Chapter 5

Cache Memories

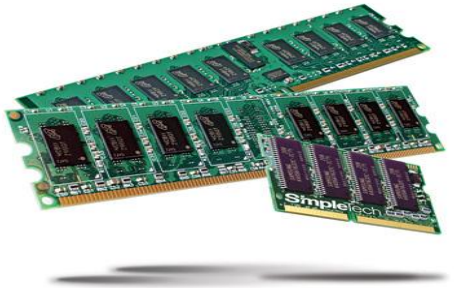
The Memory System

Cache Memories

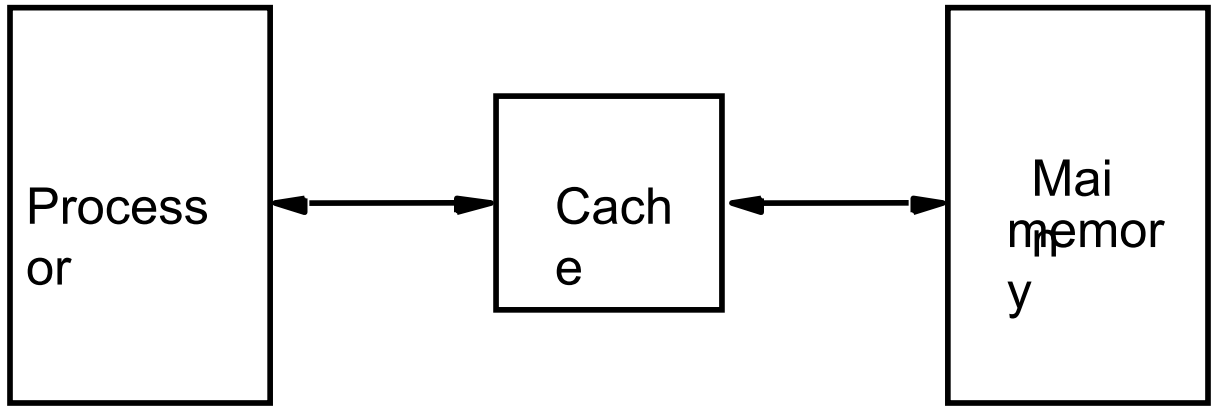
- Processor is much faster than the main memory.
  - As a result, the processor has to spend much of its time waiting while instructions and data are being fetched from the main memory.
  - Major obstacle towards achieving good performance.
- Speed of the main memory cannot be increased beyond a certain point.
- Cache memory is an architectural arrangement which makes the main memory appear faster to the processor than it really is.
- Cache memory is based on the property of computer programs known as "locality of reference".

Locality of Reference

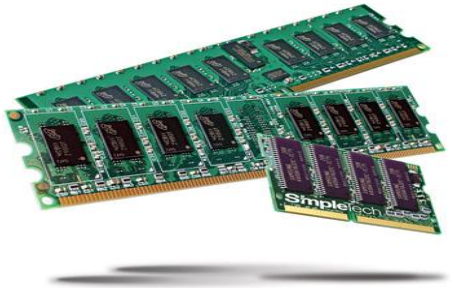
- Analysis of programs indicates that many instructions in localized areas of a program are executed repeatedly during some period of time, while the others are accessed relatively less frequently.
  - These instructions may be the ones in a loop, nested loop or few procedures calling each other repeatedly.
  - This is called "locality of reference".
- Temporal locality of reference:
  - Recently executed instruction is likely to be executed again very soon.
- Spatial locality of reference:
  - Instructions with addresses close to a recently instruction are likely to be executed soon.



Cache memories



- *Processor issues a Read request, a block of words is transferred from the main memory to the cache, one word at a time.*
- *Subsequent references to the data in this block of words are found in the cache.*
- *At any given time, only some blocks in the main memory are held in the cache. Which blocks in the main memory are in the cache is determined by a "mapping function".*
- *When the cache is full, and a block of words needs to be transferred from the main memory, some block of words in the cache must be replaced. This is determined by a "replacement algorithm".*

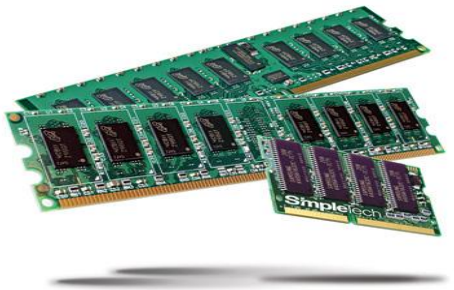


Cache hit

- *Existence of a cache is transparent to the processor. The processor issues Read and Write requests in the same manner.*
- *If the data is in the cache it is called a Read or Write hit.*
- *Read hit:*
  - *The data is obtained from the cache.*
- *Write hit:*
  - *Cache has a replica of the contents of the main memory.*
  - *Contents of the cache and the main memory may be updated simultaneously. This is the write-through protocol.*
  - *Update the contents of the cache, and mark it as updated by setting a bit known as the dirty bit or modified bit. The contents of the main memory are updated when this block is replaced. This is write-back or copy-back protocol.*

