Computer Organization and Architecture

Lecture 7: Memory Hierarchy
Cache Memory

Murtadha Hssayeni, Ph.D.

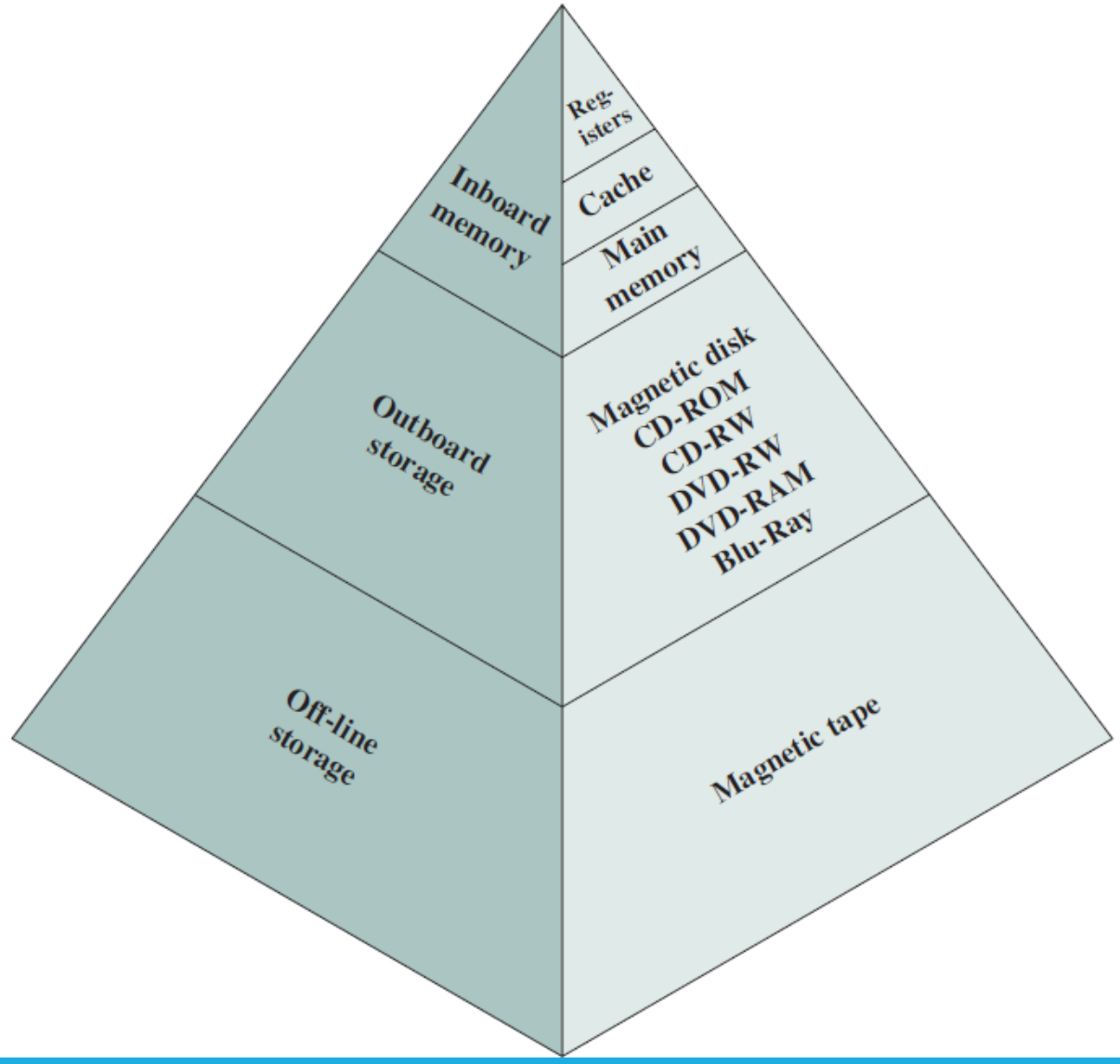
m.hssayeni@uobabylon.edu.iq

Outlines

- Memory Hierarchy
- Key Characteristics of Computer Memory Systems
- The Major Types of Semiconductor Memory
- DRAM and SRAM
- Cache Memory
- Cache/Main Memory Structure
- Cache Memory: Mapping Function

