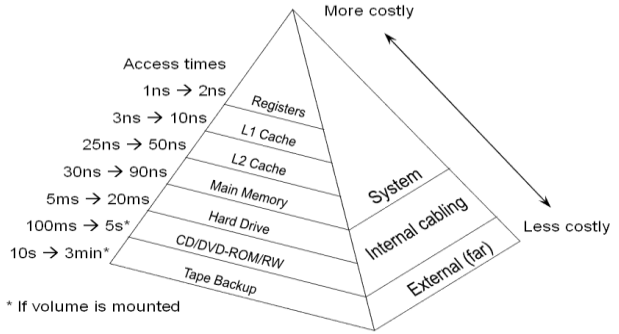
- 12-1 Memory Hierarchy
  - Memory hierarchy in a computer system
    - **Main Memory** : memory unit that communicates directly with the CPU (RAM)
    - **Auxiliary Memory** : device that provide backup storage (Disk Drives)
    - **Cache Memory** : special very-high-speed memory to increase the processing speed (Cache RAM)




  - Multiprogramming
    - enable the CPU to process a number of independent program concurrently.



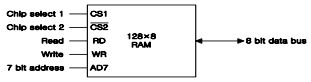








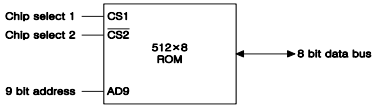

- 12-2 Main Memory
  - Bootstrap Loader
    - A program whose function is to start the computer software operating when power is turned on
  - RAM and ROM Chips
    - Typical RAM chip
      - » 128 X 8 RAM : $2^7$ = 128 (7 bit address lines)
    - Typical ROM chip
    - 512 X 8 ROM : $2^9$ = 512 (9 bit address lines)




(a) Block diagram

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

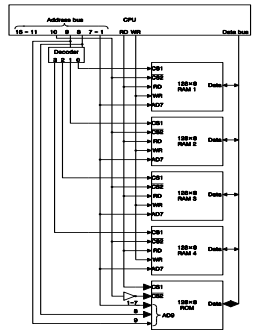
(b) Function table



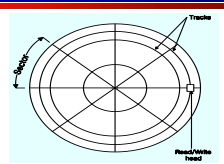


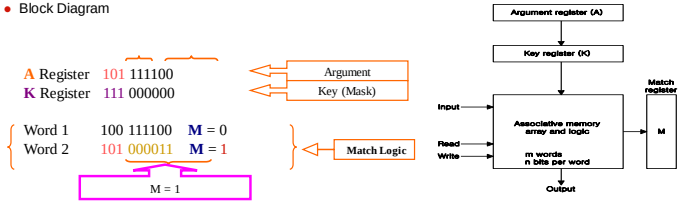



- Memory Address Map
  - Memory Configuration : **512** bytes **RAM** + **512** bytes **ROM**
    - » **1** x **512** byte **ROM** + **4** x **128** bytes RAM
  - Memory Address Map
    - » Address line 9 8
      - RAM 1 **0 0** : 0000 - 007F
      - RAM 1 **0 1** : 0080 - 00FF
      - RAM 1 **1 0** : 0100 - 017F
      - RAM 1 **1 1** : 0180 - 01FF
    - » Address line 10
      - ROM **1** : 0200 - 03FF
  - Memory Connection to CPU
    - » 2 x 4 Decoder : RAM select (**CS1**)
    - » Address line 10
      - RAM select : **CS2**
      - ROM select : **CS2** Invert



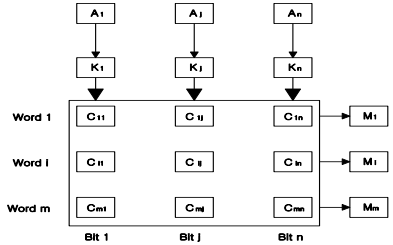
- 12-3 Auxiliary Memory
  - Magnetic Disk : FDD, HDD
  - Magnetic Tape : Backup or Program
  - Optical Disk : CDR, ODD, DVD
- 12-4 Associative Memory
  - Content Addressable Memory (CAM)
    - A memory unit accessed by content
    - Block Diagram












- 12-5 Cache Memory
  - Locality of Reference
    - the references to memory *tend to be confined within a few localized areas* in memory
  - Cache Memory : **a fast small memory**
    - keeping the most frequently accessed instructions and data in the fast cache memory
  - Cache
    - cache size : 256 K byte
    - mapping method : 1) associative, 2) direct, 3) set-associative
    - replace algorithm : 1) LRU, 2) LFU, 3) FIFO
    - write policy : 1) write-through, 2) write-back
  - Hit Ratio
    - the ratio of the number of hits divided by the total CPU references (hits + misses) to memory
      - » ***hit*** : the CPU finds the word in the cache (0.9)
      - » ***miss*** : the word is not found in cache (CPU must read main memory)
    - cache memory access time = 100 ns, main memory access time = 1000 ns, hit ratio = 0.9
      - » 1 miss : 1 x 1000 ns + penalty time(1 x 100 ns)
      - » 9 hit : 9 x 100 ns