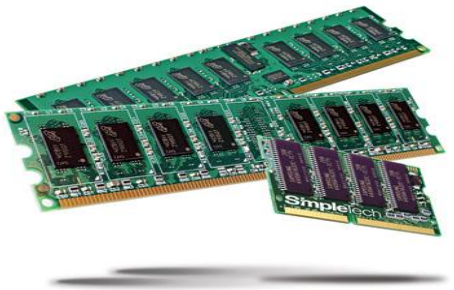
Cache miss

- *If the data is not present in the cache, then a Read miss or Write miss occurs.*
- *Read miss:*
  - *Block of words containing this requested word is transferred from the memory.*
  - *After the block is transferred, the desired word is forwarded to the processor.*
  - *The desired word may also be forwarded to the processor as soon as it is transferred without waiting for the entire block to be transferred. This is called load-through or early-restart.*
- *Write-miss:*
  - *Write-through protocol is used, then the contents of the main memory are updated directly.*
  - *If write-back protocol is used, the block containing the addressed word is first brought into the cache. The desired word is overwritten with new information.*

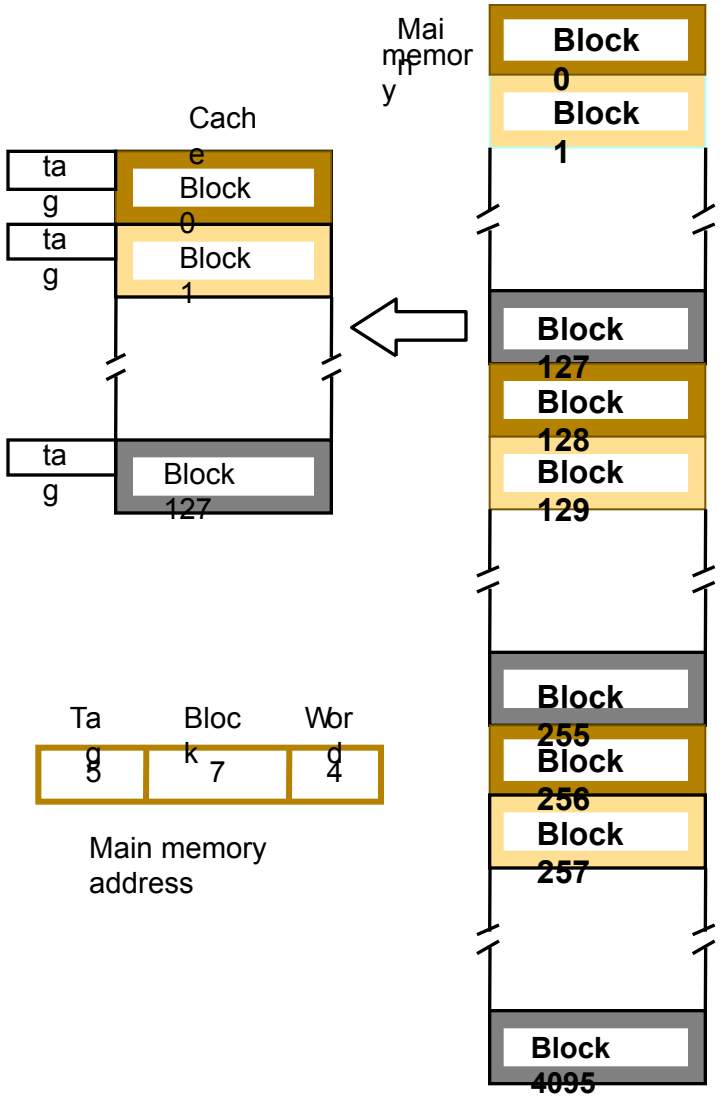


Mapping functions

- Mapping functions determine how memory blocks are placed in the cache.
- A simple processor example:
  - Cache consisting of 128 lines of 16 words each.
  - Total size of cache is 2048 (2K) words.
  - Main memory is addressable by a 16-bit address.
  - Main memory has 64K words.
  - Main memory has 4K blocks of 16 words each.
- Three mapping functions:
  - Direct mapping
  - Associative mapping
  - Set-associative mapping.



