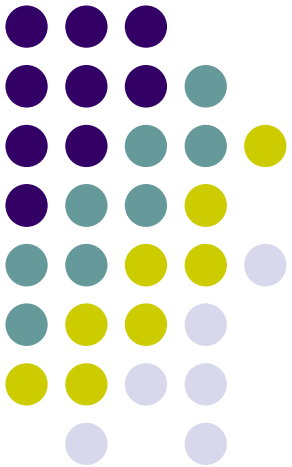
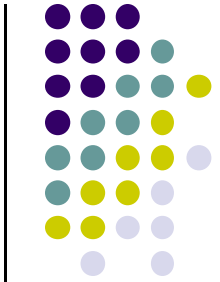
Unit IV – Memory System

C.Bala Subramanian,
AP/CSE,
KLU

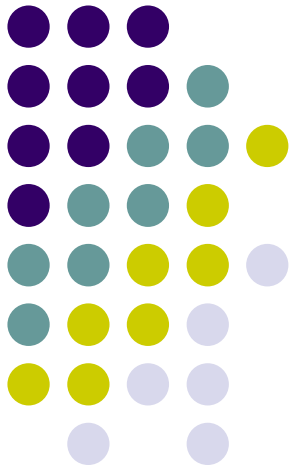


Overview



- Basic memory circuits
- Organization of the main memory
- Cache memory concept
- Virtual memory mechanism
- Secondary storage

Some Basic Concepts

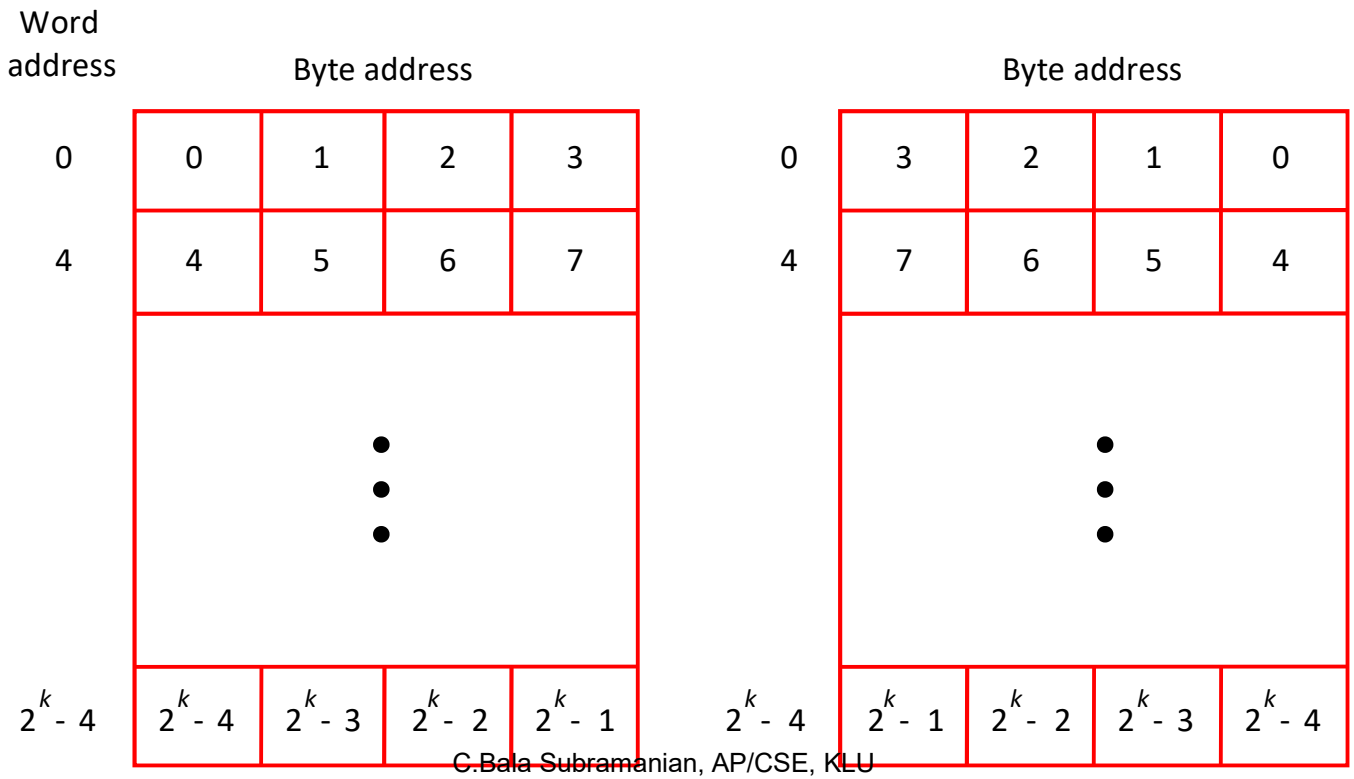


Basic Concepts

- 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.



(a) Big-endian assignment (b) Little-endian assignment

Traditional Architecture

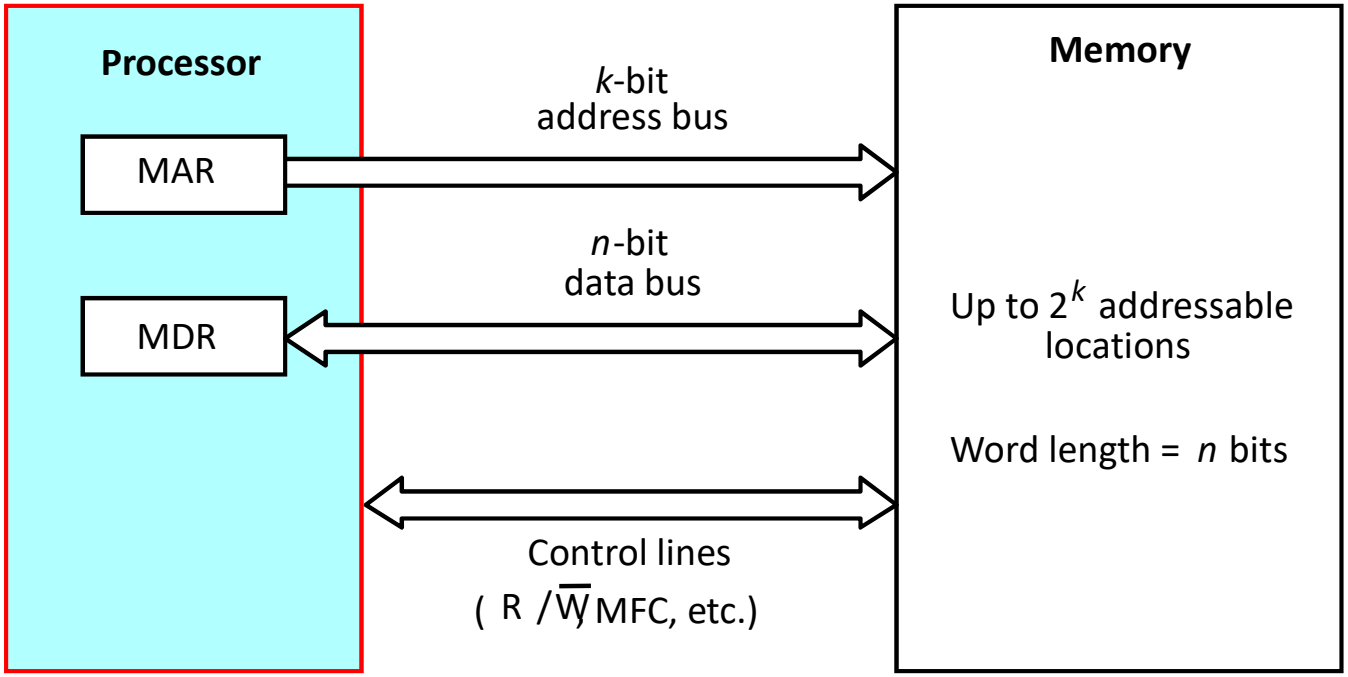
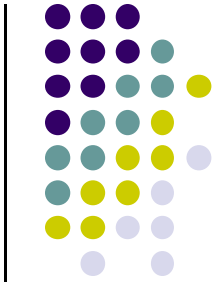


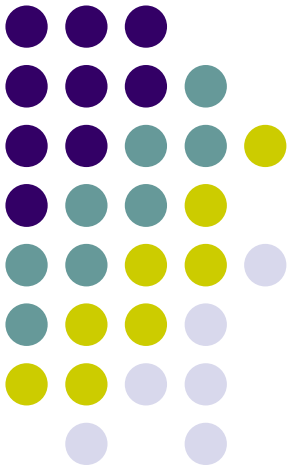
Figure 5.1. Connection of the memory to the processor.

