UNIT IV MEMORY SYSTEM

8+3

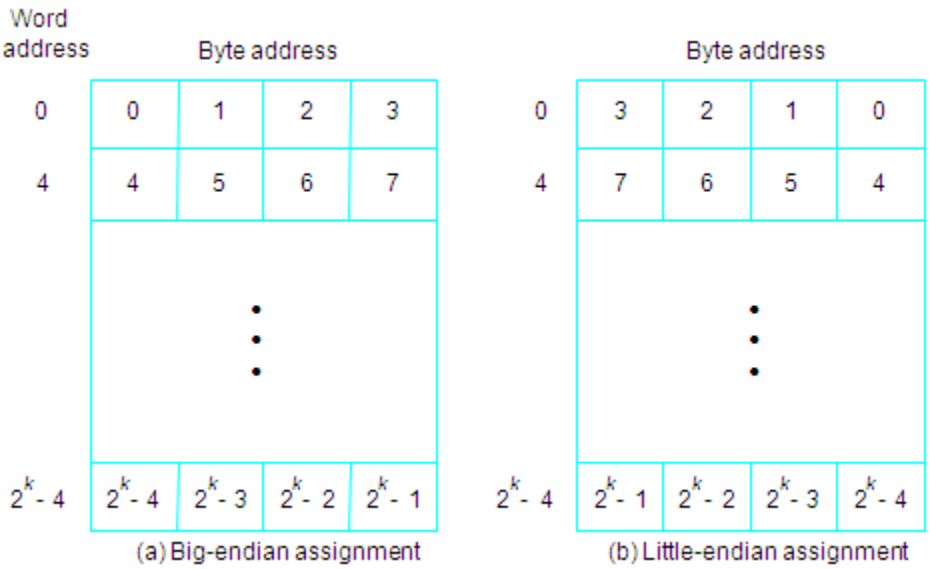
Basic concepts – Semiconductor RAMs - ROMs – Speed - size and cost – Cache memories - Performance consideration – Virtual memory- Memory Management requirements – Associative memories – Secondary storage – Case Study: Multi core processor and its memory.

Basic Concepts

- The maximum size of the memory that can be used in any computer is determined by the addressing scheme.

  16-bit addresses = 2^{16} = 64K memory locations

- Most modern computers are byte addressable.



Traditional Architecture

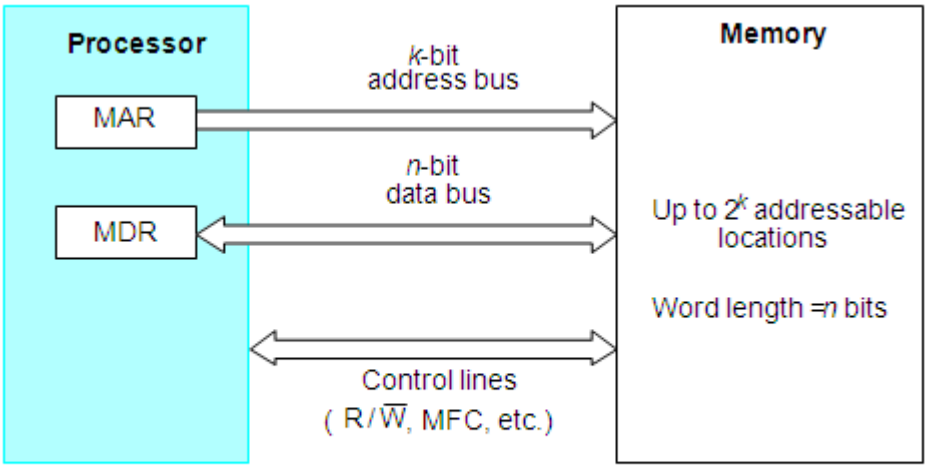


Figure 5.1. Connection of the memory to the processor.

- Block transfer" – bulk data transfer
- *Memory access time*
- *Memory cycle time*
- RAM – any location can be accessed for a Read or Write operation in some fixed amount of time that is independent of the location's address.
- Cache memory
- Virtual memory, memory management unit

Semiconductor RAM Memories

Internal Organization of Memory Chips

16 words of 8 bits each: 16x8 memory org.. It has 16 external connections: addr. 4, data 8, control: 2, power/ground: 2

1K memory cells: 128x8 memory, external connections: ? 19(7+8+2+2)

1Kx1:? 15 (10+1+2+2)

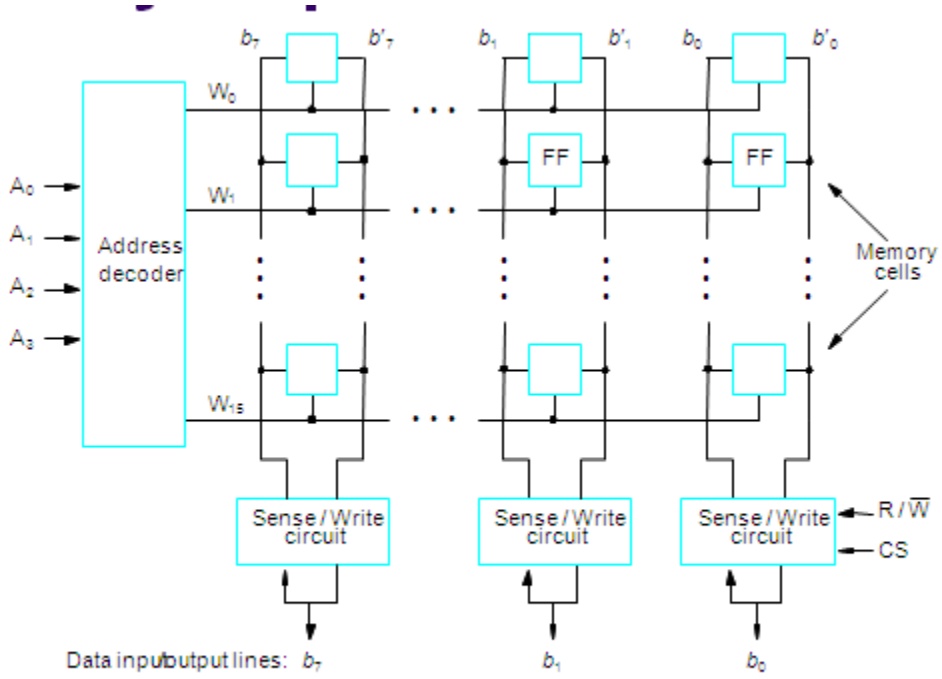


Figure 5.2. Organization of bit cells in a memory chip.

A Memory Chip

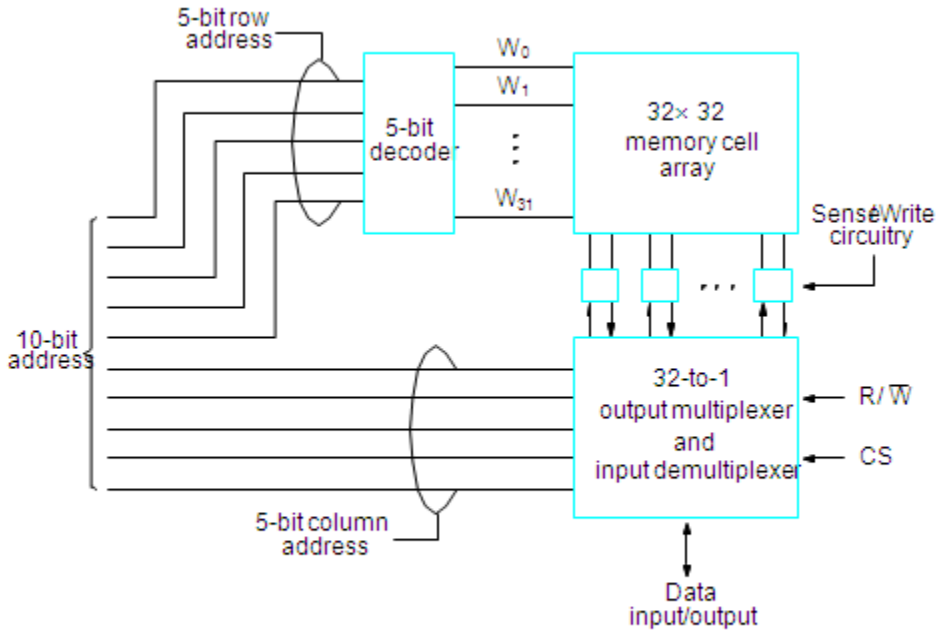


Figure 5.3. Organization of a 1K × 1 memory chip.

Static Memories

- The circuits are capable of retaining their state as long as power is applied.

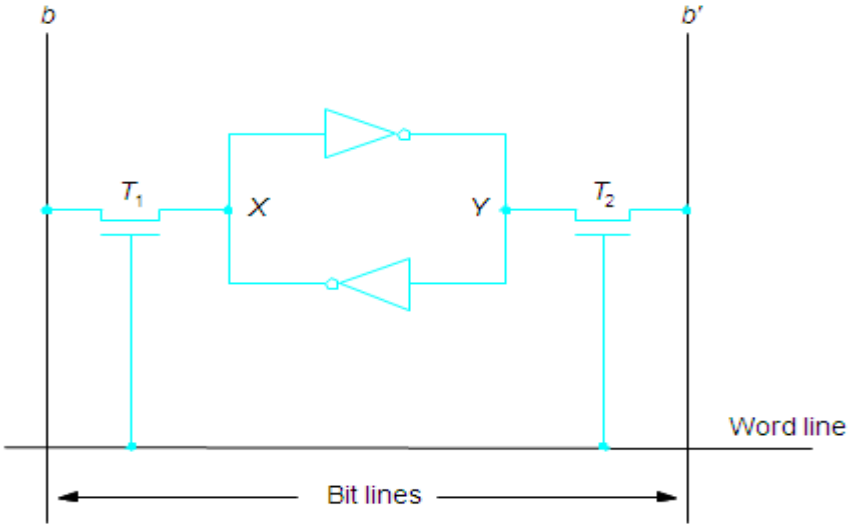


Figure 5.4. A static RAM cell.

- CMOS cell: low power consumption

Asynchronous DRAMs

- Static RAMs are fast, but they cost more area and are more expensive.
- Dynamic RAMs (DRAMs) are cheap and area efficient, but they can not retain their state indefinitely – need to be periodically refreshed.