Direct mapping



- *Block j of the main memory maps to j modulo 128 of the cache. 0 maps to 0, 129 maps to 1.*
- *More than one memory block is mapped onto the same position in the cache.*
- *May lead to contention for cache blocks even if the cache is not full.*
- *Resolve the contention by allowing new block to replace the old block, leading to a trivial replacement algorithm.*
- *Memory address is divided into three fields:*
  - *Low order 4 bits determine one of the 16 words in a block.*
  - *When a new block is brought into the cache, the the next 7 bits determine which cache block this new block is placed in.*
  - *High order 5 bits determine which of the possible 32 blocks is currently present in the cache. These are tag bits.*
- *Simple to implement but not very flexible.*

numericals

Q1. Consider a direct mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find-

1. Number of bits in tag
2. Tag directory size

Q2. Consider a direct mapped cache of size 512 KB with block size 1 KB. There are 7 bits in the tag. Find-

1. Size of main memory
2. Tag directory size

Q3.Consider a direct mapped cache with block size 4 KB. The size of main memory is 16 GB and there are 10 bits in the tag. Find-

1. Size of cache memory
2. Tag directory size

numericals

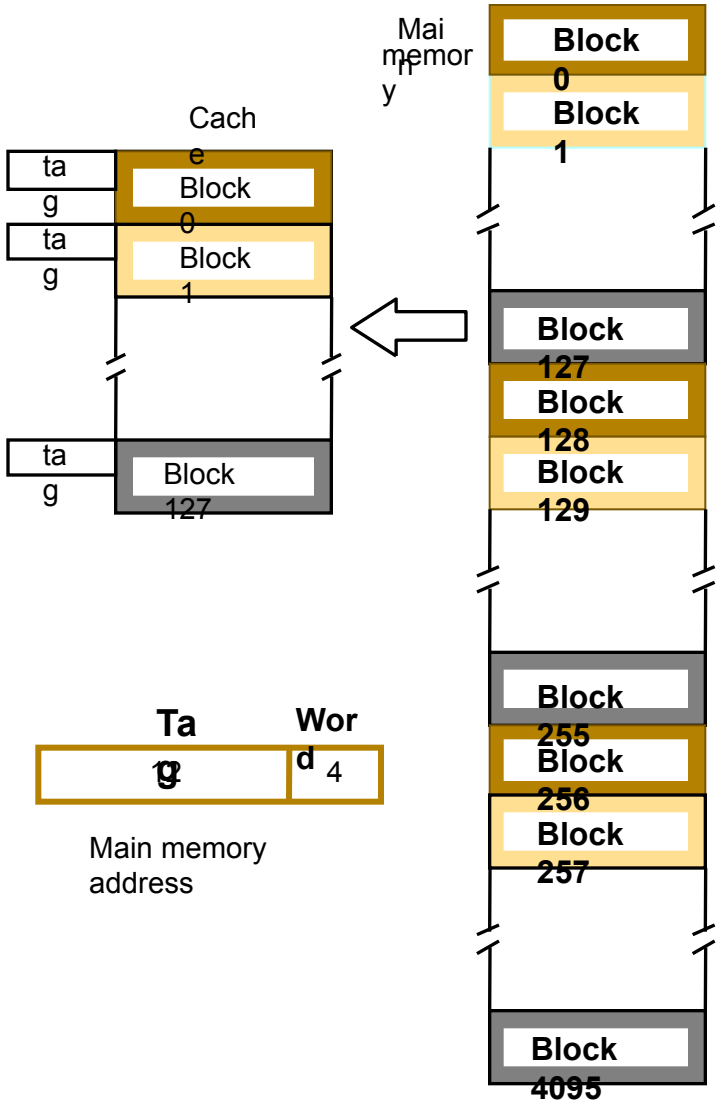
Q4. Consider a machine with a byte addressable main memory of 2^{32} bytes divided into blocks of size 32 bytes. Assume that a direct mapped cache having 512 cache lines is used with this machine. The size of the tag field in bits is _____.

Q5. An 8 KB direct-mapped write back cache is organized as multiple blocks, each of size 32 bytes. The processor generates 32 bit addresses. The cache controller maintains the tag information for each cache block comprising of the following-

- 1 valid bit
- 1 modified bit
- As many bits as the minimum needed to identify the memory block mapped in the cache

What is the total size of memory needed at the cache controller to store meta data (tags) for the cache?