Memory Hierarchy

- As one goes down the memory hierarchy, the following occur:
  1. Decreasing cost per bit
  2. Increasing capacity
  3. Increasing access time
  4. Decreasing frequency of access of the memory by the processor



Key Characteristics of Computer Memory Systems

1. **Location**
   - Internal (e.g., processor registers, cache, main memory)
   - External (e.g., optical disks, magnetic disks, tapes)
2. **Capacity**
   - Number of words
   - Number of bytes
3. **Unit of Transfer**
   - Word
   - Block
4. **Access Method**
   - Sequential
   - Direct
   - Random
5. **Performance**
   - Access time
   - Cycle time
   - Transfer rate
6. **Physical Type**
   - Semiconductor
   - Magnetic
   - Optical
   - Magneto-optical
7. **Physical Characteristics**
   - Volatile/nonvolatile
   - Erasable/nonerasable

The Major Types of Semiconductor Memory

- The basic element of a semiconductor memory is the **memory cell**.
- All semiconductor **memory cells** share certain **properties**:
  - They exhibit **two stable states** for binary 1 and 0.
  - They are capable of being **written into** to set the state.
  - They are capable of being **read to** sense the state.
- The most common semiconductor memory is referred to as random-access memory (RAM).

Memory Type	Category	Erasure	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Nonvolatile
Programmable ROM (PROM)			
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level	
Electrically Erasable PROM (EEPROM)		Electrically, byte-level	
Flash memory		Electrically, block-level	

DRAM and SRAM

- A **dynamic RAM (DRAM)** is made with cells that store data as charge on capacitors.
  - It is commonly known as **main memory**, is where programs and data are kept when a program is running.
  - It is inexpensive, but **must be refreshed** every millisecond to avoid losing its contents.
  - Some systems use ECC (error checking and correcting) memory.
- A **static RAM (SRAM)** is used primarily for expensive, high-speed **cache memory**.
  - It uses the same logic elements used in the processor.
  - It does not have to be refreshed.

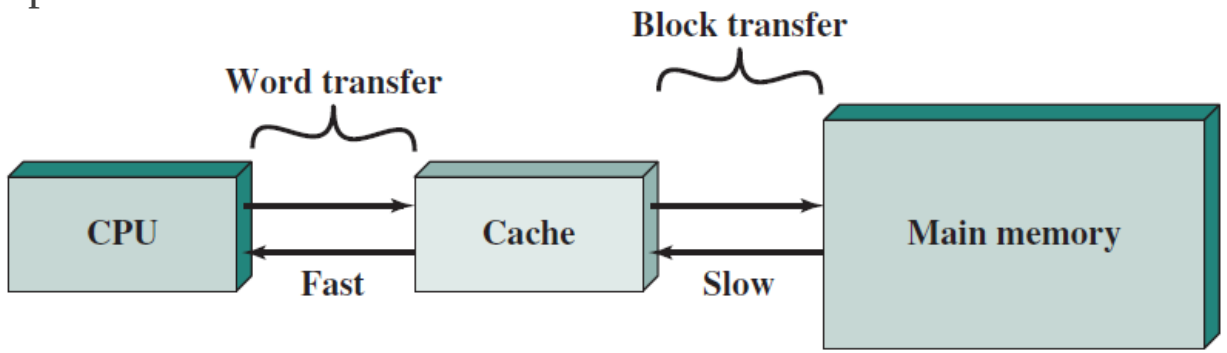
DDR Characteristics

- A **synchronous DRAM (SDRAM)** exchanges data with the processor synchronized to an external clock signal.
  - It is running at the full speed of the processor/memory bus without imposing wait states.
- A **double data-rate DRAM (DDR DRAM)** provides several features that dramatically increase the data rate.
  - First, the data transfer is synchronized to both the rising and falling edge of the clock.
  - Second, DDR uses higher clock rate on the bus to increase the transfer rate.
  - Third, a buffering scheme is used.