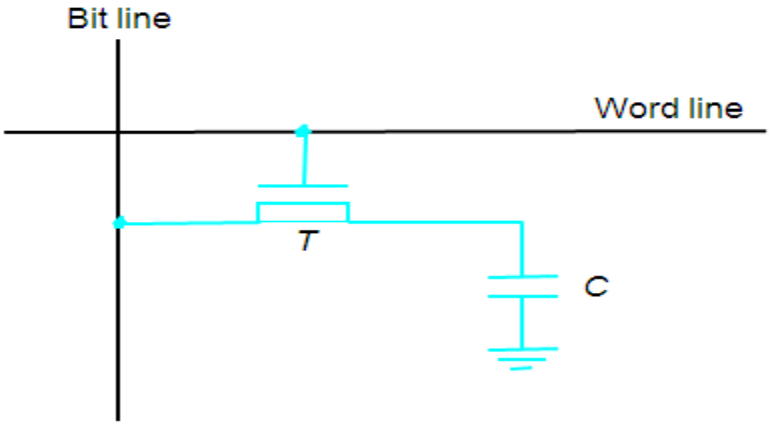


Figure 5.6. A single-transistor dynamic memory cell

A Dynamic Memory Chip

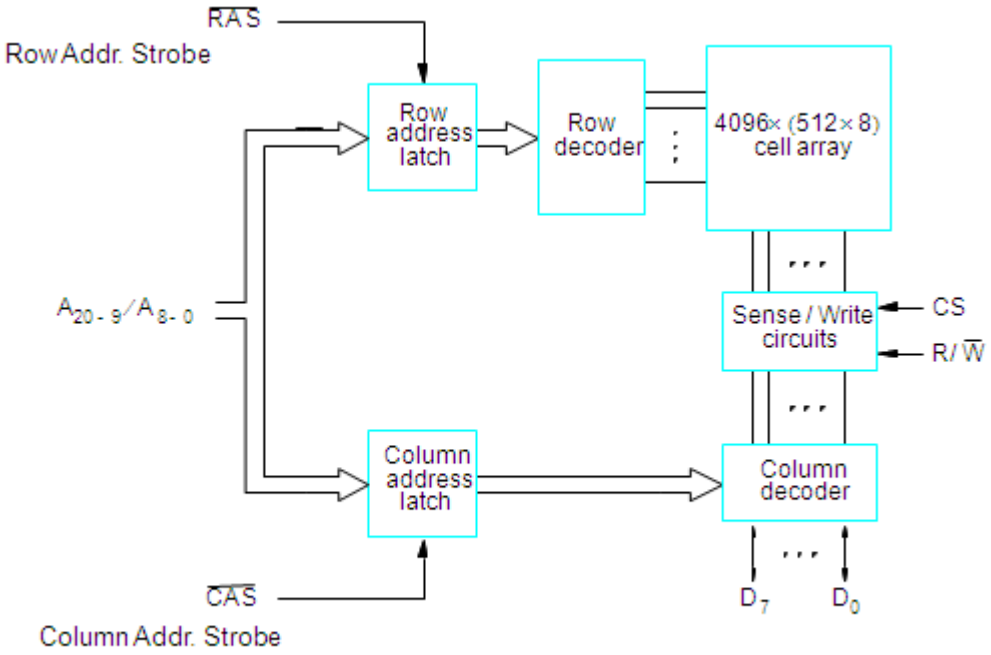


Figure 5.7. Internal organization of a 2M × 8 dynamic memory chip.

Synchronous DRAMs

- The operations of SDRAM are controlled by a clock signal.

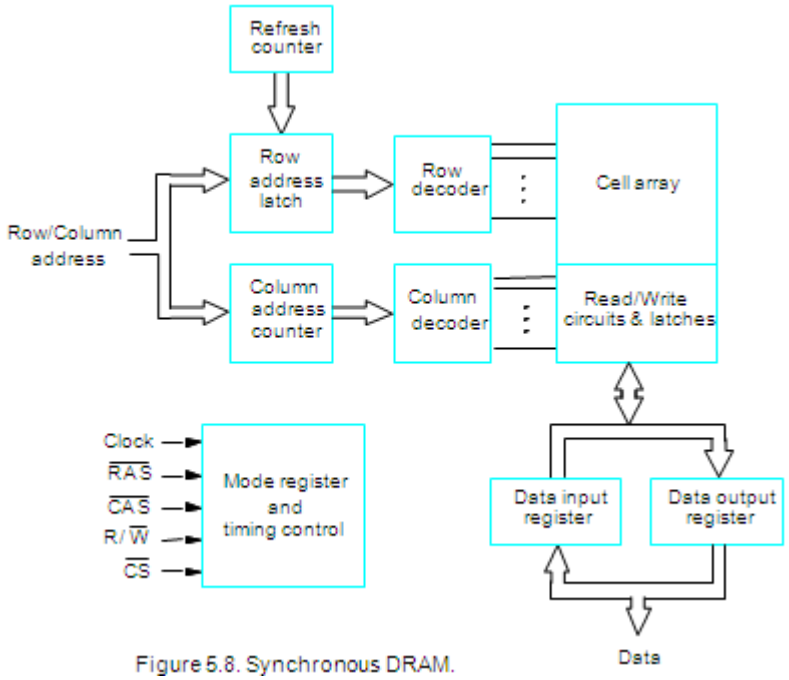


Figure 5.8. Synchronous DRAM.

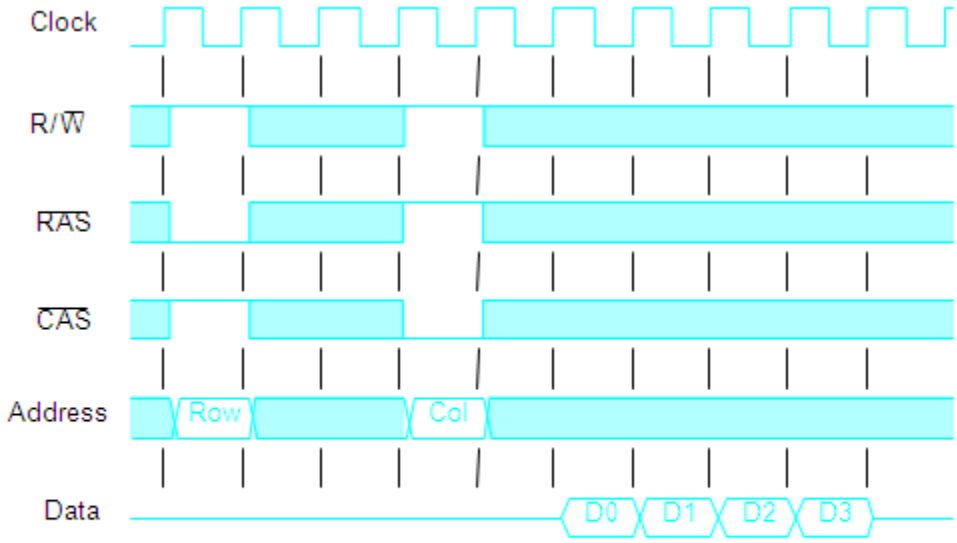


Figure 5.9. Burst read of length 4 in an SDRAM.

- Refresh circuits are included (every 64ms).
- Clock frequency > 100 MHz

Intel PC100 and PC133

Latency and Bandwidth

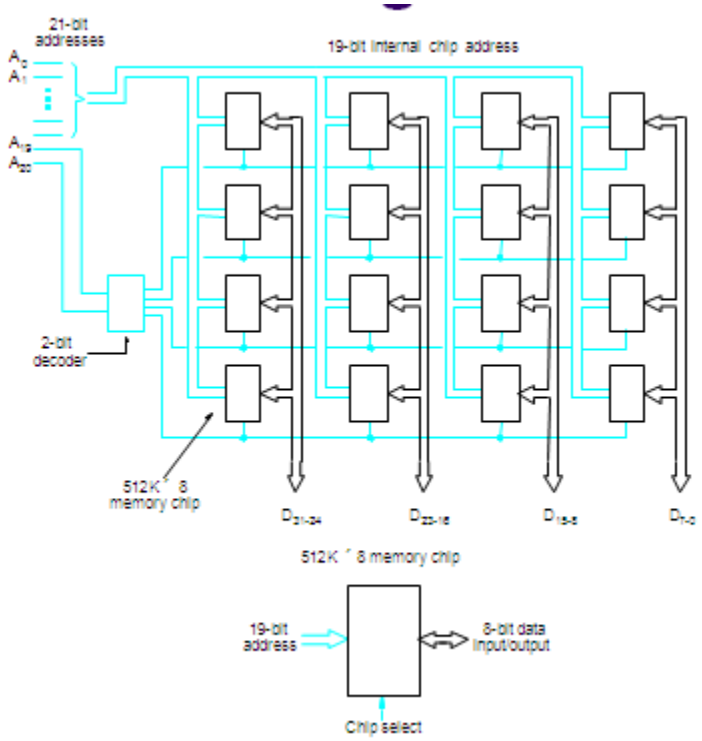
- The speed and efficiency of data transfers among memory, processor, and disk have a large impact on the performance of a computer system.
- Memory latency – the amount of time it takes to transfer a word of data to or from the memory.
- Memory bandwidth – the number of bits or bytes that can be transferred in one second. It is used to measure how much time is needed to transfer an entire block of data.
- Bandwidth is not determined solely by memory. It is the product of the rate at which data are transferred (and accessed) and the width of the data bus.

DDR SDRAM

- Double-Data-Rate SDRAM
- Standard SDRAM performs all actions on the rising edge of the clock signal.
- DDR SDRAM accesses the cell array in the same way, but transfers the data on both edges of the clock.
- The cell array is organized in two banks. Each can be accessed separately.

- DDR SDRAMs and standard SDRAMs are most efficiently used in applications where block transfers are prevalent.

Structures of Larger Memories



Memory System Considerations

- The choice of a RAM chip for a given application depends on several factors:

  Cost, speed, power, size…

- SRAMs are faster, more expensive, smaller.
- DRAMs are slower, cheaper, larger.
- Which one for cache and main memory, respectively?
- Refresh overhead – suppose a SDRAM whose cells are in 8K rows; 4 clock cycles are needed to access each row; then it takes 8192×4=32,768 cycles to refresh all rows; if the clock rate is 133 MHz, then it takes $32{,}768/(133\times10^{-6})=246\times10^{-6}$ seconds; suppose the typical refreshing period is 64 ms, then the refresh overhead is 0.246/64=0.0038<0.4% of the total time available for accessing the memory.

