HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER ARCHITECTURE

Practical session - Week 7

Lecturers: Assoc. Prof. Dr. Cuong Pham Quoc
Mr. Kieu Do Nguyen Binh

Class: CC04 Group: 8

Members:	Nguyễn Minh Hùng	Student ID: 1952737
	Phạm Nhật Hoàng	Student ID: 1952703
	Nguyễn Bá Minh Hưng	Student ID: 1952748

Question 1. A memory uses 32 bits for address, the size of cache is 4MB, the size of block is 256B. The system supports byte access. Determine the size of tag, index, offset in each following scenarios:

1. Direct mapped
2. 4-way set associative
3. Fully associative

Answer:

4 MB = 4 x 2^{10} bytes

1. Directed mapped

Block size = 256 B = 2^8 bytes => **8 bits for offset.**

#blocks in main memory = $\frac{memorysize}{blocksize} = \frac{2^{32}}{256}$ =2^{24} blocks

-> block address = 24 bit

#blocks in cache = $\frac{cachesize}{blocksize} = \frac{4\times2^{20}}{256} = 2^{14}blocks$

-> block index = 14 bit.
Tag = block address – block index = 24 – 14 = 10 bit.

2. 4-way set associative

Block size = 256 B = 2^8 bytes => **8 bits for offset.**

#blocks in main memory = $\frac{memorysize}{blocksize} = \frac{2^{32}}{256}$ =2^{24} blocks

-> block address = 24 bit

#cache sets = $\frac{cachesize}{4\times blocksize} = \frac{4\times2^{20}}{4\times256} = 2^{12}blocks$

-> index = 12 bits
Tag = block address – block index = 24 – 12 = 12 bit

3. Fully associative

Block size = 256 B = 2^8 bytes => **8 bits for offset.**

#blocks in main memory = $\frac{memorysize}{blocksize} = \frac{2^{32}}{256}$ =2^{24} blocks

-> block address = 24 bit

#cache sets = 1 = 2^0
-> Index = 0 (fully associative)
Tag = block address – block index = 24 - 0 = 24 bit

Question 2. A memory has a size of 256MB, the size of cache is 256KB, each block contains 64 words. The system supports half-word access. Determine the tag, index, offset with the following cache's configuration:

1. Direct mapped
2. 8-way set associative.
3. Fully associative

Answer:

mem size = 256 MB = $2^8 \times 2^{20}$ bytes = 2^{28} bytes
cache size = 256 KB = $2^8 \times 2^{10}$ bytes = 2^{18} bytes
=> #bits in physical address = 28-bit address.

1. Direct mapped

#block offset = $\log_2 64 + 1 = 7 bits(half-word)$

#blocks in MM = $\frac{memsize}{blocksize} = \frac{2^{28}}{2^6} = 2^{22}$ blocks.

=> block address = 22 bits.

#block in cache = $\frac{cachesize}{blocksize} = \frac{2^{18}}{2^6} = 2^{12}$ blocks
=> block index = 12 bits.
=> Tag = #bits in physical address – (block index + block offset) = 28 - (12 + 7) = 9 bits.

2. 8-way set associative

#block offset = $\log_2 64 + 1 = 7 bits$

#blocks in MM = $\frac{memsize}{blocksize} = \frac{2^{28}}{2^6} = 2^{22}$ blocks.

=> block address = 22 bits.

#sets in cache = $\frac{cachesize}{8 \times blocksize} = \frac{2^{18}}{8 \times 2^6} = 2^9$ blocks
=> block index = 9 bits.
=> Tag = #bits in physical address – (block index + block offset) = 28 - (9 + 7) = 12 bits.

3. Fully associative

#block offset = $\log_2 64 + 1 = 7 bits$

#blocks in MM = $\frac{memsize}{blocksize} = \frac{2^{28}}{2^6} = 2^{22}$ blocks.

=> block address = 22 bits.