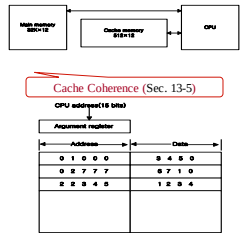


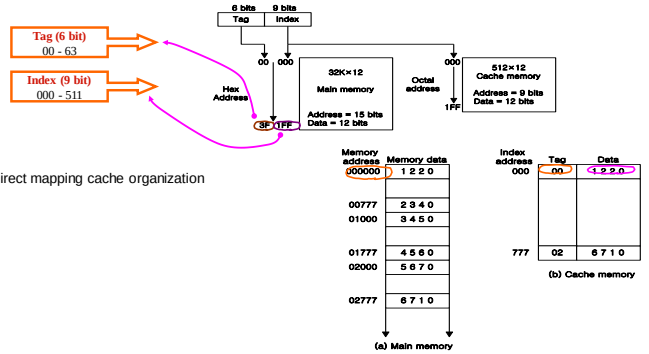


- Mapping
  - The transformation of data from main memory to cache memory
    - » 1) Associative mapping
    - » 2) Direct mapping
    - » 3) Set-associative mapping
- Example of cache memory

  main memory : **32 K** x 12 bit word (15 bit address lines)

  cache memory : **512** x 12 bit word
  - » CPU sends a 15-bit address to cache
    - **Hit** : CPU accepts the 12-bit data from cache
    - **Miss** : CPU reads the data from main memory (then data is written to cache)
- Associative mapping
  - Cache memory associative memory
  - Address Data Cache memory
  - Tag field (**n - k**) Index field (**k**)
  - $2^k$ words cache memory + $2^n$ words main memory
    - **Tag** = 6 bit (15 - 9), **Index** = 9 bit



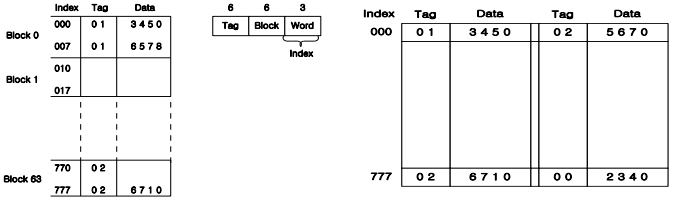



- Direct mapping cache organization





- Direct mapping cache with block size of 8 words
- 64 block x 8 word = 512 cache words size
  - 8 word block update




◆ Set-associative mapping : *Fig. 12-15* (two-way)
- Direct mapping Index tag ( **02777**, **01777** )





◆ Replacement Algorithm : cache miss or full
- 1) **LRU** (Least Recently Used): On a miss, the frame that was least recently used in replaced.
- 2) **LFU** (Least Frequently Used): It looks forward in time to see which frame to replace on a cache miss.
- 3) **FIFO** (First-In First-Out): On a miss, the frame that has been in memory the longest is replaced.
- Ex: 1 2 3 2 1 5 2 1 6 2 5 6 3 1 3 6 1 2 4 3
- Writing to Cache : Cache Coherence
  - » 1) **Write-through**
  - » 2) **Write-back**

◆ Cache Initialization
- Cache is initialized
  - » 1) when power is applied to the computer
  - » 2) when main memory is loaded with a complete set of programs from auxiliary memory
- valid bit
  - » indicate whether or not the word contains valid data



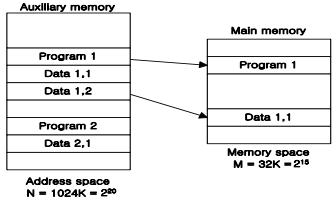


■ 12-6 Virtual Memory

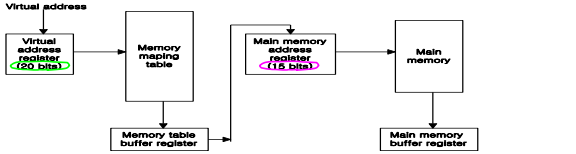
◆ Virtual Memory : Auxiliary memory → Main memory
- Translate program-generated (Aux. Memory) address into main memory location
  - » Give programmers the illusion that they have a very large memory, even though the computer actually has a relatively small main memory
- Intel Pentium Processor
  - » Physical Address Lines = $A_0$ - $A_{31}$ : $2^{32} = 2^{30} \times 2^2 = 4$ Giga
  - » Logical Address = 46 bits address : $2^{46} = 2^{40} \times 2^6 = 64$ Tera

◆ Address Space & Memory Space
- Address Space : ***Virtual Address***
  - » Address used by a programmer
- Memory Space : ***Physical Address***(Location)
  - » Address in main memory

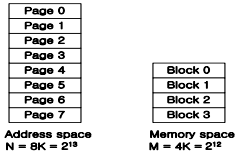
◆ ***Figure***
- address space (**N**) = 1024 K = $2^{20}$
  - » Auxiliary Memory
- memory space (**M**) = 32 K = $2^{15}$
  - » main Memory



- Memory table for mapping a virtual address
  - Translate the *20 bits Virtual address* into the *15 bits Physical address*




- Address Mapping Using Pages
  - Address mapping Address space
  - memory space
  - fixed size
    - Address space : 1 K page
    - Memory space : 1 k block



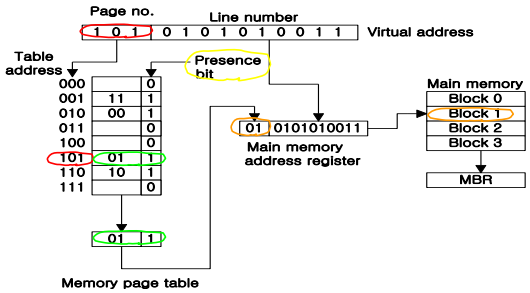








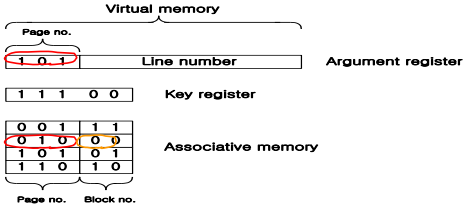
- Memory table in a paged system








- Associative memory page table
- Associative memory block number(**01**)




- Page(Block) Replacement
  - Page Fault : the page referenced by the CPU is not in main memory
    - » a new page should be transferred from auxiliary memory to main memory
  - Replacement algorithm