Memory Controller

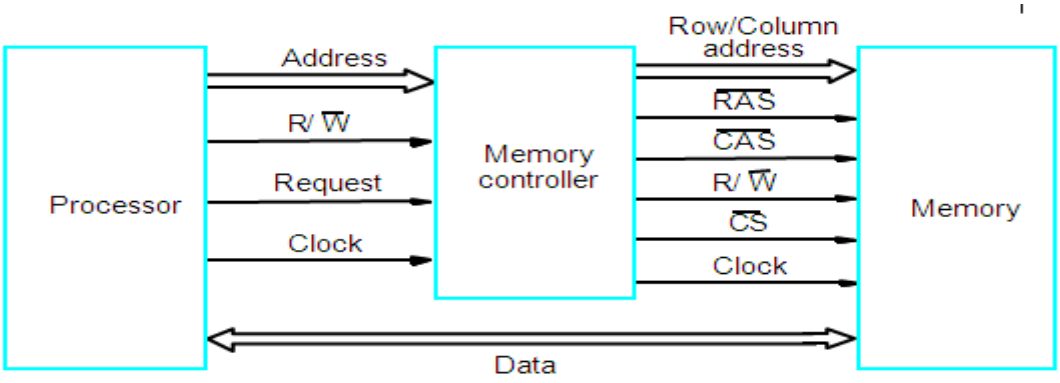


Figure 5.11. Use of a memory controller.

Read-Only Memories

- Volatile / non-volatile memory
- ROM
- PROM: programmable ROM

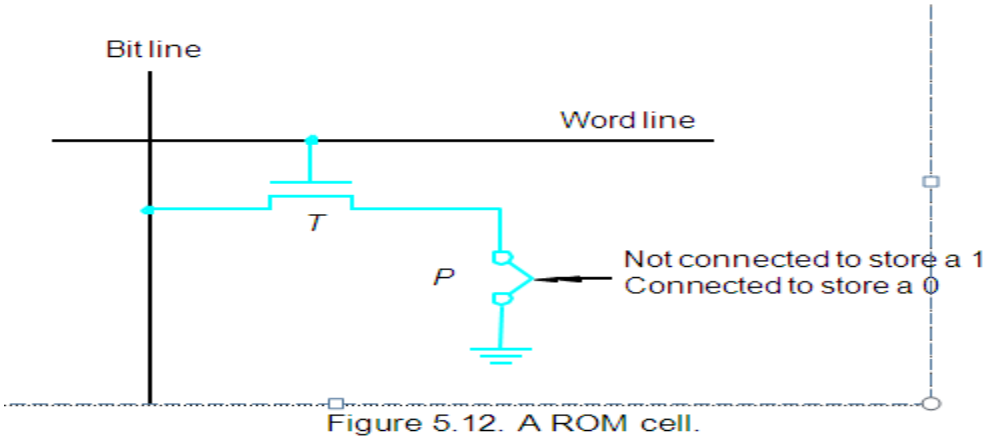


Figure 5.12. A ROM cell.

Flash Memory

- Similar to EEPROM
- Difference: only possible to write an entire block of cells instead of a single cell
- Low power
- Use in portable equipment
- Implementation of such modules
    - Flash cards
    - Flash drives

Speed, Size, and Cost

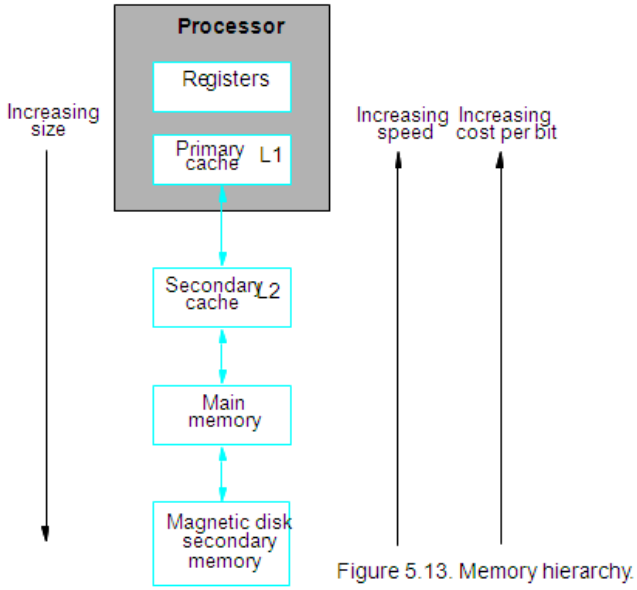


Figure 5.13. Memory hierarchy.

Cache Memories

- What is cache?
- Why we need it?
- Locality of reference (very important)

  - temporal

  - spatial

- Cache block – *cache line*
  - *A set of contiguous address locations of some size*
- Replacement algorithm
- Hit / miss
- Write-through / Write-back
- Load through

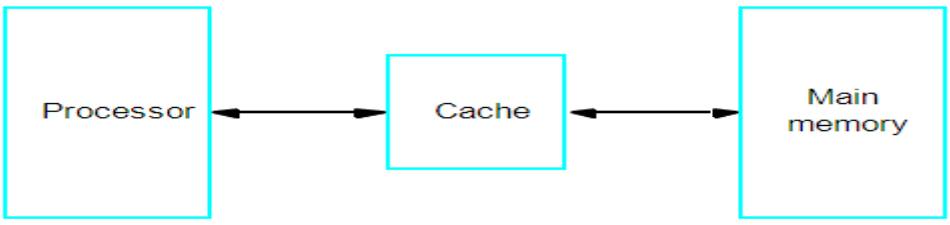
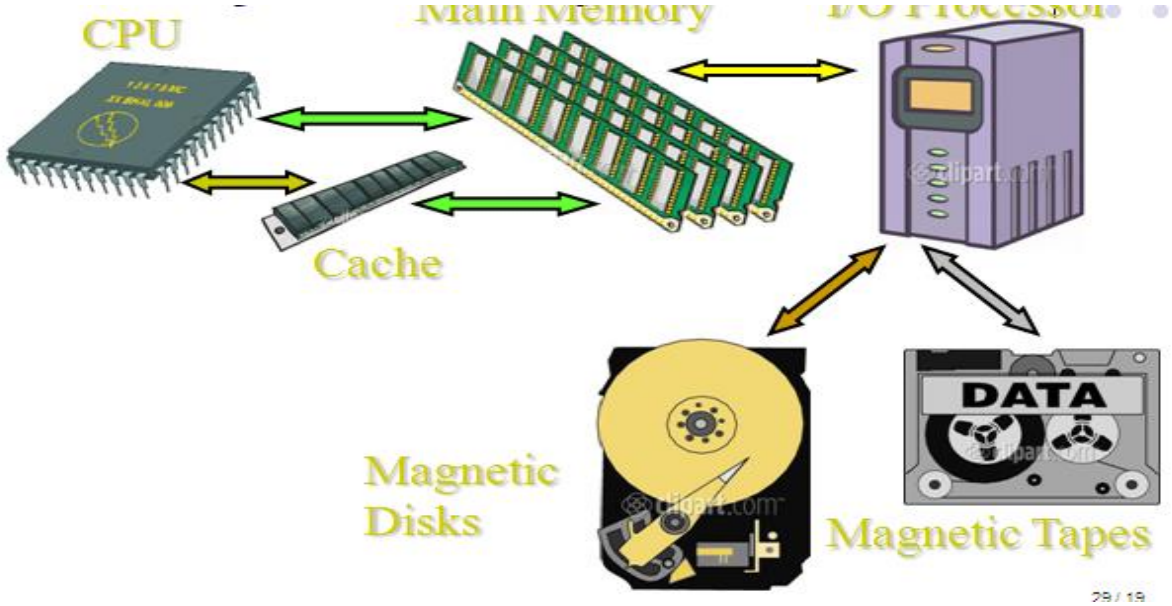
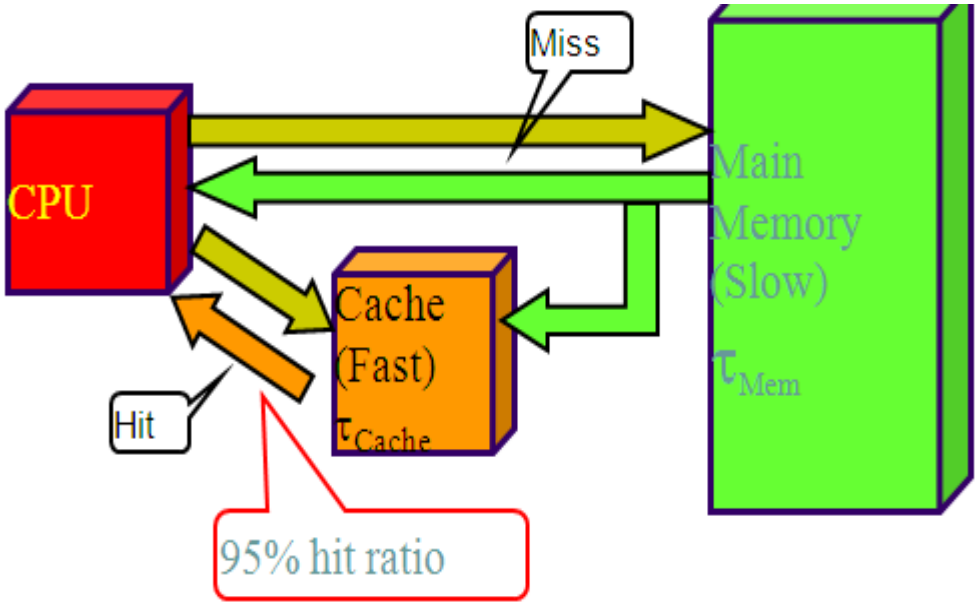


Figure 5.14. Use of a cache memory.