Basic Concepts

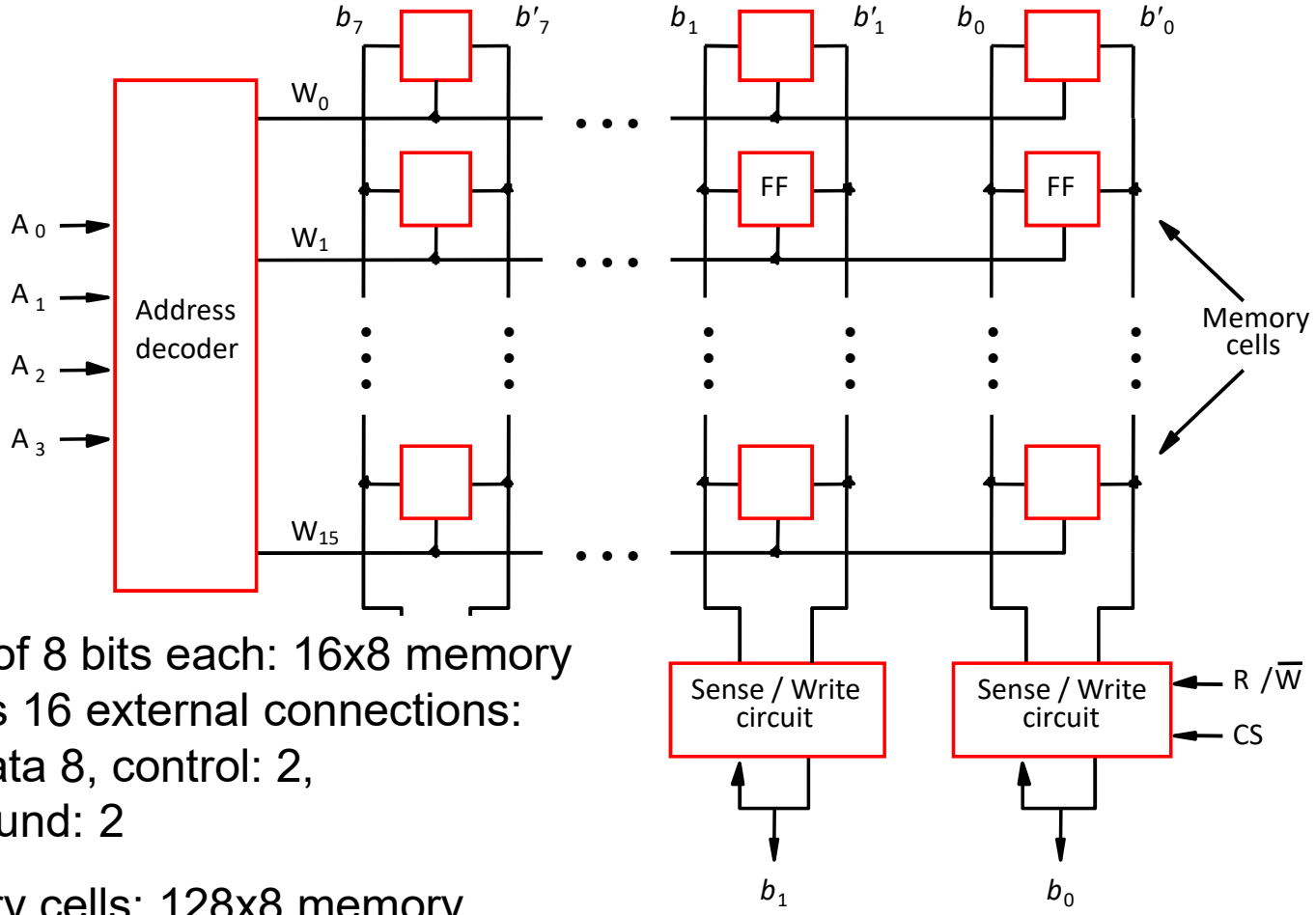


- "Block transfer" – bulk data transfer
- *Memory access time*
- *Memory cycle time*
- RAM – any location can be accessed for a Read or Write operation in some fixed amount of time that is independent of the location's address.
- Cache memory
- Virtual memory, memory management unit

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

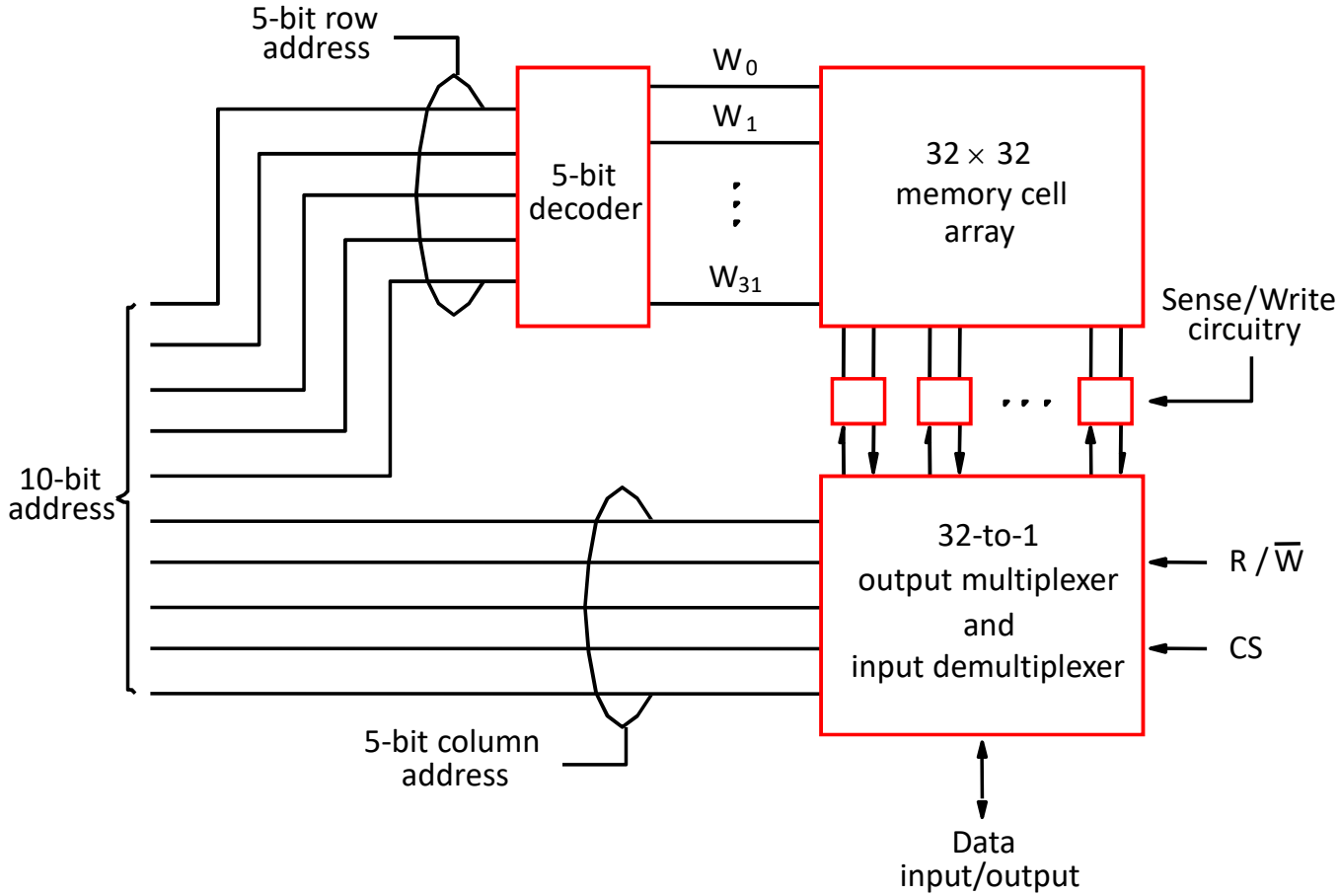


Figure 5.3. Organization of a 1K × 1 memory chip.

Static Memories

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

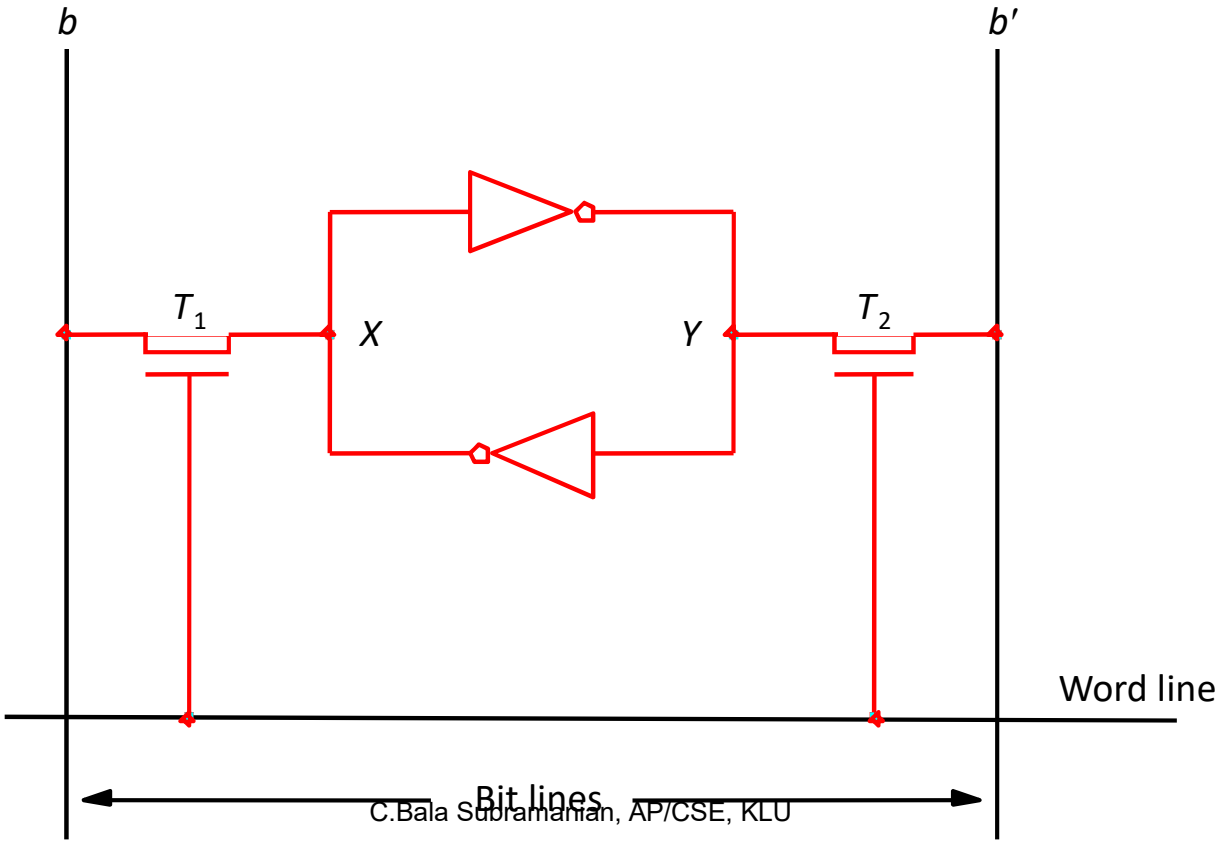


Figure 5.4. A static RAM cell.

