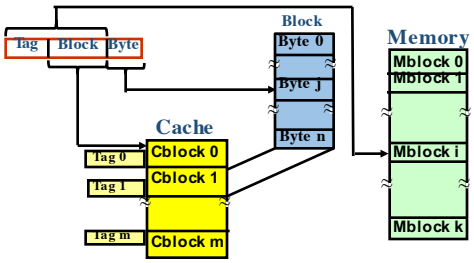
Caches: Performance



On this Lesson

- Issues regarding performance of caches
- Measuring Performance
- Performance parameters
- Performance improvement methods

Performance of a Microprocessor

- Two main factors
  - *Wasted cycles due to pipeline hazards*
  - *Memory latency*
- Memory latency effect is reduced by caches
  - *On hits there is no response delay (assuming one cycle access time)*
  - *On misses there is a response delay due to the transfer of memory blocks*
- In the analysis of performance of pipelines there were three non stated assumptions:
  - *There was a cache serving memory access*
  - *Memory hit time was one cycle*
  - *The only misses were due to missed predictions (a miss occurred when fetching the first location of the instruction stream that should be executed).*

Cache Performance Issues

- A cache may respond as quick as one CPU cycle on a hit.
- Misses introduce a miss penalty due to the memory latency of each memory access needed to transfer a memory block.
- The miss penalty could be different for instructions misses and data misses (load/store misses)
- The miss penalty could be different for read misses (instruction fetches or loads) and write misses (stores).
- Misses involving the replacement of a written cache block requires the transfer of two blocks
  - *a write-back block from cache to memory*
  - *a miss block from memory to the cache.*

Cache Performance Measurements

- Cache performance is usually measured in terms of the number of cycles per instruction (CPI).
- The CPI involves three factors:
  - *the time it takes to access a memory location in the cache*
  - *the time penalty introduced by a miss*
  - *one stall cycle in the case of load/stores accesses due to the structural pipeline hazard*

Calculating CPI

Consider a cache with an access time of one cycle and a miss penalty of 25 cycles. For the following ARM program only structural hazards are generated, and that none of the branches do branch. Determine the CPI.

Instruction	Instruction Access Time	Structural Hazard Penalty	Instruction Access Result	Instruction Miss Penalty	Load/Store Access Result	Load/Store Miss Penalty
LDR	1	1	Hit		Hit	
ADD	1		Hit			
ORRS	1		Miss	25		
BNE	1		Hit			
STR	1	1	Hit		Miss	25
LDRB	1	1	Miss	25	Hit	
LDRH	1	1	Hit		Hit	
SUBS	1		Hit			
BEQ	1		Hit			
AND	1		Hit			

CPI = Total cycles/Total instructions = (10 + 4 + 50 +25)/10 =89/10 = 8.9

Calculating CPI by Formula

$CPI = CPI_{BASE} + CPI_{MISSES}$

CPI_{BASE} : Cache access time in cycles

$CPI_{MISSES} = CPI_{IM} + CPI_{DM}$

CPI_{IM} : CPI component due to instruction misses

CPI_{DM} : CPI component due to data (loads/stores) misses

$CPI_{IM} = MissRate_{INSTRUCTIONS} \times MissPenalty_{INSTRUCTIONS}$

$CPI_{DM} = (Stall\ Cycle + MissRate_{DATA} \times MissPenalty_{DATA}) \times \%\ load/store\ instructions$

Calculating CPI by Formula

Consider a cache with an access time of one cycle and a miss penalty of 25 cycles. For the following ARM program only structural hazards are generated, and that none of the branches do branch. Determine the CPI.

