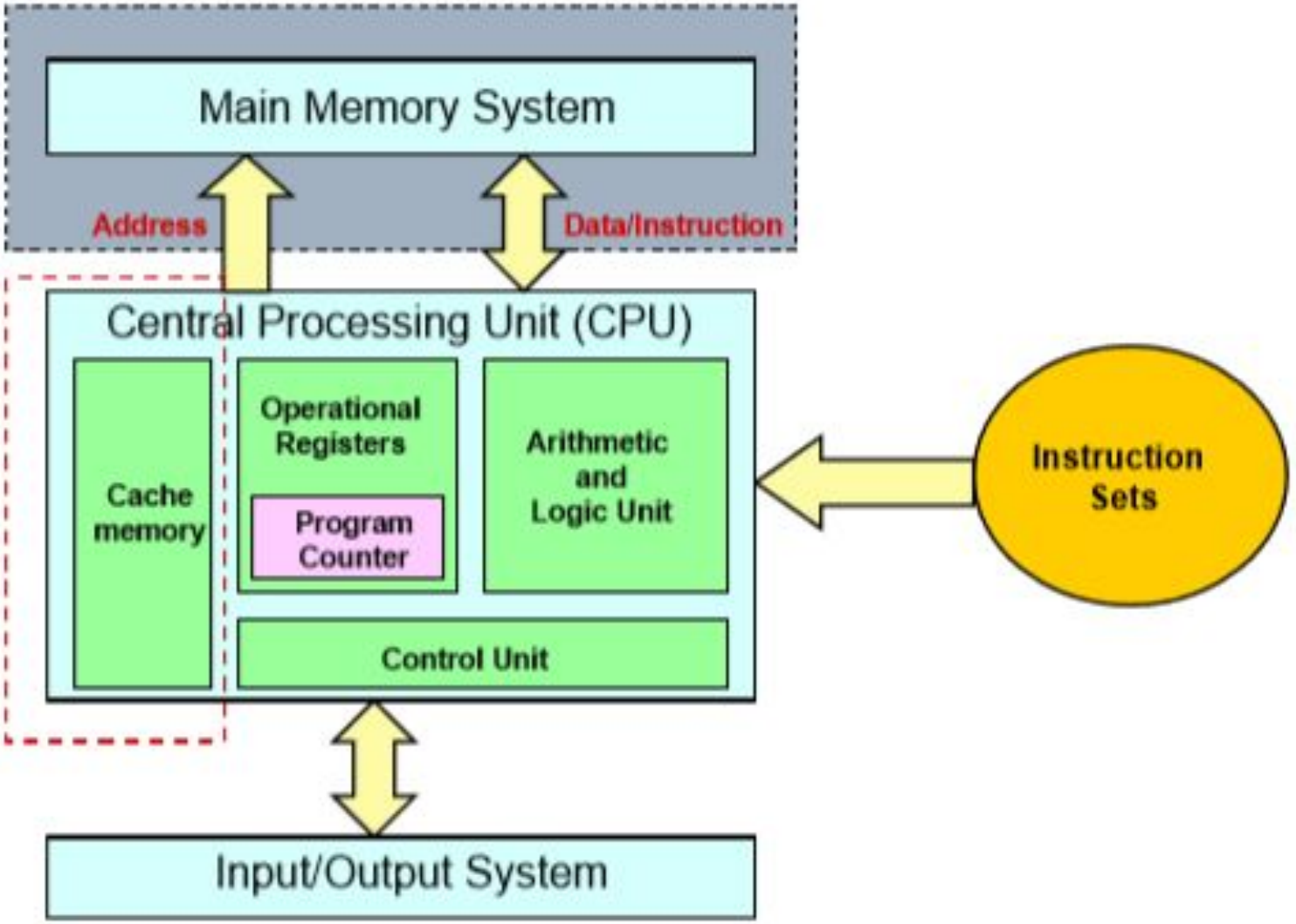
Unit-5

The Memory System

Outline

- Basic Concepts
- Semiconductor Random Access Memories
- Read Only Memories
- Speed, Size, and Cost
- Cache Memories
- Performance Considerations
- Virtual Memories

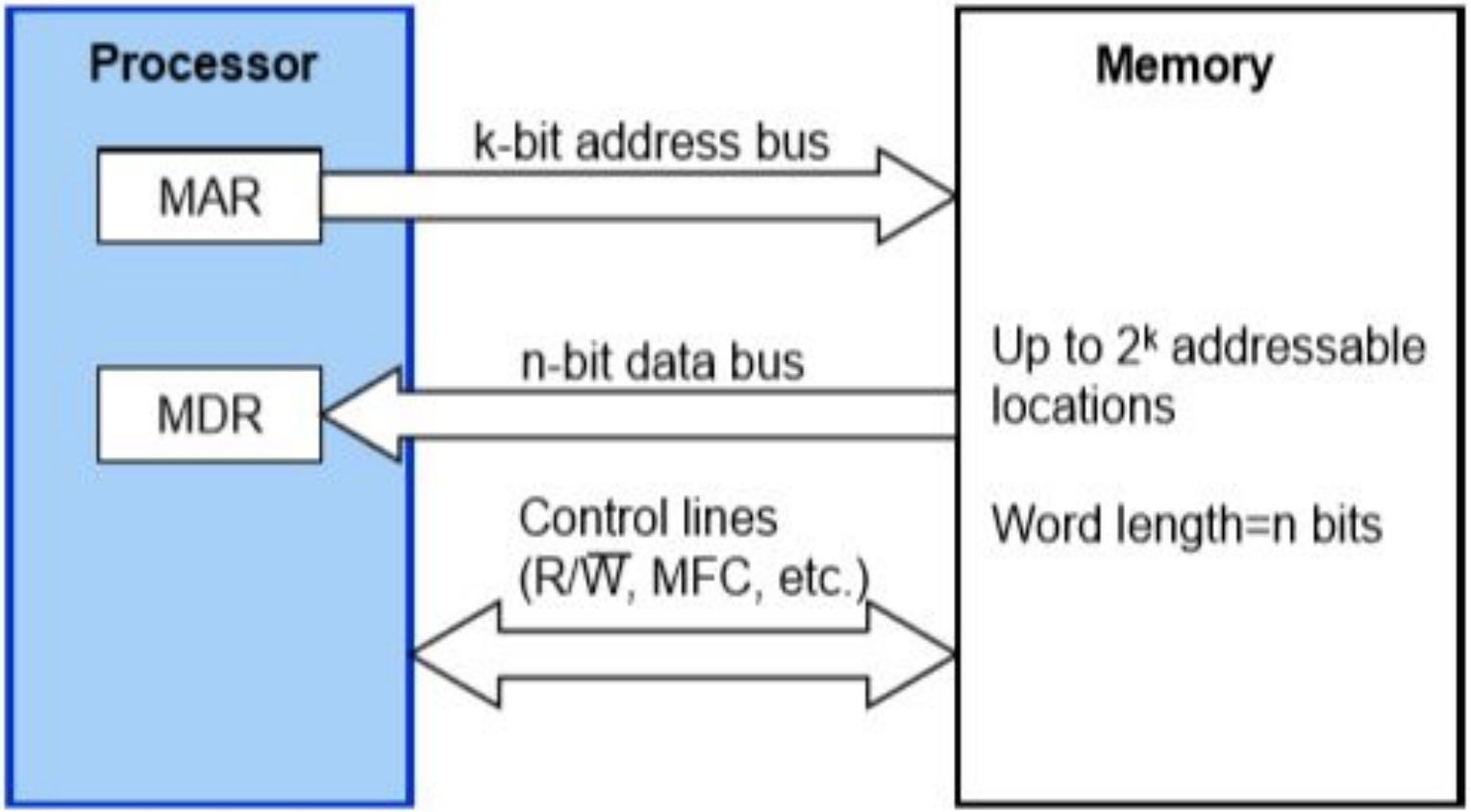
Content Coverage



Basic Concepts

- The maximum size of the memory that can be used in any computer is determined by the addressing scheme
  - For example, a 16-bit computer that generates 16-bit addresses is capable of addressing up to 2^{16}=64K memory locations.
  - Similarly, machines whose instructions generate 32-bit addresses can utilize a memory that contains up to 2^{32}=4G memory locations
- Most modern computers are byte addressable
- Form the system standpoint, we can view the memory unit as a block box
  - Data transfer between the memory and processor takes place through the use of two processor registers, MAR and MDR

Connection of the Memory to the Processor



Basic Concepts

- A useful measure of the speed of memory units is the time that elapses between the initiation of an operation and the completion of that operation. This is referred to as the memory access time.
- Another important measure is the memory cycle time, which is the minimum time delay required between the initiation of two successive memory operations
- A memory unit is called random-access memory (RAM) if 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
- The memory cycle time is the bottleneck in the system

Basic Concepts

- One way to reduce the memory access time is to use a cache memory
- Cache memory is a small, fast memory that is inserted between the larger, smaller main memory and the processor.
- Virtual memory is used to increase the apparent size of the physical memory. Data are addressed in a virtual address space that can be as large as the addressing capability of the processor. But at any given time, only the active portion of this space is mapped onto locations in the physical memory. The remaining virtual addresses are mapped onto the bulk storage devices used, such as magnetic disks

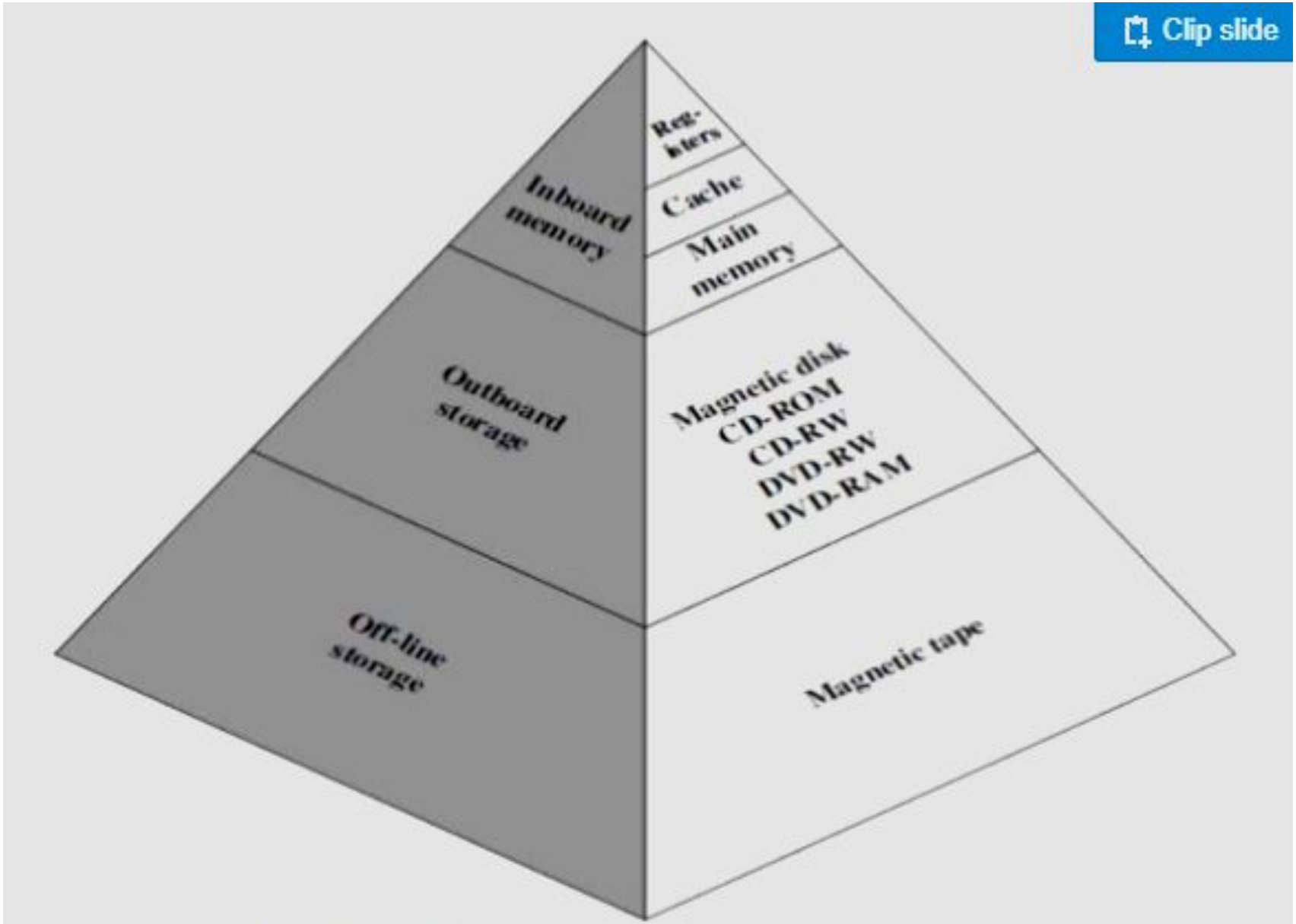
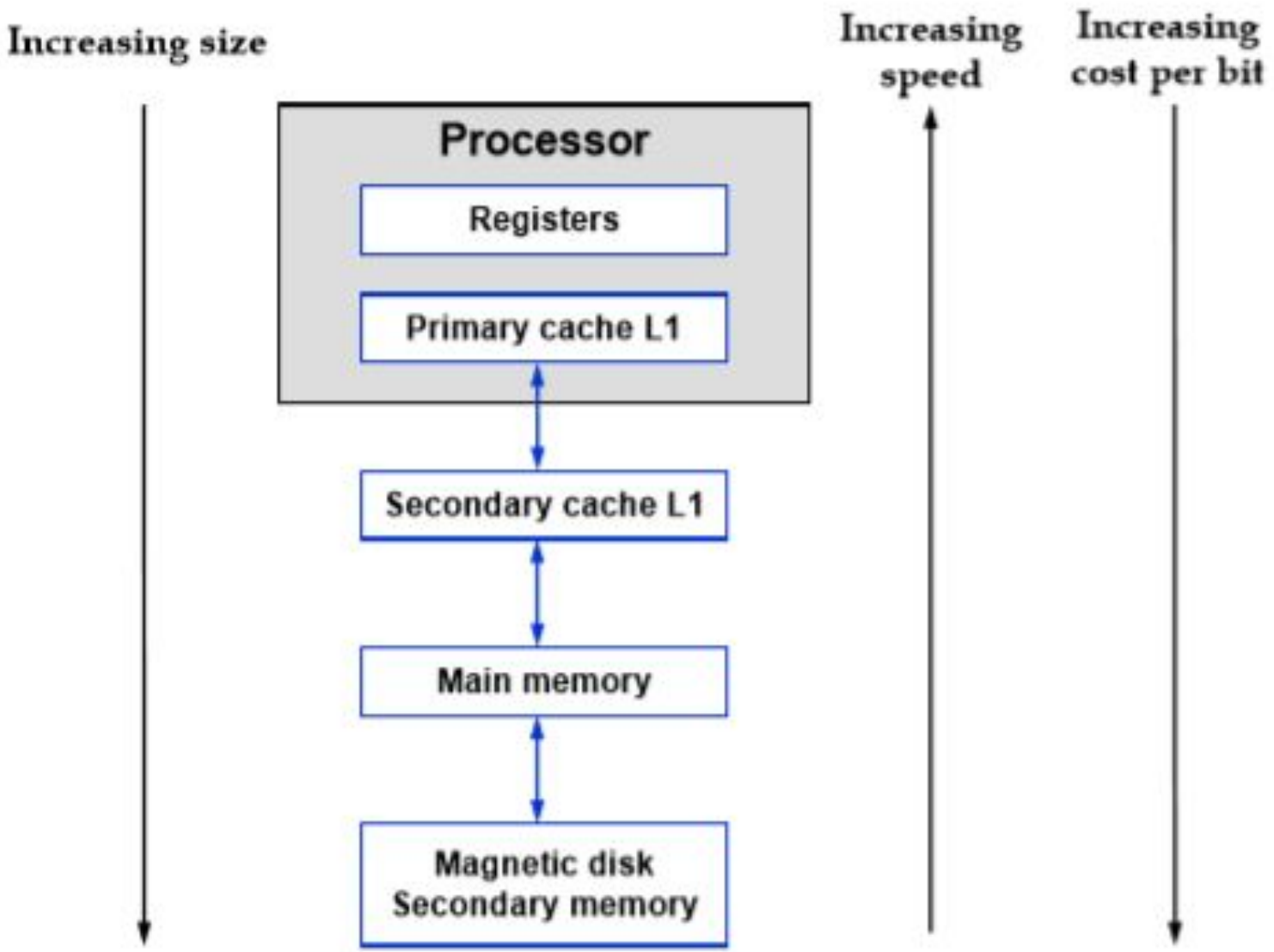
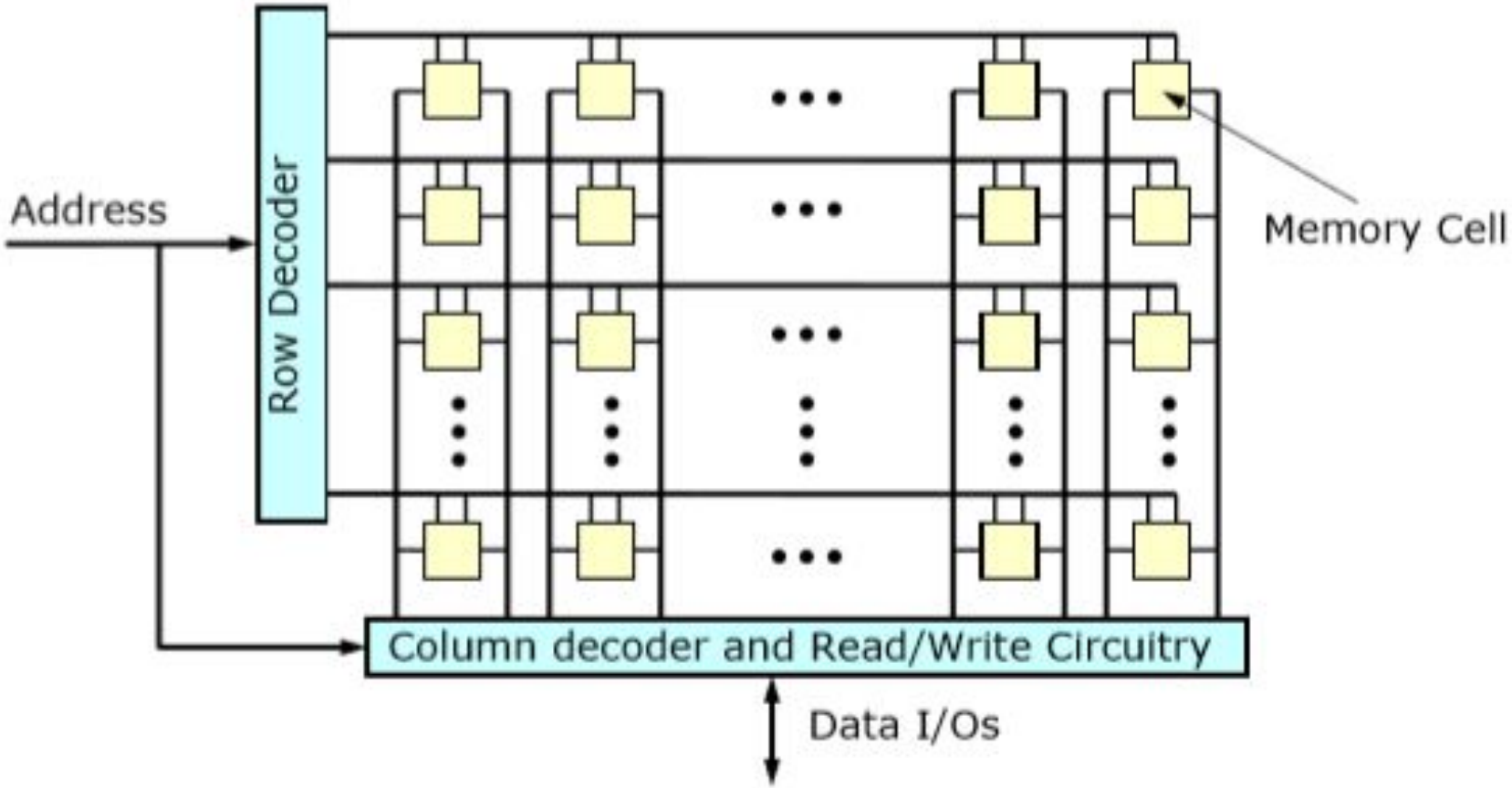