Static Memories

- CMOS cell: low power consumption

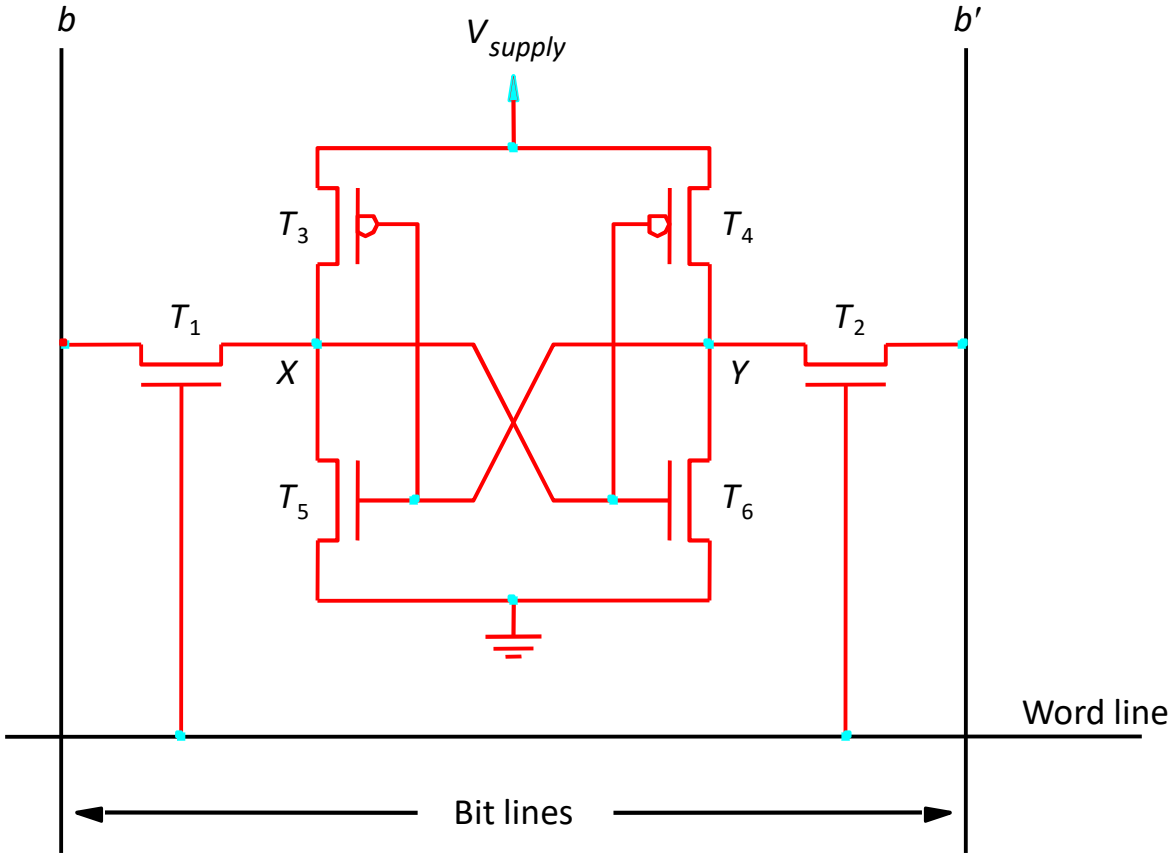
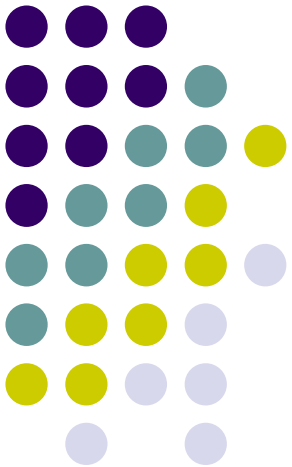


Figure 5.5. An example of a CMOS memory cell.

Read-Only Memories



Read-Only-Memory

- Volatile / non-volatile memory
- ROM
- PROM
- EPROM
- EEPROM

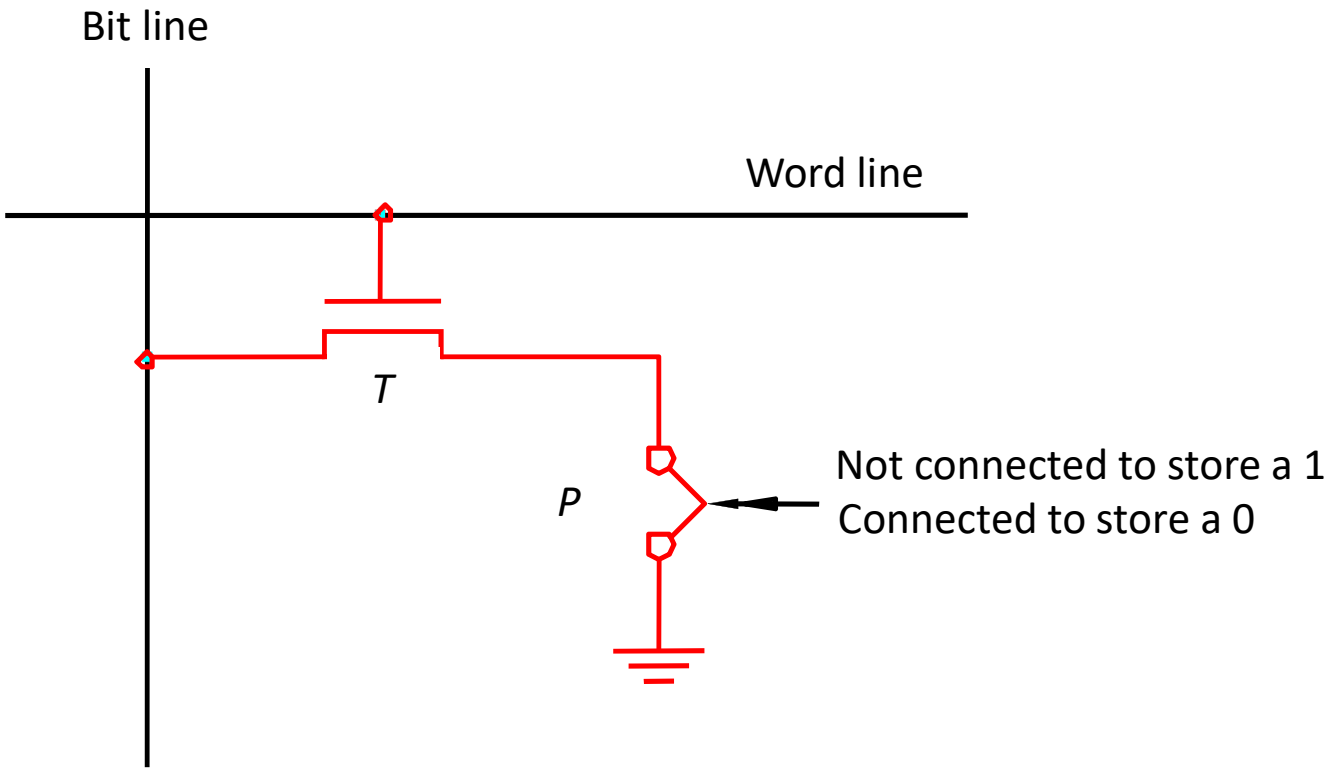
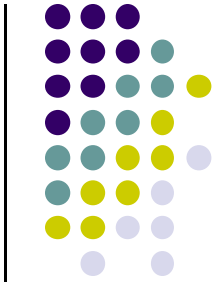


Figure 5.12. A ROM cell.

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

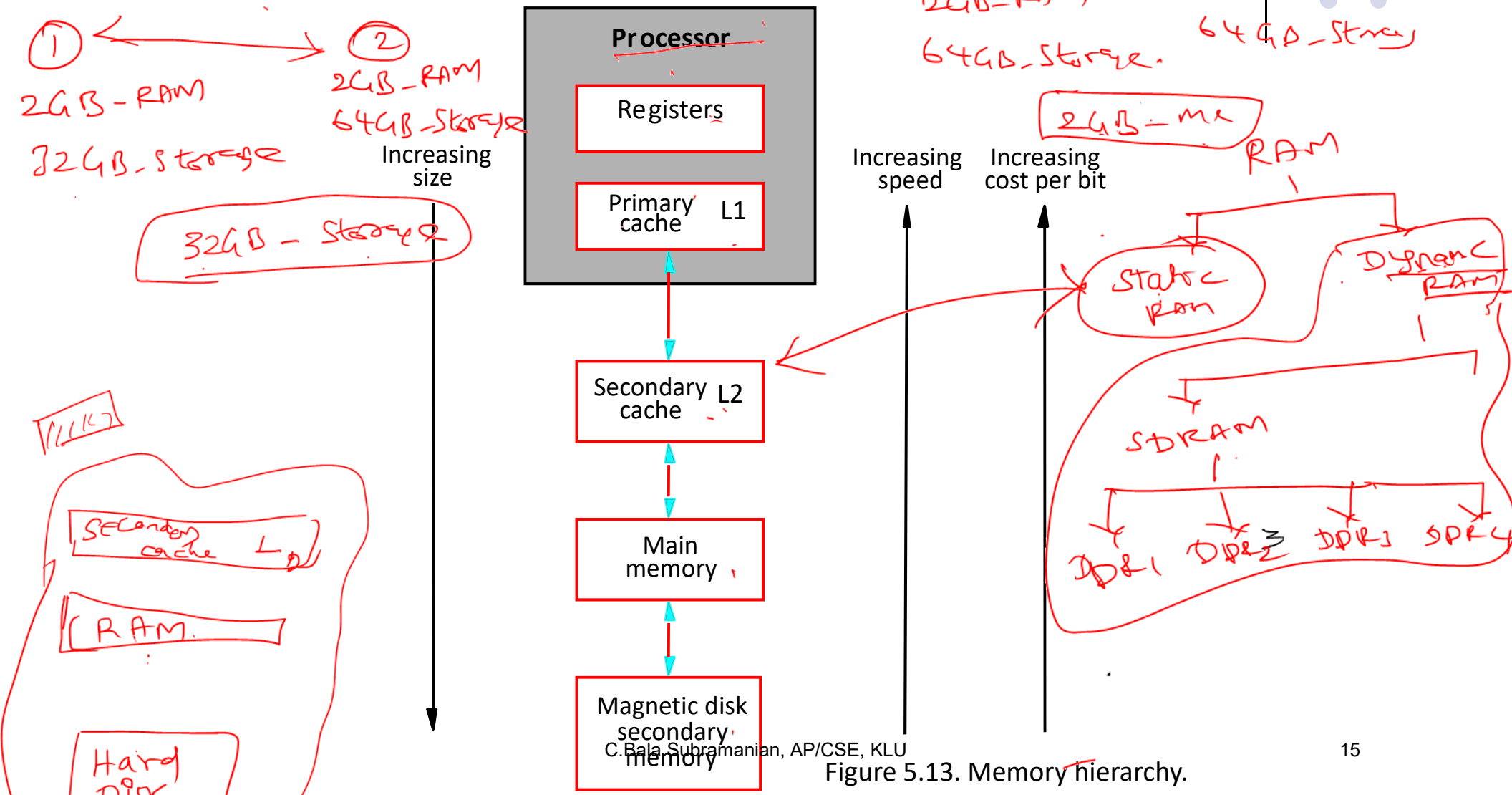
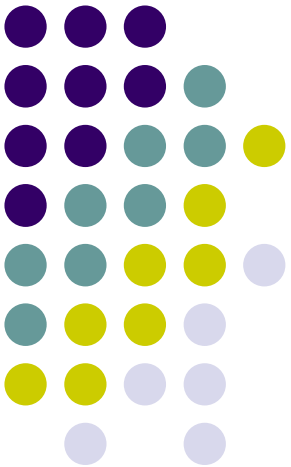
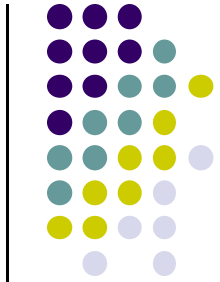


Figure 5.13. Memory hierarchy.

