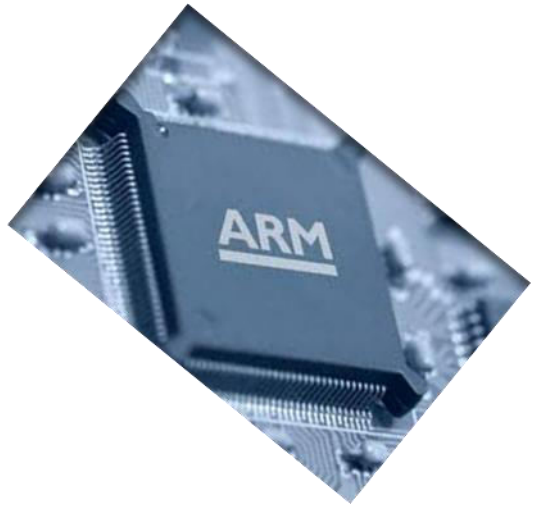
MICROPROCESSOR AND COMPUTER ARCHITECTURE

CACHE PERFORMANCE



Credits:
MPCA Team

PERFORMANCE CONSIDERATIONS

- A key design objective is to achieve the best possible performance at the lowest possible cost.
  - Price/performance ratio is a common measure.

- Performance of a processor depends on:
  - How fast machine instructions can be brought into the processor for execution.
  - How fast the instructions can be executed.

PERFORMANCE CONSIDERATIONS

- Memory hierarchy described earlier was created to increase the speed and size of the memory at an affordable cost.

- Data need to be transferred between various units of this hierarchy as well.
  - Speed and efficiency of data transfer between these various memory units also impacts the performance.

PERFORMANCE ENHANCEMENTS - PREFETCHING

- New data are brought into the processor when they are first needed.
- Processor has to wait before the data transfer is complete.
- Prefetch the data into the cache before they are actually needed, or a before a Read miss occurs.
- Prefetching should occur (hopefully) when the processor is busy executing instructions that do not result in a read miss.

PERFORMANCE ENHANCEMENTS - PREFETCHING

- Prefetching can be accomplished through software by including a special instruction in the machine language of the processor.
  - Inclusion of prefetch instructions increases the length of the programs.

- Prefetching can also be accomplished using hardware:
  - Circuitry that attempts to discover patterns in memory references and then prefetches according to this pattern.

PERFORMANCE ENHANCEMENTS - PREFETCHING

Prefetching can be accomplished through software by including a special instruction in the machine language of the processor.

- Inclusion of prefetch instructions increases the length of the programs.

Prefetching can also be accomplished using hardware:

- Circuitry that attempts to discover patterns in memory references and then prefetches according to this pattern.

PERFORMANCE ENHANCEMENTS – LOCKUP FREE CACHE

- Prefetching scheme does not work if it stops other accesses to the cache until the prefetch is completed.
- A cache of this type is said to be "locked" while it services a miss.
- Cache structure which supports multiple outstanding misses is called a lockup free cache.
- Since only one miss can be serviced at a time, a lockup free cache must include circuits that keep track of all the outstanding misses.
- Special registers may hold the necessary information about these misses.

CACHE PERFORMANCE

THE PROCESSOR PERFORMANCE EQUATION

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

$$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}$$

$$\text{CPU time} = \text{Instruction count} \times \text{Cycles per instruction} \times \text{Clock cycle time}$$

$$\frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}} = \frac{\text{Seconds}}{\text{Program}} = \text{CPU time}$$

CACHE PERFORMANCE

Different instruction types having different CPIs

$$\text{CPU clock cycles} = \sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i$$

$$\text{CPU time} = \left(\sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i \right) \times \text{Clock cycle time}$$

- One method to evaluate cache performance is to expand processor execution time.
- Add memory stall cycles - the number of cycles during which the processor is waiting for a memory access.

- **CPU Execution = (CPU clock cycles + Memory stall cycles) x Clock cycle execution time**

CACHE PERFORMANCE

- Equation assumes that CPU clock cycles include
  - The time to handle a cache hit.
  - The processor stalled during the cache miss.
- The number of memory stall cycles depends on
  - The number of misses
  - The cost per miss
  - This is called Miss penalty.

Memory stall cycles = Number of misses x Miss penalty.

$$= IC * \frac{Misses}{Instruction} * Miss\ penalty$$

where, Miss rate = Cache access that result in a miss.

$$x = \frac{\#\ of\ access\ that\ miss}{\#\ of\ access}$$

Note: miss rates and miss penalties are often different for read & writes.