Figure 4.1 The Memory Hierarchy

Memory Hierarchy

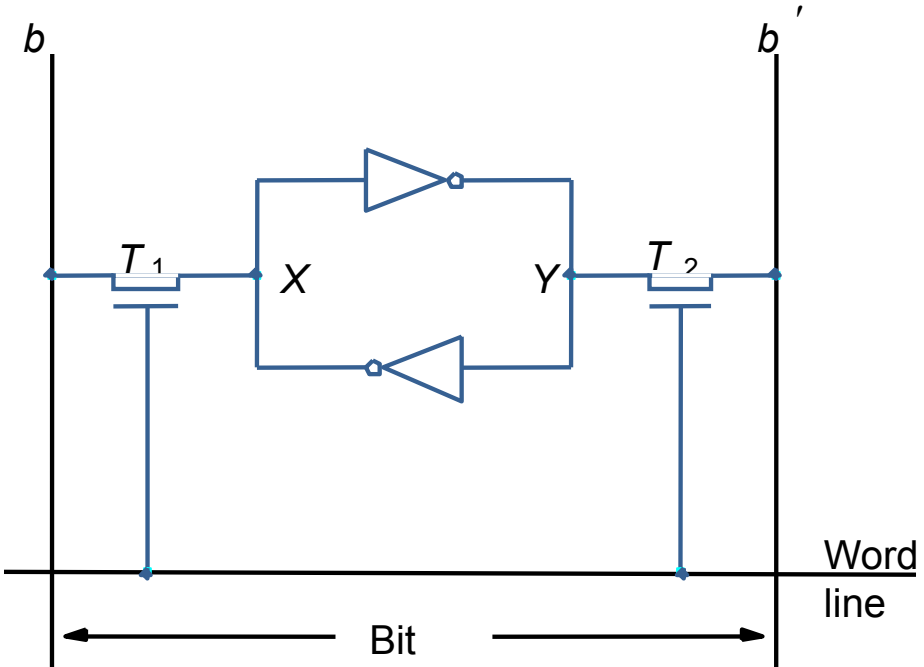


Organization of Bit Cells in a Memory

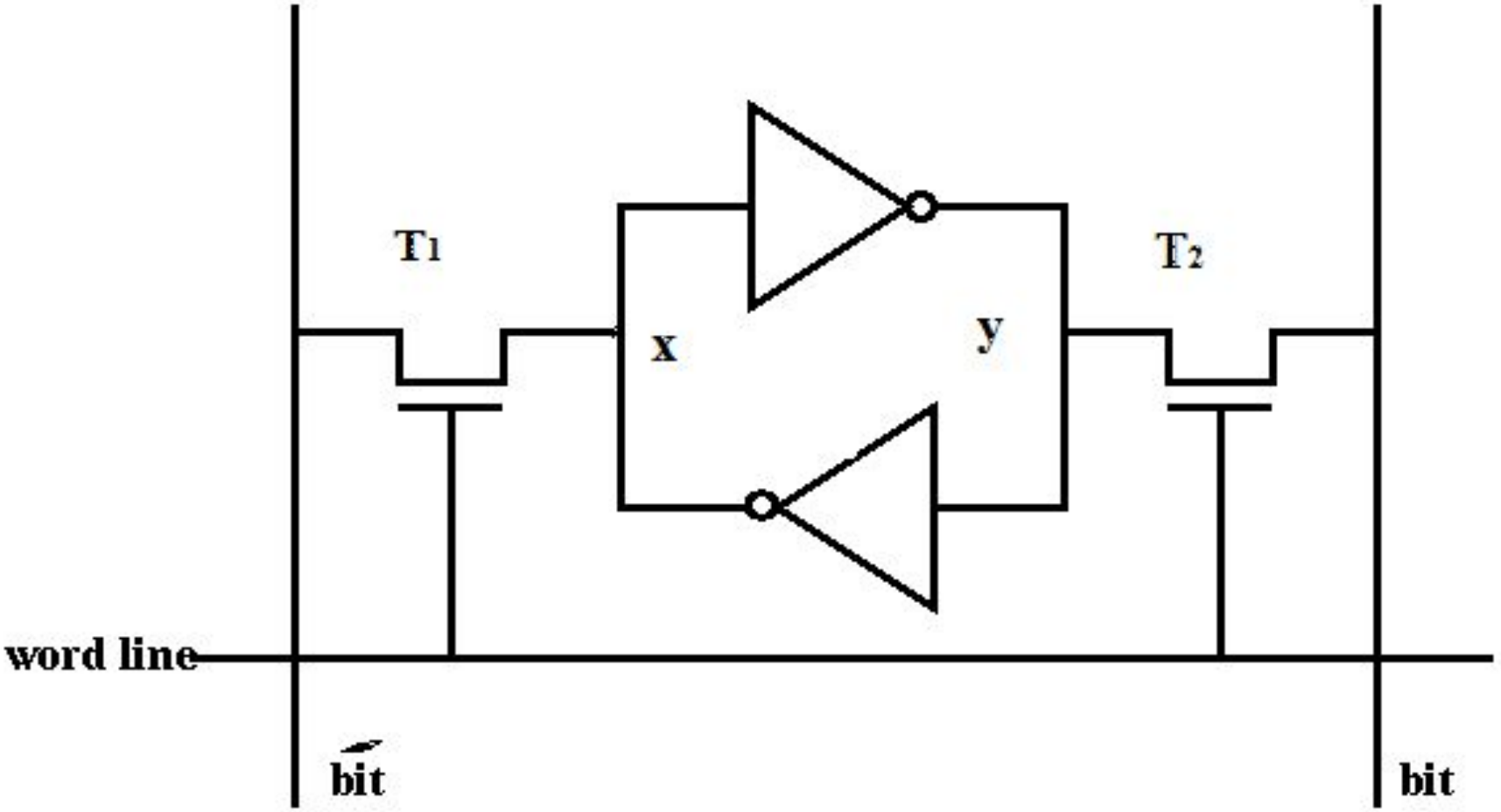


SRAM Cell

- Two transistor inverters are cross connected to implement a basic flip-flop.
- The cell is connected to one word line and two bits lines by transistors T1 and T2
- When word line is at ground level, the transistors are turned off and the latch retains its state
- Read operation: In order to read state of SRAM cell, the word line is activated to close switches T1 and T2. Sense/Write circuits at the bottom monitor the state of b and b'

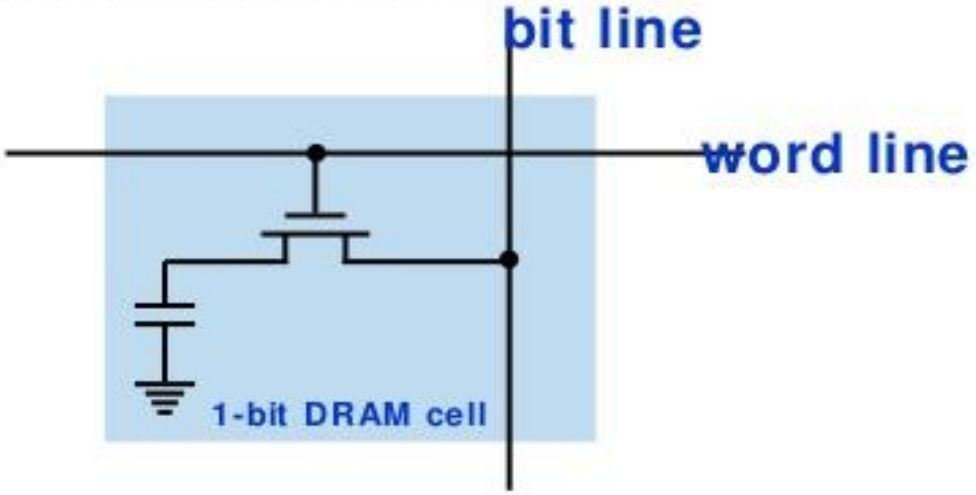


SRAM Cell

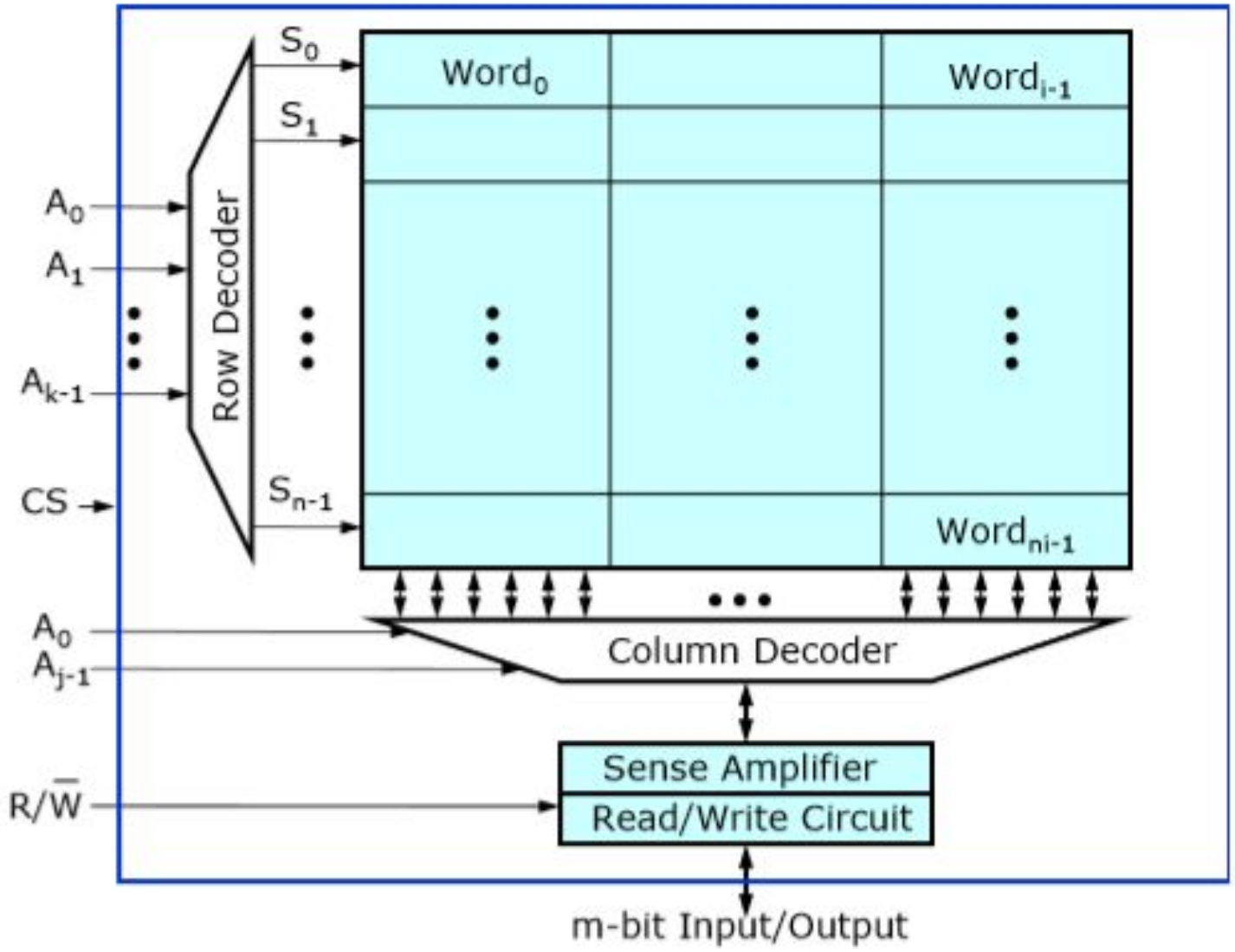


DRAM

- DRAM: Dynamic RAM
  - Uses MOS transistor and capacitor to store bit
  - More compact than SRAM
  - "Refresh" required due to capacitor leak
  - Typical refresh rate 15.625 microsecond
  - Slower to access than SRAM