	DDR1	DDR2	DDR3	DDR4
Prefetch buffer (bits)	2	4	8	8
Voltage level (V)	2.5	1.8	1.5	1.2
Front side bus data rates (Mbps)	200—400	400—1066	800—2133	2133—4266

Cache Memory

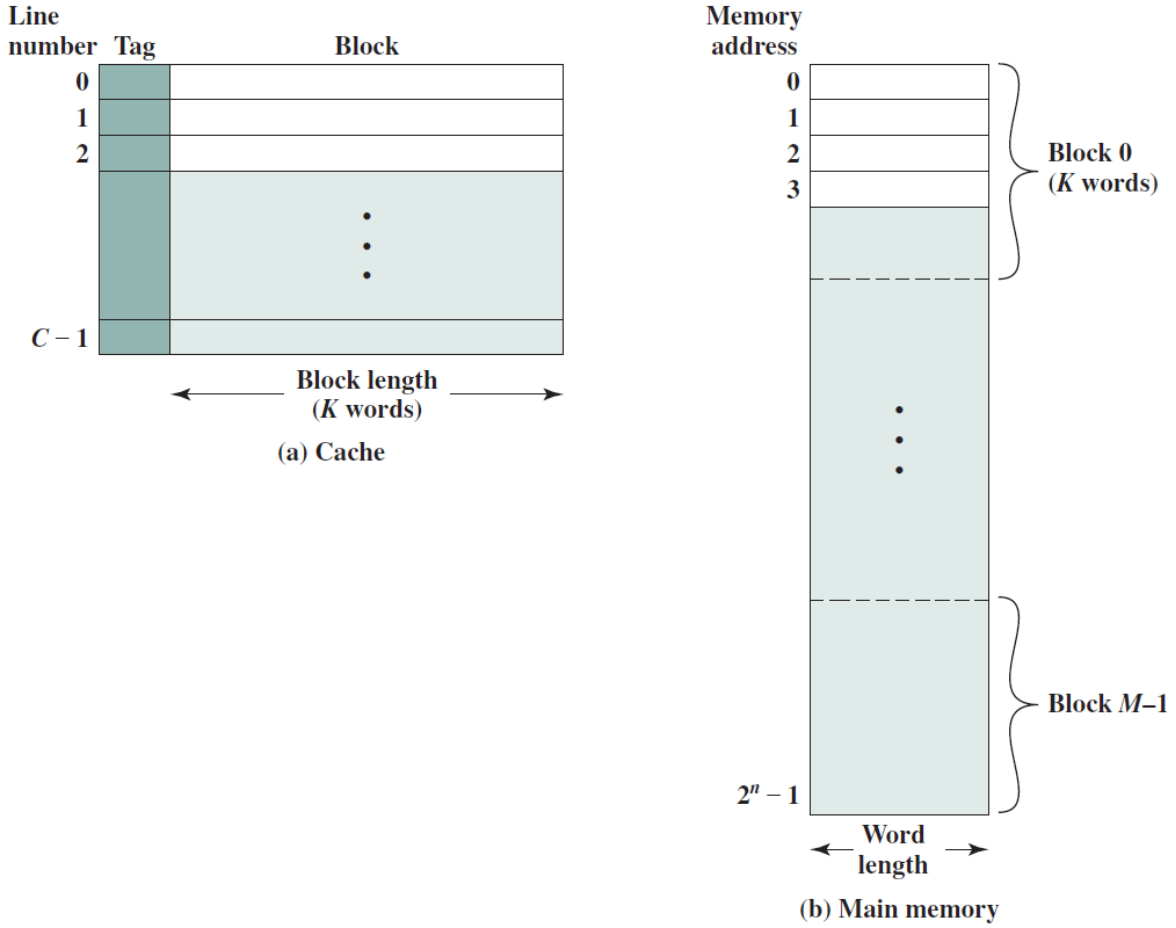
- The cache contains a copy of portions of main memory.
  - When the processor attempts to read a word of memory:
    - A check is made to determine if the word is in the cache.
    - After **hit check**, the word is delivered to the processor.
    - If the word is not in the cache (**miss check**), a block of main memory is read into the cache and then the word is delivered to the processor.