CACHE PERFORMANCE

- Memory stall cycles can be defined in terms of
  - Number of memory accesses per instruction.
  - Miss penalty (in clock cycles for reads & writes)
  - Miss rate (for reads & writes)
- That is,

$Memory\ stall\ clock\ cycles$= IC * Reads per instruction * Read miss rate + IC * writes per instruction * Write miss rate

Normally, simplifying the complete formula,

- Combining reads & writes.
- Finding the average miss rates and miss penalty for reads & writes.

$$= IC * \frac{Memory\ Access}{Instruction} * Miss\ rate * Miss\ Penalty$$

Note: Miss rate is not the only most important measures of the cache design.

CACHE PERFORMANCE – EXAMPLE 1

Assume we have a computer where the cycles per instruction (CPI) is 1.0 when all memory accesses hit in the cache. The only data accesses are loads and stores, and these total 50% of the instructions. If the miss penalty is 25 clock cycles and the miss rate is 2%, how much faster would the computer be if all instructions were cache hits?

SOLUTION:

First compute the performance for the computer that always hits:

CPU execution time = (CPU clock cycles + Memory stall cycles) × Clock cycle
= (IC × CPI + 0) × Clock cycle
= IC × 1.0 × Clock cycle

Now for the computer with the real cache, first we compute memory stall cycles:

$$\text{Memory stall cycles} = IC * \frac{Memory\ Access}{Instruction} * Miss\ rate * Miss\ Penalty$$

CACHE PERFORMANCE – EXAMPLE 1

$$\text{Memory stall cycles} = IC \times (1 + 0.5) \times 0.02 \times 25$$
$$= IC \times 0.75$$

where the middle term (1 + 0.5) represents one instruction access and 0.5 data accesses per instruction. The total performance is thus

$$\text{CPU execution time cache} = (IC \times 1.0 + IC \times 0.75) \times \text{Clock cycle}$$
$$= 1.75 \times IC \times \text{Clock cycle}$$

The performance ratio is the inverse of the execution times:

$$\frac{CPU\ execution\ time\ (with\ cache)}{CPU\ execution\ time\ (without\ cache)} = \frac{1.75 * IC * Clock\ cycle}{1.0 * IC * Clock\ cycle} = 1.75$$

Hence, The computer with no cache misses is 1.75 times faster.

CACHE PERFORMANCE – EXAMPLE 2

	Cache #1	Cache #2
Block size	32-bytes	64-bytes
Miss rate	5%	4%

Which cache configuration would be better?
Assume both caches have single cycle hit times.
Memory accesses take 15 cycles, and the memory bus is 8-bytes wide

CACHE PERFORMANCE – EXAMPLE 2

Cache #1

For a 32-byte memory access takes 20 cycles:
1 (send address) + 15 (memory access) + 4 (four 8-byte transfers)
Miss Penalty=1+15+32B/8B=20cycles • AMAT=1+0.05x20=2

Cache #2

For a 64-byte memory access takes 24 cycles:
1 (send address) + 15 (memory access) + 8 (four 8-byte transfers)
Miss Penalty = 1 + 15 + 64B/8B = 24 cycles • AMAT = 1 + 0.04 x 24 = 1.9

CACHE PERFORMANCE – EXAMPLE 3

Assume that 33% of the instructions in a program are data accesses. The cache hit ratio is 97% and the hit time is one cycle, but the miss penalty is 20 cycles.

SOLUTION:

Memory stall cycles = Memory accesses x Miss rate x Miss penalty
= 0.33 x 0.03 x 20 cycles
= 0.2 cycles

CPI =[1 + 0.2] = 1.2 CPU
Execution Time = IC x 1.2 x cycle Time

CACHE PERFORMANCE – EXAMPLE 4

Consider a pipelined processor that has an average CPI of 1.8 without accounting for memory stalls. I-Cache has a hit rate of 95% and the D-Cache has a hit rate of 98%. Assume that memory reference instructions account for 30% of all the instructions executed. Out of these 80% are loads and 20% are stores. On average, the read-miss penalty is 20 cycles and the write-miss penalty is 5 cycles. Compute the effective CPI of the processor accounting for the memory stalls.

CACHE PERFORMANCE – EXAMPLE 4

- Cost of instruction misses = cache miss rate * read miss penalty
  = 0.05* 20
  = 1 cycle per instruction
- Cost of data read misses = fraction of memory reference instructions in program * fraction of memory reference instructions that are loads * D-cache miss rate * read miss penalty
  = 0.3 * 0.8 * 0.02 * 20
  = 0.096 cycles per instruction
- Cost of data write misses = fraction of memory reference instructions in the program * fraction of memory reference instructions that are stores * D-cache miss rate * write miss penalty
  = 0.3 * 0.2 * 0.02 * 5
  = 0.006 cycles per instruction
- Effective CPI = Avg CPI + Effect of I-Cache on CPI + Effect of D-Cache on CPI
  = 1.8 + 1 + 0.096 + 0.006 = **2.902**

Q & A
Cache Performance