Computer Architecture
DAT105
Exercise Session 5

Piyumal Ranawaka
piyumal@chalmers.se
M. Wqar Azar
waqarm@chalmers.se

Chalmers University of Technology, Sweden

October 3, 2021

Table of Contents

Table of Contents

Problem 8.1

A CPU designer has to decide on whether or not to add a new microarchitecture enhancement to improve performance (ignoring power costs) of a block (coarse-grain) multithreaded processor. In this processor a thread switch occurs only on a L2 cache miss. The cost of a thread switch is 60 cycles (time before a new thread can start executing). Assume that there are always enough ready threads to switch to on a cache miss. Also, it is given that the current L2 cache hit rate is 50%. The new microarchitectural block is a cache hit/miss predictor. The new predictor predicts whether a memory reference is going to hit or miss in L2 (note not L1) cache. The predictor is used to decide when to switch threads. If the predictor predicts a cache miss thread switching is initiated early. There are four scenarios to consider:

Problem 8.1

(a)The predictor predicts a L2 cache miss and the true outcome is also a L2 cache miss. In this case thread switching is initiated early and the thread switching cost is reduced to 20 cycles (from 60 cycles in the baseline).

Problem 8.1

(a)The predictor predicts a L2 cache miss and the true outcome is also a L2 cache miss. In this case thread switching is initiated early and the thread switching cost is reduced to 20 cycles (from 60 cycles in the baseline).

(b)The predictor predicts a L2 cache miss and the true outcome is a L2 cache hit. In this case an unnecessary thread switch has been initiated which increases the thread switching overhead to 120 cycles due to unnecessary pipeline flushes.

Problem 8.1

(a)The predictor predicts a L2 cache miss and the true outcome is also a L2 cache miss. In this case thread switching is initiated early and the thread switching cost is reduced to 20 cycles (from 60 cycles in the baseline).

(b)The predictor predicts a L2 cache miss and the true outcome is a L2 cache hit. In this case an unnecessary thread switch has been initiated which increases the thread switching overhead to 120 cycles due to unnecessary pipeline flushes.

(c)The predictor predicts a L2 cache hit and the true outcome is also a L2 cache hit. In this case no thread switching is initiated and there is no gain or loss

Problem 8.1

(a)The predictor predicts a L2 cache miss and the true outcome is also a L2 cache miss. In this case thread switching is initiated early and the thread switching cost is reduced to 20 cycles (from 60 cycles in the baseline).

(b)The predictor predicts a L2 cache miss and the true outcome is a L2 cache hit. In this case an unnecessary thread switch has been initiated which increases the thread switching overhead to 120 cycles due to unnecessary pipeline flushes.

(c)The predictor predicts a L2 cache hit and the true outcome is also a L2 cache hit. In this case no thread switching is initiated and there is no gain or loss

(d)The predictor predicts a L2 cache hit and the true outcome is also a L2 cache miss. This is a case of lost opportunity for an early thread switch and the machine pays the 60 cycle baseline switching penalty.

Problem 8.1

(Q1)Given these four scenarios what should be the predictor accuracy before the designer can be certain that this new microarchitectural block leads to a break-even point in performance.

Problem 8.1

(Q1)Given these four scenarios what should be the predictor accuracy before the designer can be certain that this new microarchitectural block leads to a break-even point in performance.

(Q2)If the L2 cache hit rate of the base machine is improved from 50% to 80% how does that impact predictor's accuracy requirements before achieving break-even point in performance?

Problem 8.1 (Part-1)

Lets assume

Hit rate of the L2 cache = H

Problem 8.1 (Part-1)

Lets assume

Hit rate of the L2 cache = H

The accuracy of the L2-cache hit/miss predictor = A

Problem 8.1 (Part-1)

Lets assume

Hit rate of the L2 cache = H

The accuracy of the L2-cache hit/miss predictor = A

Hit/Miss	Prediction	Outcome	Fraction	Penalty
Hit	Hit	Success	H*A	0

Problem 8.1 (Part-1)

Lets assume

Hit rate of the L2 cache = H

The accuracy of the L2-cache hit/miss predictor = A

Hit/Miss	Prediction	Outcome	Fraction	Penalty
Hit	Hit	Success	H*A	0
Hit	Miss	Failure	H*(1-A)	120

Problem 8.1 (Part-1)

Lets assume

Hit rate of the L2 cache = H

The accuracy of the L2-cache hit/miss predictor = A

Hit/Miss	Prediction	Outcome	Fraction	Penalty
Hit	Hit	Success	H*A	0
Hit	Miss	Failure	H*(1-A)	120
Miss	Hit	Failure	(1-H)*(1-A)	60