#sets in cache = 2^0 = 1 set. => block index = 0 bits.
=> Tag = #bits in physical address – (block index + block offset) = 28 - (0 + 7) = 21 bits.

Question 3. A system integrates a 256B cache that is configured as direct

mapped. The size of each cache block is 4 words. The system supports byte access. Assume that we access consecutively the following addresses:
$0 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 65 \rightarrow 1 \rightarrow 67 \rightarrow 46 \rightarrow 1 \rightarrow 70 \rightarrow 2 \rightarrow 0$

Determine the number of HIT/MISS with the following configuration:

1. Direct mapped
2. 2-ways set associative
3. 4-ways set associative
4. Full associative

Answer:
1. Direct mapped
0 (miss) -> 4(miss) -> 1(miss) -> 5(miss) ->65(miss) -> 1(miss) ->67(miss) -> 46(miss) -> 1(hit)->70(miss)->2(miss) -> 0(hit)
=> hit/miss ratio = 2/10 = 1/5
2. 2-way set associative
0 (miss) -> 4(miss) -> 1(miss) -> 5(miss) ->65(miss) -> 1(hit) ->67(miss) -> 46(miss) -> 1(hit)->70(miss)->2(miss) -> 0(hit)
=> hit/miss ratio = 3/9 = 1/3
3. 4-ways set associative
0 (miss) -> 4(miss) -> 1(miss) -> 5(miss) ->65(miss) -> 1(hit) ->67(miss) -> 46(miss) -> 1(hit)->70(miss)->2(miss) -> 0(hit)
=> hit/miss ratio = 3/9 = 1/3
4. Fully associative
same as 4-way and 2-way
=> hit/miss ratio = 3/9 = 1/3

Question 4. Define the following concept:

- Page
- Page fault
- Cache miss
- Write bach/ Write through
- PTE
- TLB

Answer:

- Page
  A page, memory page, or virtual page is a fixed-length contiguous block of virtual memory, described by a single entry in the page table. It is the

smallest unit of data for memory management in a virtual memory operating system (wikipedia).

- Page fault
  A page fault (sometimes called #PF, PF or hard fault) is a type of exception raised by computer hardware when a running program accesses a memory page that is not currently mapped by the memory management unit (MMU) into the virtual address space of a process (wikipedia).

- Cache miss
  A cache miss is a failed attempt to read or write a piece of data in the cache, which results in a main memory access with much longer latency. There are three kinds of cache misses: instruction read miss, data read miss, and data write miss.(wikipedia).

- Write bach/ Write through
  The cache mechanism includes write-through and write-back.
  Write-through: Write is done synchronously - both to the cache and to the backing store.
  Write-back (or Write-behind): Writing is done only to the cache. A modified cache block is written back to the store, just before it is replaced.
  Write-through: When data is updated, it is written to both the cache and the back-end storage. This mode is easy foris slow in data writing because data has to be written to both the cache and the storage.
  Write-back: When data is updated, it is written only to the cache. The modified data is written to the back-end storage only when data is removed from the cache. This mode has fast data write data will be lost if a power failure occurs before the updated data is written to the storage. (stackoverflow)

- PTE
  Page table has page table entries where each page table entry stores a frame number and optional status (like protection) bits. Many of status bits used in the virtual memory system. The most important thing in PTE is frame Number. (geeksforgeeks).

- TLB
  A translation lookaside buffer (TLB) is a memory cache that is used to reduce the time taken to access a user memory location. It is a part of the chip's memory-management unit (MMU). (wikipedia).

Question 5. Calculate the average CPI of a pipeline system where the miss rate of instruction memory is 5%, the miss rate of data memory is 10%. Miss penalty is 100 cycles. Base CPI is 1.5. The proportion of load/store instructions is 36%.

Answer:
Miss cycles per instruction (miss CPI)
– I-cache: $0.05 \times 100 = 5$ cycles/ins
– D-cache: 0.36 x $0.1 \times 100 = 3.6$ cycles/ins
=> Average CPI $= 1.5 + 5 + 3.6 = 10.1$