School of
Computing Science

CSC1104 – Computer Organisation and Architecture

Tutorial 2: Computer Functions and Cache Memory

Assoc. Prof. Cao Qi
Qi.Cao@Glasgow.ac.uk

Objectives

- Practice computer functions on instruction cycles and interrupts.
- Understand memory systems and cache memory.

Questions:

1. Consider a hypothetical 32-bit processor having 32-bit instructions composed of two fields: the first byte (8-bit) contains the opcode; and the remainder 24-bit as the address to load the operands (or load the immediate operand).
   a) What is the maximum directly addressable memory capacity (4-byte data length per unique address)?
   b) How many bits are needed for the program counter (PC) register and instruction register (IR)?

Solution:

(a). As number of address bits are 32-8=24 bits, there are unique addresses as: $2^{24} = 2^{4} \times 2^{20}$ = 16 MiBytes. Each unique address is 4 bytes data length.

Hence, the addressable memory capacity is: $2^{24} \times 4$ bytes = 16 MiBytes × 4 = 64 MiBytes.

(b). The program counter (PC) register loads the next address to fetch the instruction. As 24-bit is for the address to load the operands, the program counter will need 24 bits.

The instruction length is 32-bit, the IR will store the instructions. Hence, IR needs 32 bits.

2. A computer has a cache, main memory, and a disk used for virtual memory. If a referenced word by the CPU is in the cache, 20 ns (nanoseconds) are required to access it.

   If it is in main memory but not in the cache, 60 ns are needed to load it from the memory into the cache, and then the word is referenced by the CPU from the cache.

   If the word is not in main memory, 12 ms (milliseconds) are required to fetch the word from disk to the memory, followed by 60 ns to copy it from the memory to the cache, and then the word is referenced by the CPU from the cache.

   The cache hit ratio is 0.9. The main memory hit ratio is 0.6. What is the average time in nanoseconds (ns) required to access a referenced word by the CPU on this computer?

Solution:

There are three layers of storages to consider as follows.

Location	Probability (hit ratio)	Total time for access word (ns)
In cache	0.9	20
Not in cache, but in main memory	(1 - 0.9) × 0.6 = 0.06	60 + 20 = 80
Not in cache and main memory, but in disk	(1 - 0.9) × (1 - 0.6) = 0.04	12 ms + 60 ns + 20 ns = 12,000,080

The average access time would be:

Average time = 0.9 × 20 + 0.06 × 80 + 0.04 × 12,000,080 = 480,026 ns

3. There is a computer system with the following specifications:
   - Installed main memory: 4 GiB (GibiBytes), with each byte being accessible by a unique memory address.
   - Size of a cache line: 64 B (bytes).
   - Total cache capacity: 2 MiB (MebiBytes).
   - Direct mapped cache method.

   Answer the following questions about this computer system.

   I. How many address bits are required to access each byte location in the main memory?

   II. How many cache lines are there?

   III. How many blocks in the main memory?

   IV. What is the address breakdown of the cache: _____ bits tag + _____ bits line + _____ bits word?

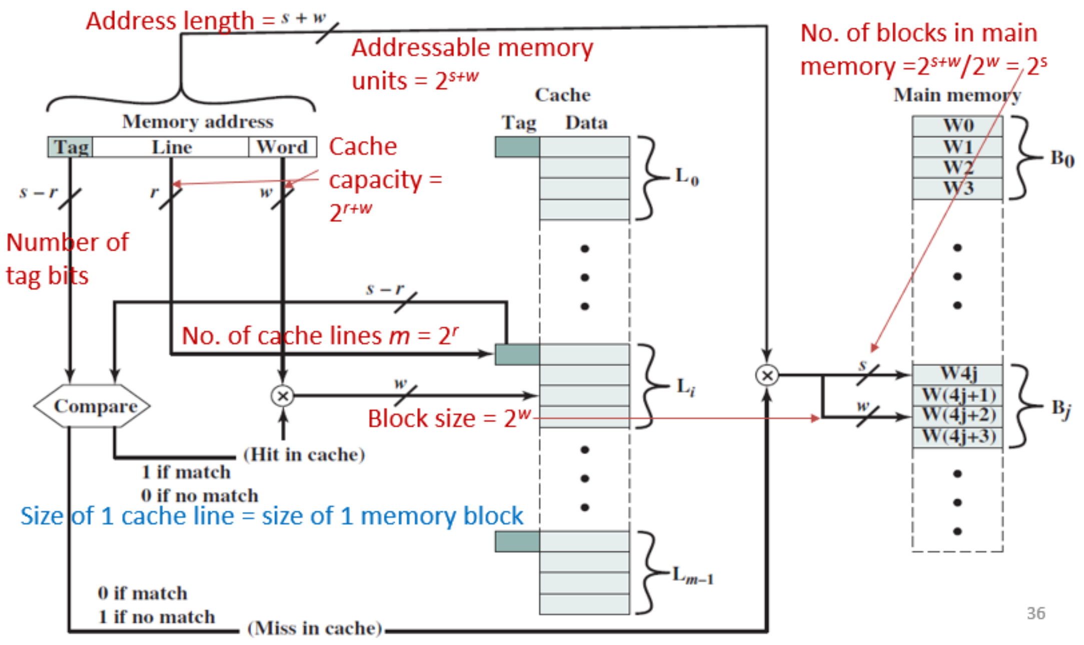
Analogy:

- The memory size can store 512 apple. Each apple has a unique serial number, such that we can access each apple following the serial number.
- In order to access each apple using the unique serial number, we can use 9 bits address (512 = 2^9) to represent each apple. The serial number of the apply starting from 0 0000 0000, 0 0000 0001, 0 0000 0010, …, 1 1111 1110, 1 1111 1111.
- All apple in the memory is divided into blocks (bags). Each bag contains 8 apple. We can access each apple in each bag using the unique serial number. We can use 3 bits address (8 = 2^3) to represent the serial number of each apple in the bag: 000, 001, 010, 011, 100, 101, 110, 111. *w* is the number of bit to represent the size of a memory block (a bag of apple). Hence, *w* = 3.
- In this case, all 512 apple in the memory can be stored into the number of bags with 8 apple per bag: 512 / 8 = 64. That is, there are 64 such bags (64 blocks) in the memory. We can use 6 bits address (64 = 2^6) to represent the number of the bags. *s* is the number of bit to represent the number of memory block (bags). Hence, *s* = 6.
- One drawer (a cache line) in the store room can hold a bag of apple, i.e., 8 apple.

- There are 32 such drawers in the store room, i.e., 32 cache lines. 32 bags of apple can be stored in the drawers. Each drawer has unique serial number. We can use 5 bits address ($32 = 2^5$) to represent the number of the drawers. *r* is the number of bit to represent the number of cache lines (drawers). Hence, *r* = 5.

Solution:

According to the figure on lecture notes Page 33 below, we can calculate the values of notations of *s, w, r, m*.



I. Based on the figure above, the addressable memory unit is 2^{s+w}, where *s* refers to address bits for number of memory blocks; *w* refers to number of address bits for one memory block.

For 4 GiB main memory (4 GiB = 2^{32}), to access each byte by a unique memory address, it means: *s* + *w* = 32 bits. **The number of address bits needed are: 32 bits.**

II. The size of 1 cache line = size of 1 memory block (i.e., 2^w). For 64 Bytes as the size of each cache line, it means $2^w = 64$ → *w* = 6 bits.

The cache capacity is 2^{r+w}. For 2 MiB cache capacity (2 MiB = 2^{21}), it means *r* + *w* = 21 bits.

Hence, $r + 6 = 21 \rightarrow r = 15$ bits.
The number of cache lines is: m = $2^r = 2^{15}$ = 32 Kibi = 32,768.

III. According to I and II, $s + w = 32$ bits, $w = 6$ bits, hence the value of s is: $s + 6 = 32 \rightarrow s = 26$ bits.
The number of blocks in the main memory is: $2^s = 2^{26}$ = 64 Mebi = 67,108,864.

IV. Based on the figure above, the address of the cache consists of 3 parts: number of tag bits (i.e., $s - r$ bits), number of line bits (r bits), and number of word bits (w bits).

As $s = 26$ bits, $r = 15$ bits, $w = 6$ bits, the address breakdown of the cache is:
__26 – 15 = 11_ bits tag + _15__ bits line + __6_ bits word.