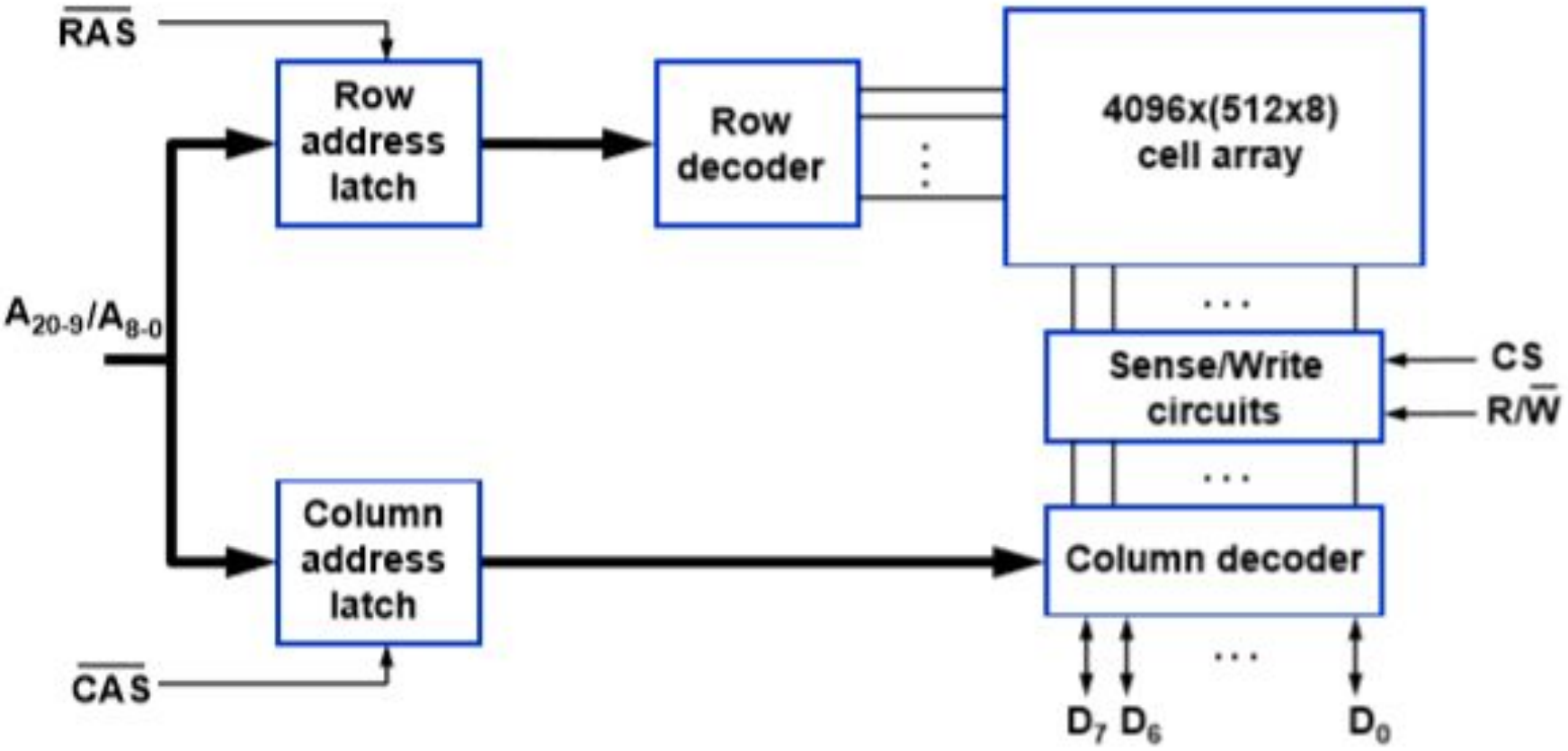


Semiconductor Random Access memories



Asynchronous DRAMs

- A 16M-bit DRAM chip, configured as 2Mx8, is shown below

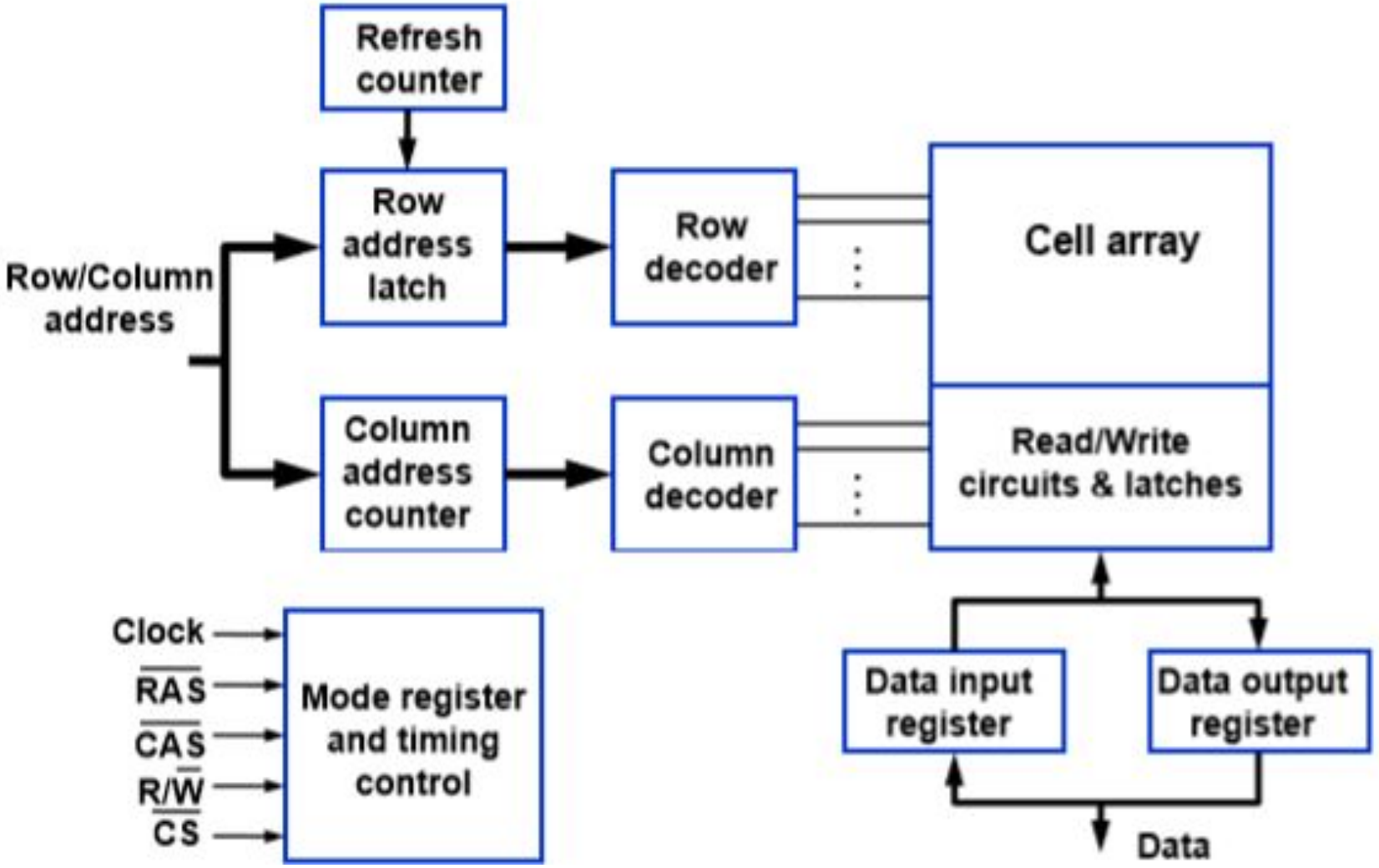


Fast Page Mode

- When the DRAM shown above is accessed, the contents of all 4096 cells in the selected row are sensed, but only 8 bits are placed on the data lines D_{7-0}. This byte is selected by the column address bits A_{8-0}.
- A simple modification can make it possible to access the other bytes in the same row without having to reselect the row
  - A latch can be added at the output of the sense amplifier in each column
  - Transfer of successive bytes is achieved by applying a consecutive sequence of column address under the control of successive CAS signals
- This block transfer capability is referred to as the *fast page mode* feature

Synchronous DRAMs

- The structure of an synchronous DRAM (SDRAM)



Latency

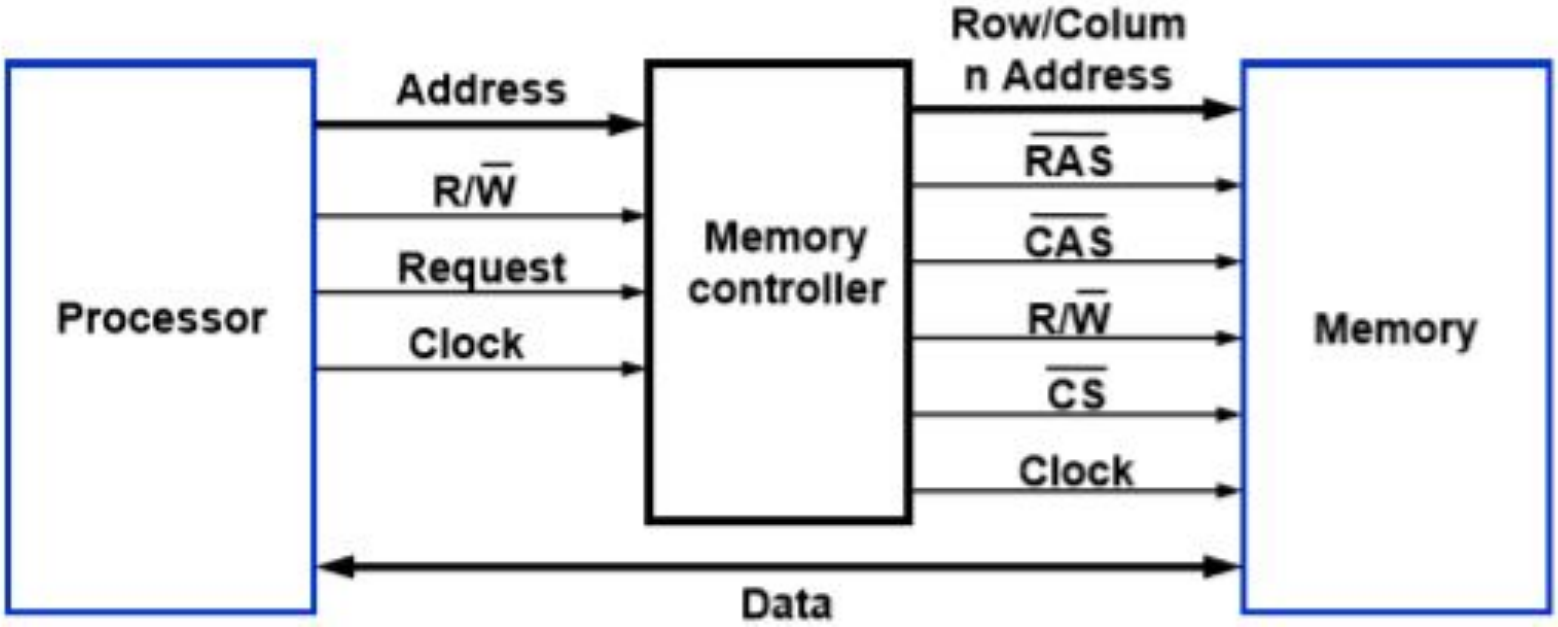
- Transfers between the memory and the processor involve single words of data or small blocks of words
- The speed and efficiency of these transfers have a large impact on the performance of a computer system
- A good indication of the performance is given by two parameters: latency and bandwidth
- The term memory latency is used to refer to the amount of time it takes to transfer a word of data to or from the memory
- In block transfers, the term latency is used to denote the time it takes to transfer first word of data

Bandwidth

- When transferring blocks of data, it is of interest to know how much time is needed to transfer an entire block
- Since blocks can be variable in size, it is useful to define a performance measure in terms of the number of bits or bytes that can be transferred in one second. This measure is often referred to as the memory bandwidth
- The bandwidth of a memory unit depends on the speed of access to the stored data and on the number of bits that can be accessed in parallel

Memory System Considerations

- Memory controller
  - To reduce the number of pins, the DRAM chips use multiplexed address inputs
  - A typical processor issues all bits of an address at the same time
  - The required multiplexing of address bits is usually performed by a memory controller circuit, which is interposed between the processor and the DRAM as shown below