- There are multiple levels of cache.
  - for instruction and data

Cache/Main Memory Structure

- **Main memory:**
  - Each address having a unique n bits
  - The memory size is 2^n addresses.
  - For mapping to Cache purposes, it consists of a number of fixed-length blocks of k words each.
  - The number of memory blocks $M = \frac{2^n}{k}$
- **The cache:**
  - It consists of m blocks, called lines.
  - Each line contains k words plus a tag of a few bits.
  - The tag identifies which particular block from memory is currently being stored in Cache.
  - The number of cache blocks $= \frac{Size\ of\ cache}{k}$



Cache/Main Memory Structure

- **Example:**
  - A cache memory can hold 64 KB.
  - Data are transferred between main memory and the cache in blocks of 4 bytes each.
  - Each byte in the main memory is directly addressable by a 24-bit address.

  - How many blocks in the cache?
  - What is the size of the main memory?
  - How many blocks in the main memory for mapping purposes?
  - What is the size of the address bus and data bus?

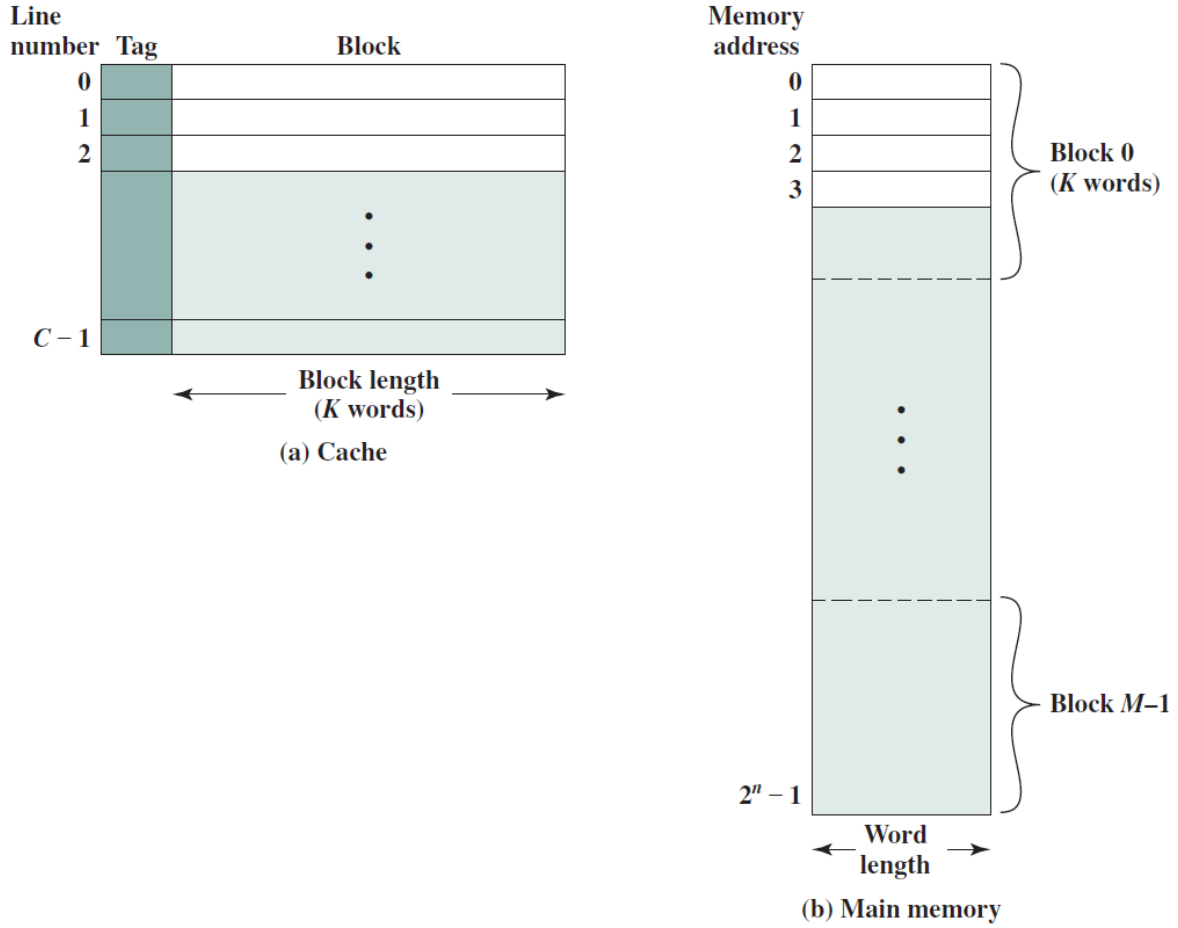
- **Solution**:
  - The number of cache blocks $= \frac{Size\ of\ cache}{k} =$ 64kB/4 = 16K
  - This means that the cache is organized as 16K = 2^{14} lines of 4 bytes each.

