Computer Organization and Architecture

Under Graduate Course
(B. Tech-Information Technology, 2nd Semester)
Jan 2021-July 2021

By

Dr. Satish Kumar Singh

Associate Professor
Indian Institute of Information Technology, Allahabad
Email:sk.singh@iiita.ac.in

Performance Enhancement Techniques

Characteristics of the memory[2]

- Location
  - CPU
  - Internal
  - External
- Capacity
  - Word size: The natural unit of organization
  - Number of words or Bytes
- Unit of transfer
  - Internal:
    - Usually governed by data bus width
  - External:
    - Usually a block which is much larger than a word
  - Addressable unit:
    - Smallest location which can be uniquely addressed
    - Word internally
    - Cluster or sector on disks

Characteristics cont....

- Access method
  - Sequential
    - Start at the beginning and read through in order
    - Access time depends on location of data and previous location
    - e.g. tape
  - Direct
    - Individual blocks have unique address
    - Access is by jumping to vicinity plus sequential search
    - Access time depends on location and previous location
    - e.g. Disk
  - Random
    - Individual addresses identify locations exactly
    - Access time is independent of location or previous access
    - e.g. RAM
  - Associative
    - Data is located by a comparison with contents of a portion of the store
    - Access time is independent of location or previous access
    - e.g. cache

Characteristics cont....

- Performance
  - Access time
    - Time between presenting the address and getting the valid data
  - Memory Cycle time
    - Time may be required for the memory to "recover" before next access
    - Cycle time is access + recovery
  - Transfer Rate
    - Rate at which data can be moved
- Physical type
  - Semiconductor
  - Magnetic
  - Optical
  - Others

Characteristics cont....

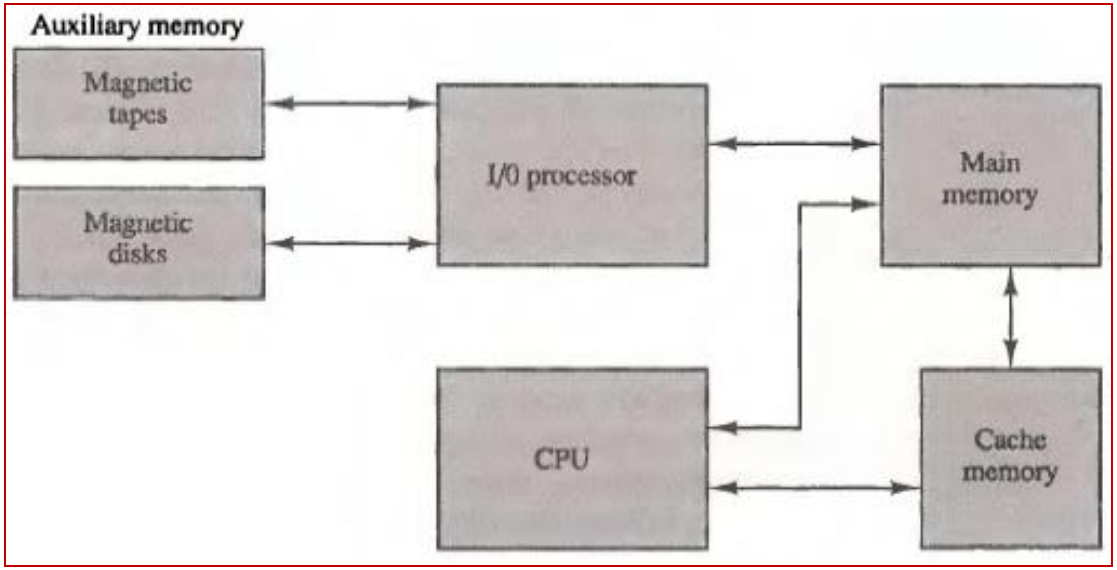
- Physical characteristics
  - Decay
  - Volatility
  - Erasable
  - Power consumption
- Organization
  - Physical arrangement of bits into words
  - Not always obvious
  - e.g. interleaved

Memory Hierarchy

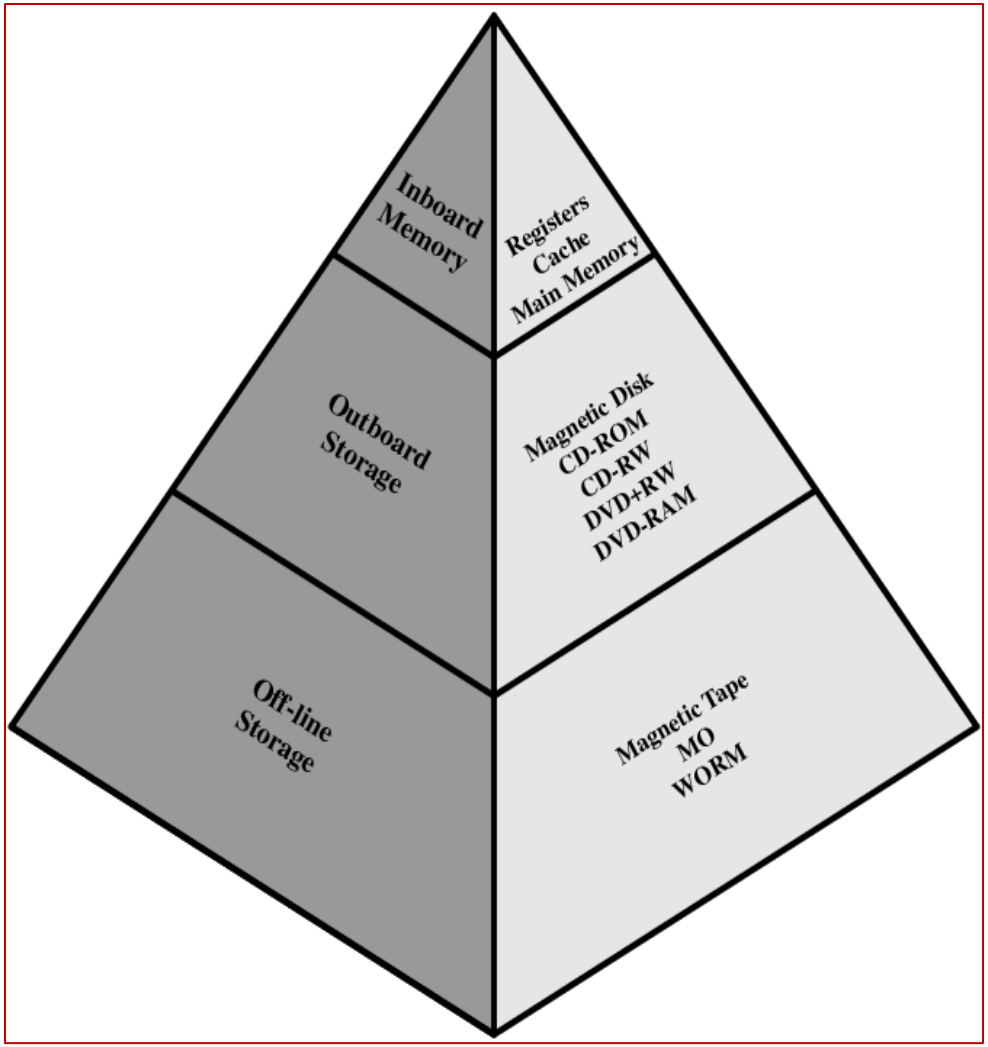
- Registers
  - In CPU
- Internal or Main memory
  - May include one or more levels of cache
  - "RAM"
- External memory
  - Backing store

Memory Hierarchy cont….

- Computational efficiency enhancement ← Using additional storage.
- General Memory used in computer systems,