Memory Hierarchy

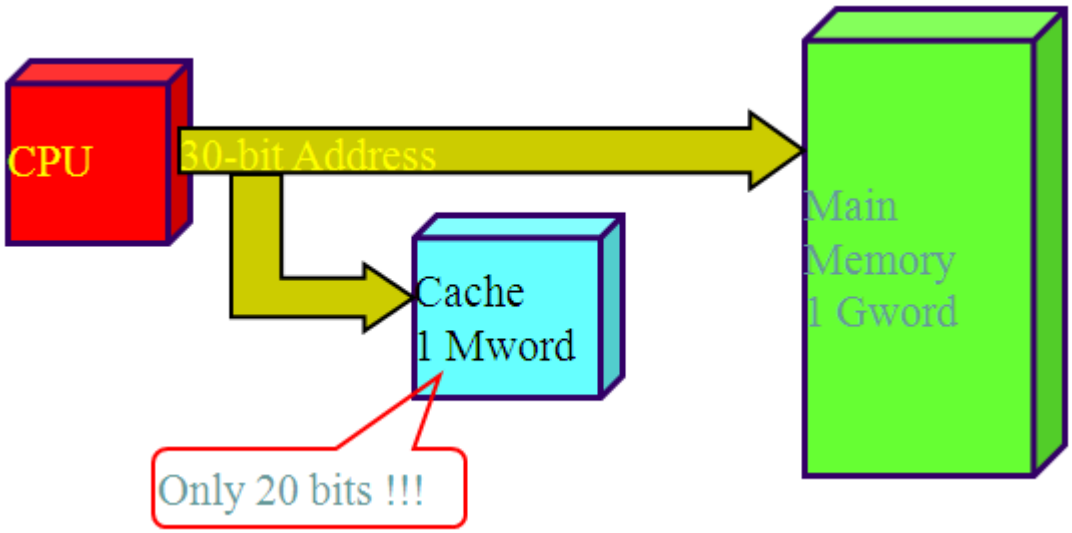


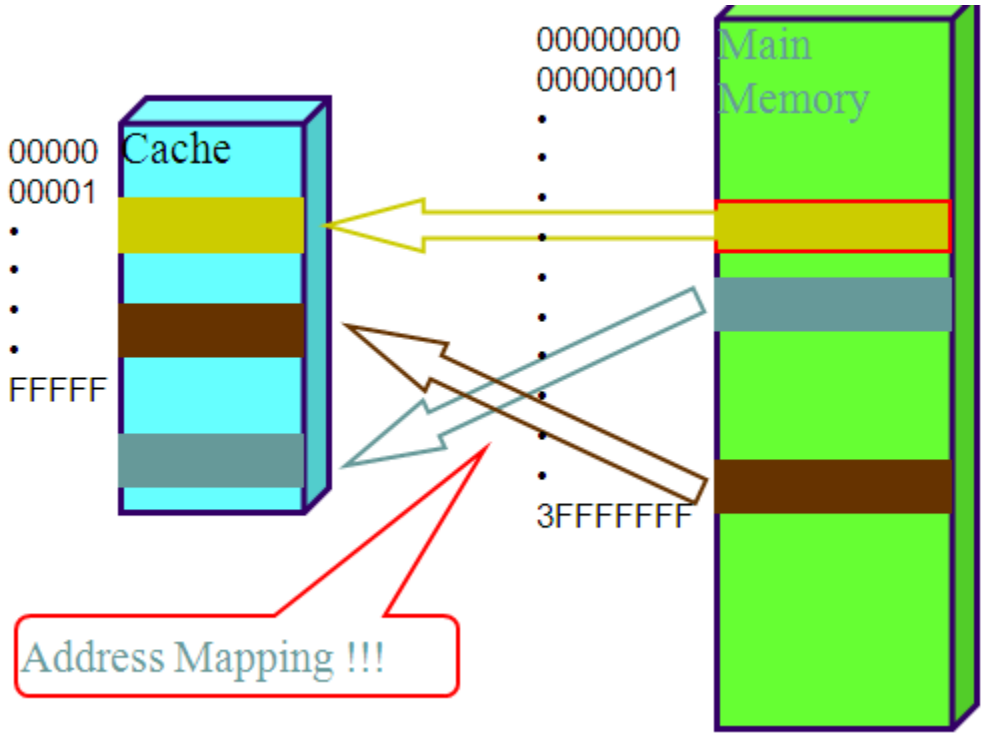
- High speed (towards CPU speed)
- Small size (power & cost)



$$\tau_{Access} = 0.95\ \tau_{Cache} + 0.05\ \tau_{Mem}$$

30 / 19





Direct Mapping

Block j of main memory maps onto block j modulo 128 of the cache

4: one of 16 words. (each block has $16=2^4$ words)

7: points to a particular block in the cache ($128=2^7$)

5: 5 tag bits are compared with the tag bits associated with its location in the cache. Identify which of the 32 blocks that are resident in the cache (4096/128).

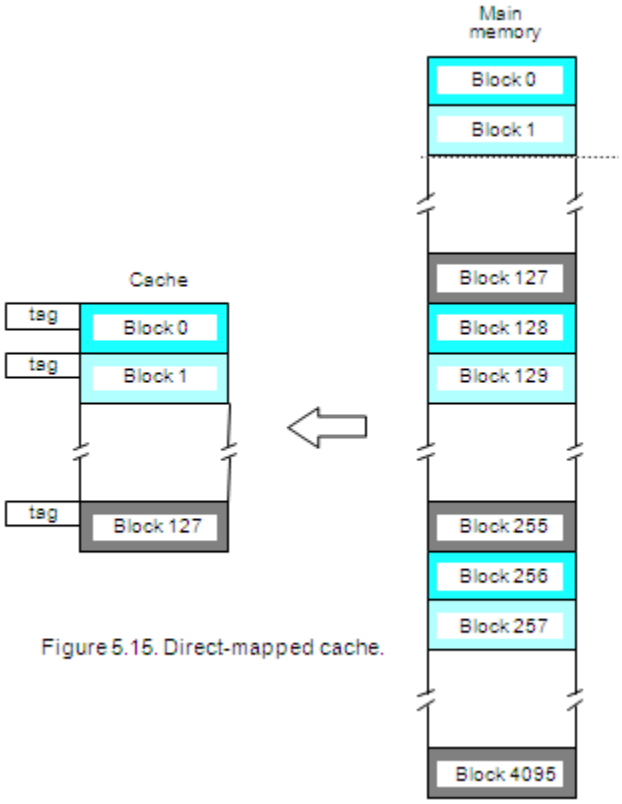
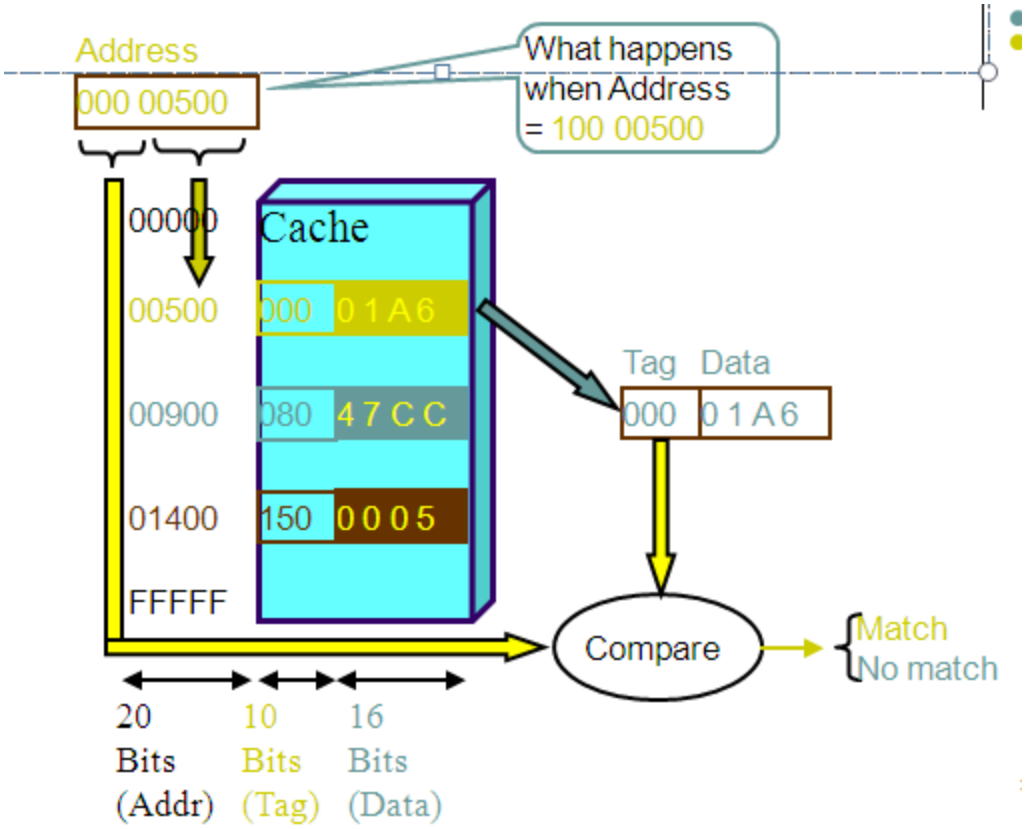
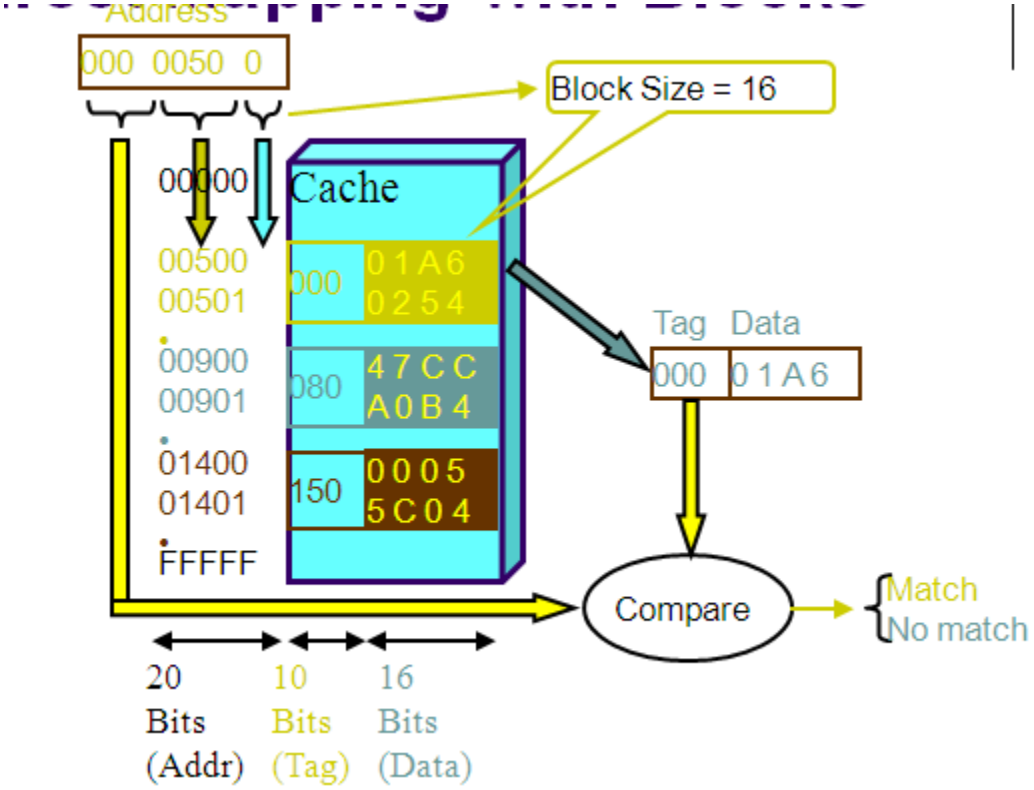


Figure 5.15. Direct-mapped cache.