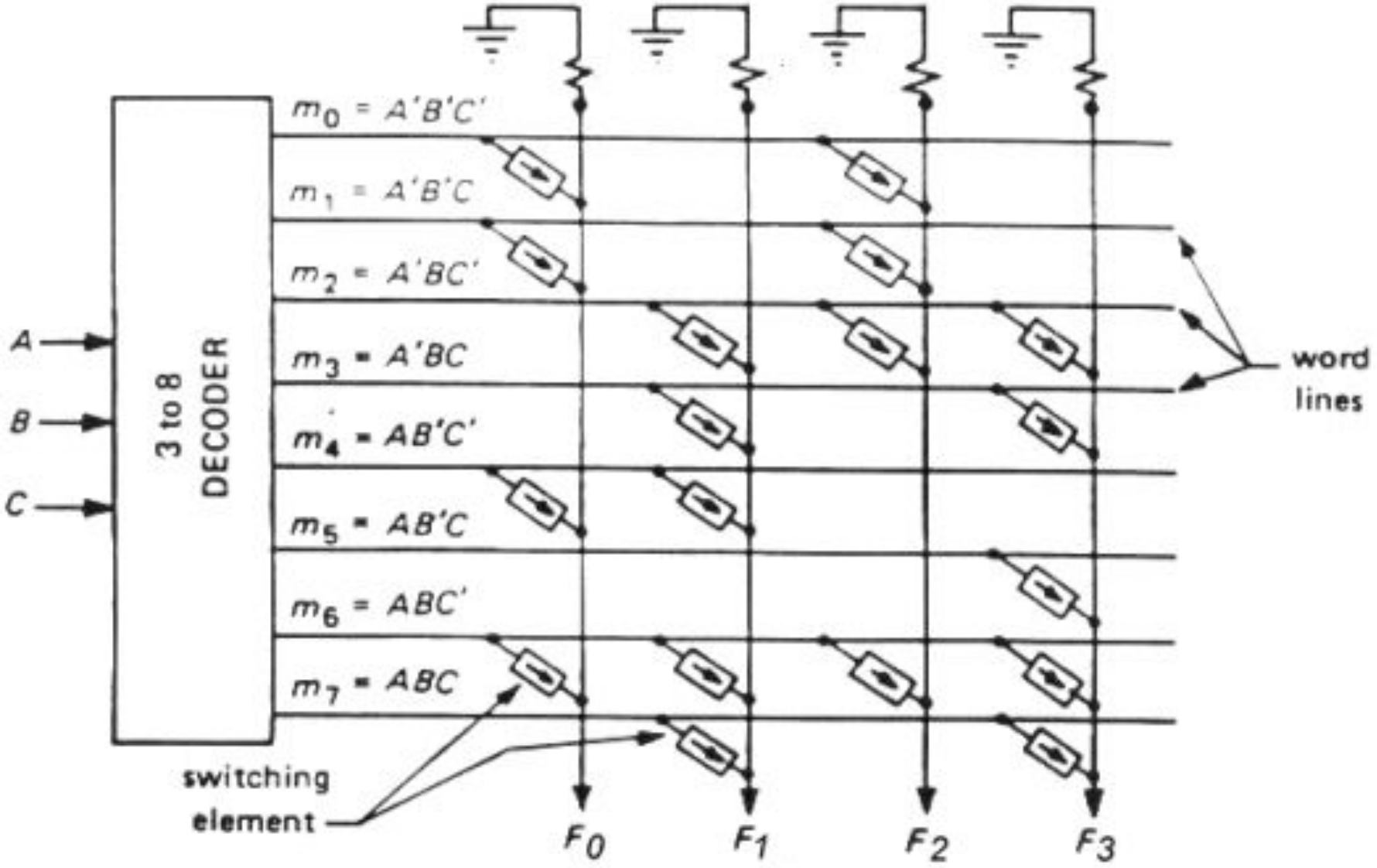


RAM vs ROM

Random Access Memory	**Read Only Memory**
Read and Write	Read Only
Data can be changed	Data **CANNOT** be changed
Data is lost when power is switched off - **Temporary**	Data is kept when power is switched off - **Permanent**
Holds all the data and programs currently in use	Holds instructions the computer and devices need to carry out their function

Read-Only Memories (ROMs)

- SRAM and SDRAM chips are volatile:
  - Lose the contents when the power is turned off.
- Many applications need memory devices to retain contents after the power is turned off.
  - For example, computer is turned on, the operating system must be loaded from the disk into the memory.
  - Store instructions which would load the OS from the disk.
  - Need to store these instructions so that they will not be lost after the power is turned off.
  - We need to store the instructions into a non-volatile memory.
- Non-volatile memory is read in the same manner as volatile memory.
  - Separate writing process is needed to place information in this memory.
  - Normal operation involves only reading of data, this type of memory is called Read-Only memory (ROM).


A
B
C
3 to 8
DECODER
$m_0 = A'B'C'$
$m_1 = A'B'C$
$m_2 = A'BC'$
$m_3 = A'BC$
$m_4 = AB'C'$
$m_5 = AB'C$
$m_6 = ABC'$
$m_7 = ABC$
word
lines
switching
element
F_0
F_1
F_2
F_3

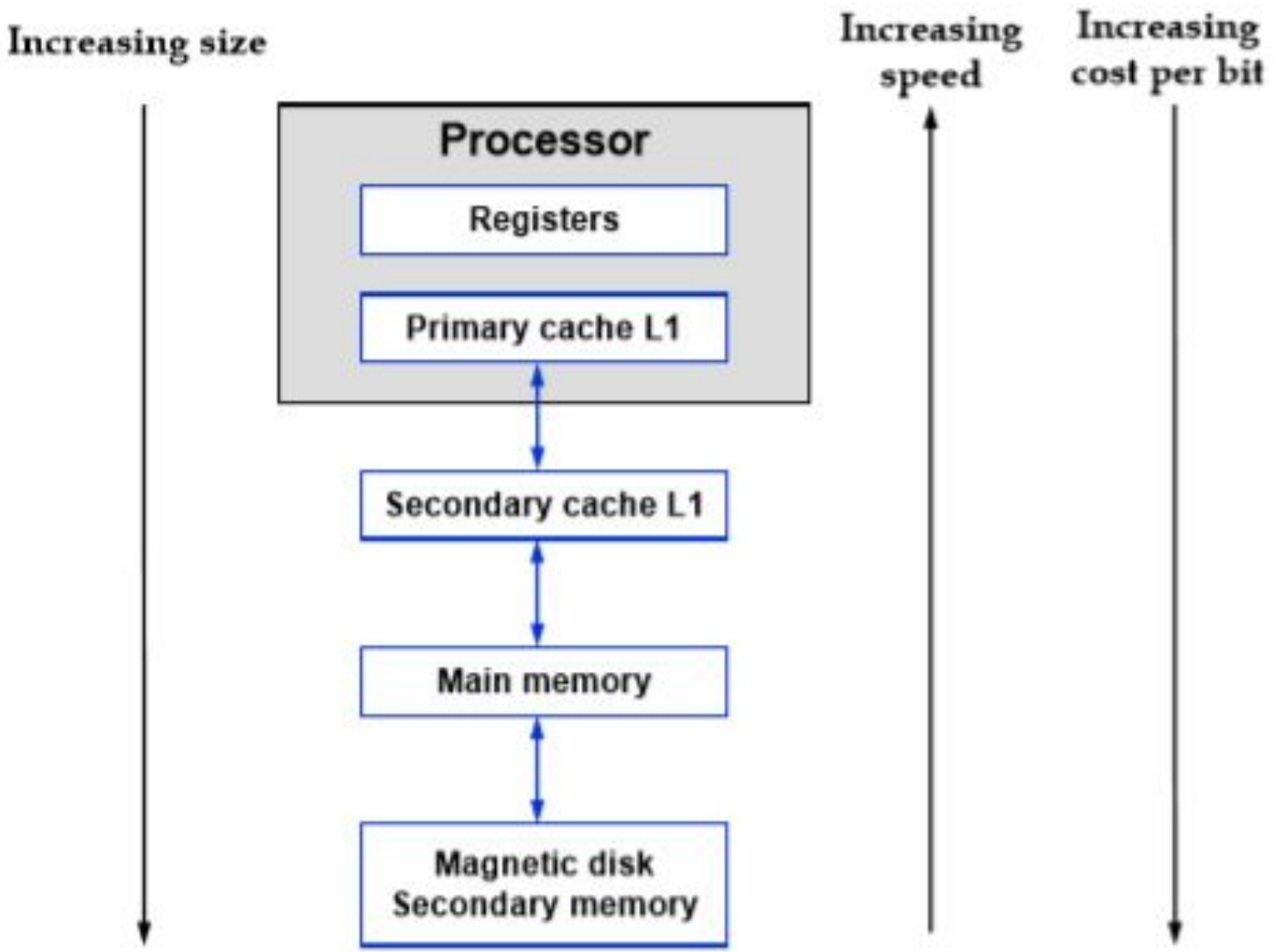
Read-Only Memories (Contd.,)

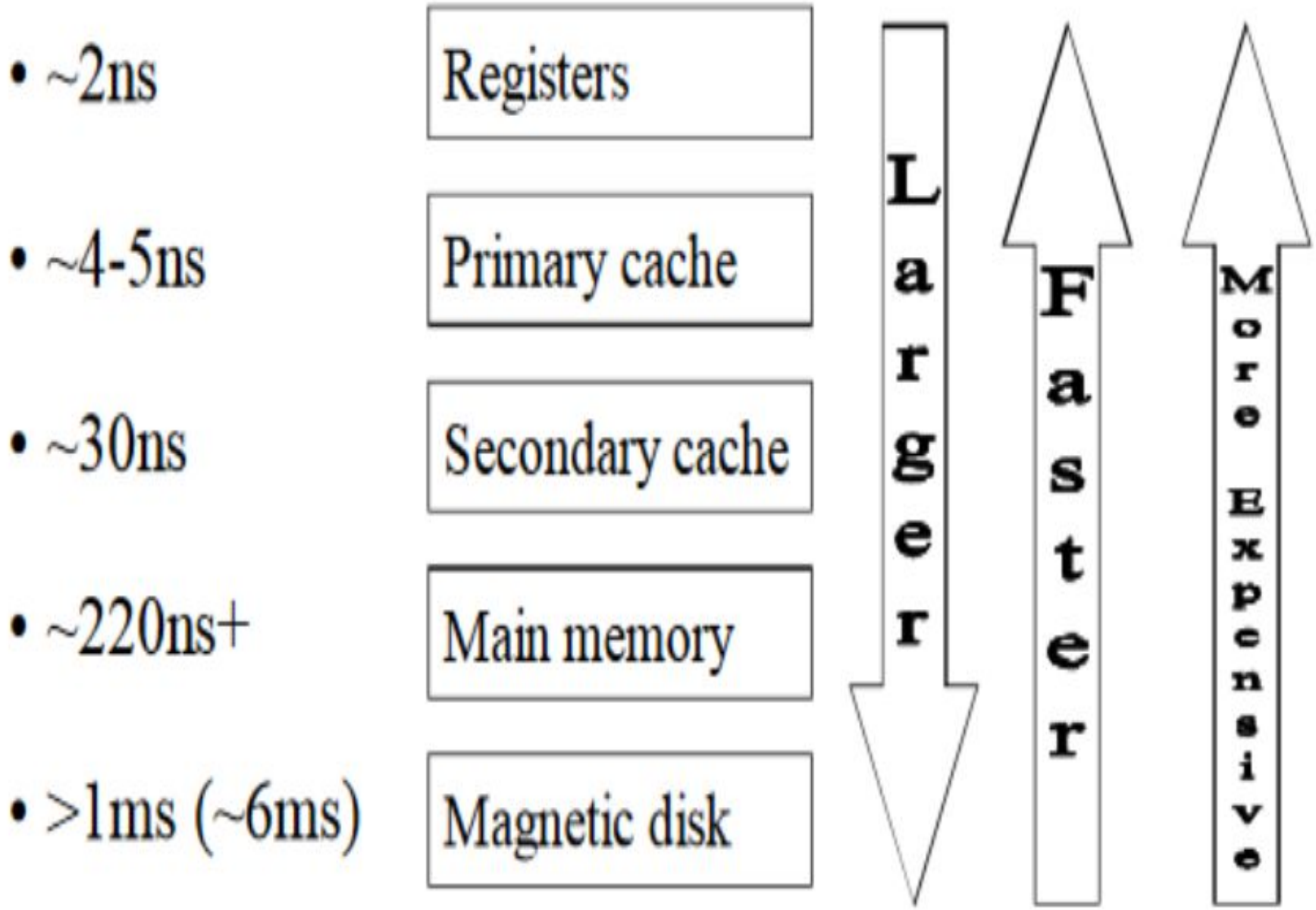
- **Basic Read-Only Memory:**
  - Data are written into a ROM when it is manufactured.
- **Programmable Read-Only Memory (PROM):**
  - Allow the data to be loaded by a user.
  - Process of inserting the data is irreversible.
  - Storing information specific to a user in a ROM is expensive.
  - Providing programming capability to a user may be better.
- **Erasable Programmable Read-Only Memory (EPROM):**
  - Stored data to be erased and new data to be loaded.
  - Flexibility, useful during the development phase of digital systems.
  - Erasable, reprogrammable ROM.
  - Erasure requires exposing the ROM to UV light.

Read-Only Memories (Contd.,)

- Electrically Erasable Programmable Read-Only Memory (EEPROM):
  - To erase the contents of EPROMs, they have to be exposed to ultraviolet light.
  - Physically removed from the circuit.
  - EEPROMs the contents can be stored and erased electrically.
- Flash memory:
  - Has similar approach to EEPROM.
  - Read the contents of a single cell, but write the contents of an entire block of cells.
  - Flash devices have greater density.
    - Higher capacity and low storage cost per bit.
  - Power consumption of flash memory is very low, making it attractive for use in equipment that is battery-driven.
  - Single flash chips are not sufficiently large, so larger memory modules are implemented using flash cards and flash drives.

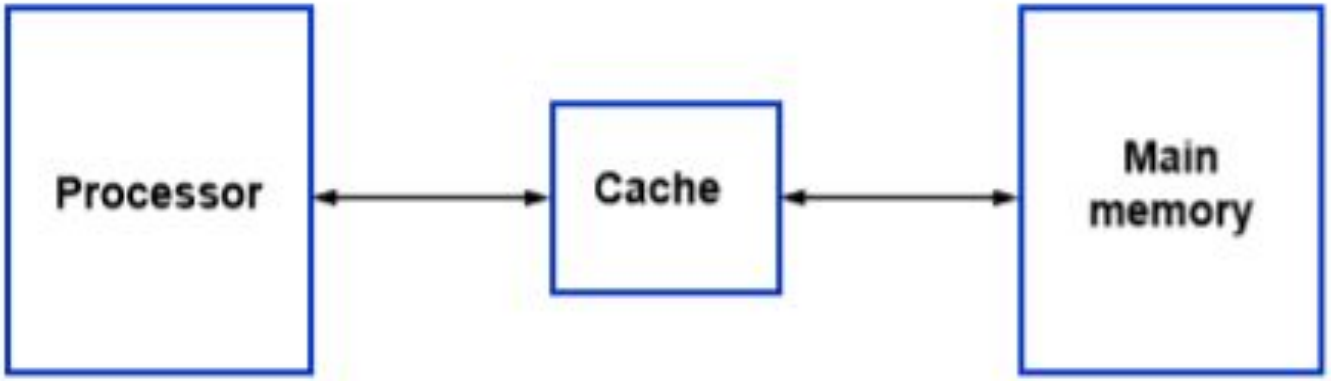
Memory Hierarchy



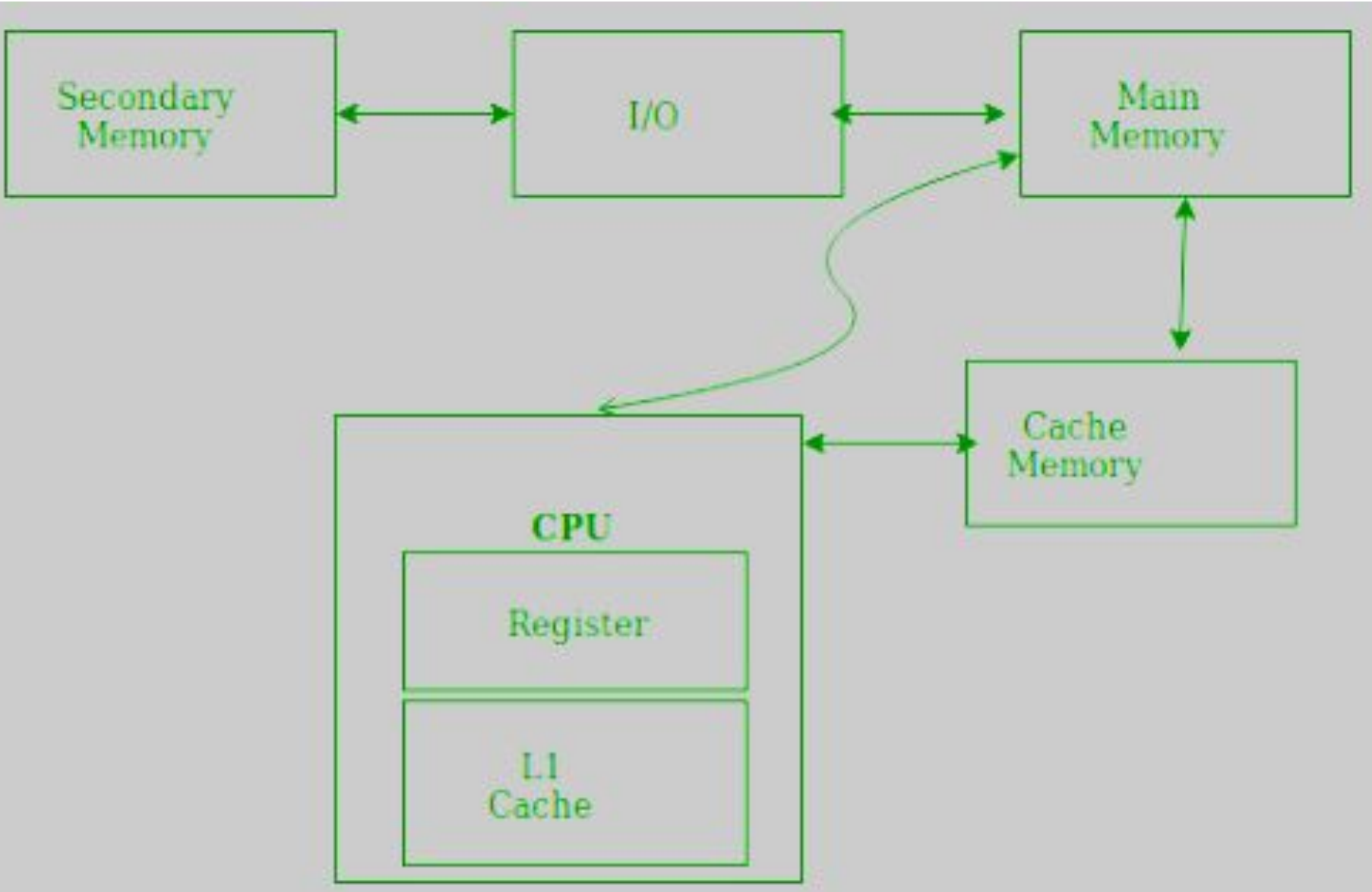

• ~2ns
Registers
• ~4-5ns
Primary cache
• ~30ns
Secondary cache
• ~220ns+
Main memory
• >1ms (~6ms)
Magnetic disk
Larger
Faster
More Expensive