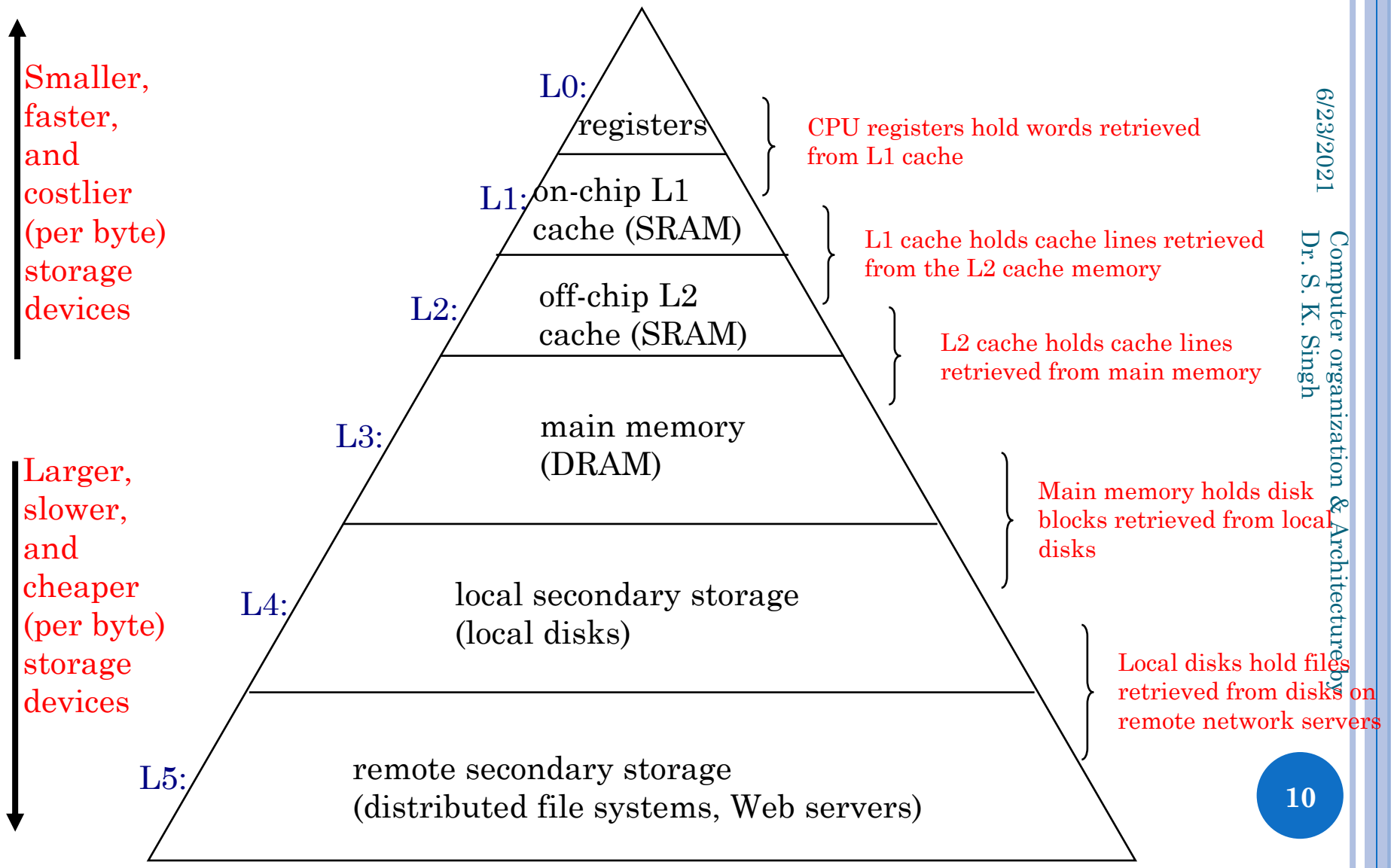


Memory Hierarchy cont….



Ref: COA, W. Stallings

An Example Memory Hierarchy

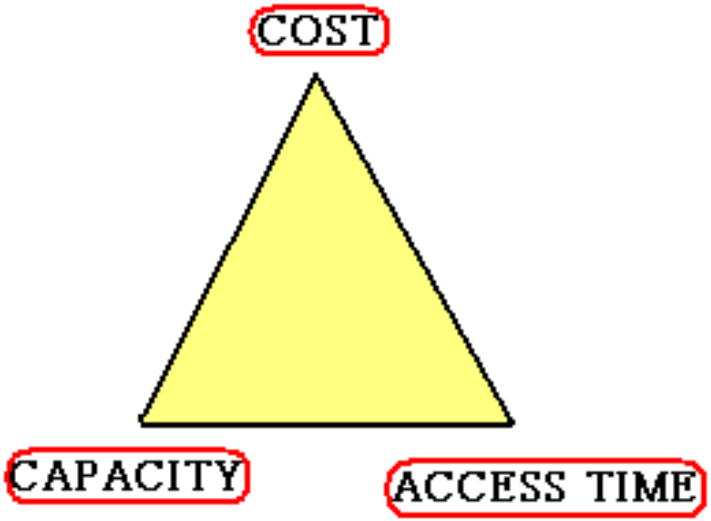


Memory Hierarchy cont….

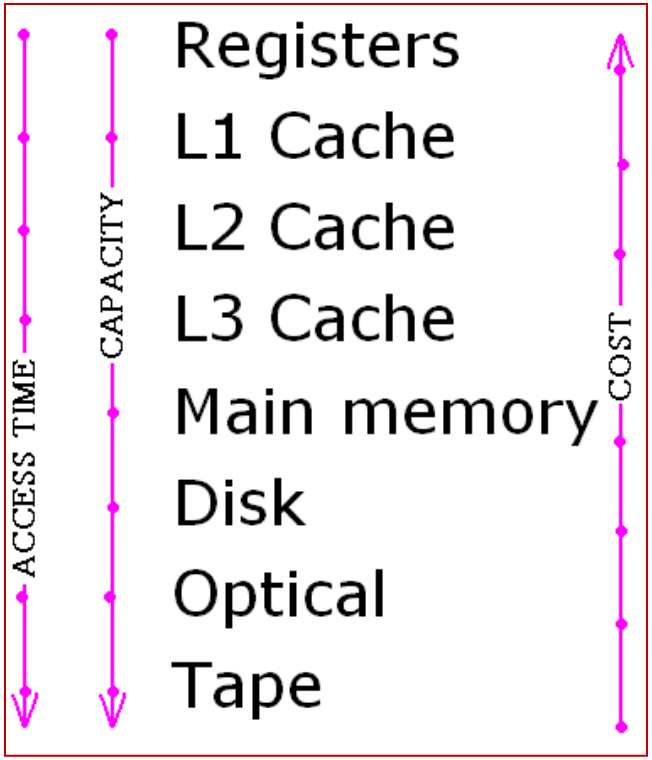
- Computer Memory
  - Main/Primary memory
    - Direct communication from CPU
    - Contains the program and data currently need by the processor
      - Processor registers (may be treated as)
      - RAM
      - ROM
      - CACHE (L1-L3)
  - Auxiliary/Secondary storage
    - Provides back-up storage for
    - System files, large data files and other backup information
      - Magnetic disks
      - Tapes
      - Optical
      - Others

Memory Hierarchy cont….

- Tradeoff triangle



Memory Hierarchy cont….



Main Memory-1[Random Access Memory]

- Key features
  - RAM is packaged as a chip
  - Basic storage unit is a cell (one bit per cell)
  - Multiple RAM chips form a memory
- Static RAM (SRAM)
  - Each cell stores bit with a six-transistor circuit
  - Retains value indefinitely, as long as it is kept powered
  - Relatively insensitive to disturbances such as electrical noise
  - ***Faster*** and ***More expensive*** than DRAM
  - Used to implement the ***Cache*** memory.
- Dynamic RAM (DRAM)
  - Each cell stores bit with a capacitor and transistor
  - Value must be refreshed every 10-100 ms
  - Sensitive to disturbances
  - ***Slower*** and ***Cheaper*** than SRAM
  - ***Reduced*** power consumption & larger ***Storage*** capacity
  - Used to implement the ***Main*** memory.

Main Memory-1[Read Only Memory]

- Key features
  - Portion of main memory is made of ROM chips.
  - ROM is packaged as a chip
  - Basic storage unit is a cell (one bit per cell)
  - Multiple ROM chips form a memory
  - ROM is also random access type.
  - ROM stores the permanent programs and tables of constants which do not required to be changed.
- Bootstrap Loader:
  - It is a initial program for starting the computer software operating after the power-switch is turned-on.
  - Due to noN-volatility the Bootstrap loader can be placed in ROM.

Main Memory-1[Read Only Memory]

- Programmable read only memory (PROM)
  - Re-programmed by using a special device called a PROM programmer.
  - Generally, a PROM can only be changed/updated once.
- Erasable Programmable read only memory (EPROM)
  - Erased by ultraviolet light
  - Reprogrammed by an RPROM programmer.
  - Erasing and programming many times however, the constant erasing and rewriting will eventually render the chip useless.
- Erasable Programmable read only memory (EEPROM)
  - Similar way to Flash memory
  - EEPROMs are used to store a computer system's BIOS, and can be updated without returning the unit to the factory.
  - BIOS updates can be carried out by computer users wishing a BIOS update.

What is BIOS?

Non-Volatile RAM (NVRAM)

- Key Feature: Keeps data when power lost
  - Several types
  - Most important is NAND flash
- NAND flash
  - Reading similar to DRAM (though somewhat slower)
  - Writing packed with restrictions:
    - Can't change existing data
    - Must erase in large blocks (e.g., 64K)
    - Block dies after about 100K erases
  - Writing slower than reading (mostly due to erase cost)
  - Chips often packaged with Flash Translation Layer (FTL)
    - Spreads out writes ("wear leveling")
    - Makes chip appear like disk drive

Locality

- Principle of Locality:
  - Programs tend to reuse data and instructions near those they have used recently, or that were recently referenced themselves
  - Spatial locality: Items with nearby addresses tend to be referenced close together in time
  - Temporal locality: Recently referenced items are likely to be referenced in the near future

Locality Example:

- Data
  - Reference array elements in succession (stride-1 reference pattern): Spatial locality
  - Reference `sum` each iteration: Temporal locality
- Instructions
  - Reference instructions in sequence: Spatial locality
  - Cycle through loop repeatedly: Temporal locality