  - The memory size is $2^n = 2^{24} = 2^4 * 2^{20} = 16M$
  - The number of memory blocks = $\frac{2^n}{k}$=16M/4=4M blocks of 4 bytes each.

  - size of the address bus = 24
  - Size of the data bus = 8

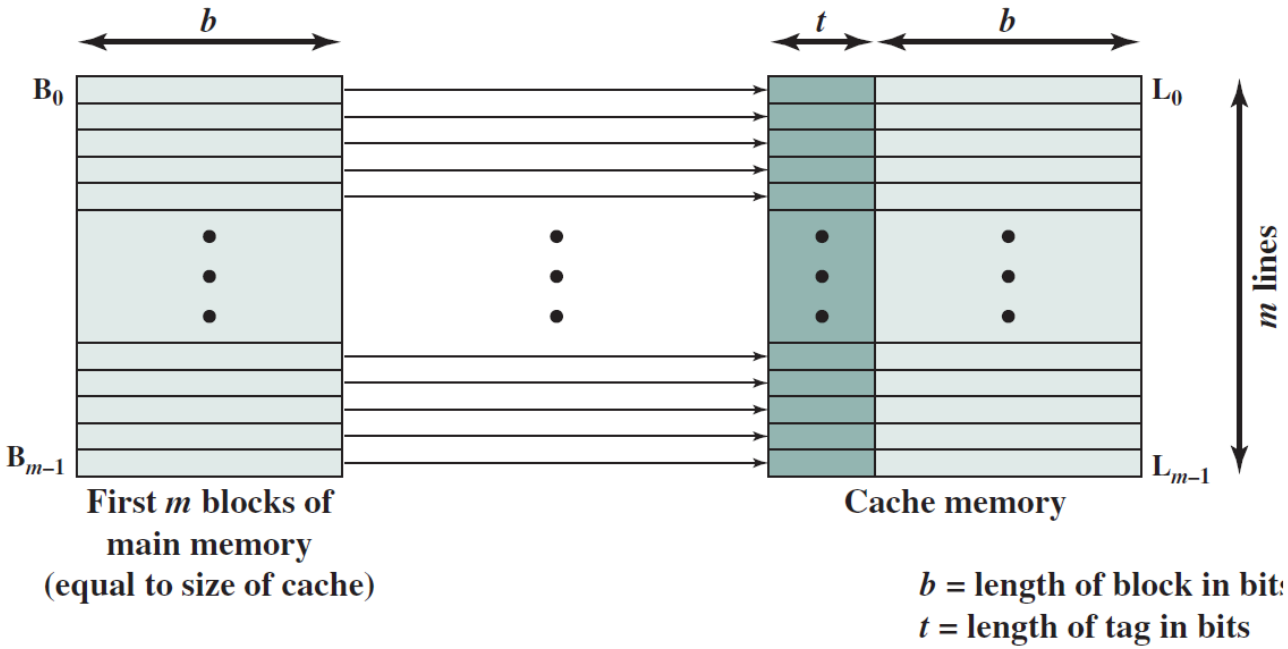
Cache Memory: Mapping Function

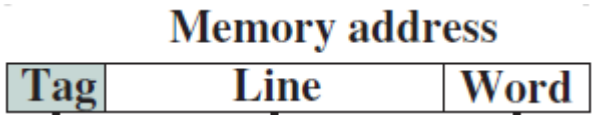
- There are fewer cache lines than main memory blocks
  - Therefore, **a technique is needed for mapping** main memory blocks into cache lines.
- **Three mapping techniques** can be used:
  - Direct mapping
  - Associative mapping
  - Set-associative mapping