From previous slides:

$MissRate_{INSTRUCTIONS} = 2/10 = .2$

$MissRate_{DATA} = 1/4 = .25$

% load/store instructions = 4/10 = .4

$MissPenalty_{DATA} = MissPenalty_{INSTRUCTIONS} = 25$

$CPI_{IM} = MissRate_{INSTRUCTIONS} \times MissPenalty_{INSTRUCTIONS} = .2 \times 25 = 5$

$CPI_{DM} = (Stall\ Cycle + MissRate_{DATA} \times MissPenalty_{DATA}) \times \%\ load/store\ instructions$

$CPI_{DM} = (1 + .25 \times 25) \times .4 = 2.9$

$CPI_{MISSES} = CPI_{IM} + CPI_{DM} = 5 + 2.9 = 7.9$

$CPI = CPI_{BASE} + CPI_{MISSES} = 1 + 7.9 = 8.9\ cycles/instruction$

Cache Performance: An Example

Consider a CPU running at 1Ghz with a cache with a hit time of one cycle, a miss penalty of 20ns, and a hit rate of 98% for instructions and data accesses. Determine the CPI if 25% of the instructions are data accesses.

Cycle period = 1/1Ghz = 1 ns

$MissPenalty_{INSTRUCTIONS} = MissPenalty_{DATA} = 20\ ns/1\ ns = 20\ cycles$

$MissRate_{INSTRUCTIONS} = MissRate_{DATA} = 100\% - 98\% = 2\% = .02$

$CPI_{IM} = .02 \times 20 = .4$

$CPI_{DM} = (1 + .02 \times 20) \times .25 = .35$

$CPI_{MISSES} = CPI_{IM} + CPI_{DM} = .4 + .35 = .75\ cycles\ per\ instruction$

$CPI_{BASE} = 1\ cycle/instruction$

$CPI = CPI_{BASE} + CPI_{MISSES} = 1 + .75 =$ ***1.75 cycles per instruction***

Another Example:

CPU running at 500Mhz. A cache with a hit time of one cycle, instruction miss penalty of 16ns, data miss penalty of 24ns, instruction hit rate of 97%, data hit rate of 96%. Determine the CPI if 30% of the instructions are data accesses.

Cycle period = 1/500Mhz = 2 ns

$MissPenalty_{INSTRUCTIONS} = 16\ ns/2\ ns = 8\ cycles$

$MissPenalty_{DATA} = 24ns/2ns = 12\ cycles$

$MissRate_{INSTRUCTIONS} = 100\% - 97\% = .03$, $MissRate_{DATA} = 100\% - 96\% = .04$

$CPI_{IM} = .03 \times 8 = .12$

$CPI_{DM} = (1 + .04 \times 12) \times .3 = .444$

$CPI_{MISSES} = CPI_{IM} + CPI_{DM} = .12 + .444 = .564\ cycles\ per\ instruction$

$CPI_{BASE} = 1\ cycles/instruction$

$CPI = CPI_{BASE} + CPI_{MISSES} = 1 + .564 =$ ***1.564*** *cycles per instruction*

Cache Performance Improvement

- Split Caches
- Prefetching
- Increasing Cache-Memory Data Bus
- Interleaved Memory

Split Caches

- Instead of using one cache to handle instructions and data, some microprocessors use a cache to handle instructions and another to handle data.
- The advantage of this configuration is that it eliminates the one cycle stall in the instruction pipeline due to load/store conflicts with instruction fetches (structural hazard).
- Data Miss Penalty

  Unified cache: *Stall Cycle + Miss Penalty x Miss Rate*

  Split cache: *Miss Penalty x Miss Rate*

Split Cache Example

Consider a CPU running at 1Ghz with a split cache with a hit time of one cycle, a miss penalty of 20ns, and a hit rate of 98% for instructions and data accesses. Determine the CPI if 25% of the instructions generate load/store accesses.

$MissRate_{INSTRUCTIONS} = MissRate_{DATA} = 100\% - 98\% = 2\% = .02$

$MissPenalty_{INSTRUCTIONS} = MissPenalty_{DATA} = 20\ ns/1\ ns = 20\ cycles$

$CPI_{IM} = .02 \times 20 = .4$

$CPI_{DM} = (.02 \times 20) \times .25 = .1$

$CPI_{MISSES} = CPI_{IM} + CPI_{DM} = .4 + .1 = .5\ cycles\ per\ instruction$

$CPI_{IDEAL} = 1\ cycle/instruction$

$CPI = CPI_{IDEAL} + CPI_{MISSES} = 1 + .5 = \mathbf{1.5\ cycles\ per\ Instruction}$