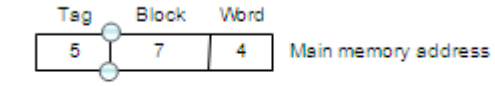
Direct Mapping



Direct Mapping with Blocks



- Tag: 11101
- Block: 1111111=127, in the 127th block of the cache
- Word:1100=12, the 12th word of the 127th block in the cache



11101,1111111,1100

Associative Mapping

4: one of 16 words. (each block has $16=2^4$ words)

12: 12 tag bits Identify which of the 4096 blocks that are resident in the cache $4096=2^{12}$.

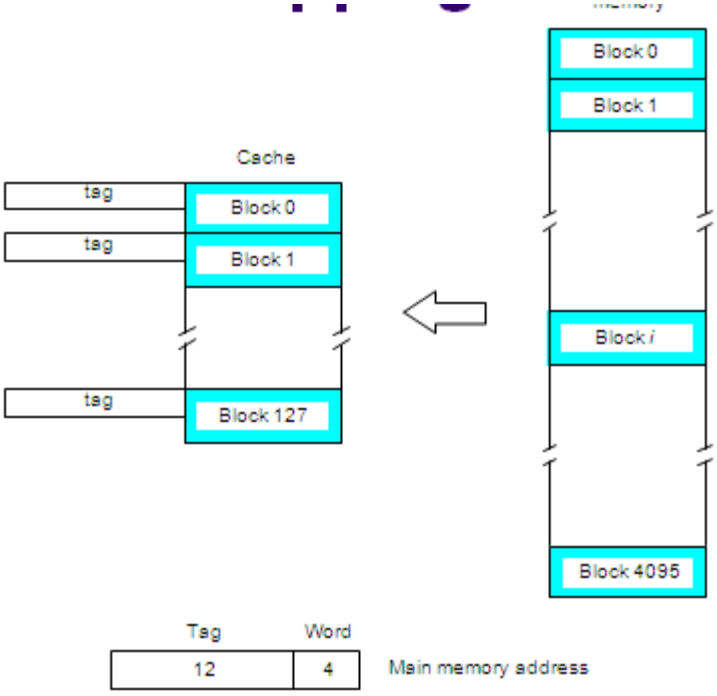
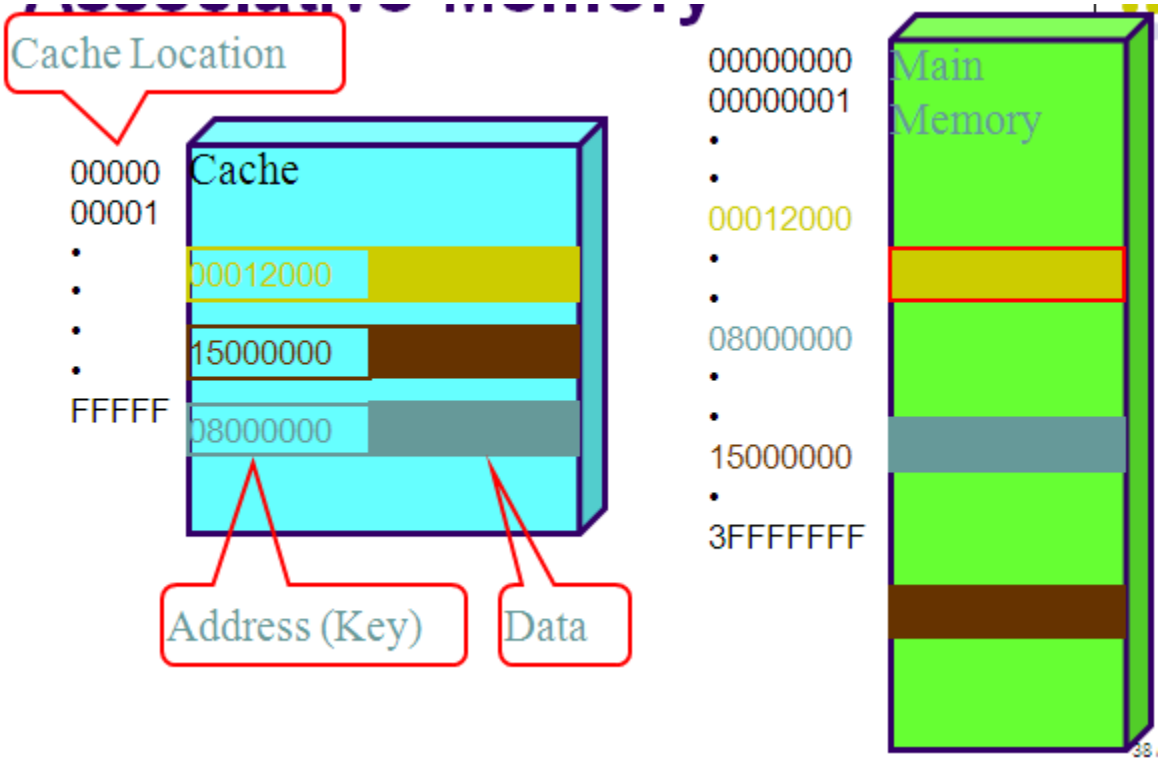
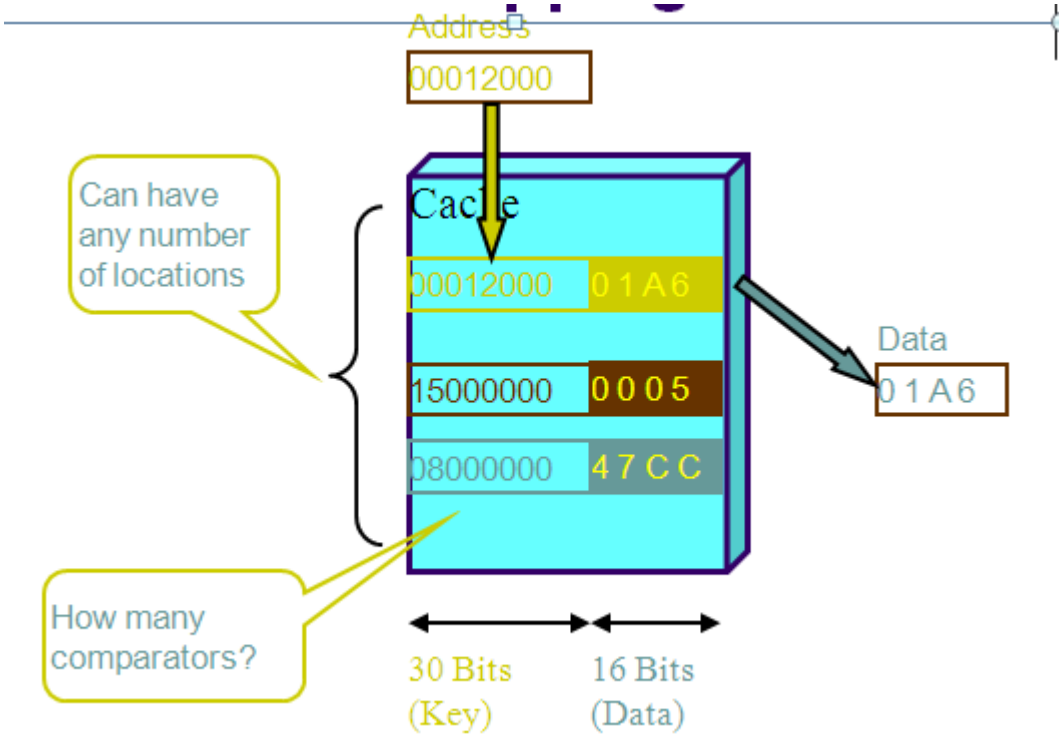
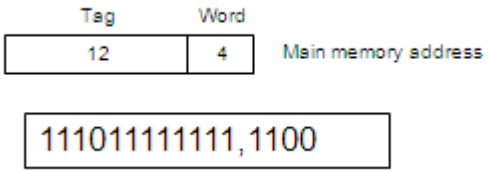


Figure 5.16. Associative-mapped cache.

Associative Memory







- Tag: 111011111111
- Word:1100=12, the 12th word of a block in the cache

Set-Associative Mapping

4: one of 16 words. (each block has $16=2^4$ words)

6: points to a particular set in the cache ($128/2=64=2^6$)

6: 6 tag bits is used to check if the desired block is present ($4096/64=2^6$).