Direct Mapping

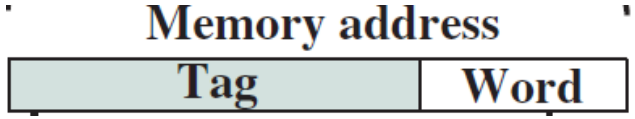
- Direct mapping maps each block of main memory into only one possible cache line.
- The mapping is expressed as
  - i = j % m
    - i = cache line number
    - j = main memory block number
    - m = number of lines in the cache



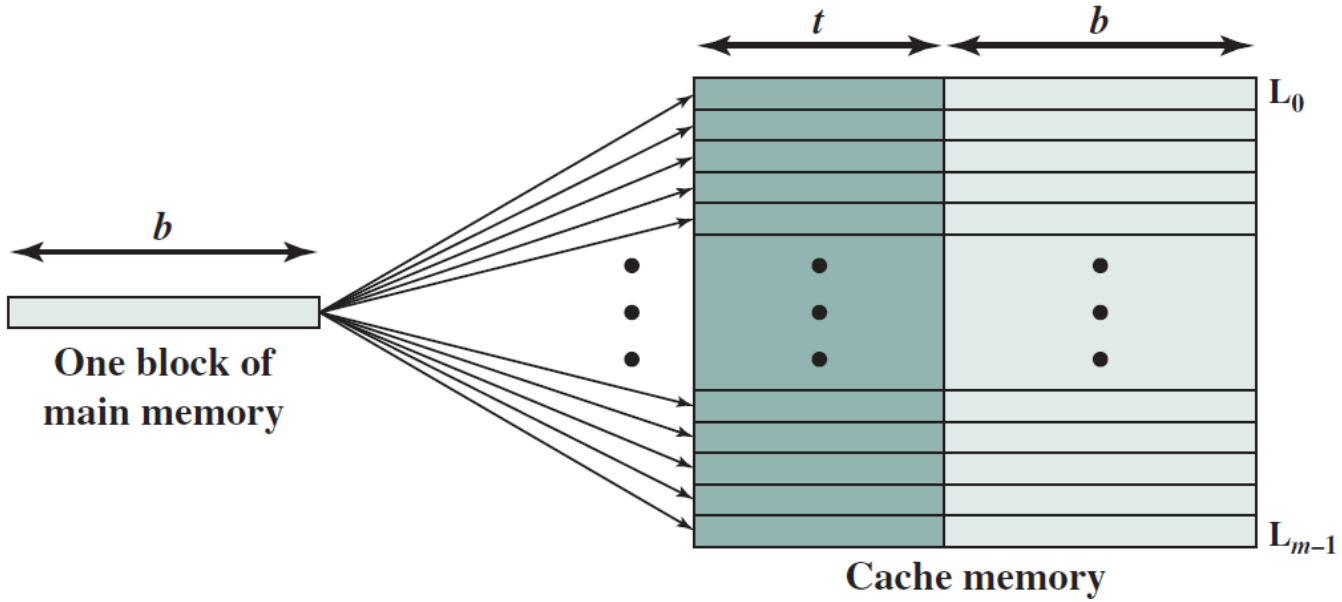


Associative Mapping

- Associative Mapping permits each main memory block to be loaded into any line of the cache.
- The cache control logic interprets a memory address simply as a **Tag and a Word field.**



- To determine whether a block is in the cache, the cache control logic must **simultaneously examine every line's tag** for a match.



Set-associative Mapping

- Set-associative Mapping exhibits the strengths of both the direct and associative approaches.
- The cache consists of **number sets**, each of which consists of a **number of lines**.
- The mapping is expressed as
  - m = v * k
  - i = j % v
    - v = number of sets
    - k = number of lines in each set
    - i = cache set number
    - j = main memory block number
    - m = number of lines in the cache