** A unified cache yields a CPI of 1.75 for the same case (slide 9).*

13

Prefetching

- Allocates cache blocks from primary memory into the cache before they are needed
- It could has the following effects:
  - *Improve hit ratio*
  - *Generate unnecessary blocks transfers*
  - *Reduce hit ratio*

14

Increasing Cache-Memory Data Bus

- *The transfer of blocks between the cache and memory takes place through a bus with a bandwidth of several bytes.*
- *Normally, the block size is larger than the bus bandwidth.*
- *Transfer of a block requires multiple memory access*
- *Typically each memory access requires a few cycles to send the address, a memory latency time, and a few cycles to transfer the data (all this account for the miss penalty).*
- *Increasing the data bus bandwidth reduces the number of memory access, what results in a reduction of the overall miss penalty.*
- *Due to hardware limitations the data bus cannot be increased arbitrarily.*

15

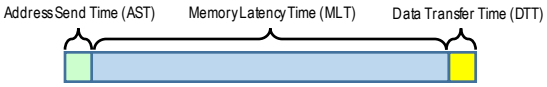
Data Transfer Between Memory and Cache

- The transfer of one block to a cache normally requires several consecutive memory accesses.
- The total transfer time of a memory access has three components:

  Address Send Time (AST) – time to send the address an initiate the access

  Memory Latency Time (MLT) – time the memory takes to complete the operation

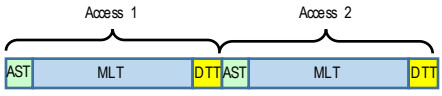
  Data Transfer Time (DTT) – Time it takes to transfer data through the data bus



16

Block Transfer on Regular Memory

If a transfer of a block requires two memory accesses then, the total transfer time is 2x(AST + MLT + DTT)



In general the total transfer time for a block of a regular memory is,

BTTT = **n** x (**AST** + **MLT** + **DTT**) = Miss Penalty,

where **n** is the number of memory accesses required to transfer a block

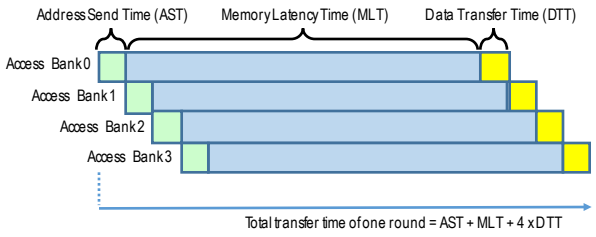
17

Interleaved Memory

- Memory is organized in banks of bytes, the size of the data bus, that can be accessed independently.
- A block is distributed among the banks.
- Transfer of a block requires rounds of consecutives accesses to each of the banks.
- Each access to successive banks is only separated by an address send time.

18

Memory Accesses on 4-Bank Interleaved Memory



19

Total Block Transfer Time of Interleaved Memories

In general the total transfer time for a block of an interleaved memory is,

BTTT = **m** x (**AST** + **MLT** + **k**•**DTT**) = **Miss Penalty**,

where

m = number of memory banks access rounds required to transfer a block

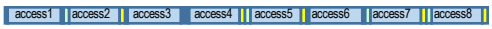
k = the number of banks

20

Interleaved vs. Non-Interleave Memory

Consider a cache-memory system with blocks of 64 bytes, an 8-byte data bus, AST = 1 cycle, DTT = 2 cycles, and MLT = 25 cycles.

A non-interleaved memory needs 8 memory accesses to transfer a block:



Miss Penalty = n x (AST + MLT + DTT) = 8 x (1 + 25 + 2) = **224 cycles**

A 4-bank interleaved memory needs 2 rounds of 4 memory accesses to transfer a block:

Miss Penalty = m x (AST + MLT + K•DTT) 2 x (1 + 25 + 4x2) = **68 cycles**

Lesson Outcomes

- Understand the factors that affect cache performance
- Determine cache performance in terms of CPI
- Know common methods for improving cache performance

22