Use of a Cache Memory

- Consider the simple arrangement shown below

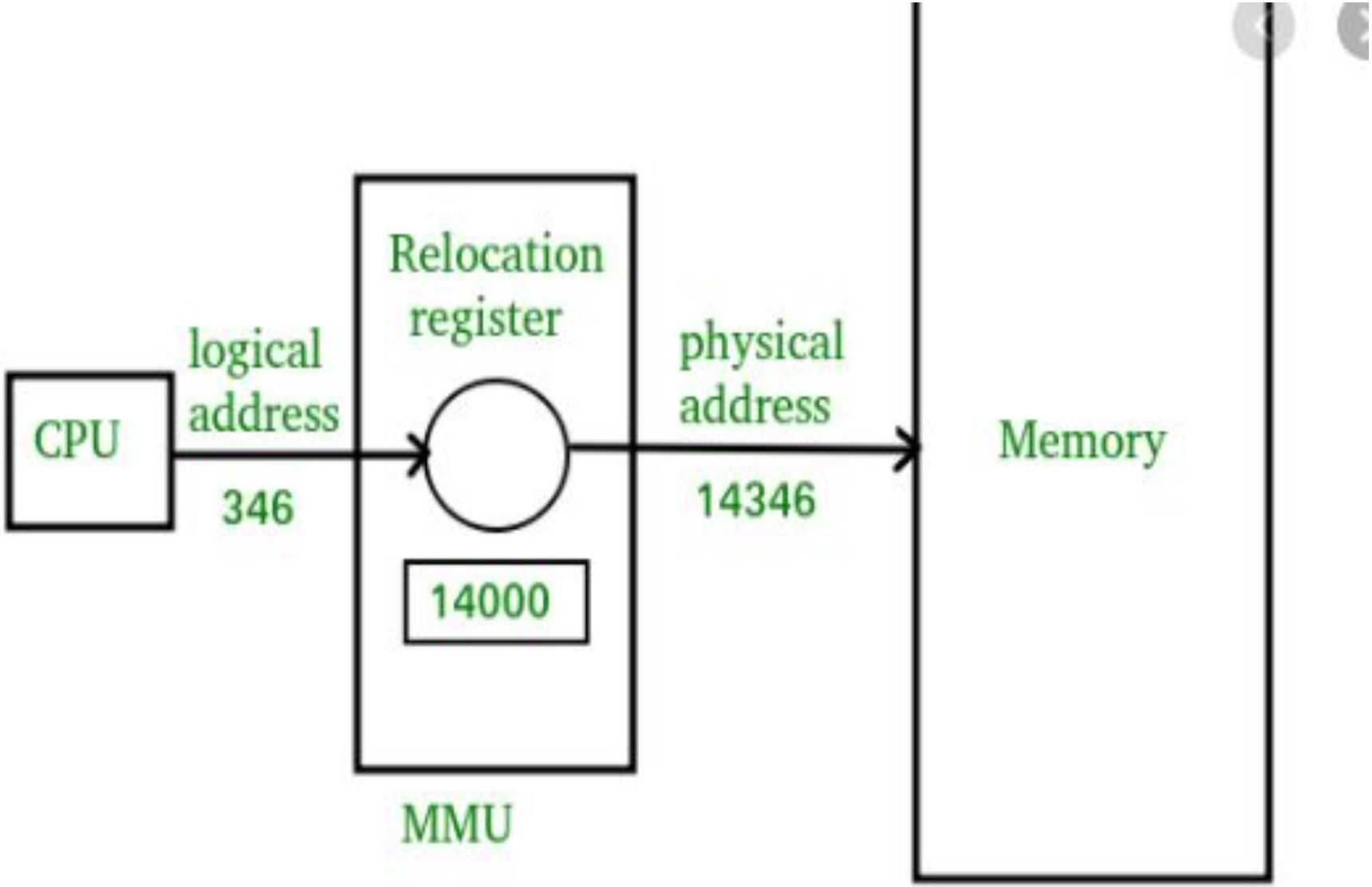


- Usually, the cache memory can store a reasonable number of blocks at any given time, but this number is small compared to the total number of blocks in the main memory
- The correspondence between the main memory blocks and those in the cache is specified by a mapping function


Secondary Memory
I/O
Main Memory
Cache Memory
CPU
Register
L1 Cache

Logical vs. Physical Address Space

- Difference between logical and physical addresses
  - **Logical address** – generated by the CPU; also referred to as *virtual address*
  - **Physical address** – address seen by the memory unit
- Logical and physical addresses are the same in compile-time and load-time address-binding schemes; logical (virtual) and physical addresses differ in execution-time address-binding scheme
- One process cannot access other process's physical address space (unless it's a shared memory)

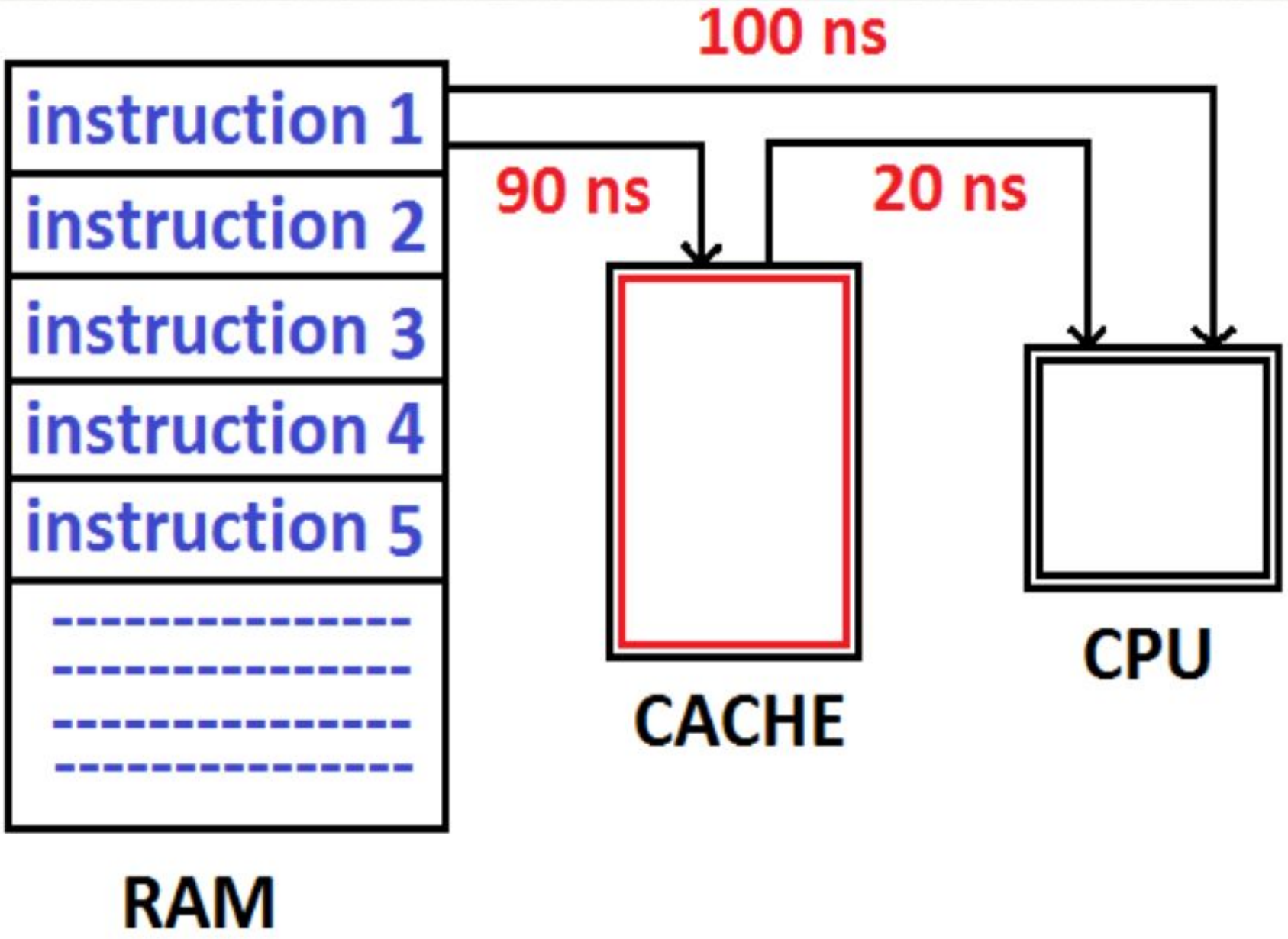

Relocation
register
logical
address
CPU
346
physical
address
14346
Memory
14000
MMU

Cache Memories

- The speed of the main memory is very low in comparison with the speed of modern processors
- Hence, it is important to devise a scheme that reduces the time needed to access the necessary information
- Since the speed of main memory unit is limited by electronic and packaging constraints, the solution must be sought in a different architectural arrangement
- An efficient solution is to use a fast cache memory which essentially makes the main memory appear to the processor to be faster than it really is

Effectiveness of the Cache Memory

- The effectiveness of the cache mechanism is based on a property of computer programs called locality of reference
- Analysis of programs shows that most of their execution time is spent on routines in which many instructions are executed repeatedly. These instructions may constitute a simple loop, nested loops, or a few procedures that repeatedly call each other
- Memory instructions in localized areas of the program are executed repeatedly during some time period, and the remainder of the program is accessed relatively infrequently. This is referred to as *locality of reference*


100 ns
instruction 1
instruction 2
instruction 3
instruction 4
instruction 5
90 ns
20 ns
CACHE
CPU
RAM

Temporal and Spatial Localities

- The temporal aspect of the locality of reference suggests that whenever an information item (instruction or data) is first needed, this item should be brought into the cache where it will hopefully remain until it is needed again
- The spatial aspect of the locality of reference suggests that instead of fetching just one item from the main memory to the cache, it is useful to fetch several items that reside at adjacent addresses as well. We will use the term *block* to refer to a set of continuous address locations of some size. Another item that is often used to refer to a cache block is *cache line*

Cache Replacement Algorithm

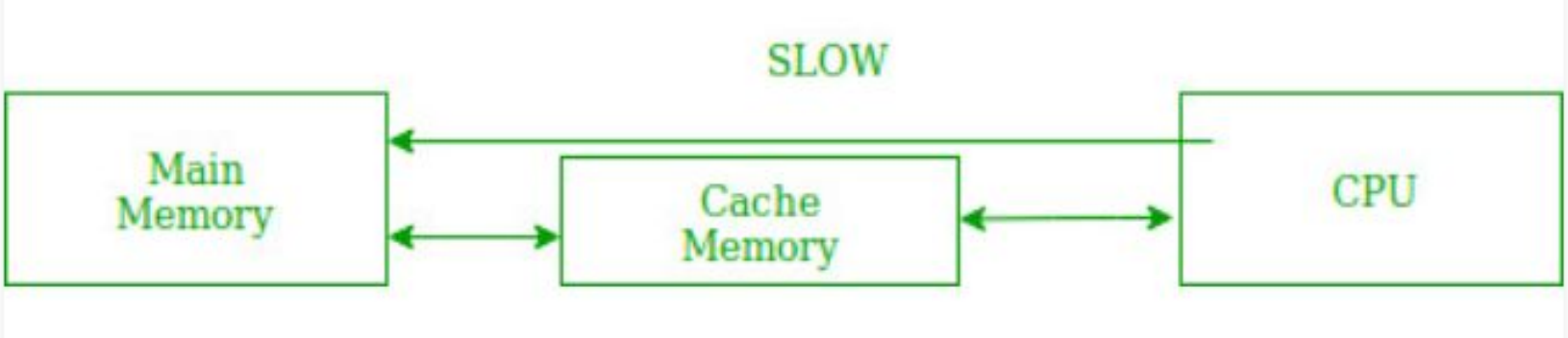
- When the cache is full and a memory word that is not in the cache is referenced, the cache control hardware must decide which block should be removed to create space for the new block that contains the referenced word
- The collection of rules for making this decision constitutes the *replacement algorithm*