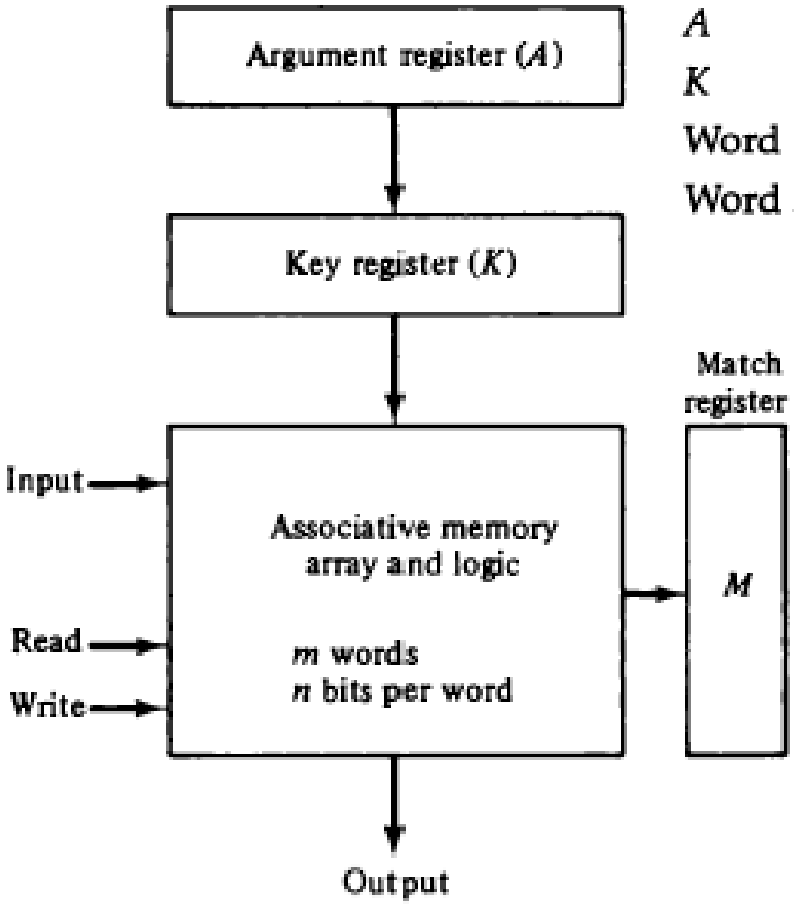
Associative memory

- Memory access time is a crucial parameter with the performance point of view.
- The memory access time can be reduced considerably by,
  - Taking very fast memory devices
  - Decreasing the number of access to memory
    - Depends on location of content
    - Search algorithm
- Searching time required for an item can be reduced is data is accessed on the basis of content rather than the address of data itself.
- Memory unit addressable by its content is know as the content addressable memory (CAM)
- Entire memory is addressed simultaneously and in parallel format while on the basis of word contents.

Associative memory cont….

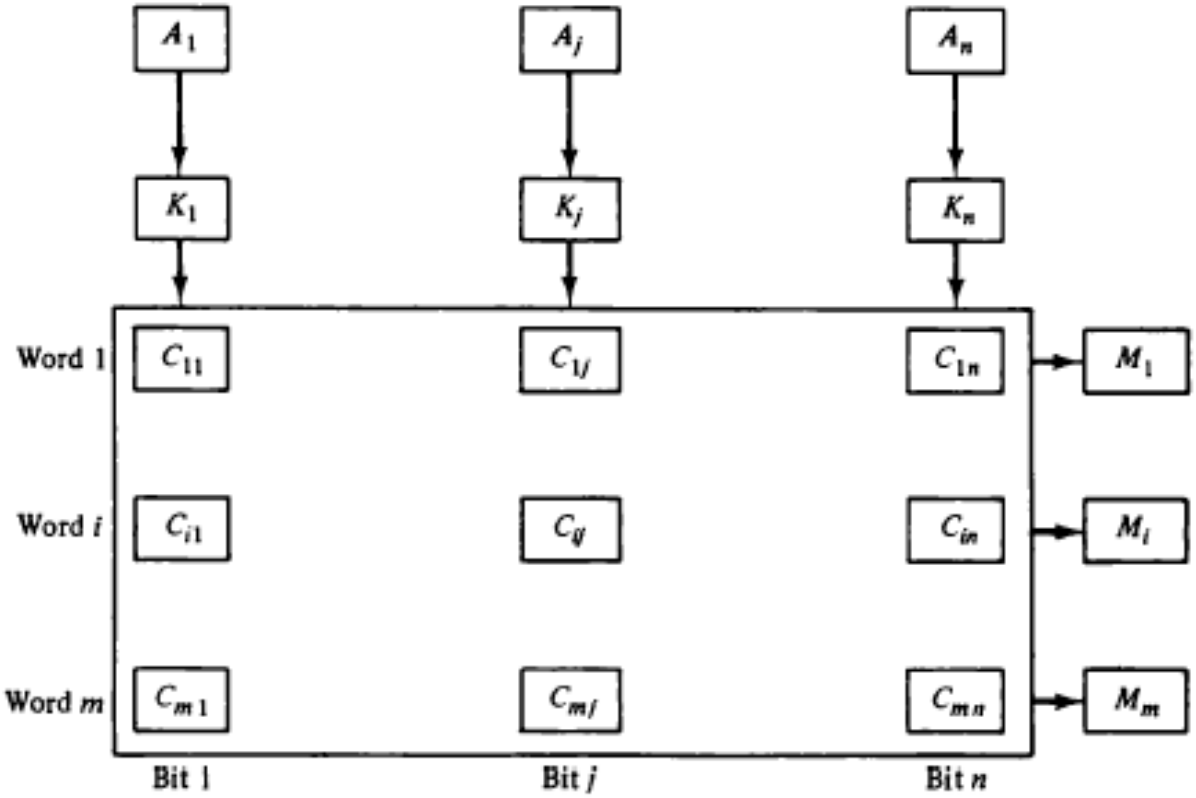
- No address is given while writing in CAM.
- Memory itself is capable of finding the unused / empty location to store a word.
- While reading the word from CAM a part of content of the word is specified,
- Memory locates all such matching words and mark them to be read.
- It is very costly because,
  - Each cell must have storage as well as matching logic capability.
- CAM are used only when the search time is very critical.

Hardware organization for associative memory



A	101 111100	
K	111 000000	
Word 1	100 111100	no match
Word 2	101 000001	match

Associative Memory with m-words and n-cells per word



ASSOCIATIVE MEMORY CONT….

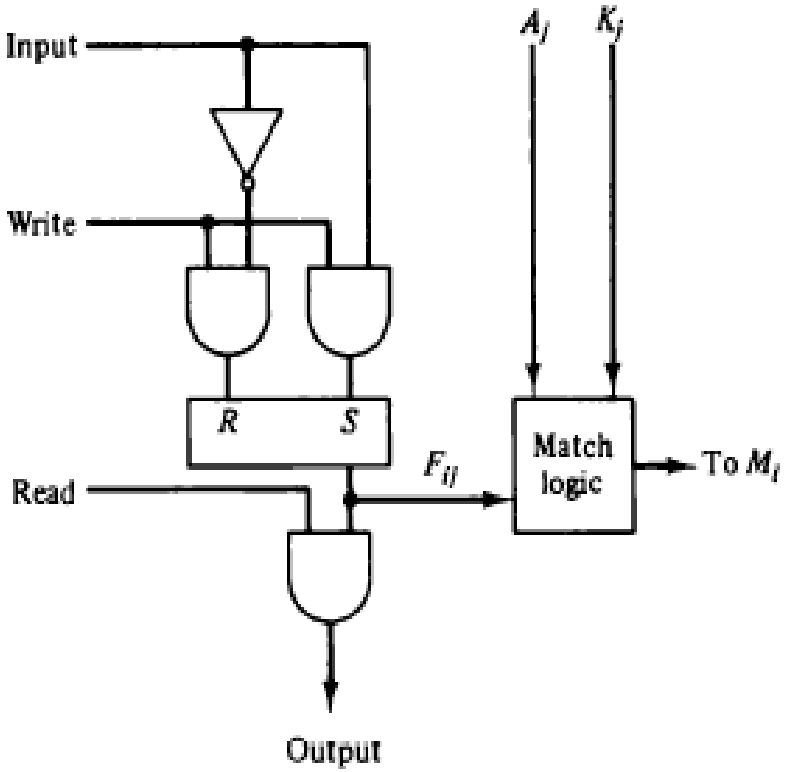
$$x_j = A_j F_{ij} + A_j' F_{ij}'$$

$$M_i = x_1 x_2 x_3 \cdots x_n$$

$$x_j + K_j' = \begin{cases} x_j & \text{if } K_j = 1 \\ 1 & \text{if } K_j = 0 \end{cases}$$