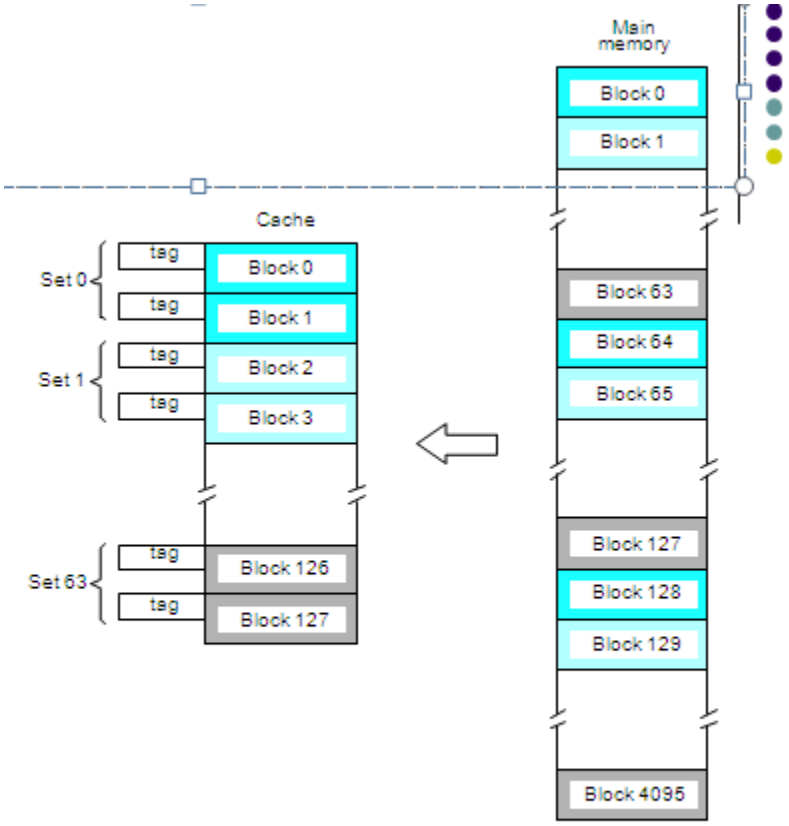
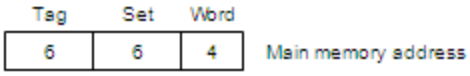
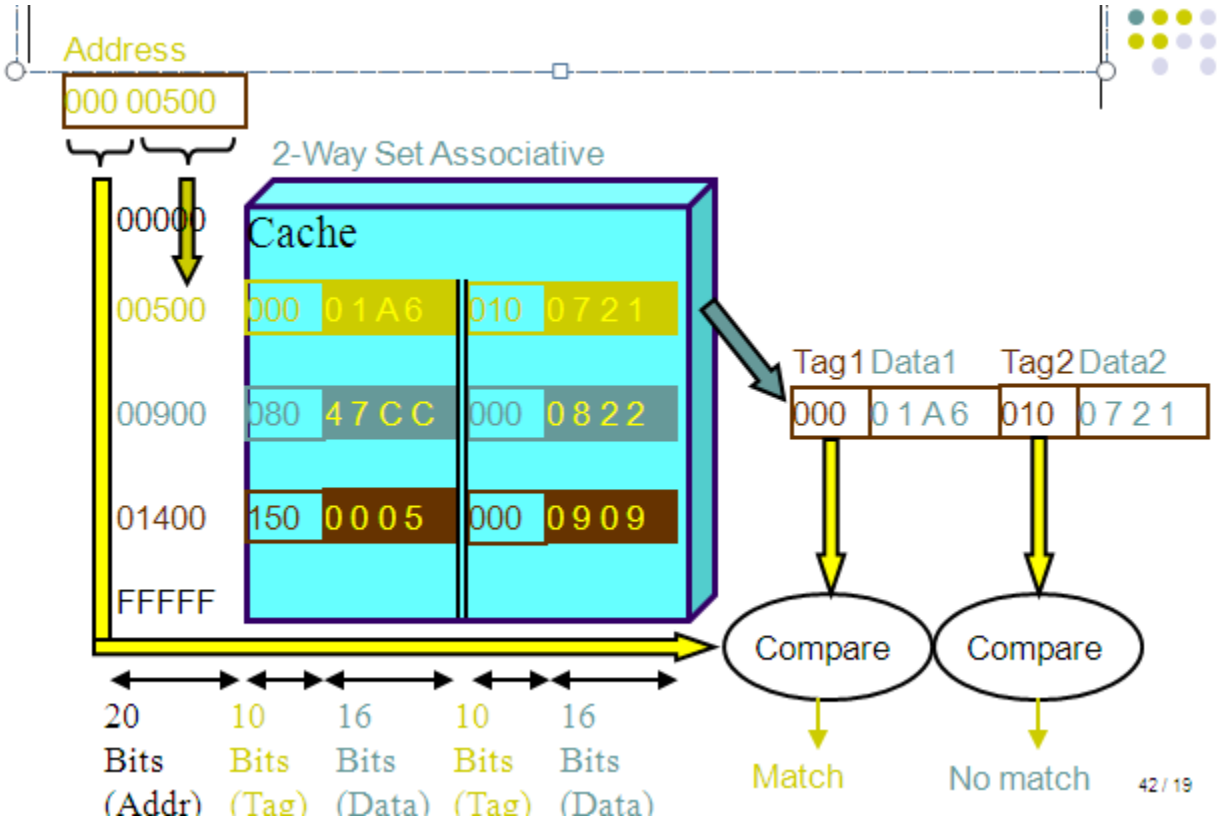
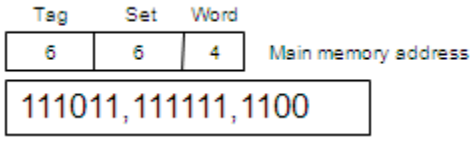


Figure 5.17. Set-associative-mapped cache with two blocks per set.



Set-Associative Mapping





- Tag: 111011
- Set: 111111=63, in the 63th set of the cache
- Word:1100=12, the 12th word of the 63th set in the cache

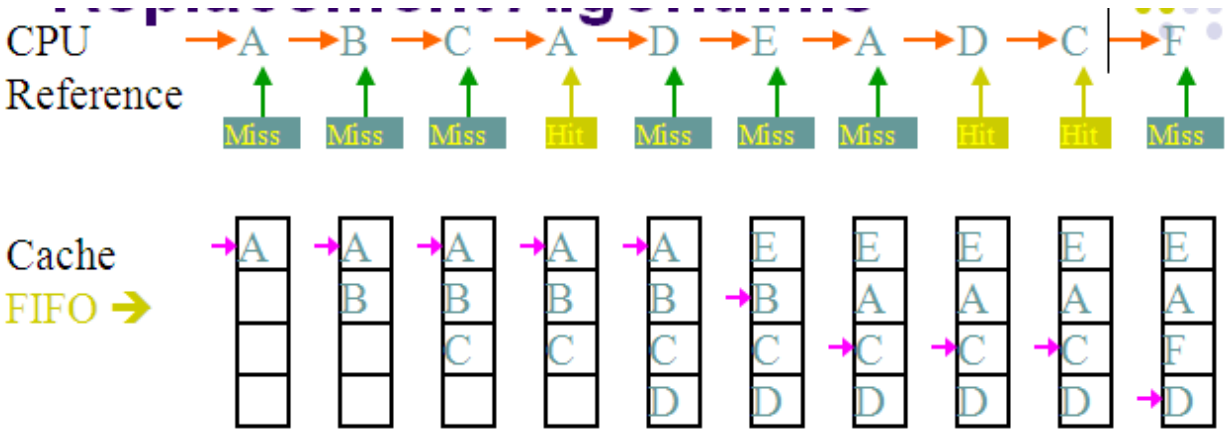
Replacement Algorithms

- Difficult to determine which blocks to kick out
- Least Recently Used (LRU) block
- The cache controller tracks references to all blocks as computation proceeds.
- Increase / clear track counters when a hit/miss occurs
- For Associative & Set-Associative Cache

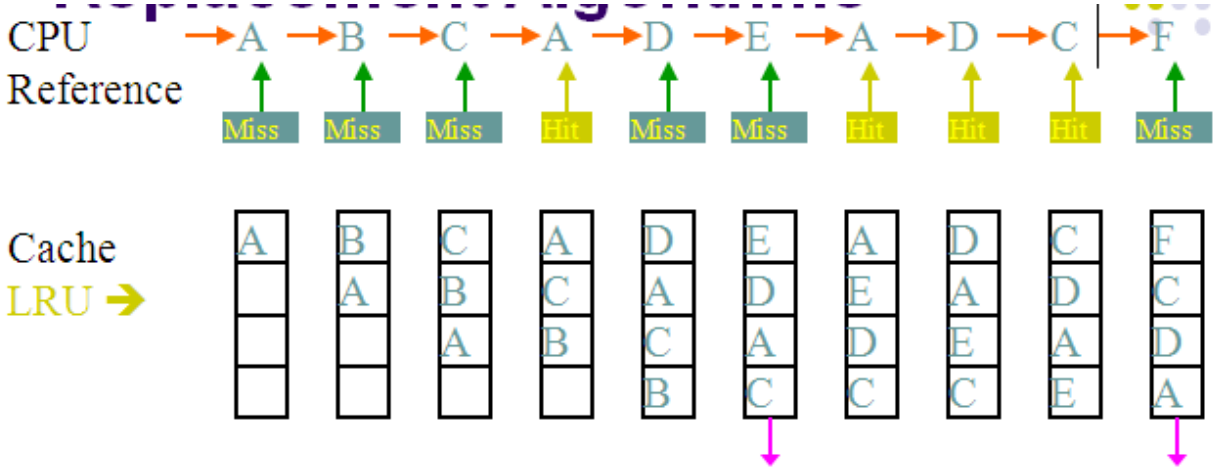
Which location should be emptied when the cache is full and a miss occurs?

  - First In First Out (FIFO)
  - Least Recently Used (LRU)

- Distinguish an *Empty* location from a *Full* one
  - Valid Bit



Hit Ratio = 3 / 10 = 0.3



Hit Ratio = 4 / 10 = 0.4

Performance Considerations

- Two key factors: performance and cost
- Price/performance ratio
- Performance depends on how fast machine instructions can be brought into the processor for execution and how fast they can be executed.
- For memory hierarchy, it is beneficial if transfers to and from the faster units can be done at a rate equal to that of the faster unit.
- This is not possible if both the slow and the fast units are accessed in the same manner.
- However, it can be achieved when parallelism is used in the organizations of the slower unit.

Interleaving

- If the main memory is structured as a collection of physically separated modules, each with its own ABR (Address buffer register) and DBR(Data buffer register), memory access operations may proceed in more than one module at the same time.

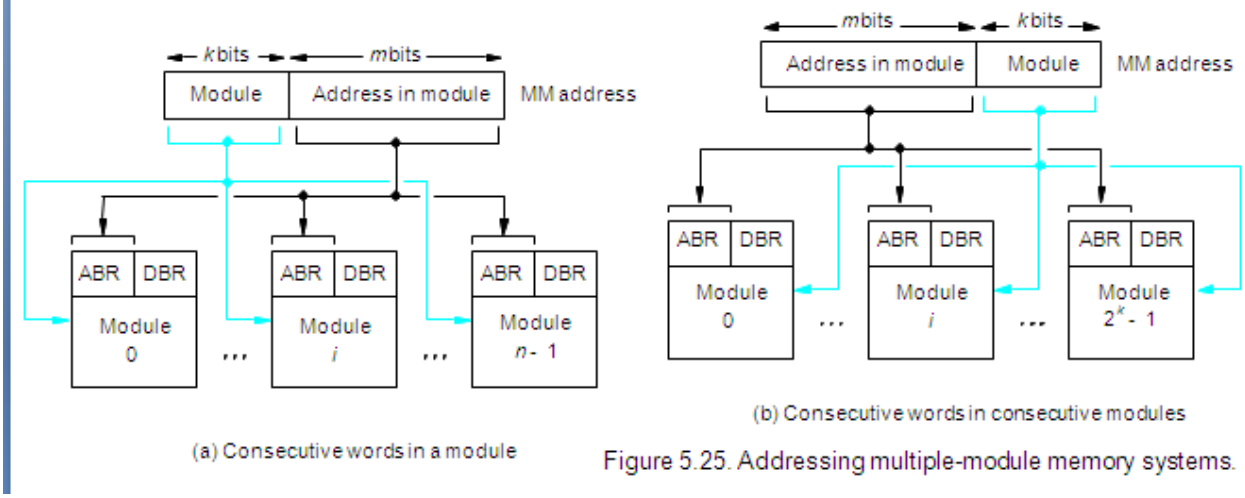


Figure 5.25. Addressing multiple-module memory systems.