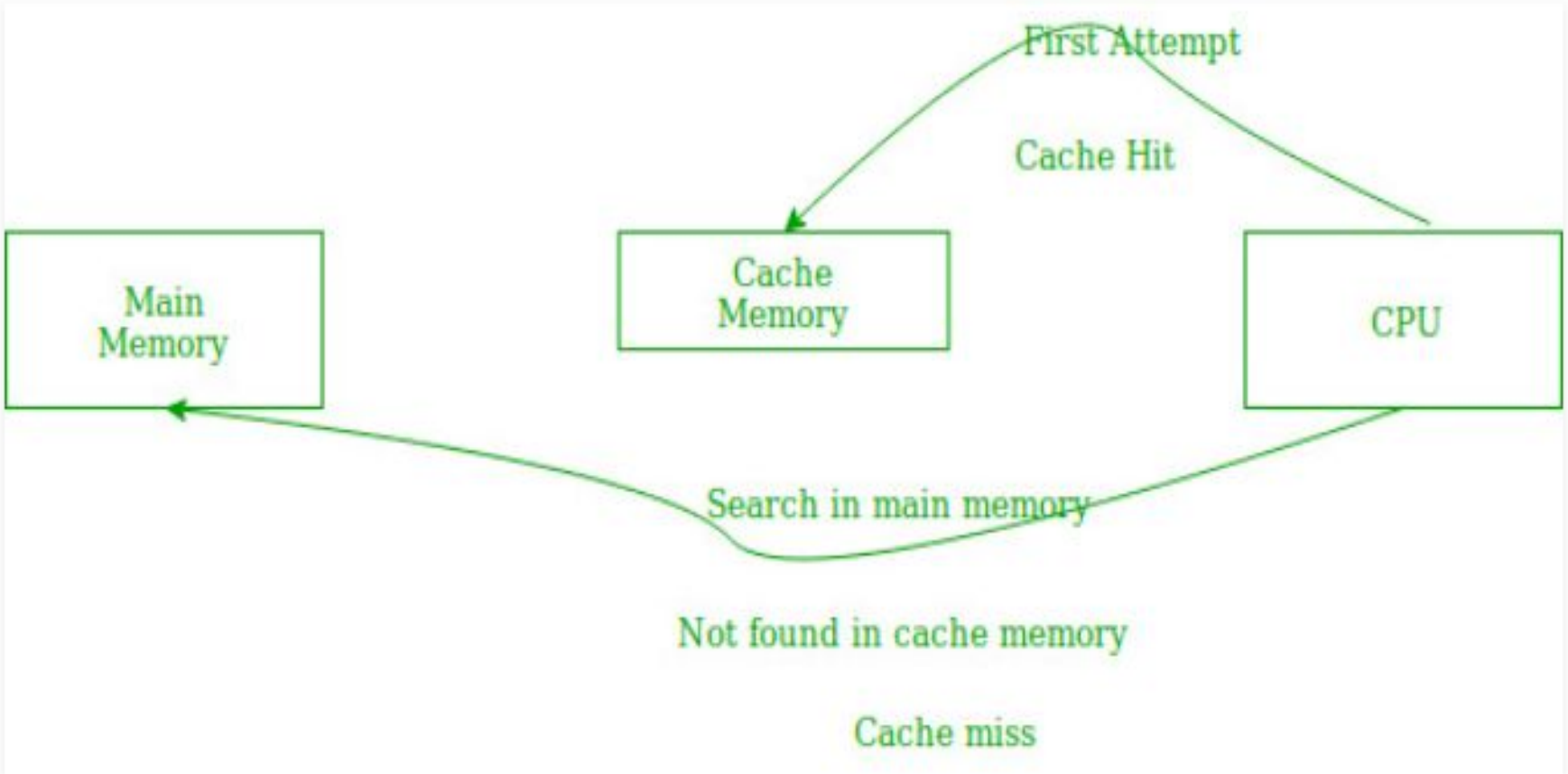
Cache Replacement Policies

- Random
  - Simple
  - Requires random generator
- First In First Out (FIFO)
  - Replace the block that has been in the cache the longest
  - Requires keeping track of the block lifetime
- Least Recently Used (LRU)
  - Replace the one that has been used the least
  - Requires keeping track of the block history

Cache Replacement Policies (Cont.)

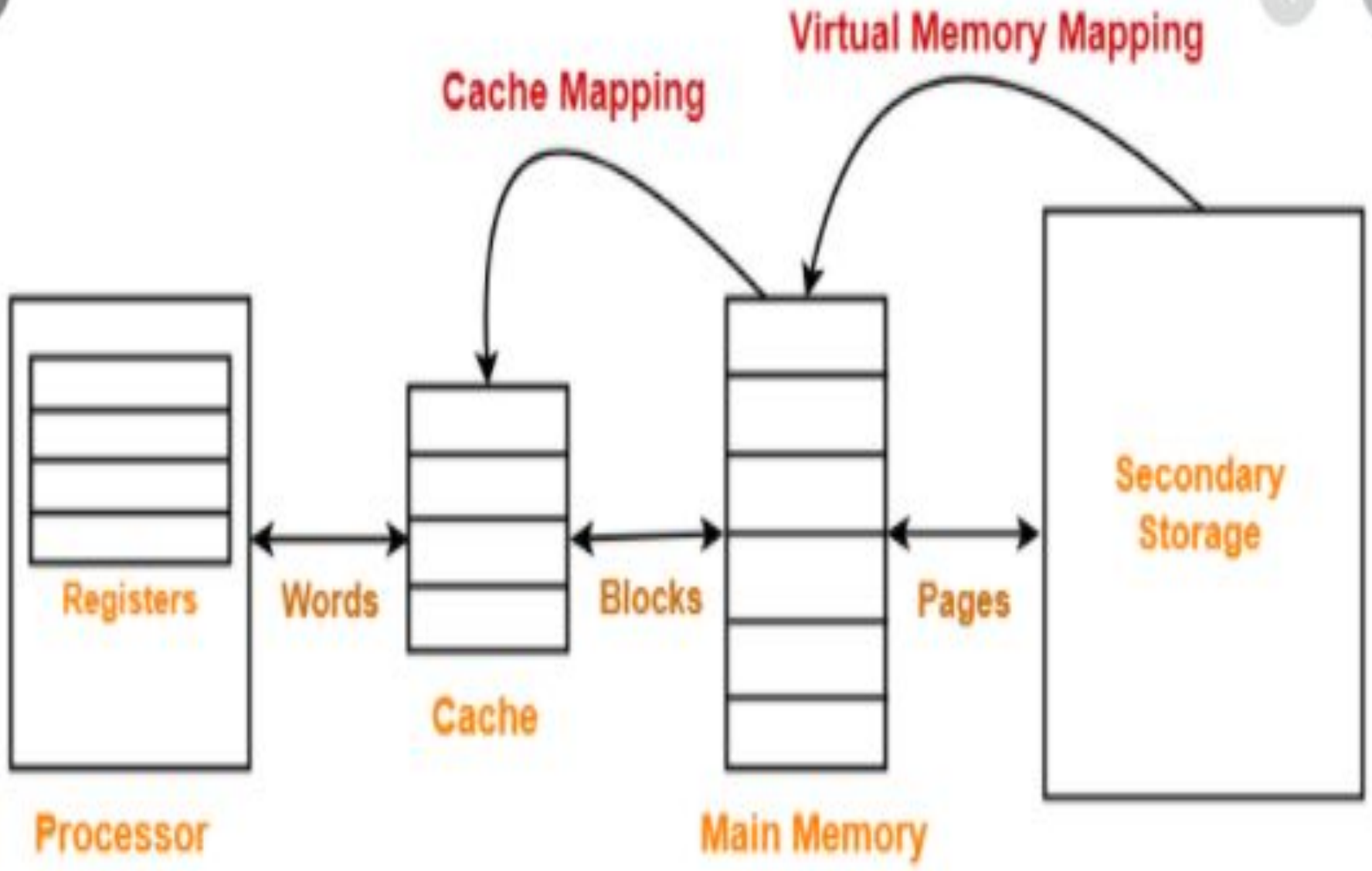
- Most Recently Used (MRU)
  - Replace the one that has been used the most
  - Requires keeping track of the block history
- Optimal
  - Hypothetical
  - Must know the future


SLOW
Main Memory
Cache Memory
CPU


First Attempt
Cache Hit
Main Memory
Cache Memory
CPU
Search in main memory
Not found in cache memory
Cache miss

Hits & Misses

- Read hits
  - this is what we want
- Read misses
  - stall the CPU, fetch block from memory, deliver to cache, restart
- Write hits:
  - can replace data in cache and memory (write-through)
  - write the data only into the cache (write-back the cache later)
- Write misses:
  - read the entire block into the cache, then write the word

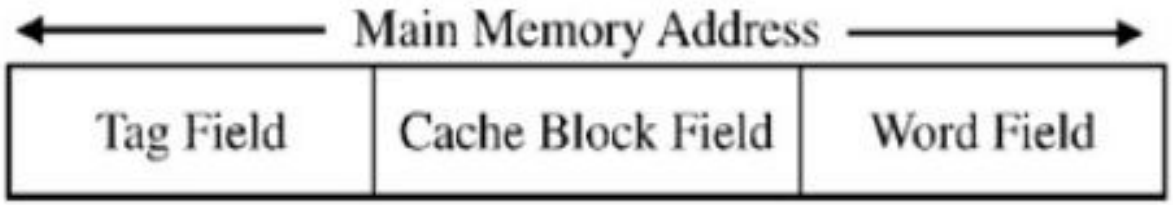

Virtual Memory Mapping
Cache Mapping
Secondary Storage
Registers
Words
Blocks
Pages
Cache
Processor
Main Memory

Mapping Functions

- To discuss possible methods for specifying where memory blocks are placed in the cache, we use a specific small example
- Consider a cache consisting of 128 blocks of 16 words each, for a total of 2048 (2K) words, and assume that the main memory is addressable by a 16-bit address. The main memory has 64K words, which we will view as 4K blocks of 16 words each
- Cache mapping functions
  - Direct mapping
  - Associative mapping
  - Set-associative mapping

Direct Mapping (Cont.)

- Bits in the main memory address are divided into three fields.



- Word ➔ identifies specific word in the block

  Word field = $\log_2 B$, where B is the size of the block in words.

- Block ➔ identifies a unique block in the cache

  Block field = $\log_2 N$, where N is the size of the cache in blocks.

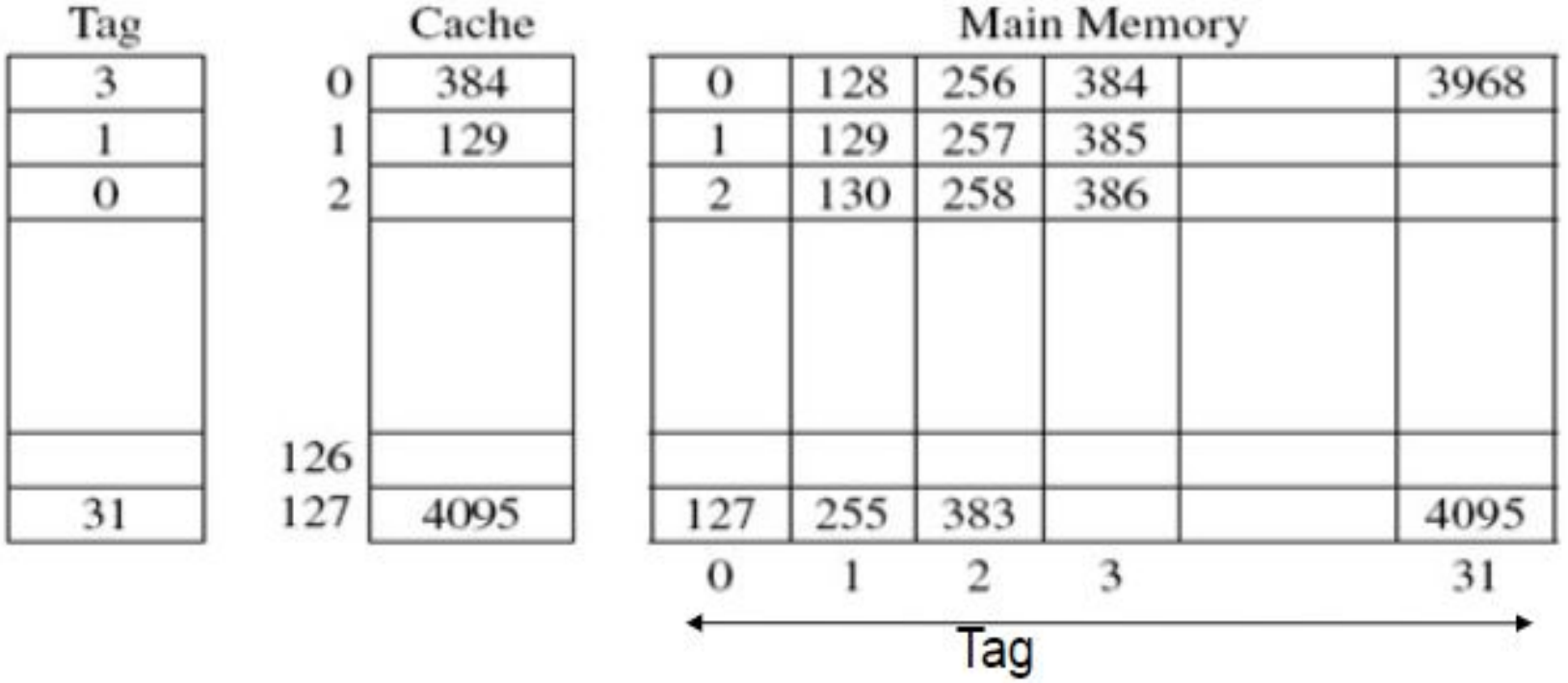
- Tag ➔ identifies which block from the main memory currently in the cache

  Tag field = $\log_2 (M/N)$, where M is the size of the main memory in blocks.

The number of bits in the main memory address = $\log_2 (B \times M)$

Example

- Consider, for example, the case of a main memory consisting of 4K blocks, a cache memory consisting of 128 blocks, and a block size of 16 words. Show the direct mapping and the main memory address format?



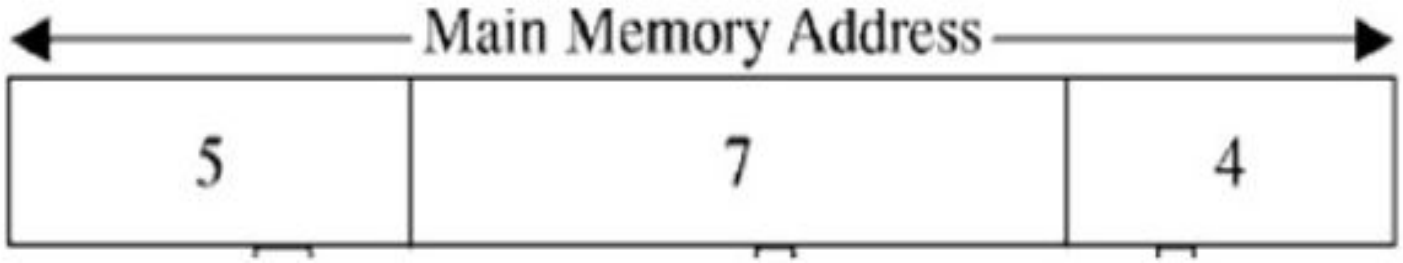
Example (Cont.)