$$M_i = (x_1 + K_1')(x_2 + K_2')(x_3 + K_3') \cdots (x_n + K_n')$$

$$M_i = \prod_{j=1}^{n} (A_j F_{ij} + A_j' F_{ij}' + K_j')$$



Read and write operations of associative memory

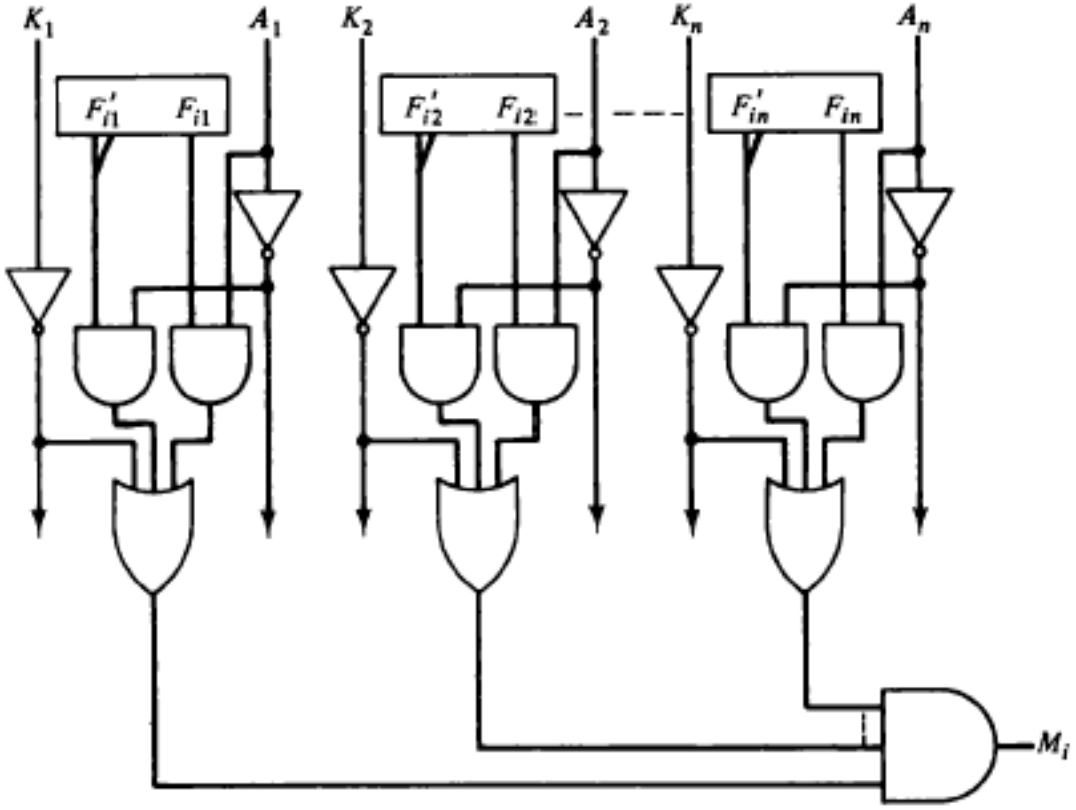
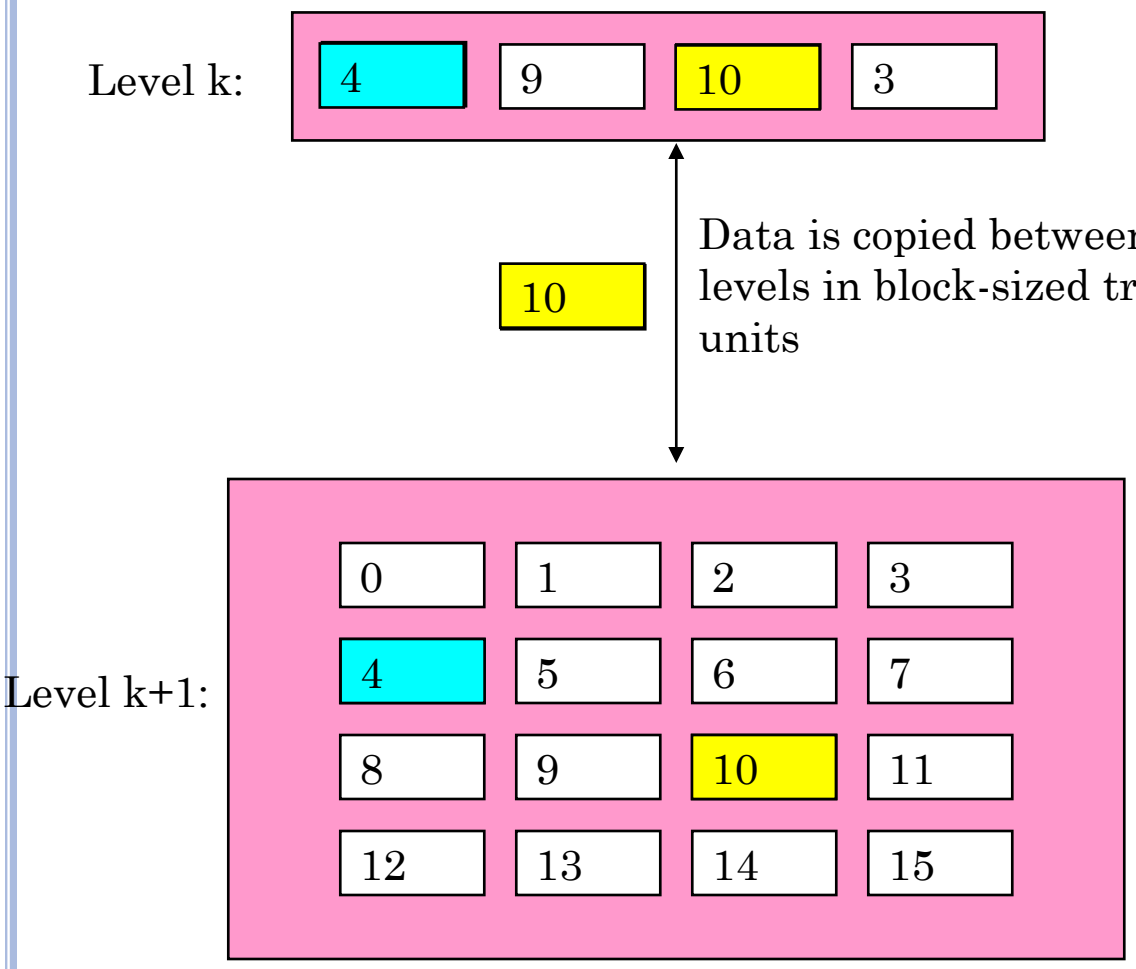


Figure 12-9 Match logic for one word of associative memory.

Caches

- Cache: Smaller, faster storage device that acts as staging area for subset of data in a larger, slower device
- Fundamental idea of a memory hierarchy:
  - For each k, the faster, smaller device at level k serves as cache for larger, slower device at level k+1
- Why do memory hierarchies work?
  - Programs tend to access data at level k more often than they access data at level k+1
  - Thus, storage at level k+1 can be slower, and thus larger and cheaper per bit
  - Net effect: Large pool of memory that costs as little as the cheap storage near the bottom, but that serves data to programs at ≈ rate of the fast storage near the top.

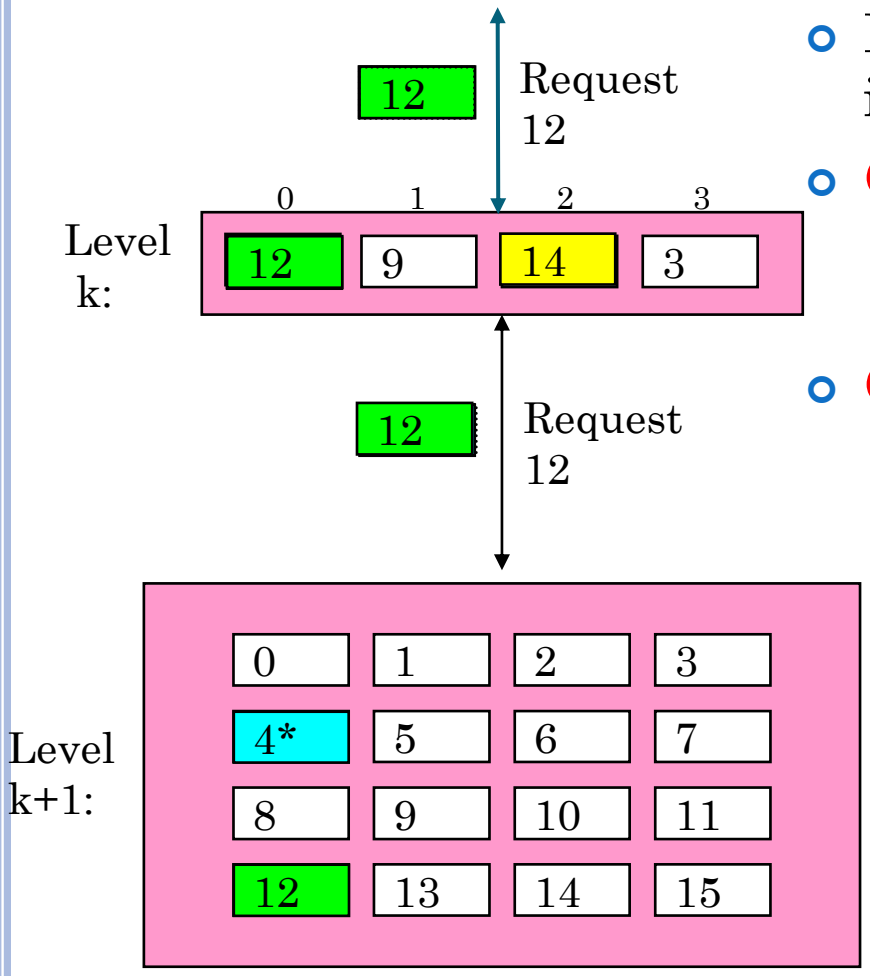
Caching in a Memory Hierarchy



Smaller, faster, more expensive device at level k caches a subset of the blocks from level k+1

Larger, slower, cheaper storage device at level k+1 is partitioned into blocks.

