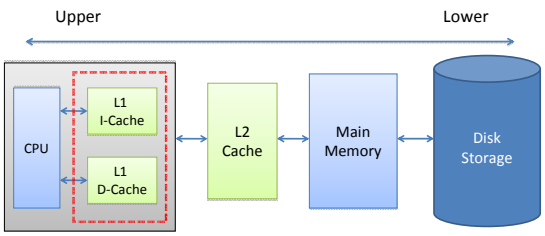
COMP4611 Tutorial 8
Memory System

Levels in Memory Hierarchy

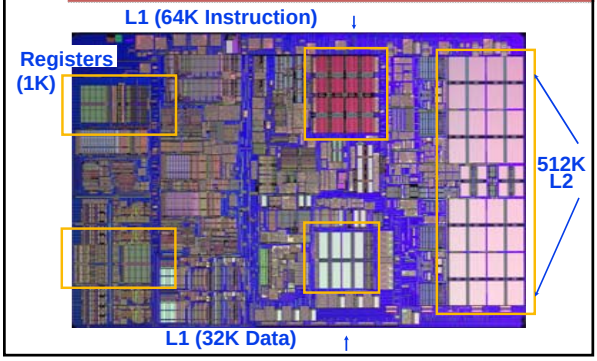
cheap technology → available memory ↑
fast technology → access speed ↑



Memory Hierarchy

Level	1	2	3	4
Name	Registers	Cache	Main Memory	Disk Storage
Typical size	<1KB	<16MB	<64GB	>1TB
Access time (ns)	0.25-0.5	0.5-25	50-250	5,000,000
Managed by	Compiler	Hardware	Operating System	Operating System/ Operator
Speed	Fastest			Slowest
Capacity	Smallest			Biggest
Cost/bit	Highest			Lowest

iMac's PowerPC 970



Design Issues

- Variable factors affect the cache design
  - Cache size
    - Larger cache -> shorter latencies
  - Cache speed and latency
    - Increasing speed shorten access latency
  - Associativity
    - one-way(direct mapping), two-way, four-way or eight-way
  - Cost

Fast and large caches are **expensive**

- Factors are *inter-related* -> Difficult to achieve the "best cache"

Cache Concepts

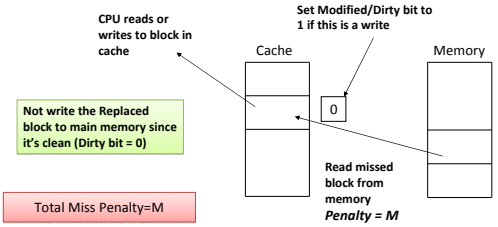
- Cache Hit
  - Data requested by the CPU is present in the cache
- Cache Miss
  - Data requested by the CPU is not present in the cache
  - On a cache miss, a block brought from the main memory may replace an existing cache block
- Hit Rate (or Hit Ratio)
  - The percentage of accesses that result in cache hits
- Cache Replacement Policies
  - FIFO (First In First Out)
  - LRU (Least Recently Used)
  - LFU (Least Frequently Used)
  - MFU (Most Frequently Used)
  - Random

Cache Concepts (cont.)

- Cache Write Policies
  - Write Through
    Data is written to both the cache block and to a block of main memory
  - Write Back
    Data is written only to the cache block ;
    Modified cache block is written to main memory when it has to be replaced
- Cache Write Miss Policies
  - Write Allocate
    The cache block is allocated on a write miss, followed by write hit actions
  - No Write Allocate
    Write misses do not affect the cache;
    The block is only modified in the main memory

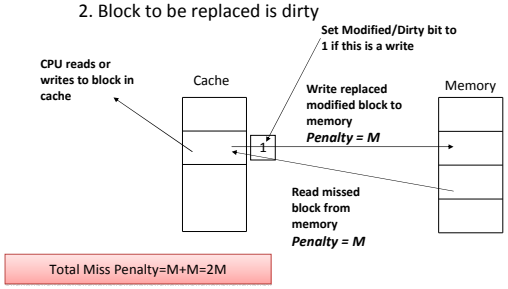
Cache Miss Operation

Assume: 1. Write-Back Cache with Write-Allocate
2. Block to be replaced is clean



Cache Miss Operation

Assume: 1. Write-Back Cache with Write-Allocate
2. Block to be replaced is dirty



Example 1

Suppose a computer's address size is 64 bits (using byte addressing), the cache size is 64 Kbytes (1 K = 2^{10} bytes), the block size is 64 bytes and the cache is 8-way set-associative.

Compute the following quantities:
(i) the number of sets in the cache
(ii) the number of index bits
(iii) the number of tag address bits in a block

Example 1 - Solution

(i) This is an 8-way set associative cache, the size of each set is
8 * Block_size = 512 bytes
Thus, # Sets = Cache_size / Set_size = 64KB / 512 B = **128 Sets**

(ii) The number of index bits is determined by the number of sets.
Since there are 128 sets, **7** bits are needed as the index bits (2^7 = 128).