Associative mapping



- *Main memory block can be placed into any cache position.*
- *Memory address is divided into two fields:*
  - *Low order 4 bits identify the word within a block.*
  - *High order 12 bits or tag bits identify a memory block when it is resident in the cache.*
- *Flexible, and uses cache space efficiently.*
- *Replacement algorithms can be used to replace an existing block in the cache when the cache is full.*
- *Cost is higher than direct-mapped cache because of the need to search all 128 patterns to determine whether a given block is in the cache.*

numericals

Q1. Consider a Fully Associative mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find-

1. Number of bits in tag
2. Tag directory size

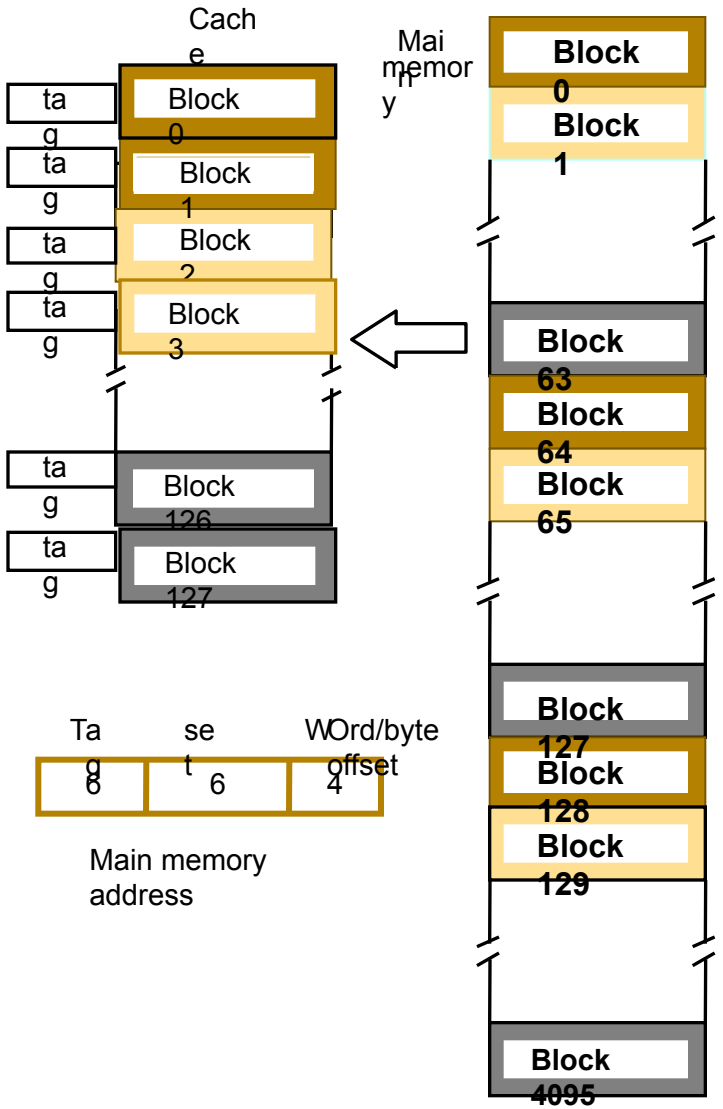
Q2. Consider a Fully Associative mapped cache of size 512 KB with block size 1 KB. There are 7 bits in the tag. Find-

1. Size of main memory
2. Tag directory size

Q3.Consider a Fully Associative mapped cache with block size 4 KB. The size of main memory is 16 GB and there are 10 bits in the tag. Find-

1. Size of cache memory
2. Tag directory size

Set-Associative mapping



Blocks of cache are grouped into sets.
Mapping function allows a block of the main memory to reside in any block of a specific set.
Divide the cache into 64 sets, with two blocks per set.
Memory block 0, 64, 128 etc. map to block 0, and they can occupy either of the two positions.
Memory address is divided into three fields:

- *6 bit field determines the set number.*
- *High order 6 bit fields are compared to the tag fields of the two blocks in a set.*

Set-associative mapping combination of direct and associative mapping.

In k-way set associative mapping,

- Cache lines are grouped into sets where each set contains k number of lines.
- A particular block of main memory can map to only one particular set of the cache.
- However, within that set, the memory block can map to any freely available cache line.
- The set of the cache to which a particular block of the main memory can map is given by-

Cache set number
= (Main Memory Block Address) Modulo (Number of sets in Cache)

- A particular block of main memory can be mapped to one particular cache set only.
- Block ‘j’ of main memory will map to set number (j mod number of sets in cache) of the cache.
- A replacement algorithm is needed if the cache is full.

After CPU generates a memory request,

- The set number field of the address is used to access the particular set of the cache.
- The tag field of the CPU address is then compared with the tags of all k lines within that set.
- If the CPU tag matches to the tag of any cache line, a cache hit occurs.
- If the CPU tag does not match to the tag of any cache line, a cache miss occurs.
- In case of a cache miss, the required word has to be brought from the main memory.
- If the cache is full, a replacement is made in accordance with the employed replacement policy.

Numericals

Q1. Consider a 2-way set associative mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find-

1. Number of bits in tag
2. Tag directory size

Q2Consider a 8-way set associative mapped cache. The size of cache memory is 512 KB and there are 10 bits in the tag. Find the size of main memory.

Q3. Consider a 4-way set associative mapped cache. The size of main memory is 64 MB and there are 10 bits in the tag. Find the size of cache memory.