General Caching Concepts



- Program needs object d, which is stored in some block b
- Cache hit
  - Program finds b in the cache at level k. E.g., block 14
- Cache miss
  - b is not at level k, so level k cache must fetch it from level k+1. E.g., block 12
  - If level k cache is full, then some current block must be replaced (evicted). Which one is the "victim"?
    - Placement policy: where can the new block go? E.g., b mod 4
    - Replacement policy: which block should be evicted? E.g., LRU

General Caching Concepts

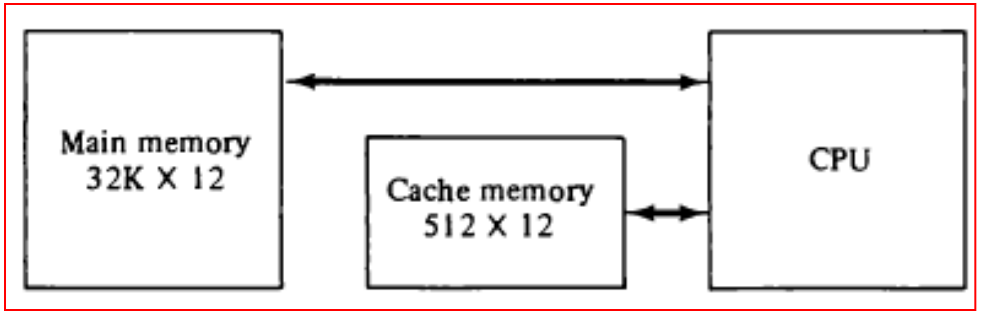
- Types of cache misses:
  - Cold (compulsory) miss
    - Cold misses occur because the cache is empty
  - Conflict miss
    - Most caches limit blocks at level k to a small subset (sometimes a singleton) of the block positions at level k+1
    - E.g. block i at level k+1 must be placed in block (i mod 4) at level k
    - Conflict misses occur when the level k cache is large enough, but multiple data objects all map to the same level k block
    - E.g. Referencing blocks 0, 8, 0, 8, 0, 8, ... would miss every time
  - Capacity miss
    - Occurs when the set of active cache blocks (working set) is larger than the cache

Cache Cont.....

The average memory access time of a computer system can be improved considerably by use of a cache. If the hit ratio is high enough so that most of the time the CPU accesses the cache instead of main memory, the average access time is closer to the access time of the fast cache memory. For example, a computer with cache access time of 100 ns, a main memory access time of 1000 ns, and a hit ratio of 0.9 produces an average access time of 200 ns. This is a considerable improvement over a similar computer without a cache memory, whose access time is 1000 ns.

Cache Cont.....

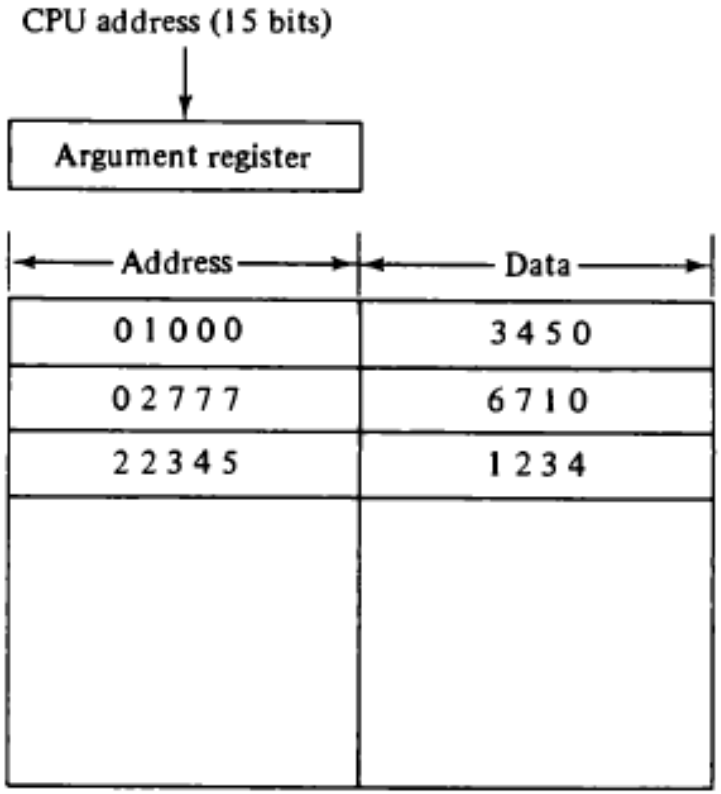
- Cache memory is very fast hence the cache search time must be the least possible.
- Memory mapping ➔ Transformation of data from main memory and cache.
- Mapping methods
  - Associative
  - Direct
  - Set-associative



Cache Cont…..(Associative mapping)

- This provides the fastest and most flexible memory organization.
- Address along with the data is stored in cache.
- Any memory word can be stored in any cache location.
- Cache is searched for the address through 15-bit CPU address register and
  - if found then corresponding 12-bit data is transferred to CPU.
  - If not found in cache then address data pair is transferred into cache form main memory.
  - If no room for the address data pair is free then first it is made some room for the same, following some algorithm.
    - Round robin order /FIFO policy.

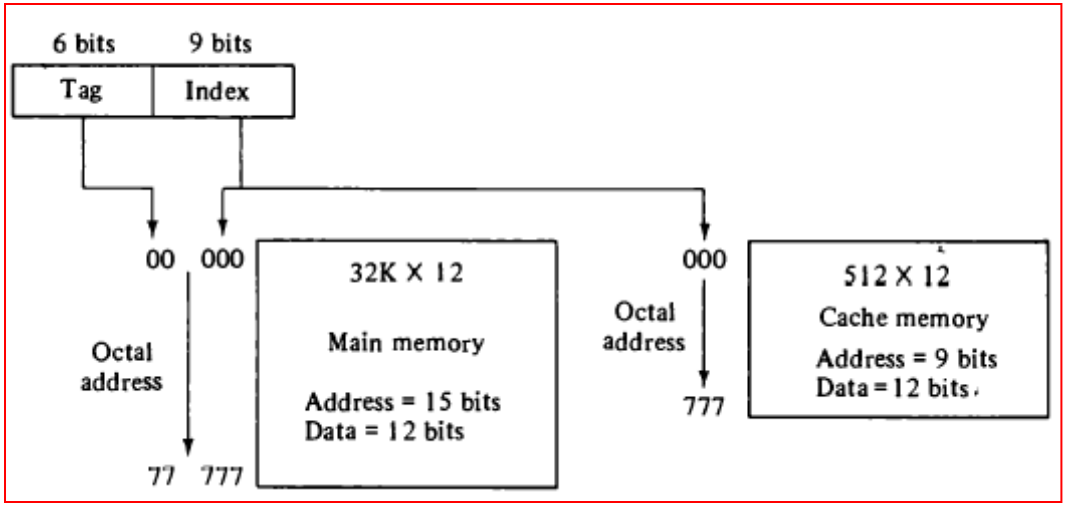
CACHE CONT…..(ASSOCIATIVE MAPPING)



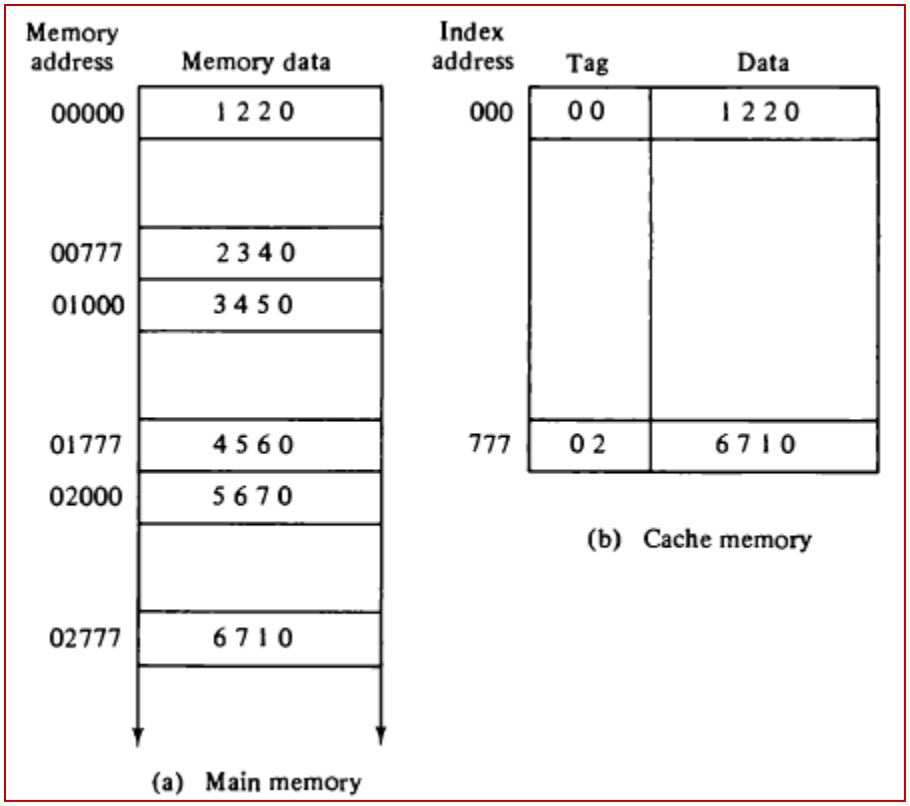
Associative memory (mapping) is costly because of the requirement of match logic for each memory/cache cell.