(iii) The number of tag address bits is determined by the total address size, the number of index bits and the number of offset bits.
(Tag address bits) = # (Address Bits) – # (index bits) - # (offset bits)
(offset bits) = 6 (block size is 64 bytes and 2^6 = 64)
Thus, # (Tag address bits) = 64 – 7 – 6 = **51**

Average Memory Access Time (AMAT)

AMAT can be expressed by Hit time, Miss rate and Miss penalty on different cache levels.

For example,

AMAT = Hit time + Miss rate x Miss penalty (1-level)

AMAT = Hit time$_{L1}$ + Miss rate$_{L1}$ x (Hit time$_{L2}$ + Miss rate$_{L2}$ x Miss penalty$_{L2}$) (2-level)

FYI: Not all the cases are included. e.g., 3-level cache

Example 2

- Suppose that in 1000 memory references there are 40 misses in the first level cache and 20 misses in the second level cache. What are the various miss rate?
- Assume the miss penalty from the L2 cache to memory is 100 clock cycles, the hit time of the L2 cache is 10 clock cycles, the hit time of L1 is 1 clock cycle, and there are 1.5 memory references per instruction. What is the average memory access time and average stall cycles per instruction? Ignore the impact of writes.

Example 2 - Solution

1. Miss rate for L1 (either Local or Global) = 40/1000 = 4%

   Local miss rate for L2 = 20/40 = 50%

   Global miss rate for L2 = 20/1000 = 2%

2. AMAT = Hit time$_{L1}$ + Miss rate$_{L1}$ x (Hit time$_{L2}$ + Miss rate$_{L2}$ x Miss penalty$_{L2}$)
   = 1 + 4% x (10 + 50% x 100)
   = 3.4 clock cycles

   Average memory stalls per instruction
   = (AMAT – 1) x 1.5
   = (3.4 – 1) x 1.5
   = 3.6

AMAT = 1 + Stall Cycles Per Memory Access

Virtual Memory

- Definition
  - It gives an application program the impression that it has contiguous working memory, while in fact it may be physically fragmented and may even overflow on to disk storage
- Motivations
  - Allows the same program to run in any location in physical memory
- Different terminology comparing with cache
  - Virtual memory block is called a page
  - A virtual memory miss is called a page fault

Virtual & physical address

- Physical address
  - Instruction or data address in *main memory*
  - 256 MB main memory -> 28-bit physical address
- Virtual address
  - Decided by ISA
- Virtual address = **virtual page #** and **page offset**
  - Page # identifies a particular page
  - Page offset identifies a byte within that page
- Physical address = **physical page #** and **page offset**
- Address translation
  - Virtual address issued by the processor needs to be translated into the physical address

Example 3 & Solution

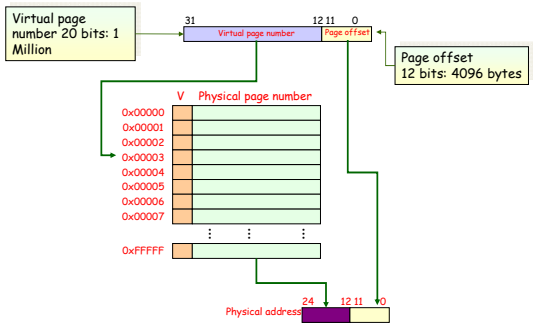
The page size on a byte-addressed machine is 16 KB. The machine has 1 GB of main memory. The virtual address of the machine has 32 bits. What are the sizes of the virtual page #, physical page #, and page offset fields?

Main memory size = 2^{30} bytes,
so physical address is 30 bits.
Page size = 2^{14} bytes,
so page offset is 14 bits.
Virtual page # size = 32 - 14 = 18 bits.
Physical page # size = 30 - 14 = 16 bits.

Paging

- Each process has its own page table
- Use page number as an index into the page table
- Each page table entry contains the physical page number of the corresponding page in main memory
- A valid bit to indicate whether the page is in main memory or not
- A modify bit to indicate whether the page has been altered or not
  - If no change, the page does not have to be written to the disk when it needs to be swapped out
- Replacement policies

Page Table



TLB

- *TLB - translation lookaside buffer*
  - Improve the speed of virtual address translation

Size: 8 - 4,096 entries
Hit time: 0.5 - 1 clock cycle
Miss penalty: 10 - 30 clock cycles
Miss rate: 0.01% - 1%

- A TLB entry is like a cache entry where the tag holds portions of the virtual address and the data portion holds the physical page number, protection field, valid bit, and dirty bit.

TLB (cont.)

- If the requested address is present in the TLB, the physical address can be used to access memory.
- If the requested address is not in the TLB, the translation proceeds using the page table, which is slower to access.

Overall operation of memory hierarchy