Chapter 6: Cache Memory and Its Impact on System Performance

6.1 Introduction to Cache Memory

Cache memory is a small, high-speed memory located close to the CPU that stores frequently accessed data.

- Acts as a **buffer between the CPU and main memory (RAM)**.
- Significantly improves system performance by reducing memory access time.
- Exploits **temporal and spatial locality** in program execution.

6.2 Characteristics of Cache Memory

- **High Speed** – Faster than RAM, closer to CPU clock speed.
- **Small Size** – Limited capacity (typically KBs to a few MBs).
- **Volatile** – Loses data when power is off.
- **Costly** – Higher cost per bit compared to RAM.

6.3 Cache Levels

Modern systems implement multiple levels of cache for performance balance:

Level	Location	Size	Speed	Shared
L1 Cache	On CPU core	16–64 KB	Fastest	Private
L2 Cache	On or near core	256 KB – 1 MB	Slower than L1	Private
L3 Cache	Shared across cores	2–30 MB	Slowest among caches	Shared

---

6.4 Cache Mapping Techniques

Cache mapping determines how data from main memory is placed in the cache.

1. Direct Mapping

● Each memory block maps to one specific cache line.
● Simple but prone to collisions.
Formula: `Cache Line = (Block Address) mod (Number of Lines)`

2. Fully Associative Mapping

● A memory block can go into any cache line.
● Uses a comparator for every line (high cost, flexible).

3. Set-Associative Mapping

● Compromise between direct and fully associative.
● Cache is divided into sets; each set has multiple lines.
● Reduces collisions and improves performance.

6.5 Cache Replacement Policies

When the cache is full, one block must be replaced.
Common policies include:

● **Least Recently Used (LRU)** – Replaces the least recently accessed block.
● **First-In First-Out (FIFO)** – Replaces the oldest loaded block.
● **Random** – Chooses a block at random.

6.6 Write Policies

Controls how data is written to cache and main memory.

1. **Write-Through**
   ● Data is written to both cache and main memory.
   ● Ensures consistency but increases memory traffic.

2. **Write-Back**
   ● Data is written only to cache initially.

- Updated to main memory later (on eviction).
- Reduces traffic but needs control logic.

6.7 Cache Performance Metrics

Performance is evaluated using:

- **Hit** – Data found in cache
- **Miss** – Data not in cache, must be fetched from memory
- **Hit Rate** = (Number of Hits) / (Total Accesses)
- **Miss Rate** = 1 – Hit Rate
- **Average Memory Access Time (AMAT)** =
`Hit Time + Miss Rate × Miss Penalty`

6.8 Impact of Cache on System Performance

A well-designed cache can greatly enhance performance by:

- Reducing average memory access time
- Increasing CPU utilization and instruction throughput
- Decreasing memory bottlenecks in pipelines
- Lowering power usage due to fewer memory accesses

6.9 Cache Coherency in Multicore Systems

In multicore processors, caches may store copies of shared memory.

- Coherency protocols (like MESI) are used to **maintain consistency**.
- Without coherency, cores may use outdated or incorrect data.

6.10 Applications of Cache Memory

- **Processor Performance** – Boosts instruction fetch and data read speeds
- **File Systems** – Disk caching improves I/O performance
- **Web Browsers** – Cache web content for faster load times
- **Databases** – Cache query results for faster access

6.11 Advantages and Disadvantages

✅ **Advantages:**

- Speeds up memory access
- Reduces processor idle time
- Improves system responsiveness
- Lowers bandwidth usage on memory bus

❌ **Disadvantages:**

- Expensive per bit
- Limited size
- Complexity in design (coherency, replacement)
- Potential inconsistency in multicore systems without proper management

---

6.12 Summary of Key Concepts

- Cache memory is a high-speed memory that enhances system performance.
- It operates using mapping, replacement, and write policies.
- A high cache hit rate reduces average memory access time.
- Multilevel cache and coherency mechanisms are vital in modern multicore processors.
- Cache design significantly affects CPU throughput and overall efficiency.