CACHE CONT…..(DIRECT MAPPING)

- It uses the random access memory rather than using the associative memory.
- Each word stored in the cache is basically the data word and associated tag.
- CPU generates the address field (15-bits) then index field will be used to access the cache and the tag filed will be matched, if it matches then the associated data is transferred to CPU otherwise data from main memory to cache is transferred.



CACHE CONT…..(DIRECT MAPPING)

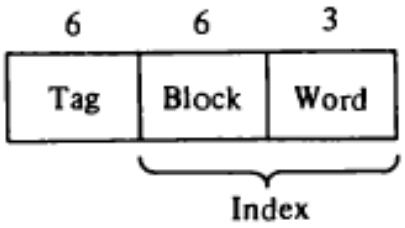


The main problem with this mapping is that the hit ration may decrease drastically if the data having same index but different tags are addressed repeatedly.

Direct mapping with block size of 8

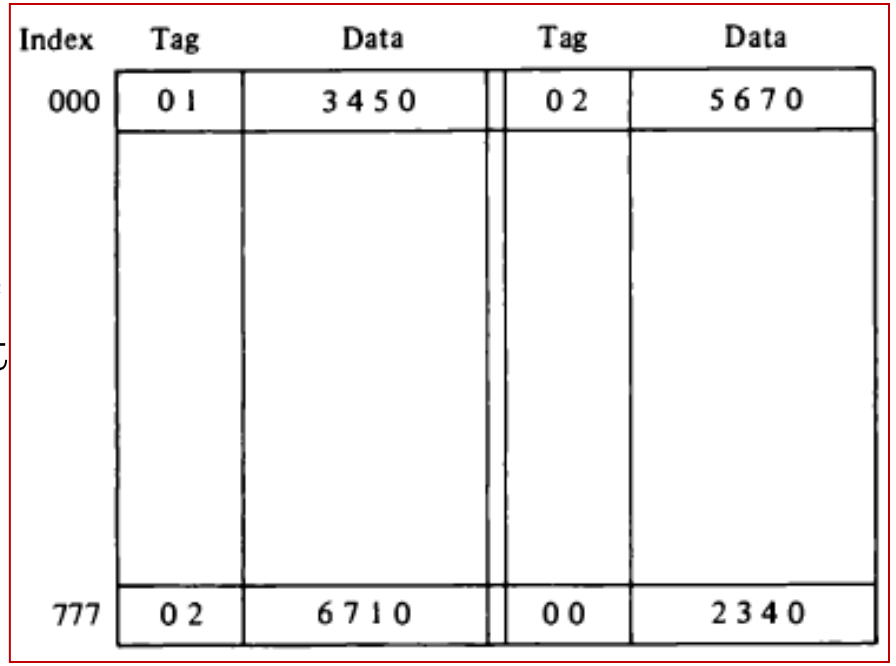
- Hit ratio is considerably improved using the block rather than taking a word only.

	Index	Tag	Data
Block 0	000	01	3450
	007	01	6578
Block 1	010		
	017		
Block 63	770	02	
	777	02	6710



Cache Cont…(Set-associative mapping)

- Associative mapping is very costly.
- Direct mapping can not store two words with the same index and different tag.
- Set-associative mapping ➔ cache can store more than one word having same index and varying tag values.
- ➔ Set of two, three or four.



Index	Tag	Data	Tag	Data
000	01	3450	02	5670
777	02	6710	00	2340

Cache data replacement writing

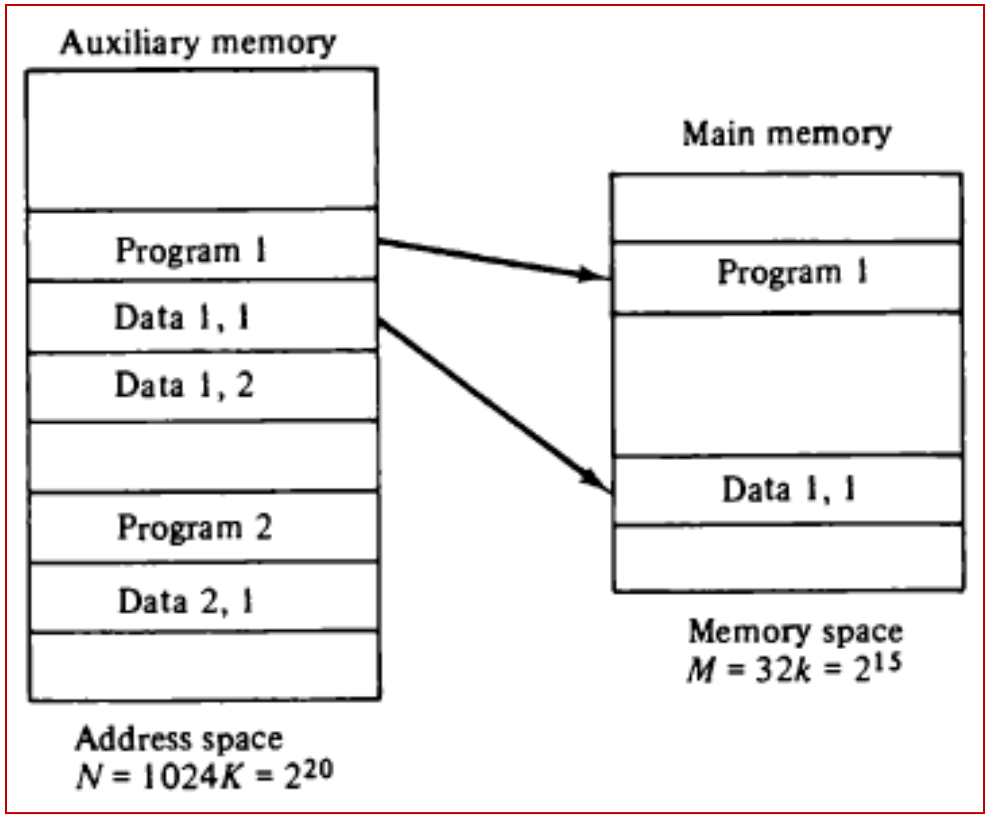
- Data replacement:
  - While miss occurs ➔ some data from cache is to be displaced from cache to main memory, if it is not free space to accommodate the word which is not present in cache.
    - Random replacement
    - Round robin (FIFO)
    - Least recently used (LRU)
- Data writing:
  - Write-through
    - Cache as well as main memory are updated with every write operation in parallel (parallel update procedure)
  - Write-back
    - Data is only updated into cache and marked if updated, and when required to be removed from cache, it is written in main memory.

VIRTUAL MEMORY

- Size of main memory is usually very small as compared to auxiliary memory.
- The VIRTUAL MEMORY is a concept that provides a flexibility of using the auxiliary memory as main memory.
- Each CPU referenced address goes through the address mapping form virtual memory to main memory.
- Address mapping requires some kind of table information, known as address mapping table.
- Address Space: (Generated by programmer)
  - Set of virtual addresses ← Auxiliary Memory
- Memory Space: (Actual address used for processing)
  - Set of physical address ← Main Memory

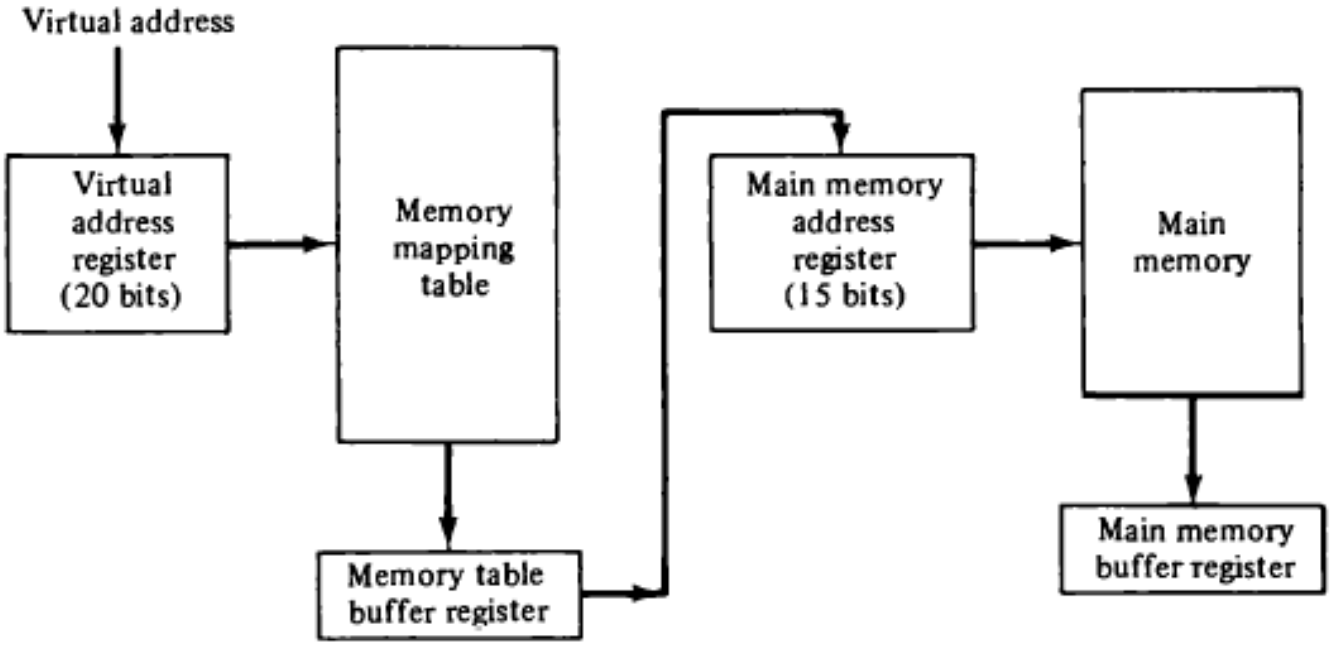
VIRTUAL MEMORY CONT….