Cache Memories



Cache



- What is cache?
- Why we need it?
- Locality of reference (very important) –The effectiveness of the cache mechanism is based on a property of a computer programs

  -Temporal :Recently executed instruction is likely to be executed very soon

  -Spatial :Instructions in close proximity to a recently executed instruction are also likely to be executed soon.
- Cache block –Set of continuous address locations of some size.
- *Cache line –To refer a cache block.*

Cache

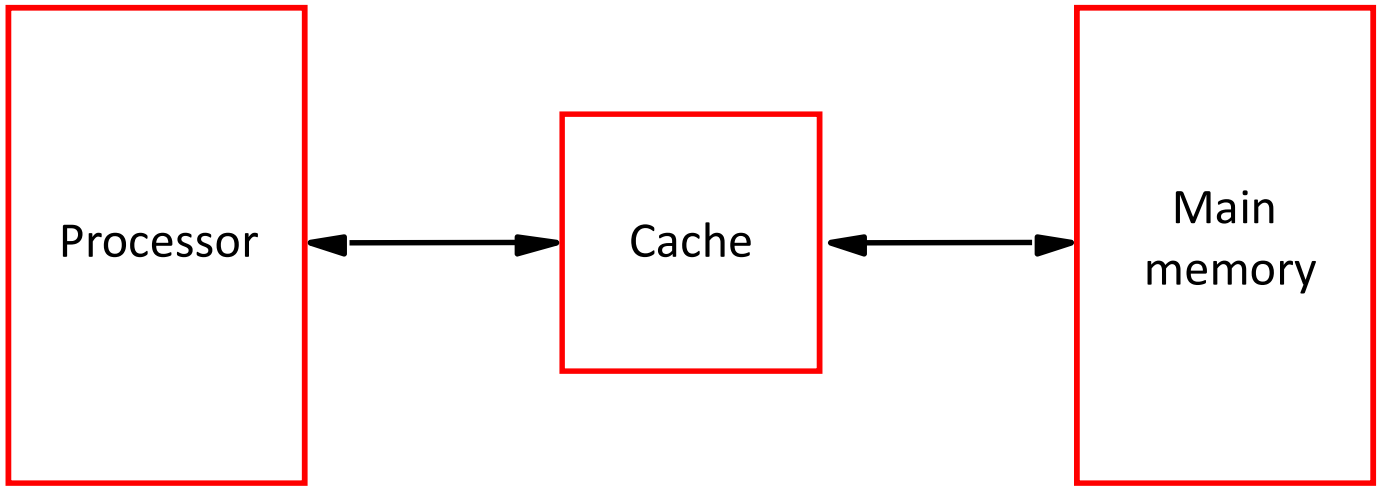


Figure 5.14. Use of a cache memory.

- Replacement algorithm
- Hit / miss
- Write-through / Write-back
- Load through

Mapping Functions

a. Direct Mapping

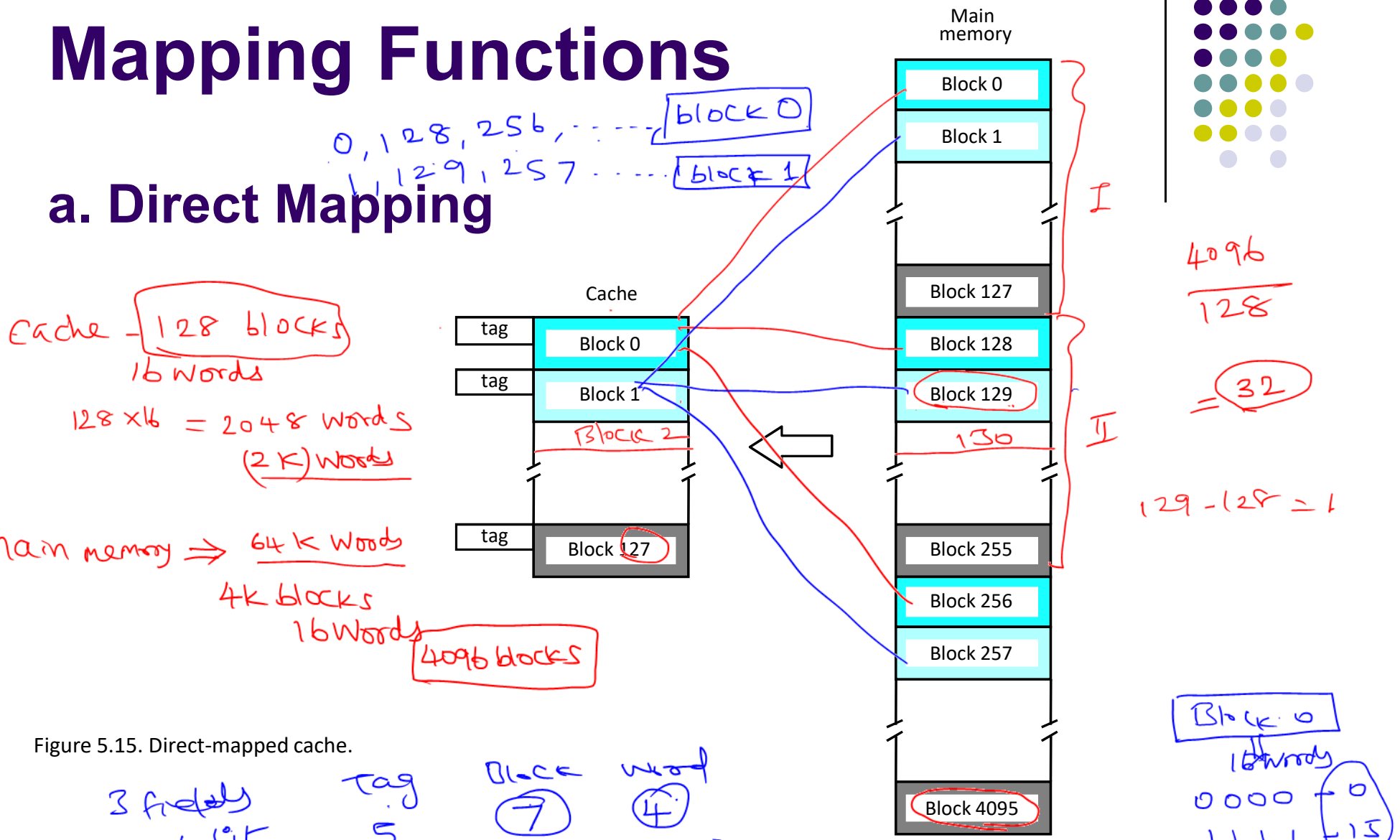


Figure 5.15. Direct-mapped cache.

b. Associative Mapping

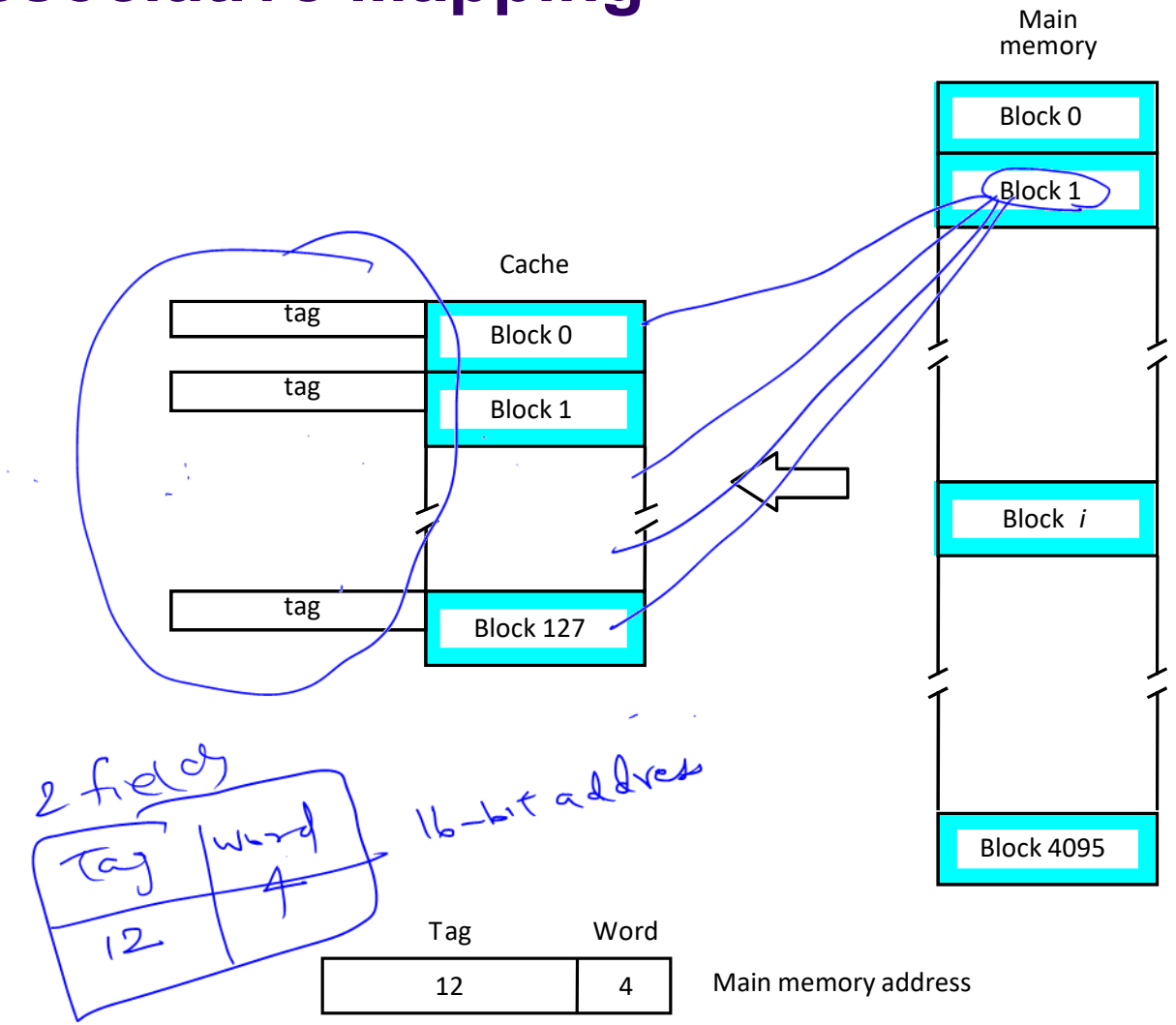


Figure 5.16. Associative-mapped cache.

