Lecture - 13 (Cache Memory)

14 February 2024 19:29

Efficieny of memory hierarchy :-

Let efficiency (e) be the factor by which t_1 differs from averge time t_{Avg}.

i.e. $e = t_1 / t_{avg}$

$$\text{Efficieny } e = \frac{t_1}{H \cdot t_1 + (1-H) t_2} \quad \text{—} ①$$

Let $r = t_2 / t_1$

which is access time ratio of two levels.

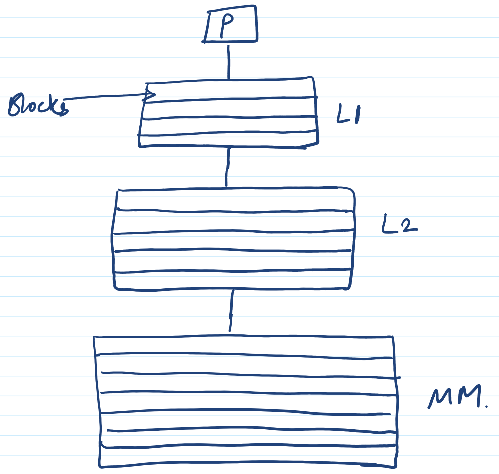
putting in ①

$$\Rightarrow e = \frac{1}{H + (1-H) r}$$

Speedup gained by Memory Hierarchy :-

$$S = \frac{t_2}{H \cdot t_1 + (1-H) t_2} = \frac{1}{H/r + (1-H)}$$

Block :- The smallest unit of information transferred between two levels.



Block placement :- where is the block placed.

Block Identification :- Identify the block if it is matched with the upper level of the memory

Block replacement :- Which block is to be replaced on a miss.

Common Memory Hierarchies :-

Cache / Main Memory Hierarchy

→ 2 to 4 levels

→ Managed by hardware

→ Objective :
To provide fast average memory access

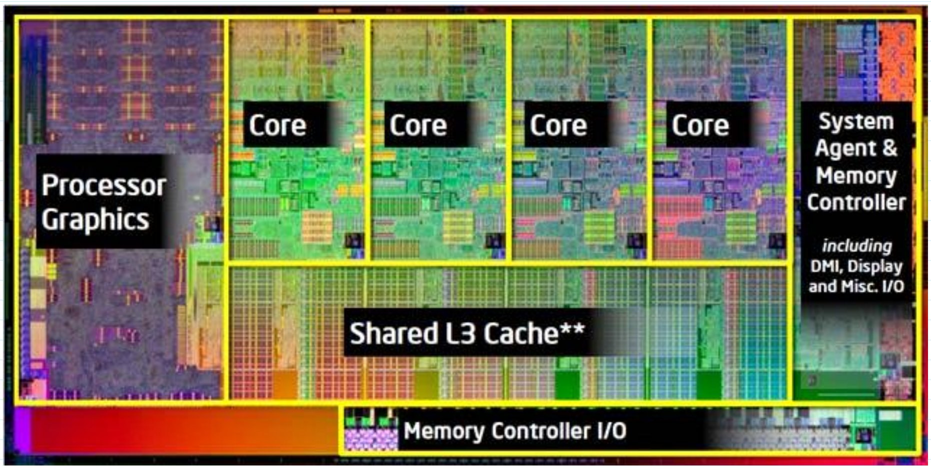
Main Memory / Secondary Memory Hierarchy

→ 2 levels

→ Managed by software (OS)

→ Objective :
To provide large memory space for users

Intel Core i7 - 4790K Processor :-



Core - A single processing unit within a CPU that can execute instructions.

4 cores - every core has its own L1 and L2 cache

L3 is shared by all the cores.

Cache Memory :-

→ A fast memory between processor and main memory.

→ [illegible] fast memory [illegible]

Read/Write strategies, Block replacement, Mapping Techniques etc.

→ For fast execution, frequently used data and instructions are brought into the cache.

→ First time there is definitely a miss and then data is brought from lower levels into Cache.

→ Cache Memory is logically divided into blocks/ lines, where every block typically contains 8 to 256 bytes.

→ When the CPU wants to access a word in memory, a special hardware first checks whether it is present in cache memory.

Present
(Cache hit)

↓

Word is directly accessed from cache memory.

Absent
(Cache Miss)

↓

The block containing the requested word is brought from main memory to cache.

Block Placement :–

Where can the block be placed in the cache?

→ determined by mapping algorithm.

which main memory blocks can reside in which cache memory blocks.

only a small subset of main memory blocks can be held in cache memory.

- Direct Mapping
- Associative Mapping
- (N-way) Set Associative Mapping.

A 2-level Cache / Main Memory Hierarchy

→ Cache Memory :-

256 blocks/lines of 32 words each

Total size = 8192 (8K) words.

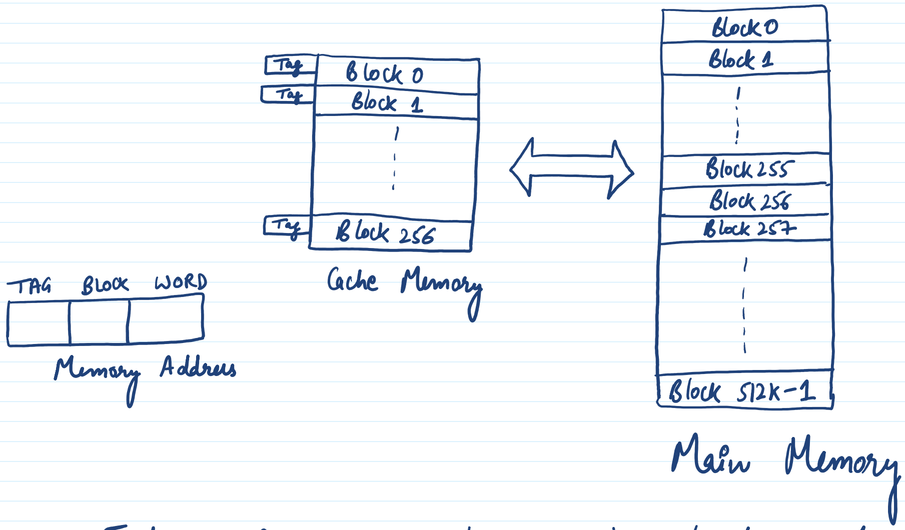
→ Main Memory :-

Total size = 16 M words = 2^{24}

⇒ 24-bit addressable

No. of 32-word blocks = $\frac{16M}{32}$ = 512K

1. Direct Mapping :-



→ Each main memory block can be placed in only one block in the cache.

The mapping function is :-

$$\text{Cache Block} = \frac{(\text{Main Memory Block})}{(\text{No. of cache blocks})}$$

→ Direct Memory Address

TAG	BLOCK	WORD
11	8	5

32 Word Memory → $\boxed{2^5}$ ⇒ 5 bits for each word

256 Blocks → $\boxed{2^8}$ → 8 bits to represent each block.

$$TAG = \frac{\text{\# of blocks in Main Memory}}{\text{\# of blocks in Cache Memory}}$$

TAG → which block of main memory is mapped to a particular block of cache memory.

$$TAG = \frac{512K}{256} = \frac{2^{19}}{2^{8}} = \boxed{2^{11}}$$