In virtual memory systems the programmers having the total address space at their disposal.

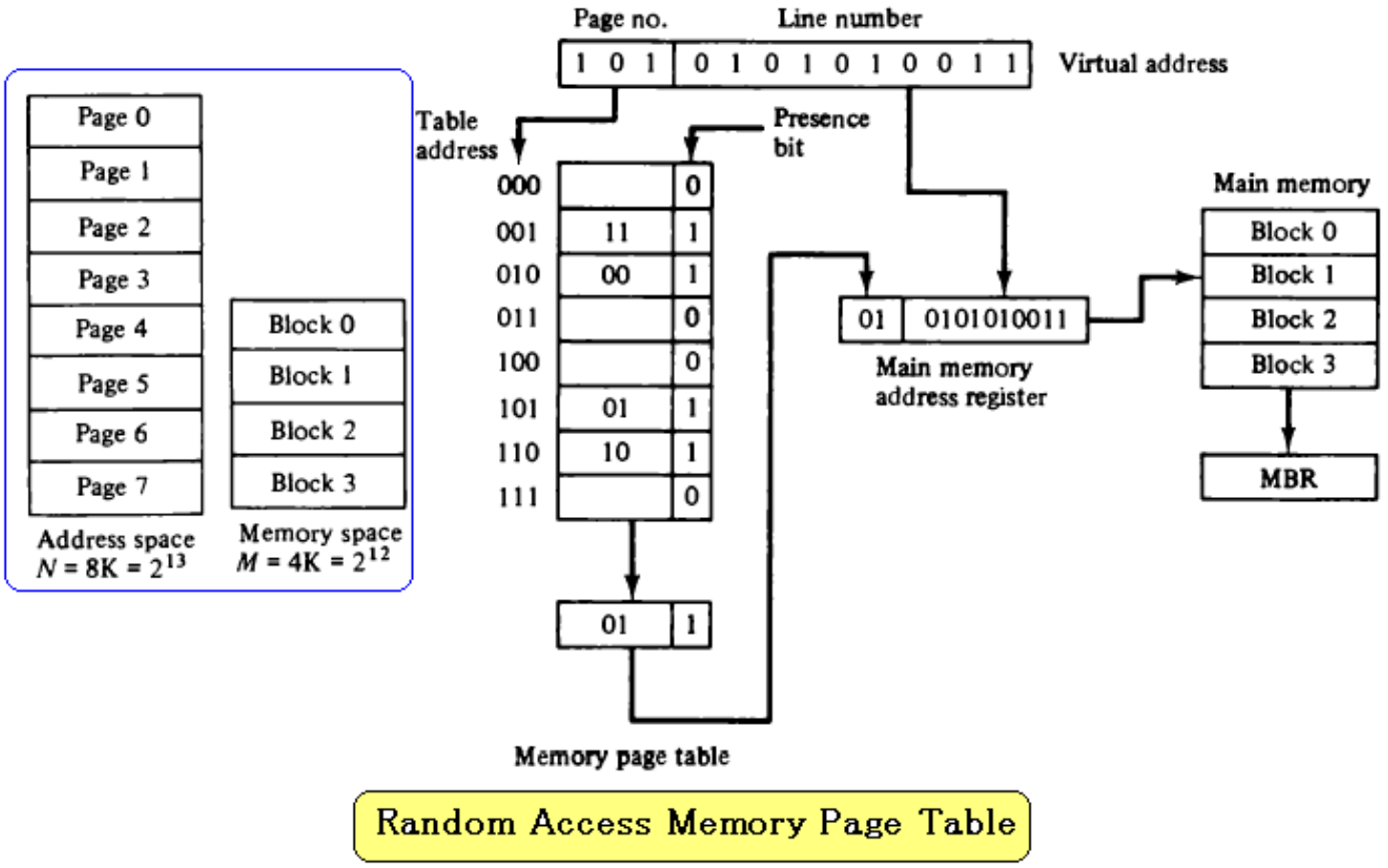


MEMORY MAPPING TABLE

- 20 bit address of address space to be mapped with the 15 bit address of the memory space ➔ Memory mapping table requirement.
- Dynamic mapping process.