c. Set-Associative Mapping

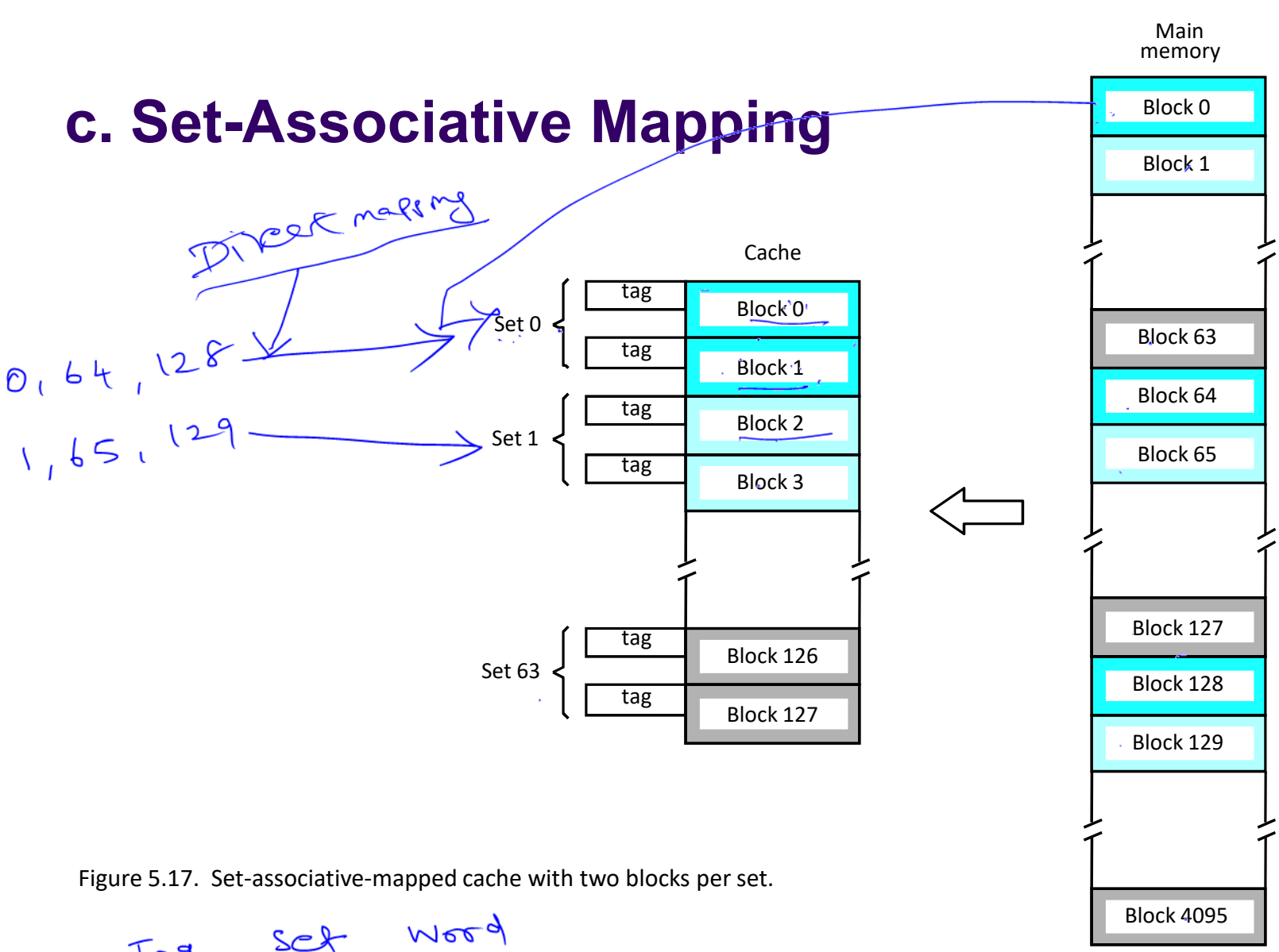
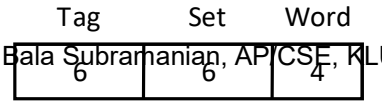


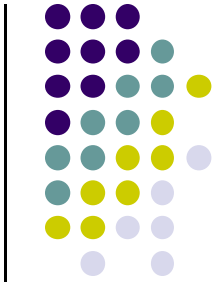
Figure 5.17. Set-associative-mapped cache with two blocks per set.

Tag 6 | Set 6 | Word 4

16-bit address

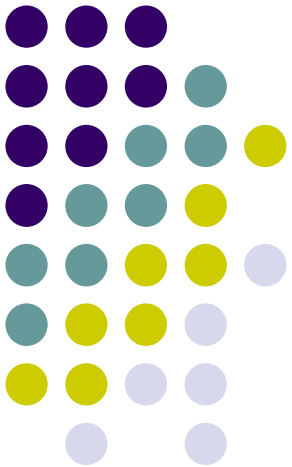


Replacement Algorithms

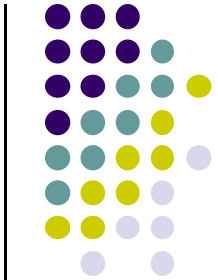


- Difficult to determine which blocks to kick out
- Least Recently Used (LRU) block
- The cache controller tracks references to all blocks as computation proceeds.
- Increase / clear track counters when a hit/miss occurs

Performance Considerations



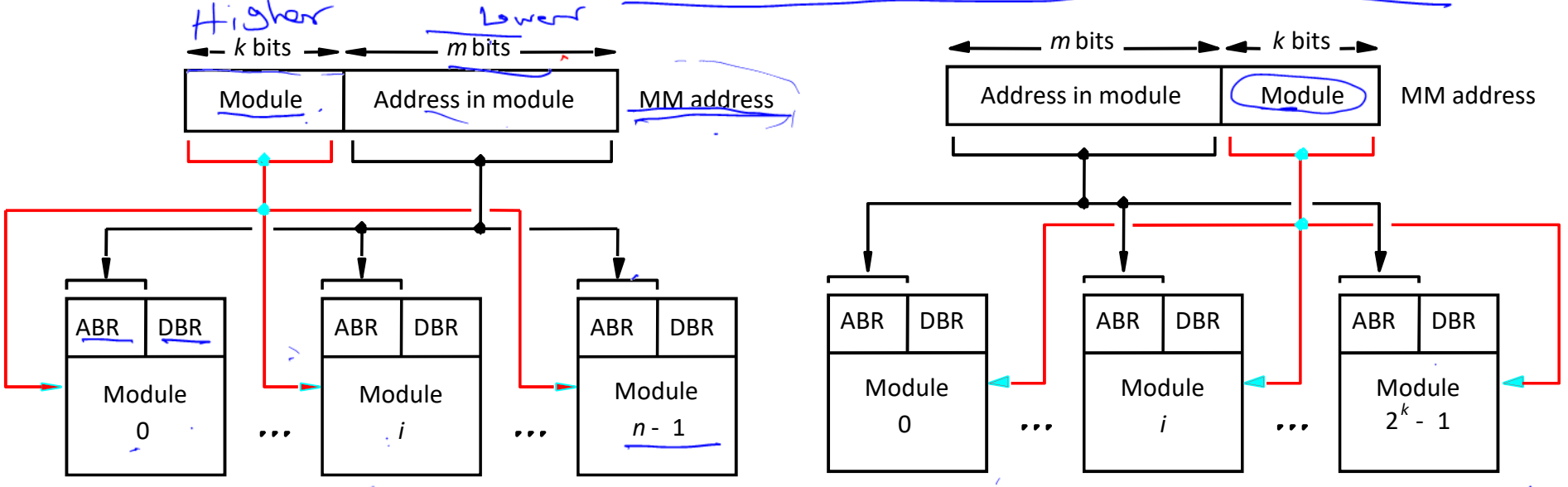
Overview



- 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 and DBR, memory access operations may proceed in more than one module at the same time.

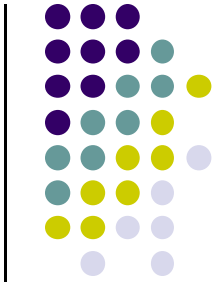


(a) Consecutive words in a module

(b) Consecutive words in consecutive modules

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 hierarch 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.

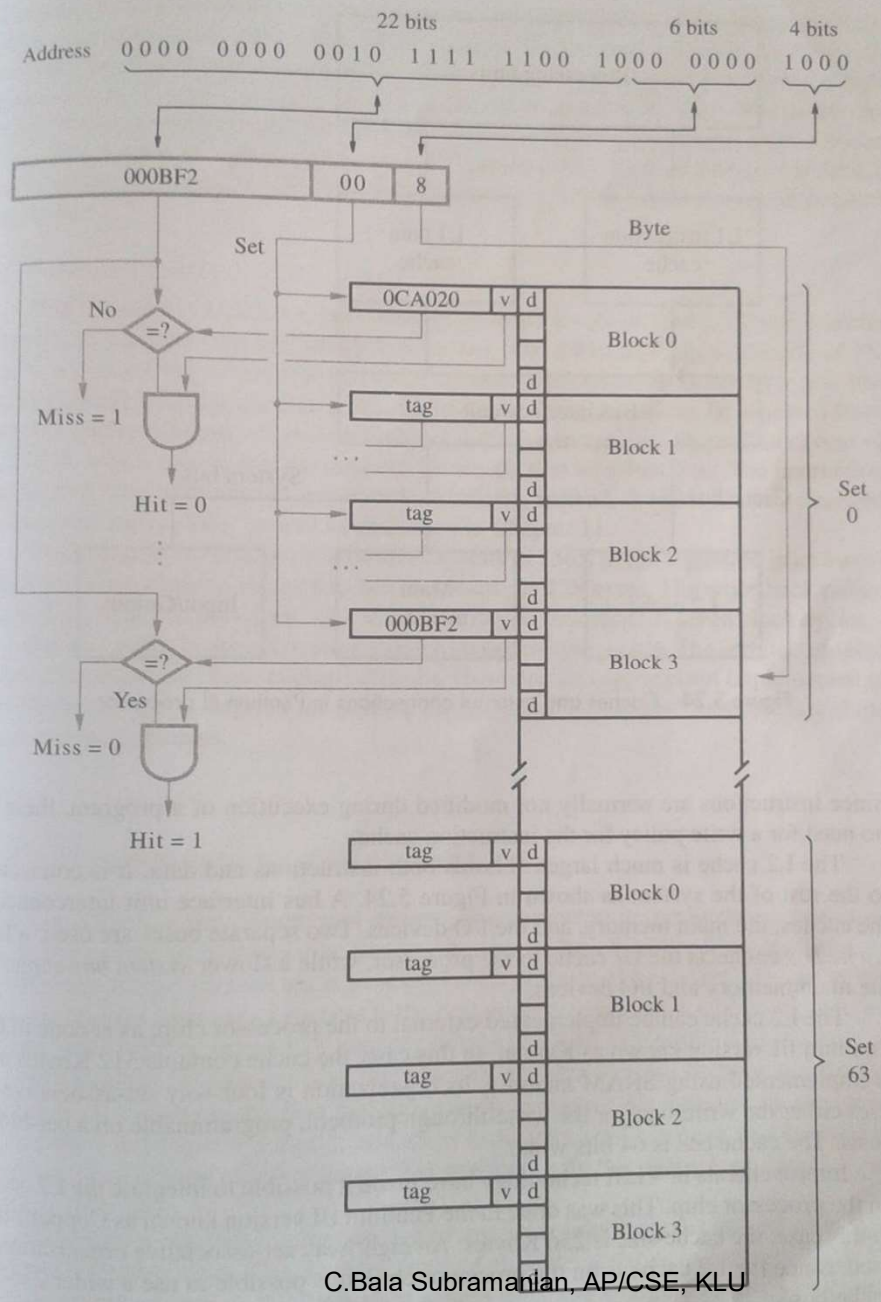
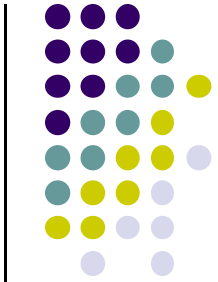


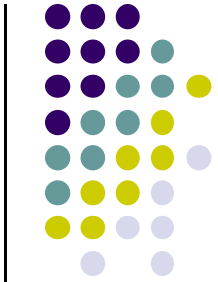
Figure 5.23 Data cache organization in the 68040 microprocessor.