Problem 8.1 (Part-1)

Lets assume

Hit rate of the L2 cache = H

The accuracy of the L2-cache hit/miss predictor = A

Hit/Miss	Prediction	Outcome	Fraction	Penalty
Hit	Hit	Success	H*A	0
Hit	Miss	Failure	H*(1-A)	120
Miss	Hit	Failure	(1-H)*(1-A)	60
Miss	Miss	Success	(1-H)*A	20

Problem 8.1

Penalty with out predictor = (1-H)*60

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120 + (1-H)*(1-A)*60

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

The predictor is beneficial if

Penalty with out predictor > Total Penalty with Predictor

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

The predictor is beneficial if

Penalty with out predictor > Total Penalty with Predictor

(1-H)*60 > H*(1-A)*120

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

The predictor is beneficial if

Penalty with out predictor > Total Penalty with Predictor

(1-H)*60 > H*(1-A)*120 + (1-H)*(1-A)*60

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

The predictor is beneficial if

Penalty with out predictor > Total Penalty with Predictor

(1-H)*60 > H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Problem 8.1

Penalty with out predictor = (1-H)*60

Total Penalty with Predictor = H*A*0 + H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

Total Penalty = H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

The predictor is beneficial if

Penalty with out predictor > Total Penalty with Predictor

(1-H)*60 > H*(1-A)*120 + (1-H)*(1-A)*60 + (1-H)*A*20

$60 - 60H > 60H - 40A - 80H * A + 60$

$A > \frac{3H}{2*H+1}$

Problem 8.1 (part-1)

Q1: Given these four scenarios what should be the predictor accuracy before the designer can be certain that this new microarchitectural block leads to a break-even point in performance.

so $H = 50\% = 0.5$

$A > 0.75$

The predictor is beneficial if A is at least 75%

Problem 8.1 (part-2)

Q2: If the L2 cache hit rate of the base machine is improved from 50% to 80% how does that impact predictor's accuracy requirements before achieving break-even point in performance?

so $\text{H} = 80\% = 0.8$

$\text{A} > 0.923$

The predictor is beneficial if A is at least 92.3%

Table of Contents

Problem 8.1

Problem 8.8

Problem 8.8 (a)

Consider a simple 5-stage pipeline that is single threaded. The pipeline treats every cache miss as a hazard and freezes the pipeline. While executing a benchmark assume that a L1 cache miss occurs every 100 cycles, and each L1 cache miss takes 10 cycles to satisfy if the block is found in L2 or 50 cycles if L2 misses as well. An L2 cache miss occurs after 200 cycles of computation. Assume that the CPI in the absence of cache misses is one. What is the actual CPI taking into account cache miss latencies?

Solution 8.8 (a)

Base CPI (CPI0)$=1$

Solution 8.8 (a)

Base CPI (CPI0)$=1$

L1 cache Misses $=1$ per 100 cycles

Solution 8.8 (a)

Base CPI (CPI0)$=1$

L1 cache Misses $=1$ per 100 cycles
L1 cache miss penalty $=10$ cycles

Solution 8.8 (a)

Base CPI (CPI0)$=1$

L1 cache Misses $=1$ per 100 cycles
L1 cache miss penalty $=10$ cycles

L2 cache Misses $=1$ per 200 cycles

Solution 8.8 (a)

Base CPI (CPI0)$=1$

L1 cache Misses $=1$ per 100 cycles
L1 cache miss penalty $=10$ cycles

L2 cache Misses $=1$ per 200 cycles
L2 cache miss penalty $=50$ cycles

Solution 8.8 (a)

Base CPI (CPI0)= 1

L1 cache Misses = 1 per 100 cycles
L1 cache miss penalty = 10 cycles

L2 cache Misses = 1 per 200 cycles
L2 cache miss penalty = 50 cycles

Lets assume 200 instructions

Solution 8.8 (a)

Base CPI (CPI0)$=1$

L1 cache Misses $=1$ per 100 cycles
L1 cache miss penalty $=10$ cycles

L2 cache Misses $=1$ per 200 cycles
L2 cache miss penalty $=50$ cycles

Lets assume 200 instructions

Cycles $=$ Base CPI $*$ No of instructions $+$ Additional cycles due to cache misses

Solution 8.8 (a)

Base CPI (CPI0)$=1$

L1 cache Misses $=1$ per 100 cycles
L1 cache miss penalty $=10$ cycles

L2 cache Misses $=1$ per 200 cycles
L2 cache miss penalty $=50$ cycles

Lets assume 200 instructions

Cycles $=$ Base CPI $*$ No of instructions $+$ Additional cycles due to cache misses