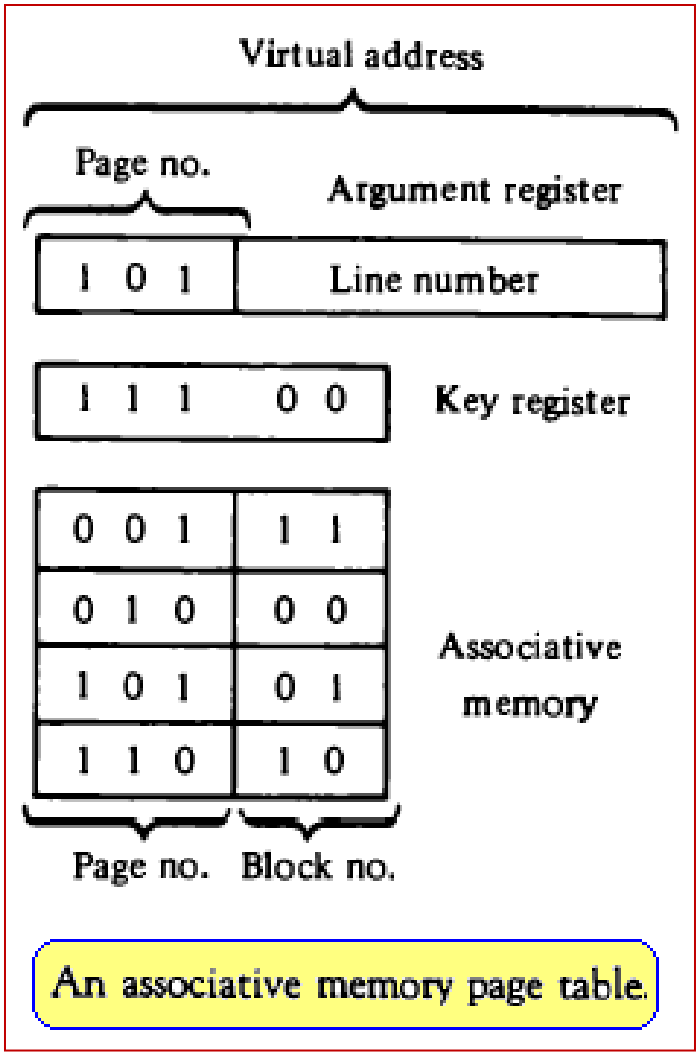
Address mapping via concept of pages



Random Access Memory Page Table

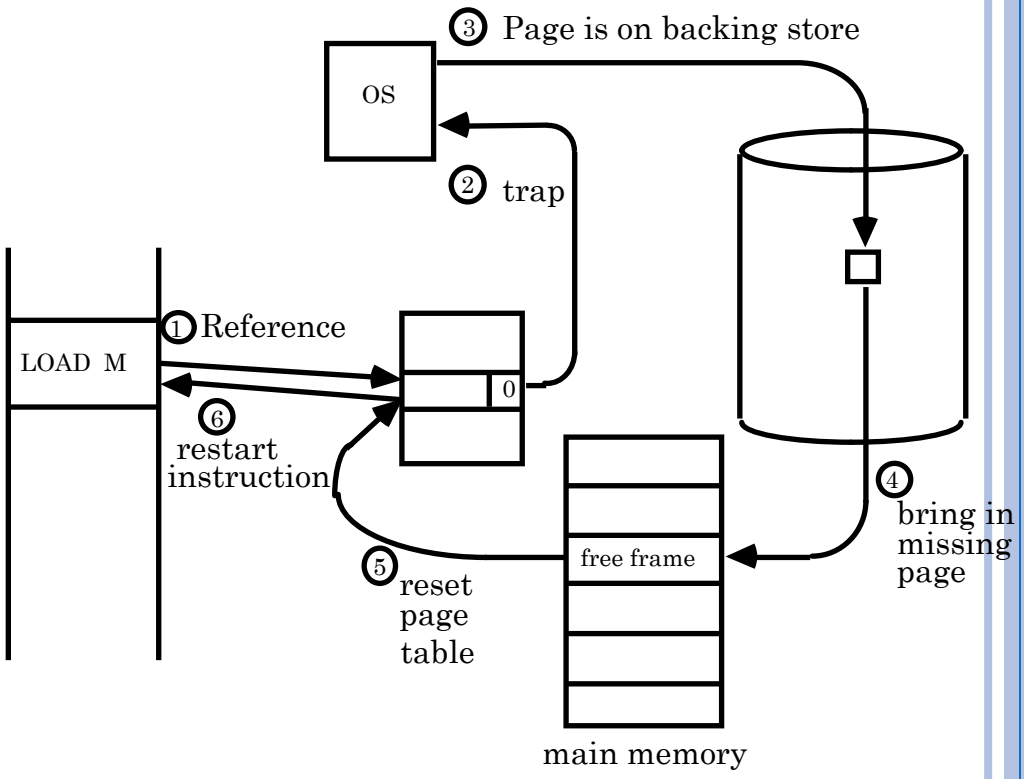
ADDRESS MAPPING VIA CONCEPT OF PAGES

- Random access memory page table is inefficient due to under utilization of storage capacity.
- It may be used number of words equal to number of blocks in main memory.
  - Size reduces and full utilization of memory.
- Achieved by using the associative memory containing each word a page number and associated block number.



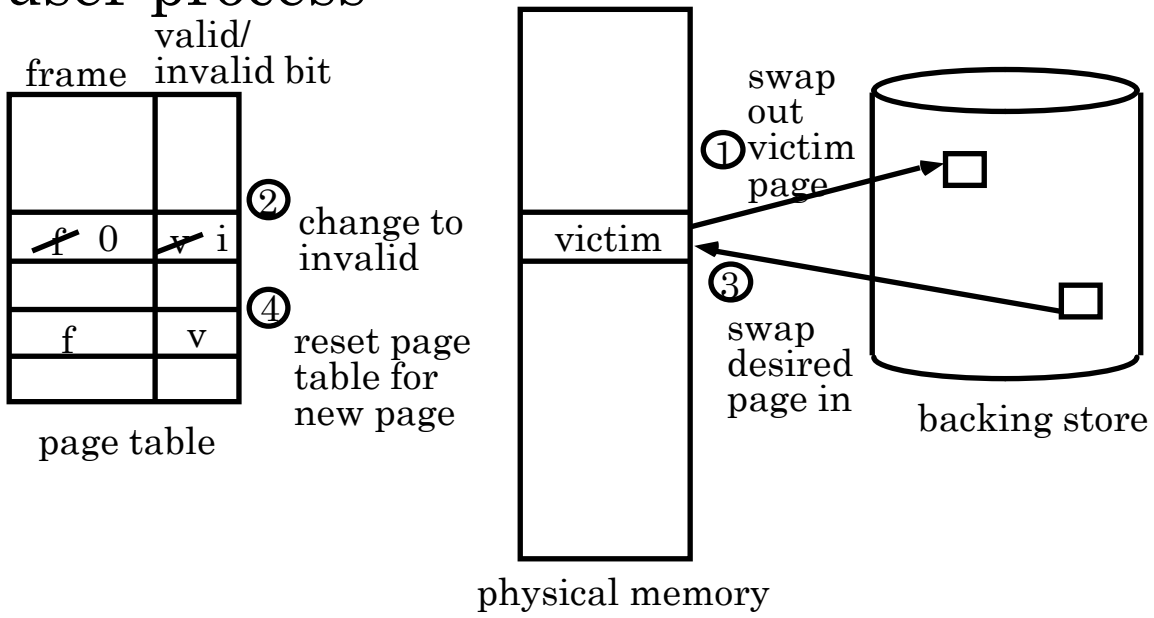
PAGE FAULT

1. CPU reference any page, in memory page table the presence bit is invalid.
2. Trap to the Operating system.
3. OS goes to the backing storage device where page is stored.
4. OS bring the page in main empty memory frame (block) , if any frame is not empty then replace any frame (by using replacement algo.)
5. Reset the page table.
6. CPU restart the execution.



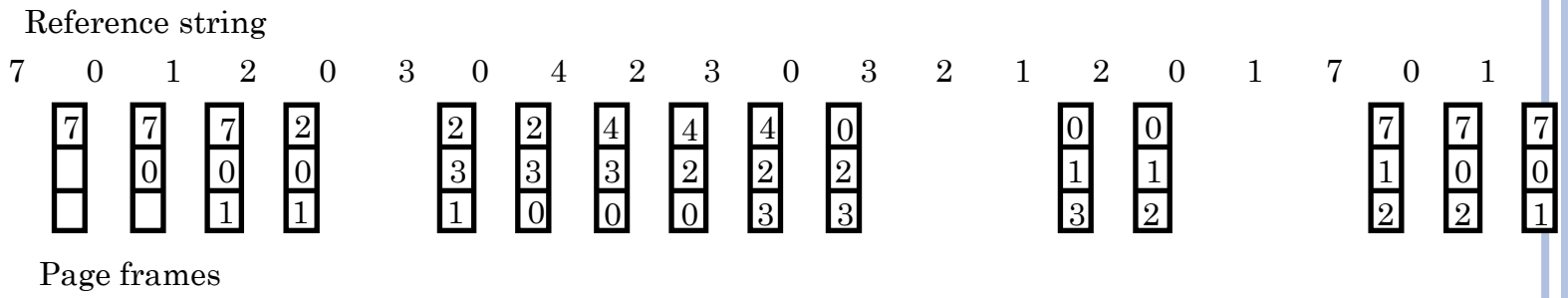
PAGE REPLACEMENT

1. Find the location of the desired page on the backing store
2. Find a free frame
   - If there is a free frame, use it
   - Otherwise, use a page-replacement algorithm to select a *victim* frame
   - Write the victim page to the backing store
3. Read the desired page into the (newly) free frame
4. Restart the user process



PAGE REPLACEMENT ALGORITHMS

FIFO

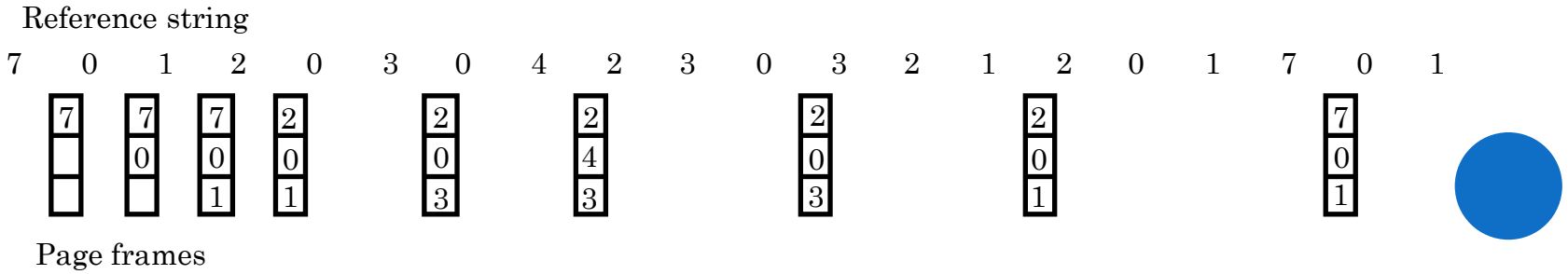


FIFO algorithm selects the page that has been in memory the longest time
Using a queue - every time a page is loaded, its
- identification is inserted in the queue

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

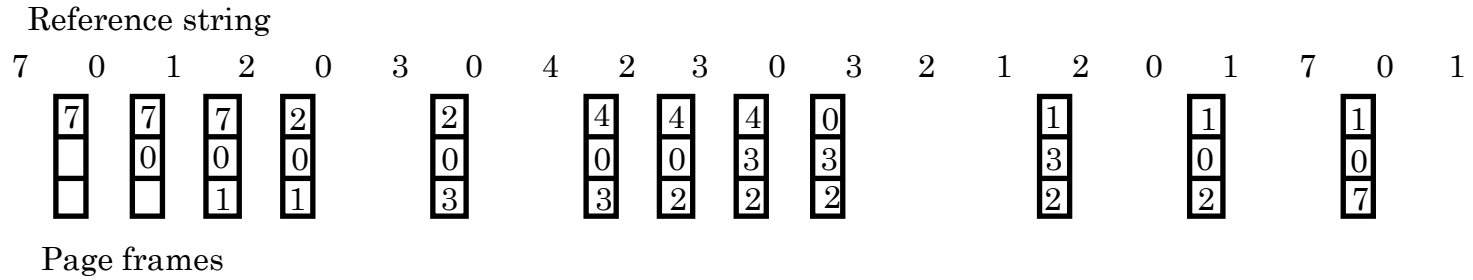


PAGE REPLACEMENT ALGORITHMS

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



MRU (Most Recently used): Replace the page which is used most Recently.

Thank you