How to Improve Hit Rate?



- Use larger cache
- 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

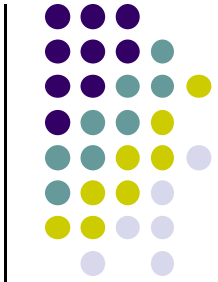


- On chip vs. off chip
- Two separate caches for instructions and data, respectively
- Single cache for both
- Which one has better hit rate?
- What's the good for separating caches?
- Level 1 and Level 2 caches
- L1 cache – faster and smaller. Access more than one word simultaneously and let the processor 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.

one cache

$t_{ave} = hC + (1-h)M$

Other Enhancements

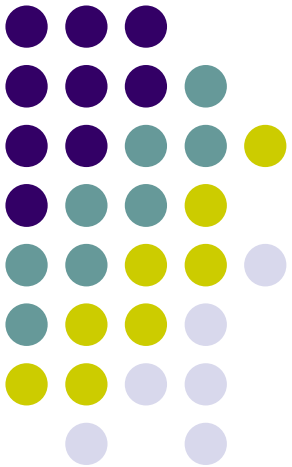


- Write buffer – processor doesn't need to wait for the memory write to be completed

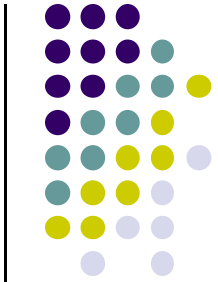
- Prefetching – prefetch the data into the cache before they are needed

- Lockup-Free cache – processor is able to access the cache while a miss is being serviced.

Virtual Memories



Overview



- 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.

Overview

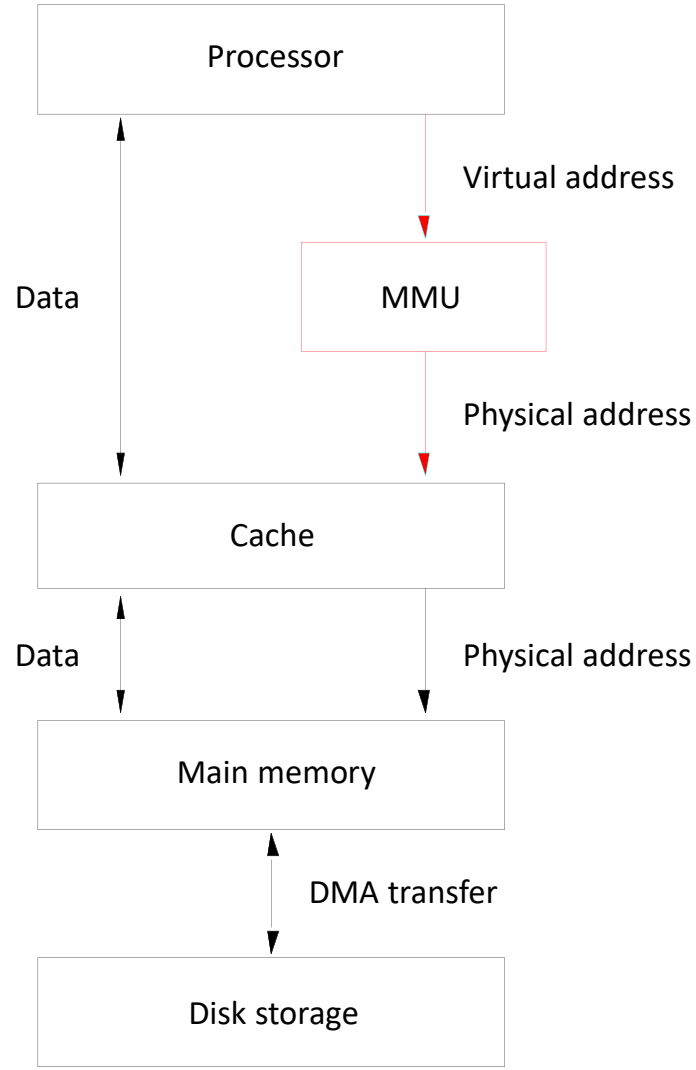
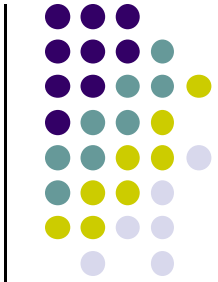


Figure 5.26. Virtual memory organization.

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.

Address Translation

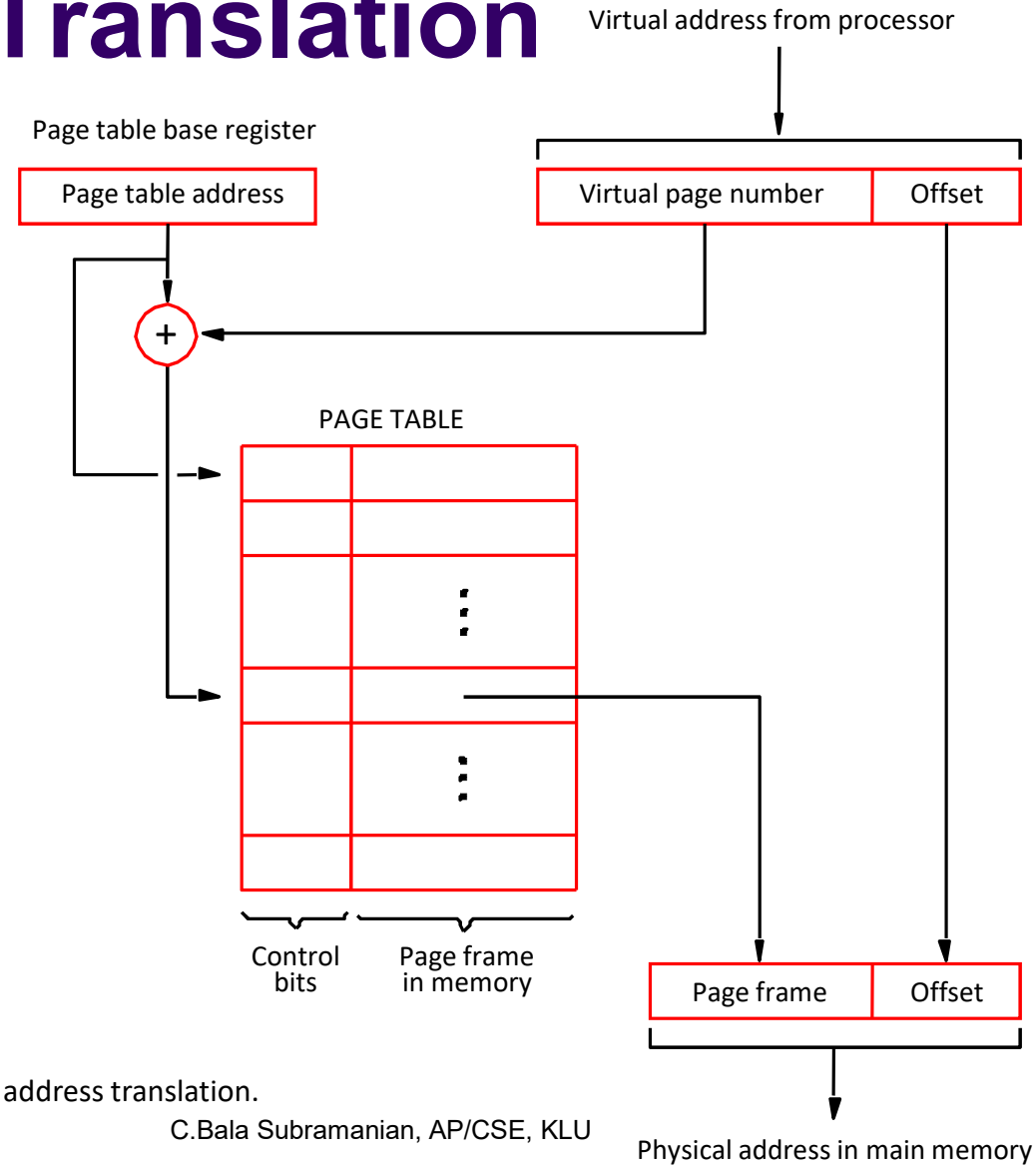
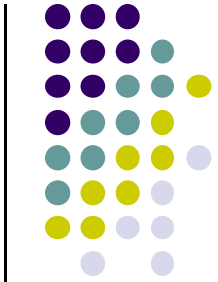


Figure 5.27. Virtual-memory address translation.