Cycles $= 1 * 200 +$ Additional cycles due to cache misses

Solution 8.8 (a)

Cycles $=$ 200 $+$ Additional cycles due to cache misses

Solution 8.8 (a)

Cycles = 200 + Additional cycles due to cache misses
In 200 cycles, we will have two L1 cache misses

Solution 8.8 (a)

Cycles = 200 + Additional cycles due to cache misses
In 200 cycles, we will have two L1 cache misses
First will miss in L-1 cache, but hit in L-2 cache so it will have 10 cycles of penalty
Second will misses in L-1 cache and L-2 cache so it will have 50 cycles of penalty.
Cycles = 200 + 10 + 50 = 260

Solution 8.8 (a)

Cycles $=$ 200 $+$ Additional cycles due to cache misses
In 200 cycles, we will have two L1 cache misses
First will miss in L-1 cache, but hit in L-2 cache so it will have 10 cycles of penalty
Second will misses in L-1 cache and L-2 cache so it will have 50 cycles of penalty.
Cycles $= 200 + 10 + 50 = 260$
$CPI = \frac{260}{200} = 1.3$

Problem 8.8 (b)

Now consider the same example above but assume that hardware is now 2-way multi-threaded, similar to Figure 8.3. Assume that switching overhead is zero and there are two threads with identical cache miss behavior as described in the first case. What is the CPI of the each of the two programs on the 2-way multi-threaded machine? Did the CPI improve? If yes, explain how? If not, explain why one should bother with 2-way multi-threaded machine?

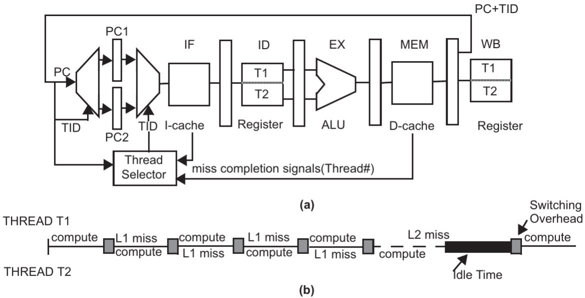


Figure 8.3 5-stage pipeline with two threads (a) and its execution timeline (b)

Solution 8.8 (b)

Cache access latencies are hidden because of the total overlap of thread executions with cache misses. The CPI improves to 1.

Problem 8.1 (c)

Consider the above case but the switching overhead is 5 cycles. Again compute the CPI of each thread and explain why it increases or decreases or stays the same?

Solution 8.1 (c)

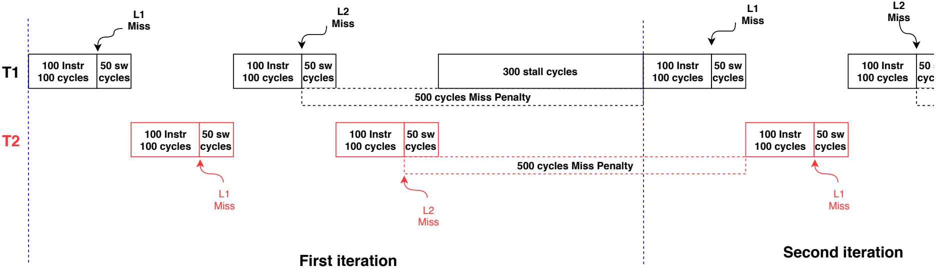
Again assume we execute 200 instructions.
After first 100 instruction, we will encounter ache miss and we context switch and have to pay a penalty of 5.
Cycles $= 100 + 5 + 100 + 5 = 210$

Solution 8.1 (c)

Again assume we execute 200 instructions.
After first 100 instruction, we will encounter ache miss and we context switch and have to pay a penalty of 5.
Cycles = $100 + 5 + 100 + 5 = 210$
$CPI = \frac{210}{200} = 1.05$
This increase reflects the overhead of switching threads.

Problem 8.1 (d)

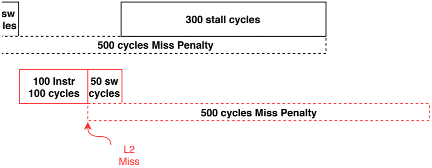
Now consider the case that L2 miss latency jumped from 50 cycles to 500 cycles and switching overhead jumped from 5 cycles to 50 cycles. Compute the CPI in this machine?



Total Cycles = 100 + 50 +100 + 50 + 100+50 +100 +50 + 300 = 900

Total Instructions = 100 +100 +100 +100 = 400

CPI = 900/400 = 2.25


sw
les
300 stall cycles
500 cycles Miss Penalty
100 Instr
100 cycles
50 sw
cycles
500 cycles Miss Penalty
L2
Miss