Word field $= \log_2 B = \log_2 16 = \log_2 2^4 = 4$ bits

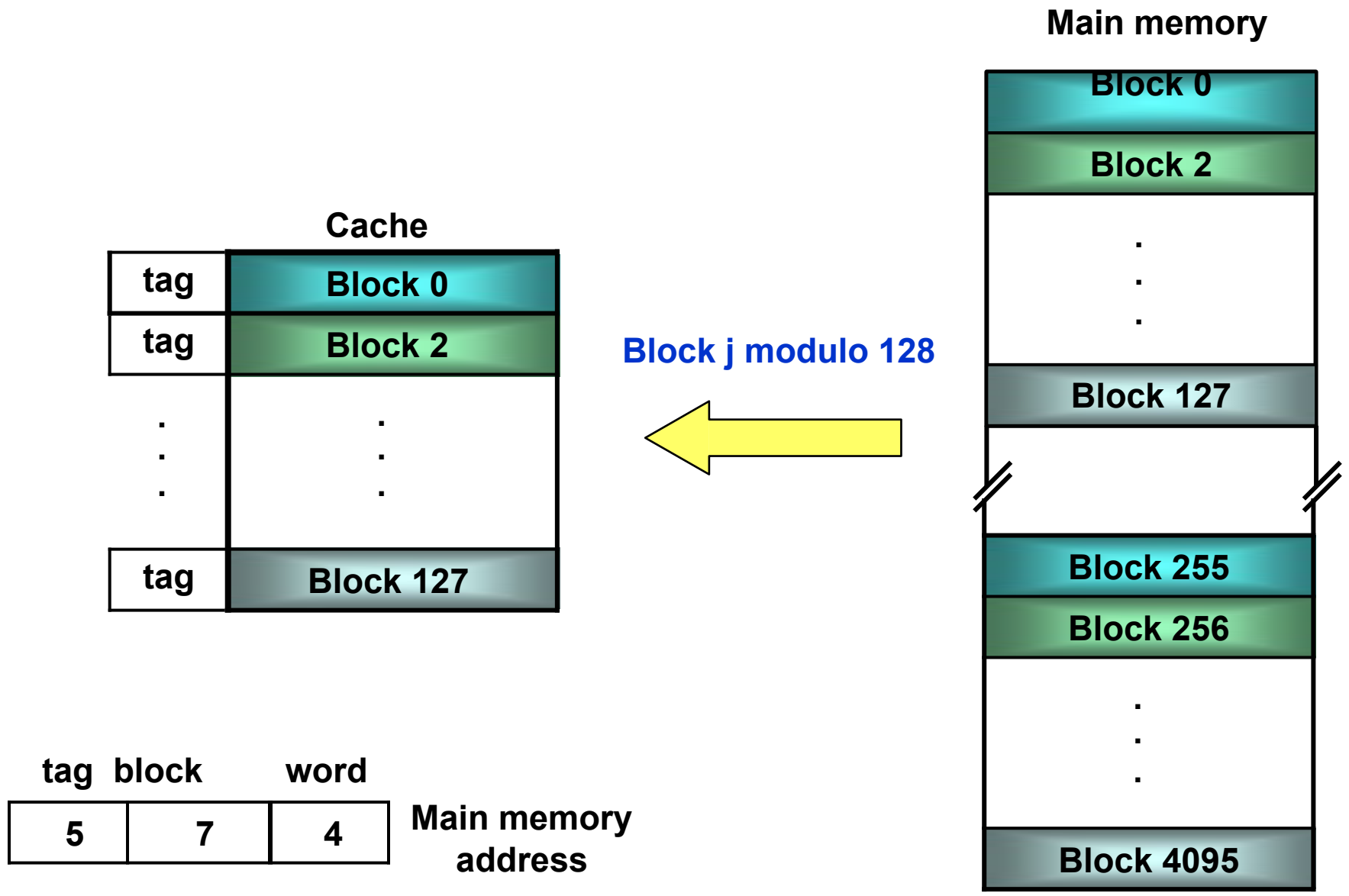
Block field $= \log_2 N = \log_2 128 = \log_2 2^7 = 7$ bits

Tag field $= \log_2(M/N) = \log_2(2^2 \times 2^{10}/2^7) = 5$ bits

The number of bits in the main memory address $= \log_2 (B \times M) = \log_2 (2^4 \times 2^{12}) = 16$ bits.



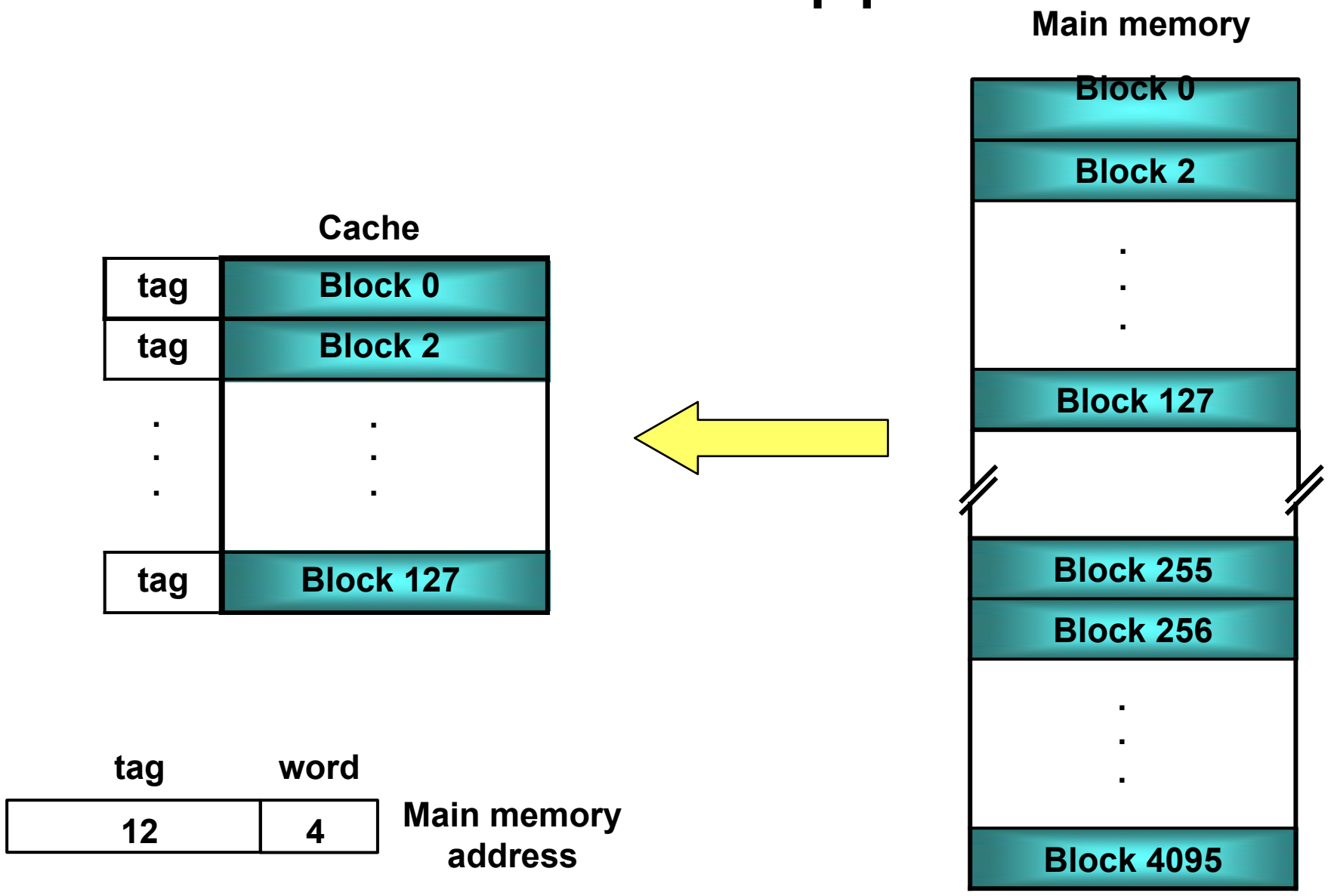
Direct-Mapped Cache



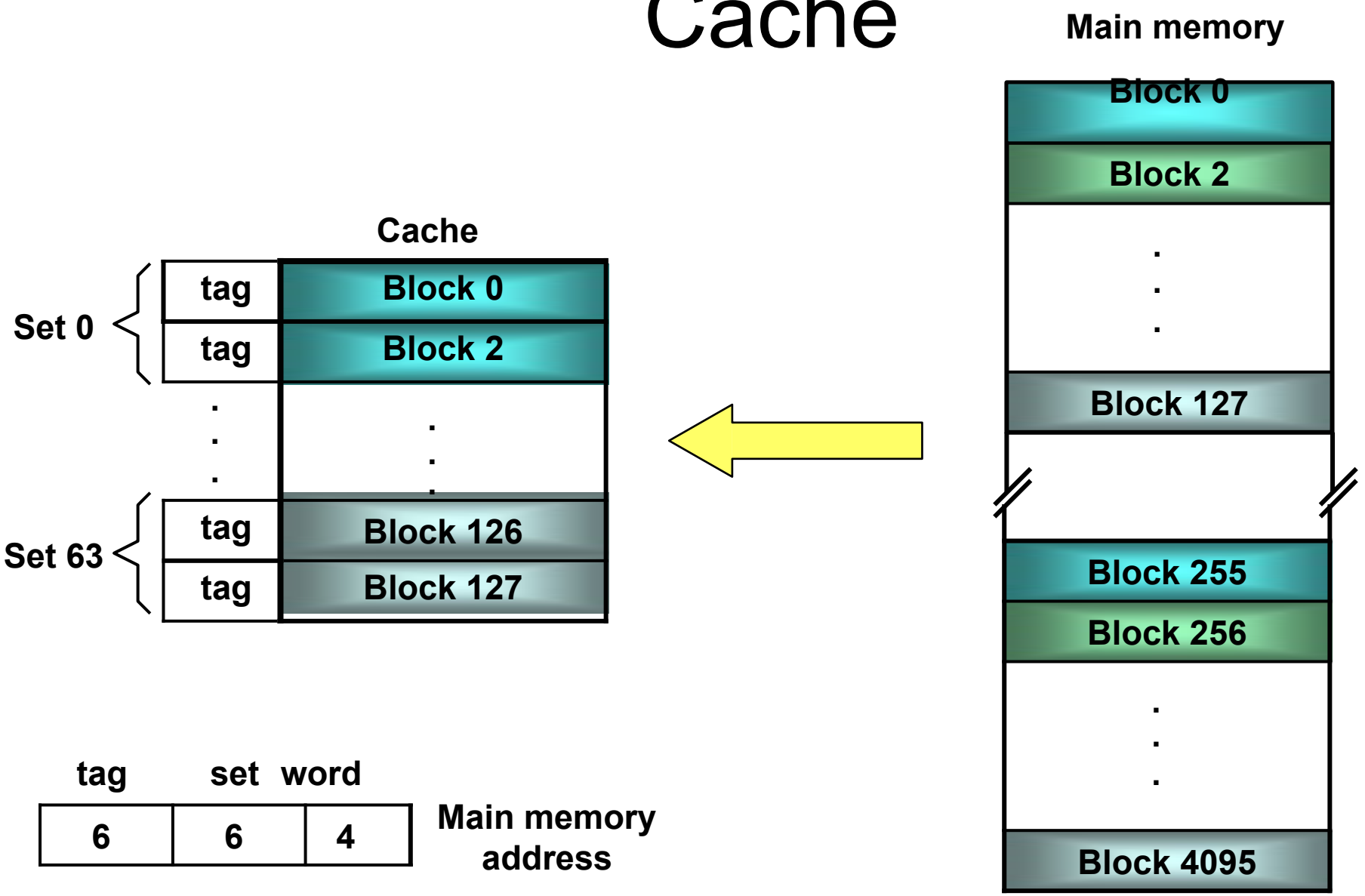
Direct Mapping

- Advantage
  - Easy
  - Does not require any search technique to find a block in cache
  - Replacement is a straight forward
- Disadvantages
  - Many blocks in MM are mapped to the same cache block
  - We may have others empty in the cache
  - Poor cache utilization

Associative-Mapped Cache



Set-Associative-Mapped Cache

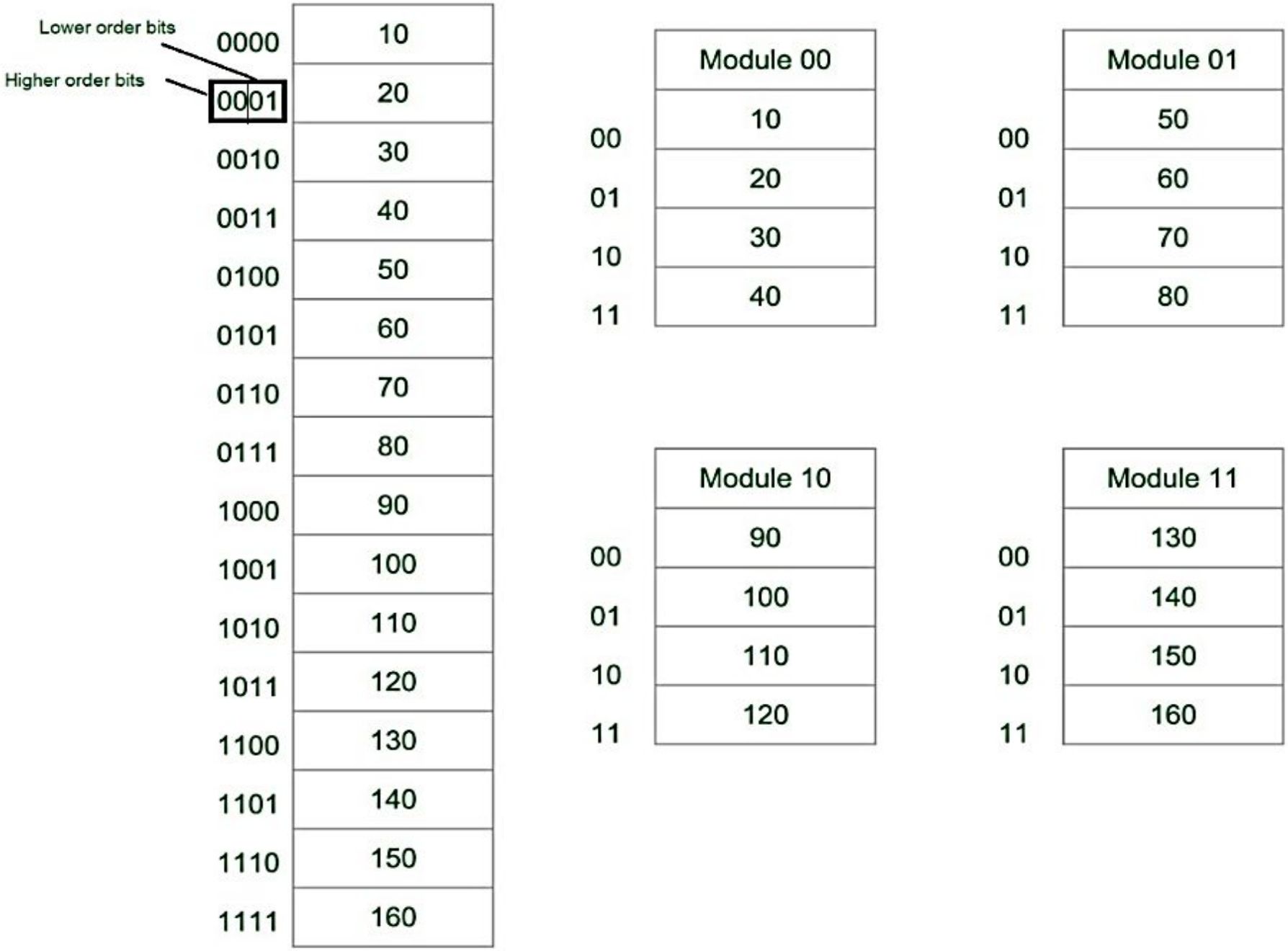


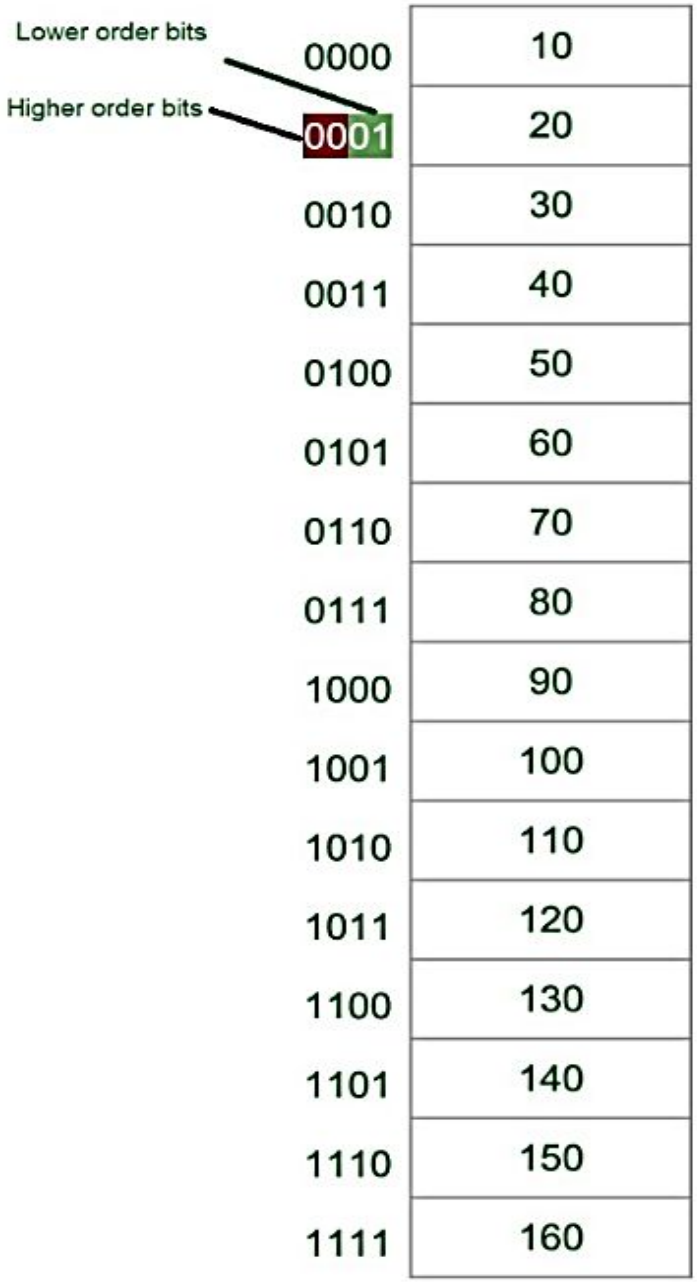
Comparison of Different Mapping Scheme

- Direct-mapped cache
  - Cost: low
  - Flexibility: low
- Associative-mapped cache
  - Cost: high
  - Flexibility: high
- Set-associative-mapped cache
  - Cost: medium
  - Flexibility: medium

Memory Interleaving

- **Memory Interleaving** is less or More an Abstraction technique.
- It is a Technique which divides memory into a number of modules such that Successive words in the address space are placed in the Different module.