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)

TLB

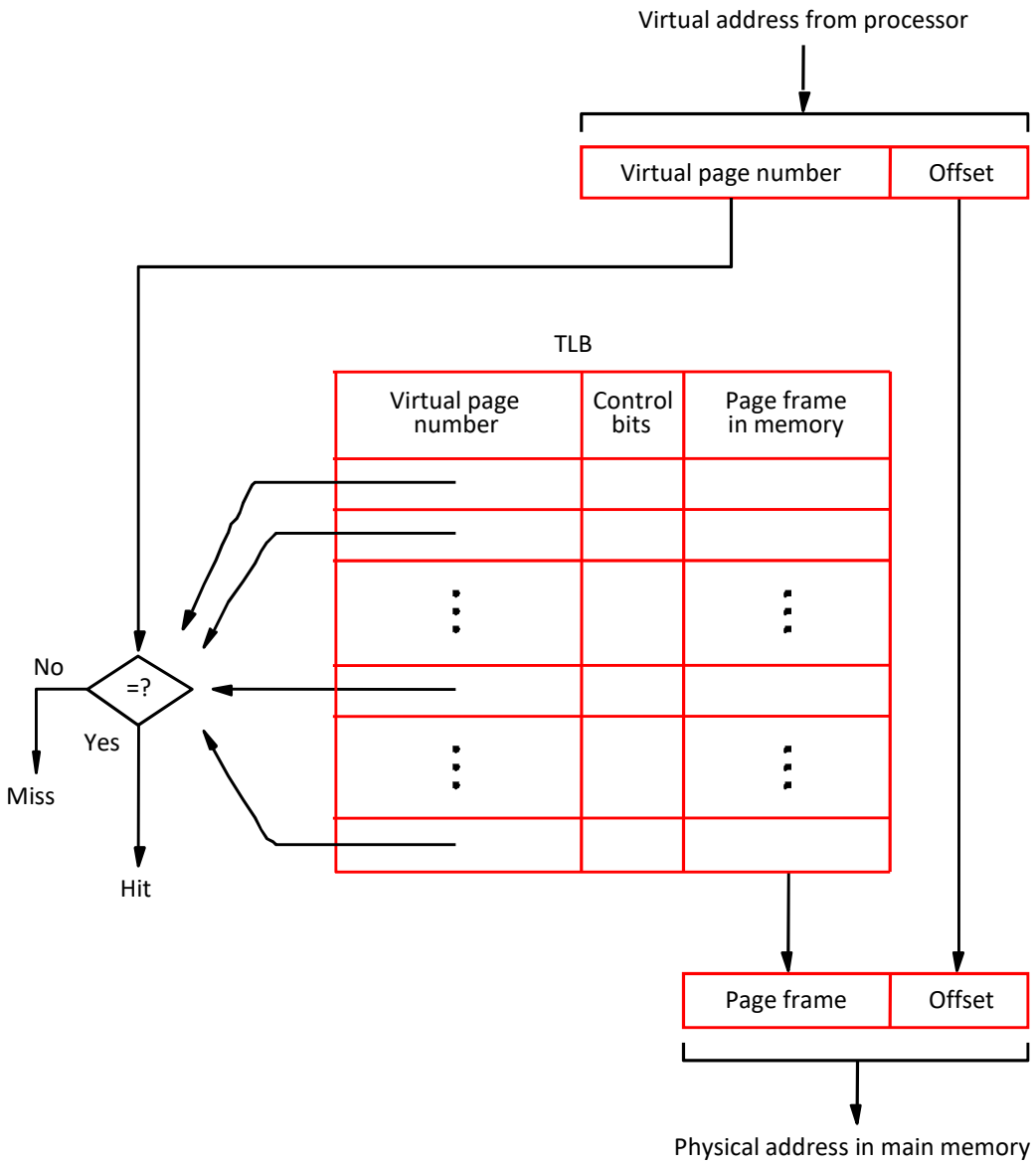
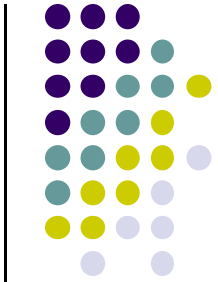


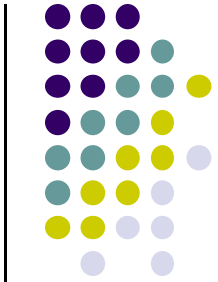
Figure 5.28. Use of an associative-mapped TLB.

TLB



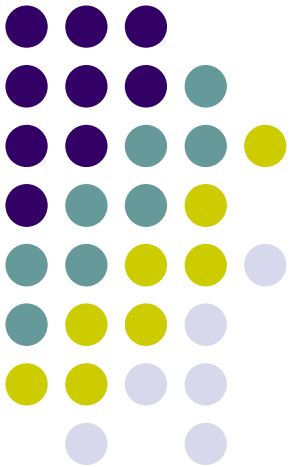
- The contents of TLB must be coherent with the contents of page tables in the memory.
- Translation procedure.
- Page fault
- Page replacement
- Write-through is not suitable for virtual memory.
- Locality of reference in virtual memory

Memory Management Requirements

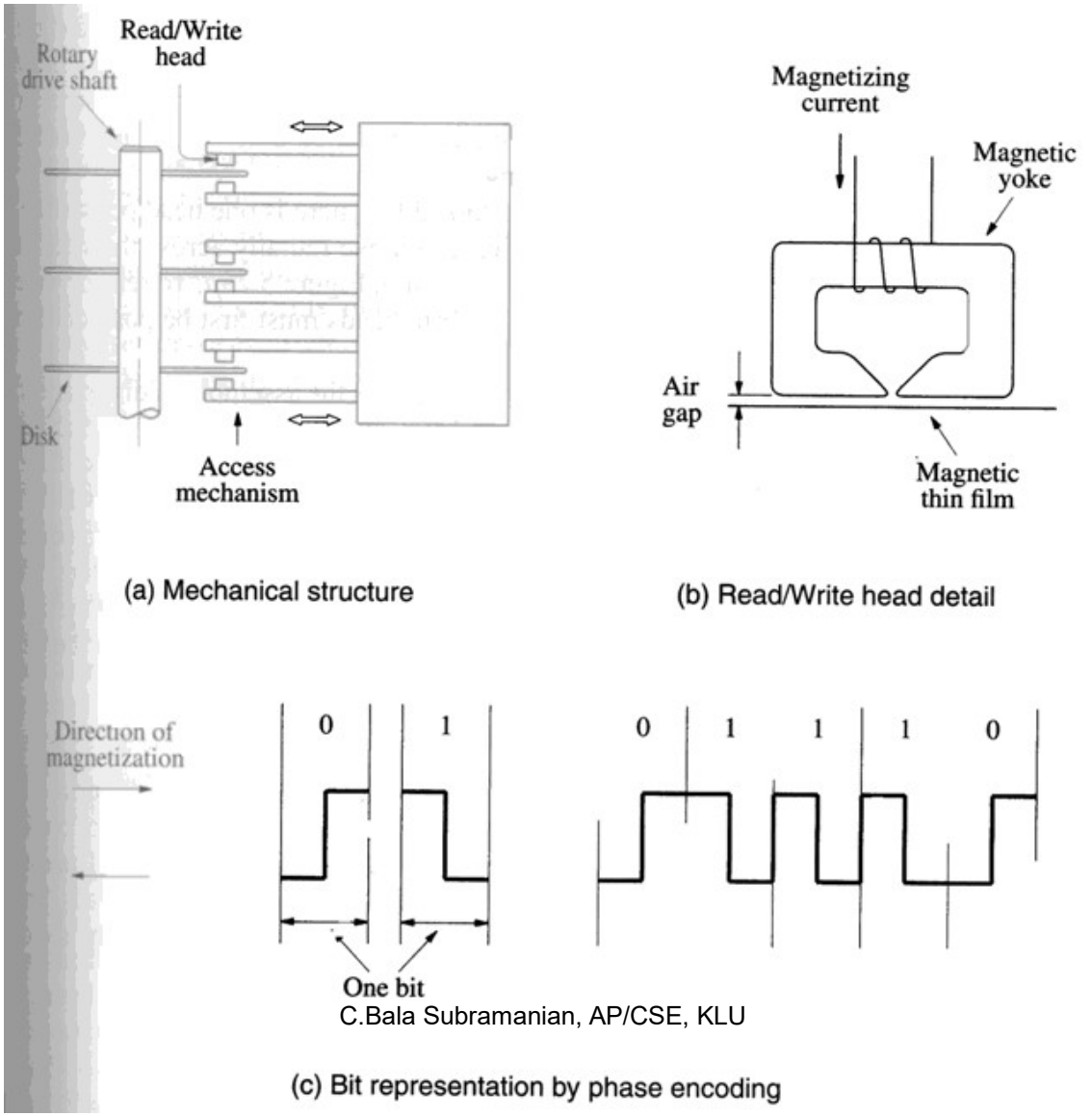


- Multiple programs
- System space / user space
- Protection (supervisor / user state, privileged instructions)
- Shared pages

Secondary Storage



Magnetic Hard Disks



Disk

Disk drive

Disk controller

Organization of Data on a Disk

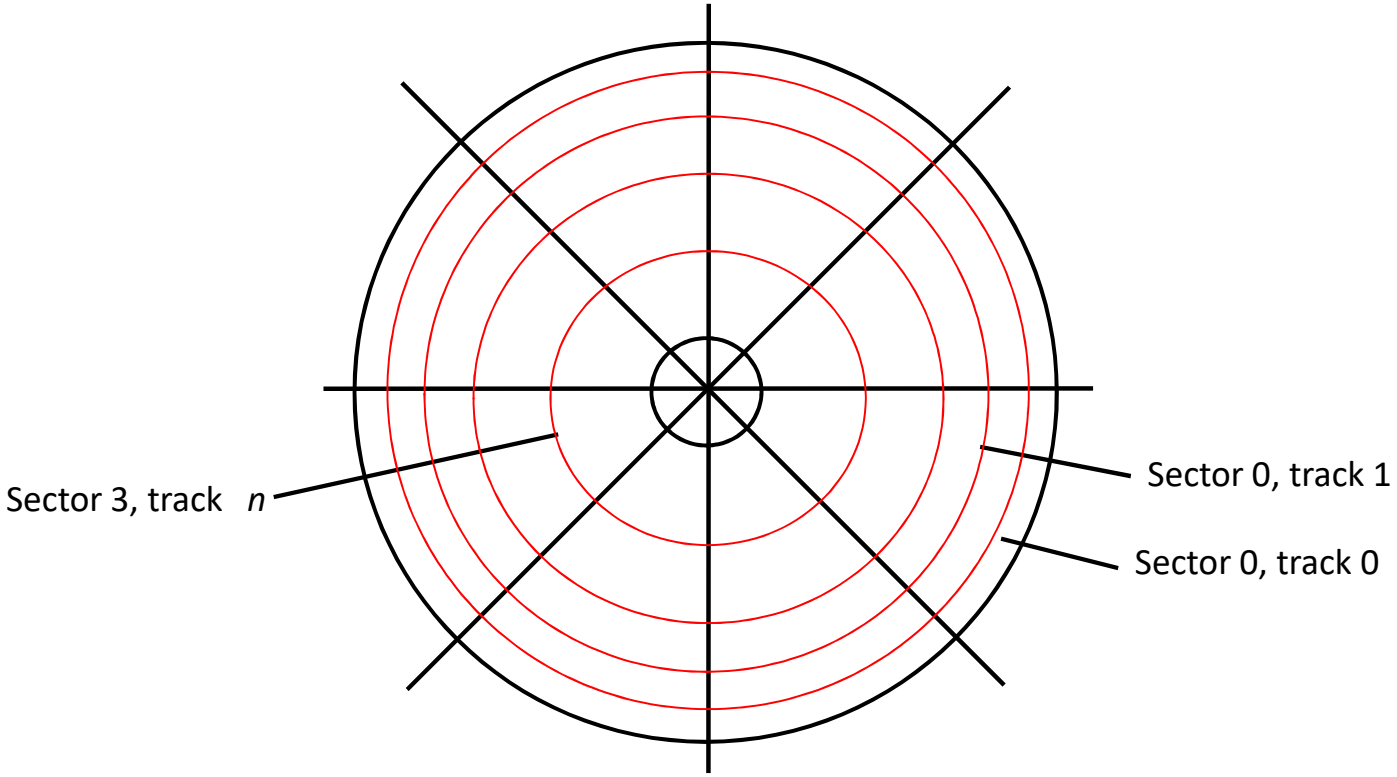
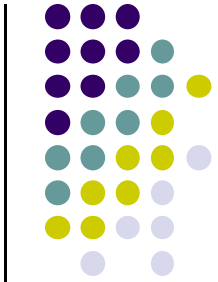


Figure 5.30. Organization of one surface of a disk.