Hit Rate and Miss Penalty

- The success rate in accessing information at various levels of the memory hierarchy – hit rate / miss rate.
- Ideally, the entire memory hierarchy would appear to the processor as a single memory unit that has the access time of a cache on the processor chip and the size of a magnetic disk – depends on the hit rate (>>0.9).
- A miss causes extra time needed to bring the desired information into the cache.
- Example 5.2, page 332.
- $T_{ave}=hC+(1-h)M$
  - $T_{ave:}$ average access time experienced by the processor
  - h: hit rate
  - M: miss penalty, the time to access information in the main memory
  - C: the time to access information in the cache
- Example:
  - Assume that 30 percent of the instructions in a typical program perform a read/write operation, which means that there are 130 memory accesses for every 100 instructions executed.
  - h=0.95 for instructions, h=0.9 for data
  - C=10 clock cycles, M=17 clock cycles, interleaved memory

Time without cache 130x10

Time with cache 100(0.95x1+0.05x17)+30(0.9x1+0.1x17)

- The computer with the cache performs five times better

How to Improve Hit Rate?

- Use larger cache – increased cost
- Increase the block size while keeping the total cache size constant.
- However, if the block size is too large, some items may not be referenced before the block is replaced – miss penalty increases.
- Load-through approach

Caches on the Processor Chip

- On chip vs. off chip
- Two separate caches for instructions and data, respectively
- Single cache for both
- Which one has better hit rate? -- Single cache
- What's the advantage of separating caches? – parallelism, better performance
- Level 1 and Level 2 caches
- L1 cache – faster and smaller. Access more than one word simultaneously and let the processor use them one at a time.
- L2 cache – slower and larger.
- How about the average access time?
- Average access time: $t_{ave} = h_1C_1 + (1-h_1)h_2C_2 + (1-h_1)(1-h_2)M$

where h is the hit rate, C is the time to access information in cache, M is the time to access information in main memory.

- Write buffer – processor doesn't need to wait for the memory write to be completed
- Prefetching – prefetch the data into the cache before they are needed
- Lockup-Free cache – processor is able to access the cache while a miss is being serviced.

Virtual Memories

- Physical main memory is not as large as the address space spanned by an address issued by the processor.

  2^{32} = 4 GB, 2^{64} = …

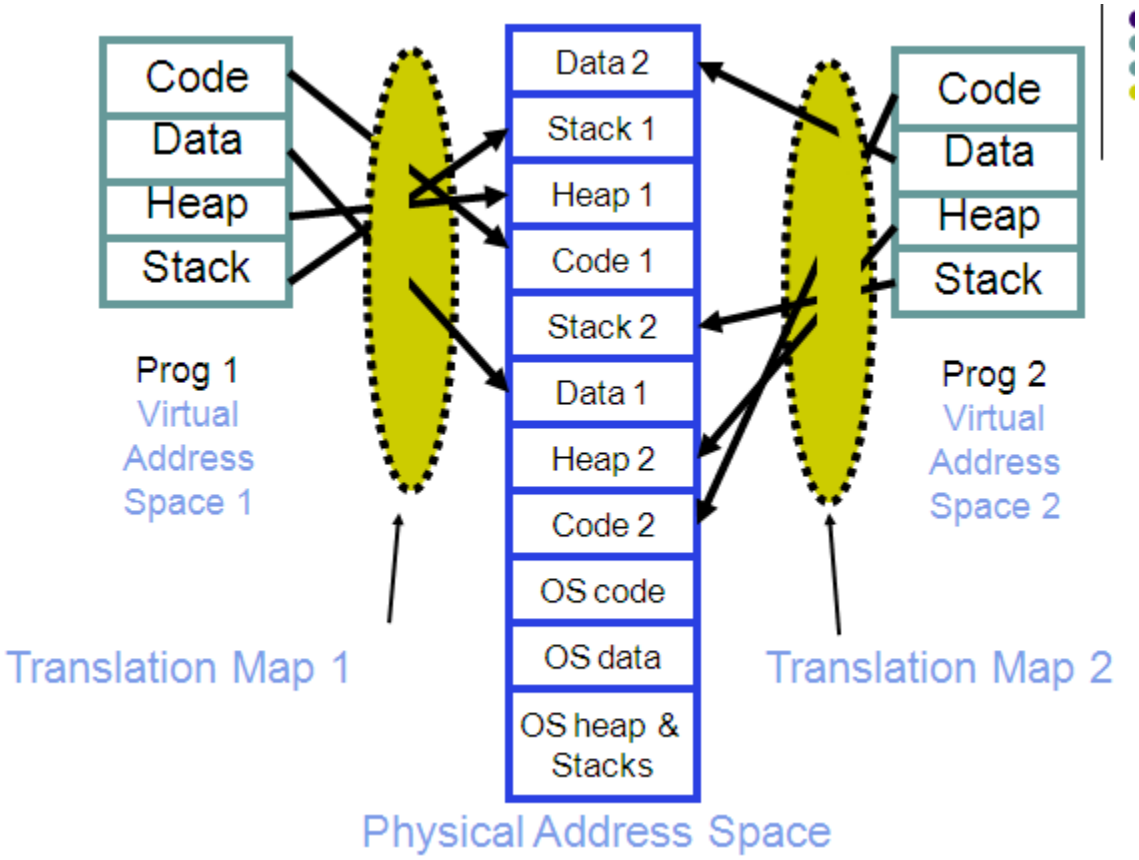
- When a program does not completely fit into the main memory, the parts of it not currently being executed are stored on secondary storage devices.
- Techniques that automatically move program and data blocks into the physical main memory when they are required for execution are called virtual-memory techniques.

Virtual addresses will be translated into physical addresses

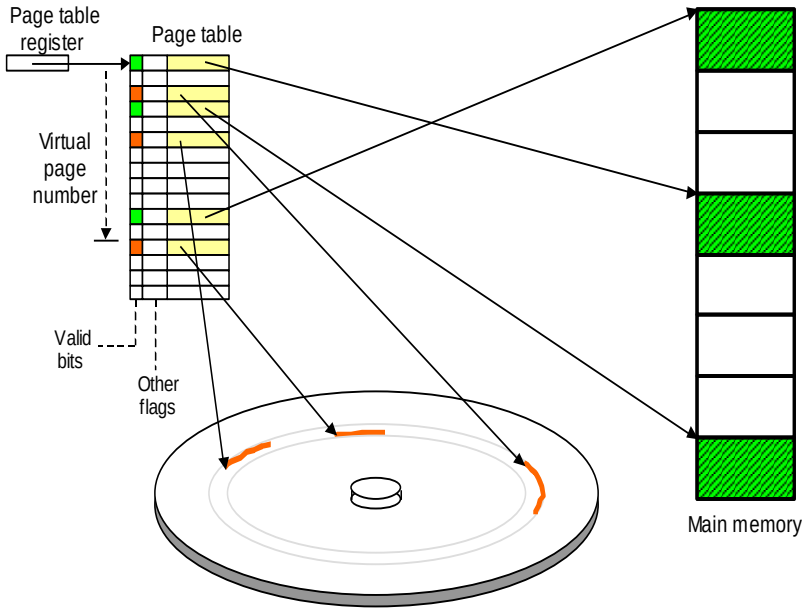
Address Translation

- All programs and data are composed of fixed-length units called pages, each of which consists of a block of words that occupy contiguous locations in the main memory.
- Page cannot be too small or too large.
- The virtual memory mechanism bridges the size and speed gaps between the main memory and secondary storage – similar to cache.

Example: Example of Address Translation



Page Tables and Address Translation



The role of page table in the virtual-to-physical address translation process.

Address Translation

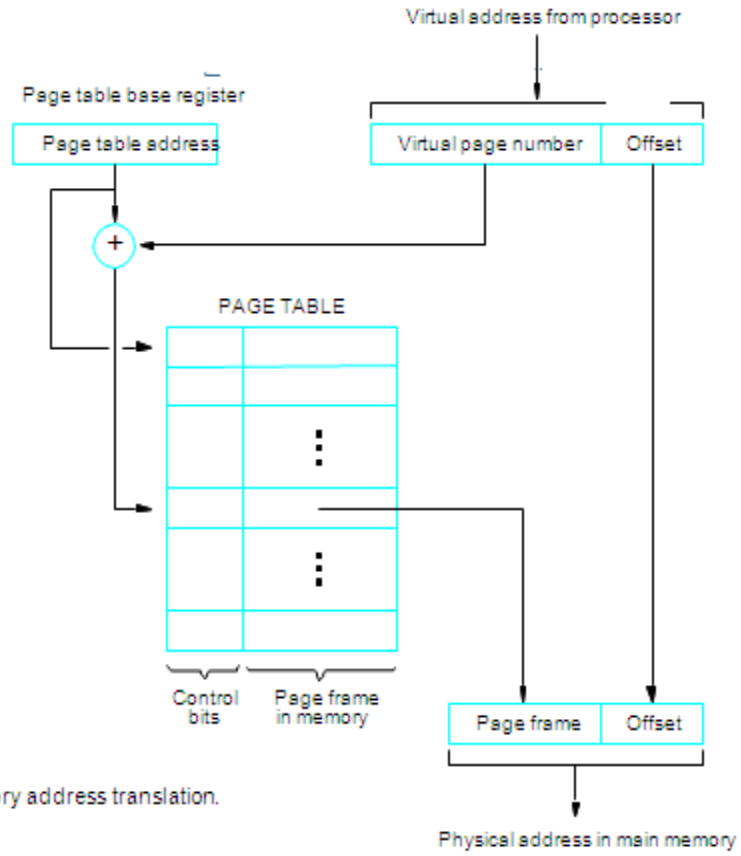


Figure 5.27. Virtual-memory address translation.