Lower order bits
Higher order bits
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
Module 00
00
01
10
11
10
20
30
40
Module 01
00
01
10
11
50
60
70
80
Module 10
00
01
10
11
90
100
110
120
Module 11
00
01
10
11
130
140
150
160



	Module 00
00	10
01	50
10	90
11	130

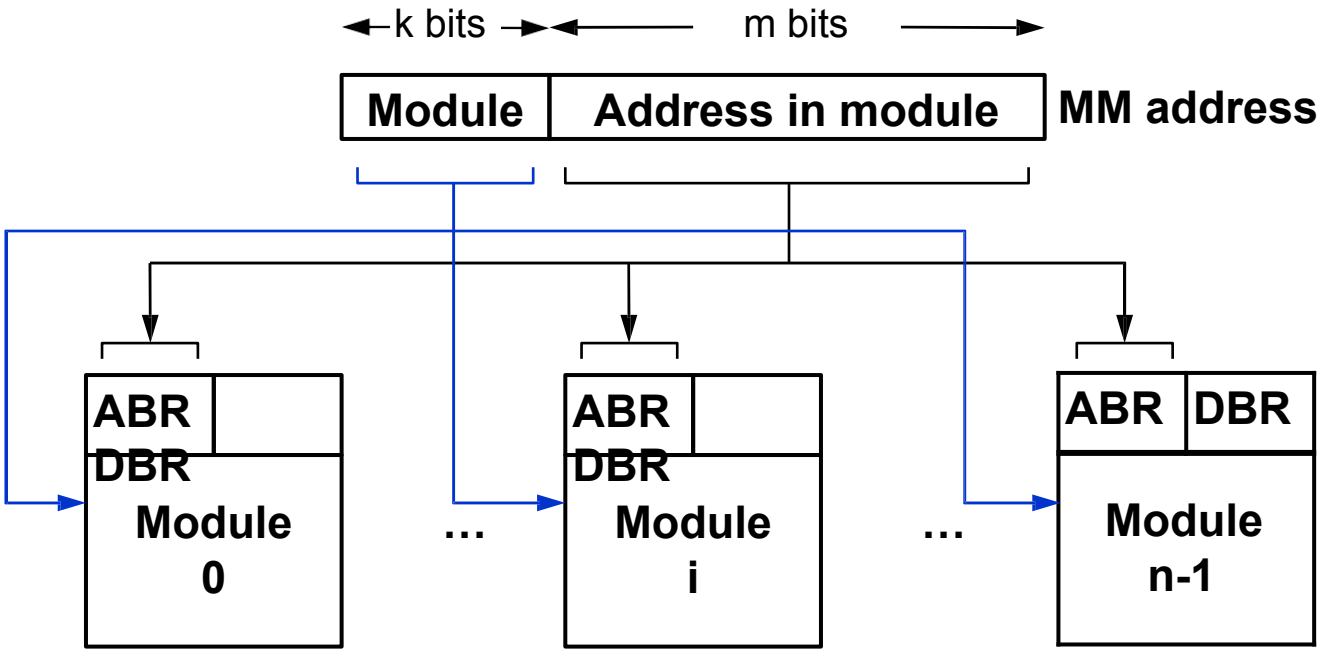
	Module 01
00	20
01	60
10	100
11	140

	Module 10
00	30
01	70
10	110
11	150

	Module 11
00	40
01	80
10	120
11	160

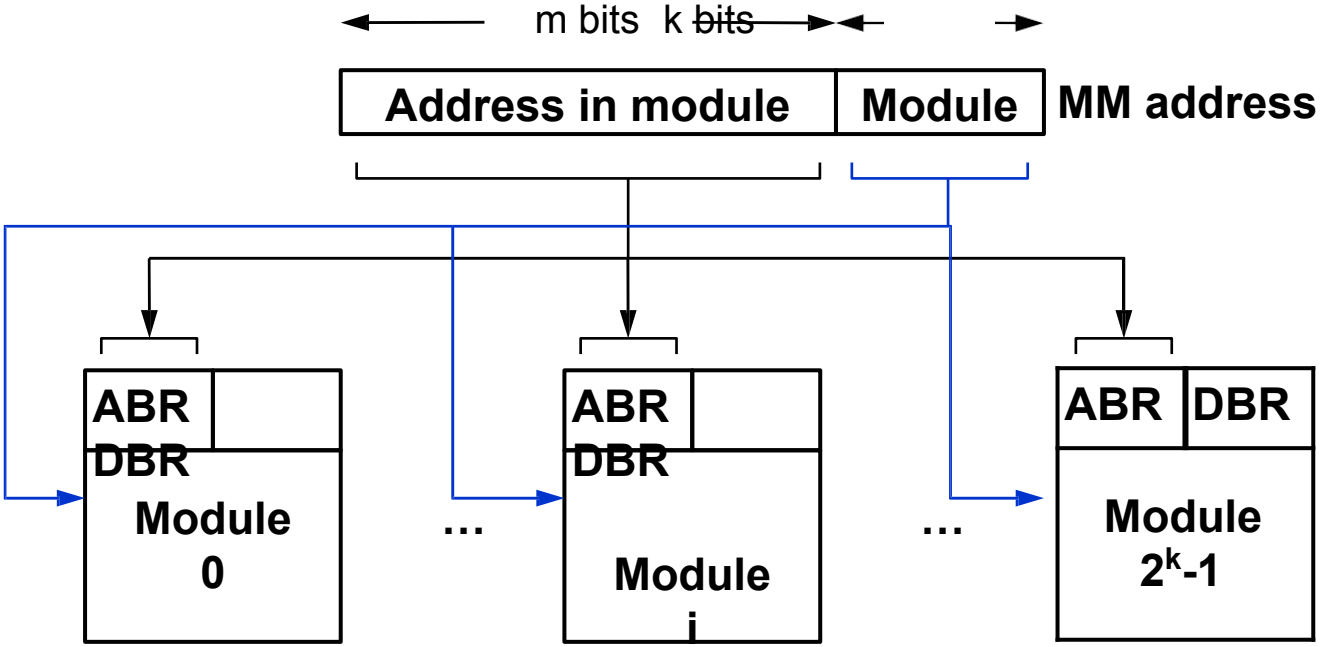
Addressing Multiple-Module Memories

- Consecutive words in a module



Addressing Multiple-Module Memories

- Consecutive words in consecutive modules



- This memory structure is called *memory interleaving*. The low-order k bits of the memory address select a module, and the high-order m bits name a location within that module.

Access Time Reduction

- Consider the time needed to transfer a block of data from the main memory to the cache when a read miss occurs.
  - Suppose that a cache with 8-word blocks is used. On a read miss, the block that contains the desired word must be copies from the memory into the cache
  - Assume that the hardware has the following properties. It takes one clock cycle to send an address to the main memory. The main memory allows the first word to be accessed in 8 cycles, but subsequent words of the block area accessed in 4 cycles per word
  - Also, one clock cycle is needed to send one word to the cache
- If a single memory module is used, then the time needed to load the desired block into the cache is
  - 1+8+(7x4)+1=38 cycles

Access Time Reduction

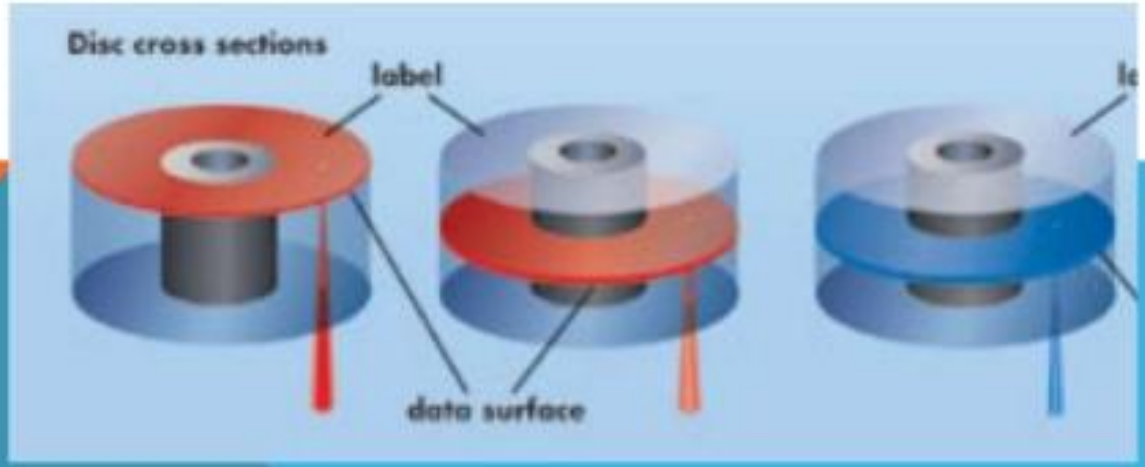
- Suppose now that the memory is constructed as four interleaved modules, using the scheme shown above.
- When the starting address of the block arrives at the memory, all four modules begin accessing the required data, using the high-order bits of the address. After 8 cycles, each module has one word of data in its data buffer register (DBR). These words are transferred to the cache, one word at a time, during the next 4 cycles. During this time, the next word in each module is accessed. Then it takes another 4 cycles to transfer these words to the cache
- Therefore, the total time needed to load the block from the interleaved memory is
  - 1+8+4+4=17 cycles

Hit Rate and Miss Penalty

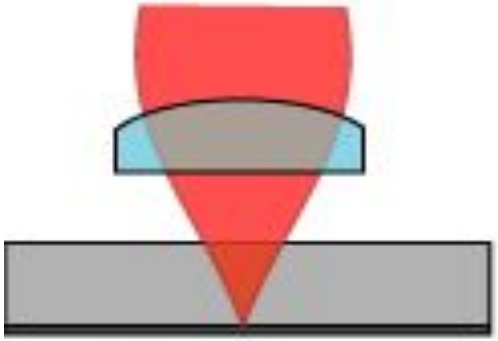
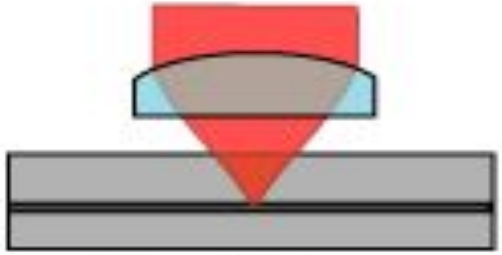
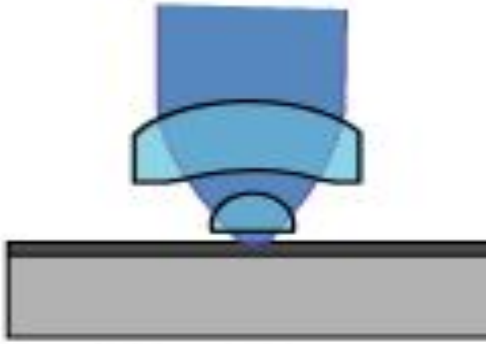
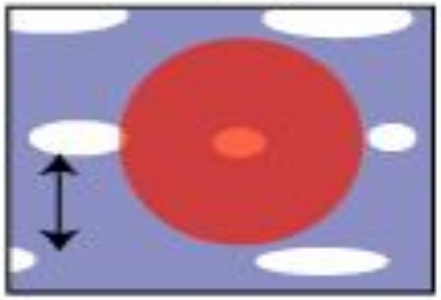
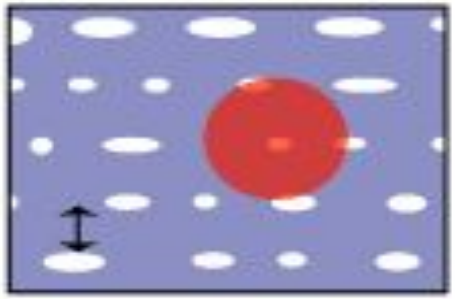
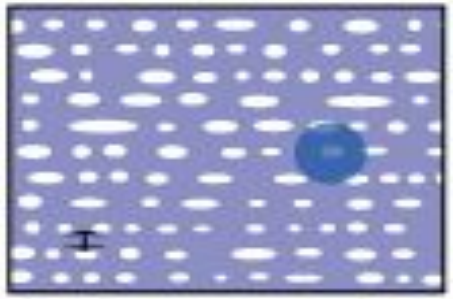
- A successful access to data in a cache is called a hit. The number of hits stated as a fraction of all attempted accesses is called the hit rate, and the miss rate is the number of misses stated as a fraction of attempted accesses
- High hit rates, well over 0.9, are essential for high-performance computers
- Performance is adversely affected by the actions that must be taken after a miss. The extra time needed to bring the desired information into the cache is called the *miss penalty*

- DVD

  - 12 cm disk with plastics substrate, aluminum coating and plastic cover
  - 0.4 micron wide pits, 1.2 micron spacing
  - Data storage capacity 4.7 GB
  - Two sided, 2 levels/side top level partially reflecting
  - Increase to 8.5 GB (2 sided), 17 GB (2 levels, 2 sided)



CD vs. DVD vs. Blu-ray Writing

CD	DVD	BD
780-nm Red Laser Lens Aperture = 0.45	650-nm Red Laser Lens Aperture = 0.6	405-nm Blue Laser Lens Aperture = 0.8
		
One 1.2-mm polycarbonate layer	Two 0.6-mm polycarbonate layers	One 1.1-mm polycarbonate layer
		
track pitch = 1.6µm	track pitch = .74µm	track pitch = .30µm