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

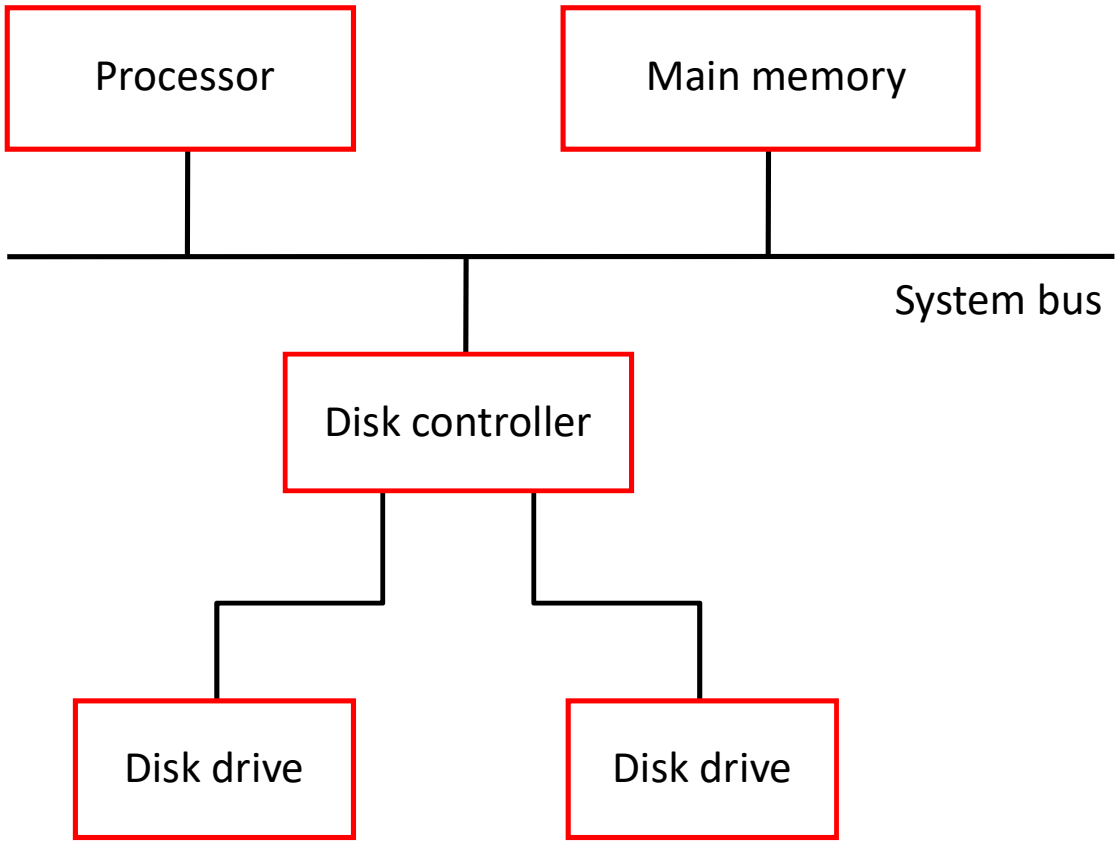
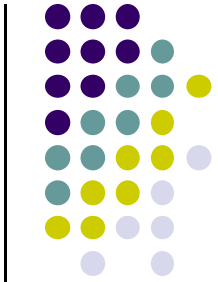


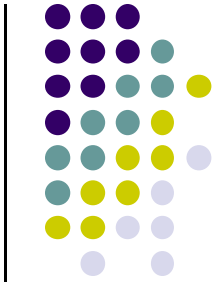
Figure 5.31. Disks connected to the system bus.

Disk Controller



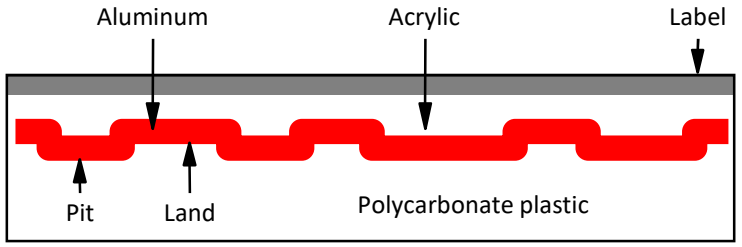
- Seek
- Read
- Write
- Error checking

RAID Disk Arrays

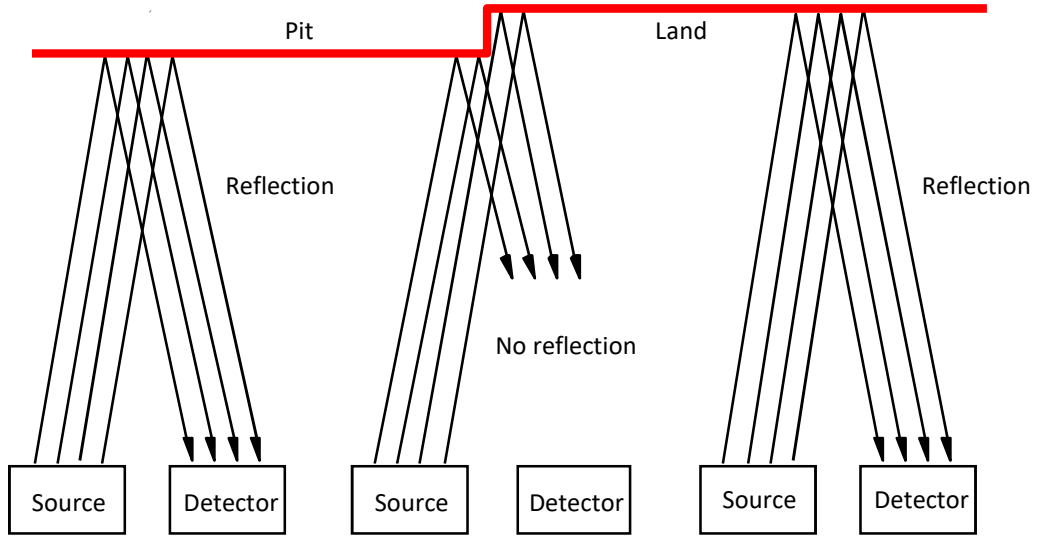


- Redundant Array of Inexpensive Disks
- Using multiple disks makes it cheaper for huge storage, and also possible to improve the reliability of the overall system.
- RAID0 – data striping
- RAID1
- RAID2, 3, 4
- RAID5 – parity-based error-recovery

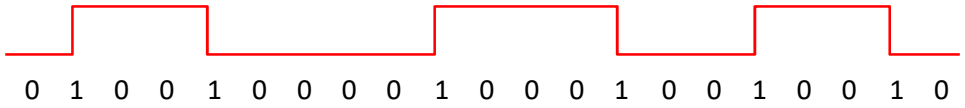
Optical Disks



(a) Cross-section



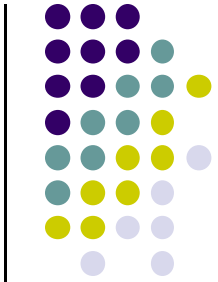
(b) Transition from pit to land



(c) Stored binary pattern

Figure 5.32. Optical disk.

Optical Disks



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

Magnetic Tape Systems

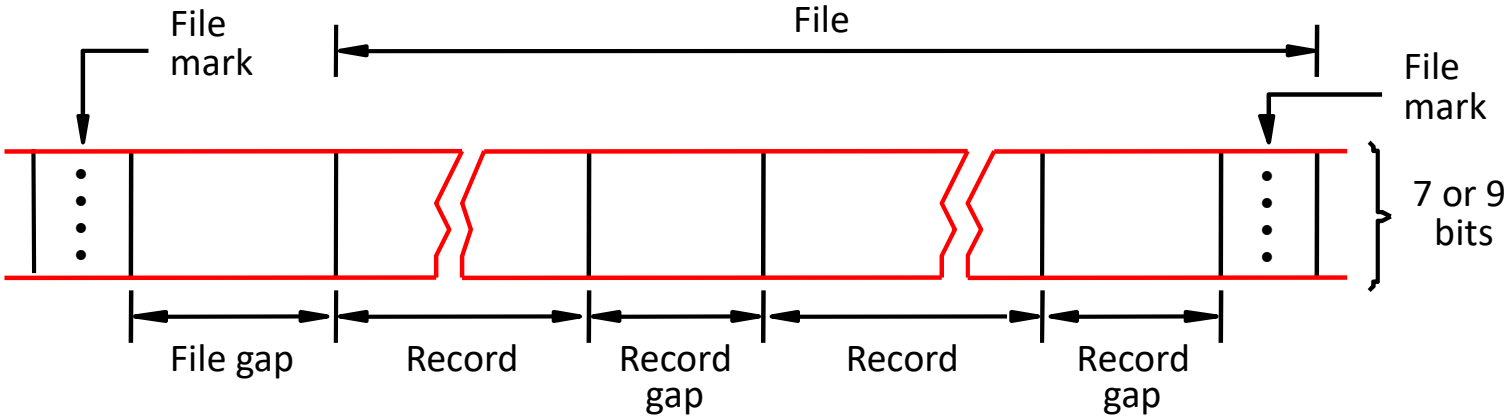


Figure 5.33. Organization of data on magnetic tape.