- The page table information is used by the MMU for every access, so it is supposed to be with the MMU.
- However, since MMU is on the processor chip and the page table is rather large, only small portion of it, which consists of the page table entries that correspond to the most recently accessed pages, can be accommodated within the MMU.
- Translation Lookaside Buffer (TLB)

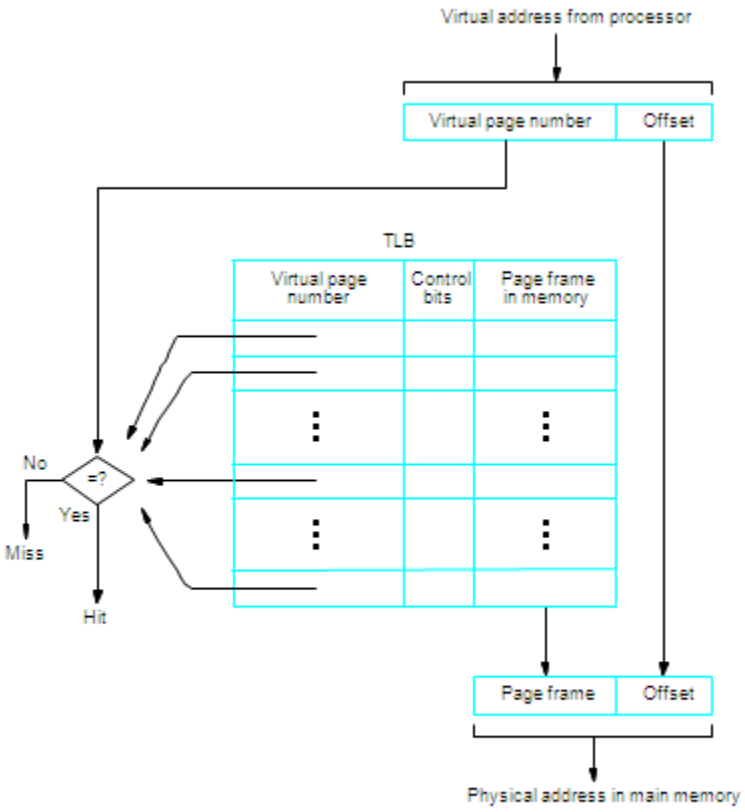


Figure 5.28. Use of an associative-mapped TLB.

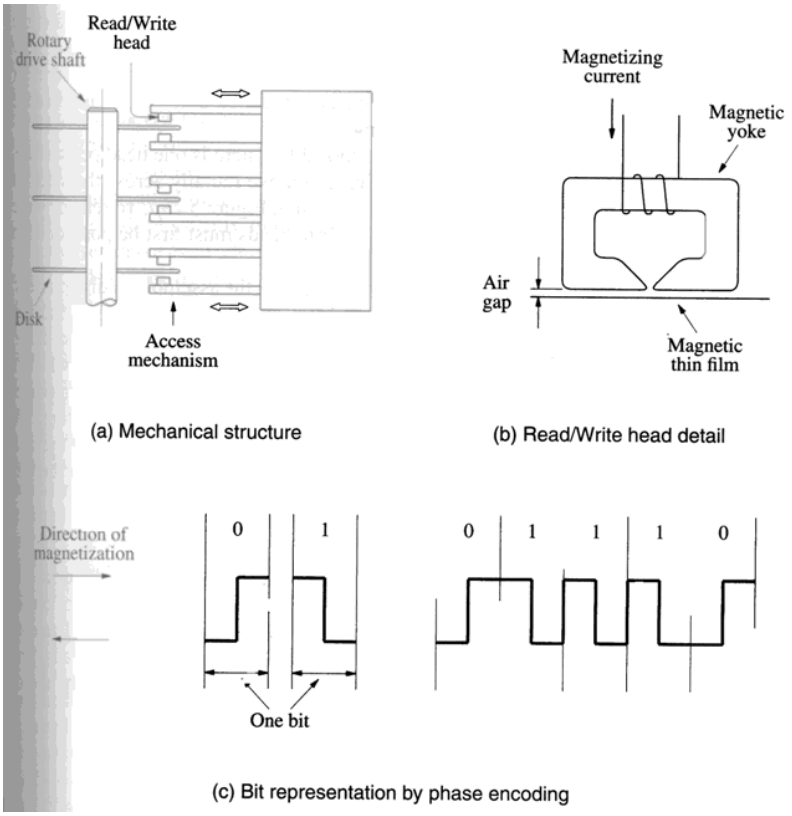
- The contents of TLB must be coherent with the contents of page tables in the memory.
- Translation procedure.
- Page fault
- Page replacement
- Write-through is not suitable for virtual memory.
- Locality of reference in virtual memory

Memory Management Requirements

- Multiple programs
- System space / user space
- Protection (supervisor / user state, privileged instructions)
- Shared pages

Secondary Storage

Magnetic Hard Disks



Disk

Disk drive

Disk controller

Organization of Data on a Disk

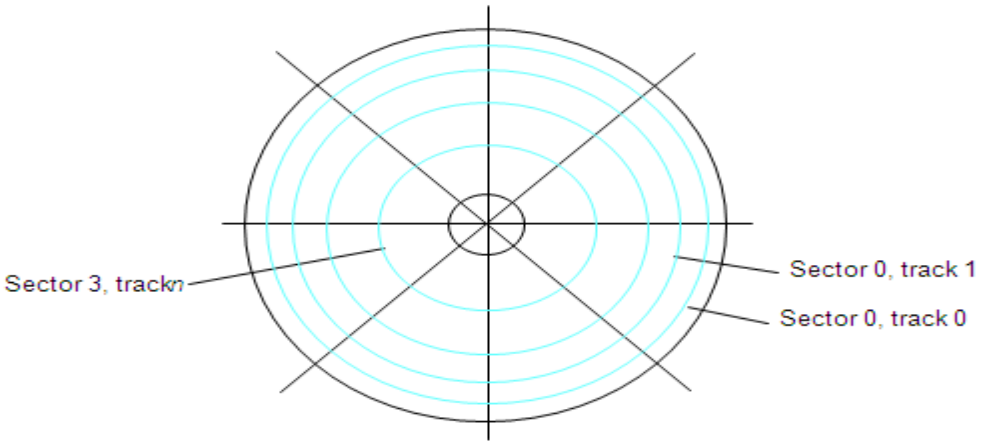


Figure 5.30. Organization of one surface of a disk.

Access Data on a Disk

- Sector header
- Following the data, there is an error-correction code (ECC).
- Formatting process
- Difference between inner tracks and outer tracks
- Access time – seek time / rotational delay (latency time)
- Data buffer/cache

Disk Controller

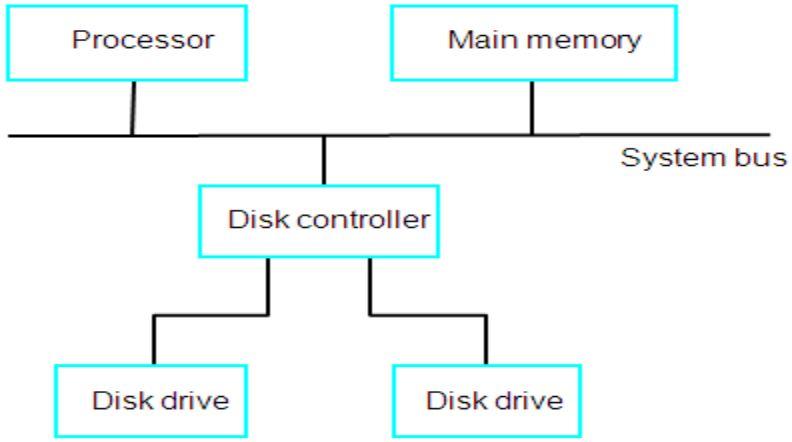


Figure 5.31. Disks connected to the system bus.

- Seek
- Read
- Write
- Error checking

RAID Disk Arrays

- Redundant Array of Inexpensive Disks
- Using multiple disks makes it cheaper for huge storage, and also possible to improve the reliability of the overall system.
- RAID0 – data striping
- RAID1 – identical copies of data on two disks
- RAID2, 3, 4 – increased reliability

- RAID5 – parity-based error-recovery

Optical Disks

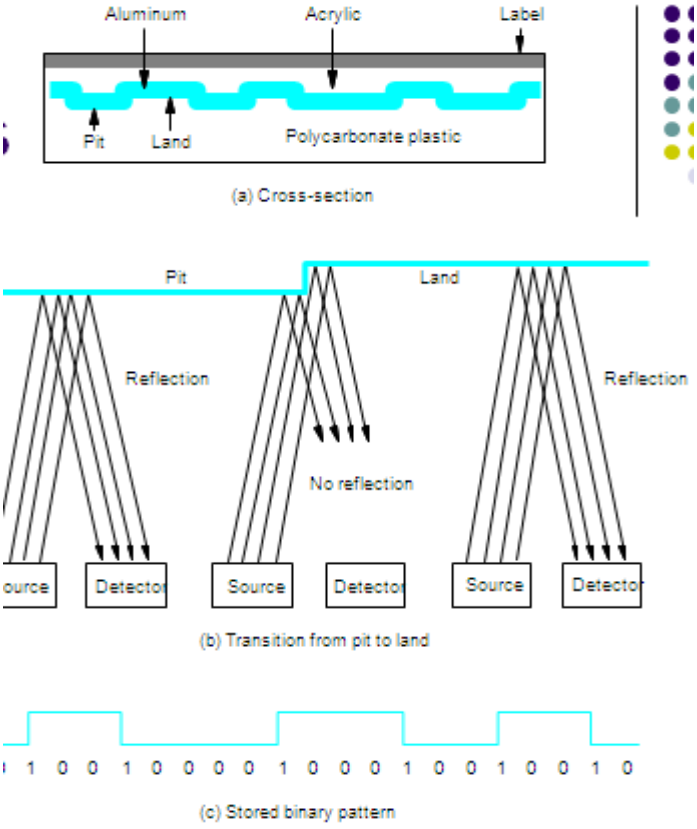


Figure 5.32. Optical disk.

- CD-ROM
- CD-Recordable (CD-R)
- CD-ReWritable (CD-RW)
- DVD
- DVD-RAM

Magnetic Tape Systems

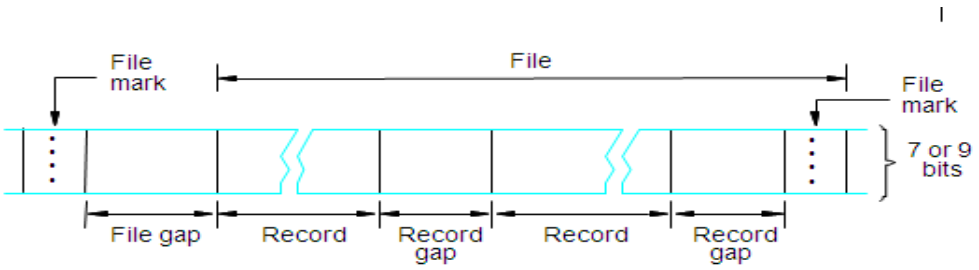


Figure 5.33. Organization of data on magnetic tape.