Fall 2006

Problem 11 Computer Architecture Page 1 of 1

Fall 2006, Problem 11 Computer Architecture

Consider the three architectures below:

1- A single-core processor running at a 1GHz clock frequency. If this processor had an ideal cache (i.e., assuming no misses at all), then it would have a core CPI (Clocks Per Instruction) of 1.0 . However, the cache of this processor yields 0.01 misses per instruction.

2- A pipelined version of the above processor that runs at 1.2GHz. If this processor had an ideal cache, then it would have a core CPI of 1.2. However, the cache system of this processor yields 0.015 misses per instruction on average (smaller cache used compared to the above architecture).

3- A multi-core processor with speculative execution running at 800MHz. If this processor had an ideal cache, its core CPI would have been 0.22. However, the cache in this system is small and results in 0.02 misses per instruction. The designers of this processor have used dynamic scheduling to successfully hide 10% of the miss penalty on every miss on average. (Note: the miss rate of 0.02 does not model the 10% improvement. Instead, the 10% improvement can be modeled by assuming every miss incurs only 0.9 of the penalty compared to a processor which does not have dynamic scheduling).

a) Assuming the miss penalty is 100ns, determine the MIPS (Million Instructions Per Second) execution rate of these three processors.

b) For what miss penalty values is architecture 3 faster than architecture 1 in terms of MIPS?

c) Assuming the miss penalty is 100ns, which system benefits the most (in terms of percentage MIPS improvement, i.e., (newMIPS – oldMIPS)/oldMIPS) if its cache CPI is improved by 10%?

d) Assuming the miss penalty is 100ns, which system benefits the most MIPS percentage improvement if its core CPI gets improved by 10%?

Fall 2006 Solutions- Computer Architecture

Fall 2006, Problem 11 Computer Architecture – Solutions

PART (a)

CPI = coreCPI + cacheCPI

Clock cycle times are:
cycle1 = 1/1GHz = 1ns
cycle2 = 1/1.2GHz = 0.83ns
cycle3 = 1/0.8GHz = 1.25ns

missPenalty1 = 100ns/1ns = 100 cycles
missPenalty2 = 100ns/0.83ns = 120 cycles
missPenatly3 = 0.9*100ns / 1.25ns = 72 cycles // 0.9 multiplied because dynamic
// scheduling hides 10% of the
// miss penalty

cacheCPI = missRate * missPenalty
cacheCPI1 = 0.01 * 100 = 1
cacheCPI2 = 0.015 * 120 = 1.8
cacheCPI3 = 0.02 * 72 = 1.44

effectiveCPI = cacheCPI + coreCPI
CPI1 = 1 + 1 = 2
CPI2 = 1.2 + 1.8 = 3
CPI3 = 0.22 + 1.44 = 1.66

execrate = clock / CPI
execRate1 = 1000MHz / 2 = 500 MIPS
execRate2 = 1200MHz / 3 = 400 MIPS
execRate3 = 800MHz / 1.66 = 482 MIPS

Part (b)

Same as above, but instead of a 100ns miss penalty, we use variable x and solve for execRate3 > execRate1.

missPenalty1 = x / 1ns = x cycles
missPenatly3 = 0.9 * x / 1.25ns = 0.72 x cycles

cacheCPI = missRate * missPenalty
cacheCPI1 = 0.01 x
cacheCPI3 = 0.02 * 0.72 * x = 0.0144 x

effectiveCPI = cacheCPI + coreCPI

CPI1 = $1 + 0.01x$
CPI3 = $0.22 + 0.0144x$

execrate = clock / CPI
execRate1 = 1000MHz / ($1 + 0.01x$)
execRate3 = 800MHz / ($0.22 + 0.0144x$)

execRate3 > execRate1
$800 / (0.22 + 0.0144x) > 1000 / (1 + 0.01x)$ ➔ $x < 90.625$ns

Part (c)

newCPI1 = 1 + 1*0.9 = 1.9
newCPI2 = 1.2 + 1.8*0.9 = 2.82
newCPI3 = 0.22 + 1.44*0.9 = 1.516

newExecRate1 = 1000MHz / 1.9 = 526 ➔ improvement1 = (526-500)/500 = 5.2%
newExecRate2 = 1200MHz / 2.82 = 426 ➔ improvement2 = (426-400)/400 = 6.5%
newExecRate3 = 800MHz / 1.516 = 528 ➔ improvement3 = (528-482)/482 = 9.5%

Architecture 3 benefits the most in terms of percentage improvement. and it will beat the other architectures in terms of overall execution rate.

Part (d)

newCPI1 = 1*0.9 + 1 = 1.9
newCPI2 = 1.2*0.9 + 1.8 = 2.88
newCPI3 = 0.22*0.9 + 1.44 = 1.638

newExecRate1 = 1000MHz / 1.9 = 526 ➔ improvement = (526-500)/500 = 5.2%
newExecRate2 = 1200MHz / 2.88 = 417 ➔ improvement2 = (417-400)/400 = 6.1%
newExecRate3 = 800MHz / 1.638 = 488 ➔ improvement3 = (488-482)/482 = 1.2%

Architecture 2 benefits the most in terms of percentage improvement, but architecture 1 is the best in terms of execution rate.