Cache Memory

Computer architecture: A quantitative approach
by
J. L. Hennessy & D.A. Patterson

Prof.S.Meenatchi, SITE, VIT.

Cache Memory

- Introduction
- Parameters of Cache memory
- Performance of Cache Memory System
- Block placement
  - Direct associative
  - Set associative
  - Fully associative

Introduction to Cache memory

- **Principle of locality**
  - Programs tend to reuse the data and instructions they have used recently.
- **Types of locality**
  - **Temporal locality**
    - Recently accessed items are likely to be accessed in the near future.
  - **Spatial locality**
    - Items whose addresses are near one another tend to be referenced close together in time.

Parameters of Cache memory

- **Cache Hit**
  - A referenced item is found in the cache by the processor.
- **Cache Miss**
  - A referenced item is not present in the cache
- **Hit ratio**
  - Ratio of number of hits to total number of references
  - **number of hits/(number of hits + number of Miss)**
- **Miss penalty**
  - Additional cycles required to serve the miss.

Parameters of Cache memory

- Memory access time is the best measure of the benefit of different cache organizations
- **Memory access time = hit time + miss rate × miss penalty**
- **Miss rate**
  - Fraction of cache accesses that result in a miss => number of miss/number of accesses
- **Hit time**
  - Time to hit in the cache
- **Miss penalty**
  - Time to replace the block from memory (cost of a miss).

Parameters of Cache Memory

- Time required for the cache miss depends on both the latency and bandwidth
- **Latency**
  – Time to retrieve the first word of the block
- **Bandwidth**
  – Time to retrieve the rest of this block

Performance of Cache Memory System

- **$T_e = T_c + (1 - h)\ T_m$**

  Where

  Te: Effective memory access time in Cache memory system

  Tc: Cache access time

  Tm: Main memory access time

- Example:

  Tc = 0.4 ns,

  Tm = 1.2 ns,

  h = 0.85%

  Te= 0.4 + (1 - 0.85) * 1.2 = 0.58 ns

Block Placement Strategies

- **Cache organizations**
  - **Direct mapped:**
    - Each block has only one place in the cache.
    - Mapping: (Block address) MOD (Number of block in cache)
  - **Set associative:**
    - A block can be placed in a restricted set of places in the cache.
    - A set is a group of blocks in the cache.
    - Mapping: (Block address) MOD (Number of sets in the cache)
    - If there are n blocks in a set, the cache placement is called n-way set associative.
  - **Fully associative:**
    - A block can be placed any where in the cache.

Example

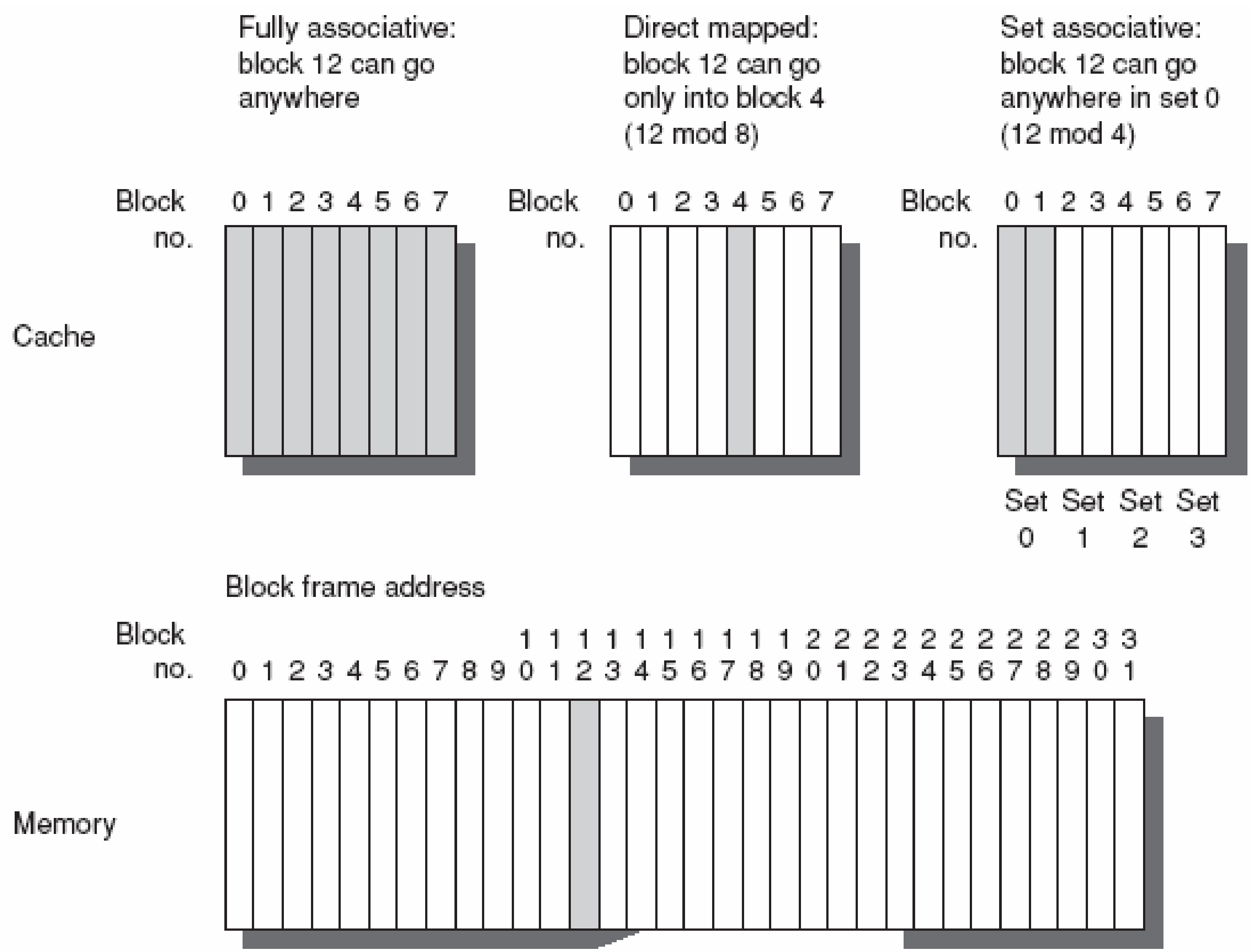

Figure C.2 This example cache has eight block frames and memory has 32 blocks.

Block Identification

Block address		Block offset
Tag	Index	

- **Set-associative cache**
  - Tag is used to check all the blocks in the set
  - Index or set no. is used to select the set.
  - Block offset or word is the address of the desired word within the block.
- **Direct-mapped cache**
  - Index or line no. is used to select the line.
  - Tag is used to compare the tag of the line.
  - Block offset or word is the address of the desired word within the block.
- **Fully associative caches** have no index field.

Virtual memory

- Similarly, not all objects referenced by a program need to reside in main memory.
- *Virtual memory* means some objects may reside on disk.
- Address space is broken into fixed-size blocks, called *pages*.
- At any time, each page resides either in main memory or on disk.
- When the processor references an item within a page that is not present in the cache or main memory, a *page fault* occurs, and the entire page is moved from the disk to main memory.
- Since page faults take so long, they are handled in software and the processor is not stalled.
- Processor usually switches to some other task while the disk access occurs.