Name: Hugo JONCOUR ID: 40139230

COMP5201 Assignment 4 Fall 2020
Computer Organization & Design Prof. D.K. Probst

Issued: November 10, 2020 Due: December 7, 2020

Submit electronically to Moodle. No extension will be granted.

1. [20 marks] A program has an 8-instruction loop that is executed many times. Only the last instruction is a conditional branch, whose target is the first instruction at memory address [3a5c50]. Only the first instruction is a load, whose target is a 4-byte integer that resides at a different memory location in each iteration. There are no stores in the program. The first integer has memory address [4b6d30]. These integers are contiguous and are loaded in increasing memory-address order. The size of an instruction is also 4 bytes. A unified, i.e., combined I-cache and D-cache, direct-mapped cache has 16-byte lines, and 16 frames. It is initially empty. We hope to determine the number of cache misses in the first 16 iterations.

a) How many cache misses in the first four iterations? 3

b) How many cache misses in the second four iterations? 1

c) How many cache misses in the third four iterations? 8

d) How many cache misses in the fourth four iterations? 8

Name: Hugo JONCOUR ID: 40139230

2. [20 marks] We investigate the increase in CPI (clocks per instruction) due to cache misses that occur during memory references. For simplicity, we pretend that instruction fetches never miss.

a) Suppose the processor takes an average of 1.3 clock cycles to execute an instruction when there are no cache misses. Assume that the miss penalty is 8 cycles, and that there is an average of 1 memory reference per 3 instructions. The base CPI (1.3 cycles) _includes_ the cache hit time.

Suppose the miss rate is 6%. Using the formula t_ave = ht + mr * mp, what is the CPI when cache misses are taken into account? In English, average time = hit time plus miss rate times miss penalty.

$$CacheMissTime = MissRate * MissPenalty$$
$$CacheMissTime = 0.06 * 8$$
$$CacheMissTime = 0.48$$

With only one memory reference every 3 instruction, we have:
$$MemoryReferencePerInstruction = \frac{0.48}{3}$$

Executing a memory instruction takes 1.3 clock cycles:

$$CPI = \frac{0.48}{3} + 1.3 = 1.46 \text{ cycles}$$

b) Consider the same processor with a two-level cache. The hit rates for the L1$ and the L2$ are 95% and 75%, respectively. The _local_ miss penalties are 10 cycles and 80 cycles, respectively. Assume the same frequency of memory references. If the CPI is 1.3 cycles when there are no cache misses, what is the CPI when cache misses are taken into account ?

Hint: Apply the formula recursively to find the effective miss penalty of the L1$.

$$CacheMissTime = 1.3 + \frac{1}{3}(L1\$MissRate * L1\$MissPenalty) + \frac{1}{3}(L2\$MissRate * L2\$MissPenalty)$$

$$CPI = 1.3 + \frac{1}{3}(0.05 * 10) + (0.25 * 80 * 0.05)$$
$$CPI = 1.3 + \frac{1}{6} + (\frac{1}{4} * 80 * \frac{1}{20})$$
$$CPI = 1.3 + \frac{1}{6} + 1$$
$$CPI = 2.4666\ldots$$

The new CPI is $2.4\overline{6}$ … cycles

Name: Hugo JONCOUR ID: 40139230

3. [21 marks] Consider a computer with a byte-addressable memory. A 40-bit memory address is divided as follows for cache processing.

- First, the 8 lower-order bits are masked to expose the line number.
- Second, the 15 lower-order bits of the line number are inspected to get the array index.
- Third, the remaining 17 bits encode the tag value. Any numerical answer may contain a cache parameter.

Hint: What do the direct-mapped and set-associative placement formulas have in common?

a) What is the cache size in bytes?

The cache size includes the cache line number and the cache container index:
8 $bytes$ for the line number
15 $bytes$ for the array index

$$m \cdot 2^8 \cdot 2^{15} = m \cdot 2^{23}$$

The cache size is: $2^{23} bytes$

b) What is the cache-mapping scheme?

We have:

- a tag
- an index

Since we can't precisely determine the exact number of frames in this map, we will use the cache parameter 'm' as a placeholder.

We can say the cache mapping scheme is 'm-way associative'

If $m = 1$ then we have a direct-mapped cache

c) For a given byte in the cache, how many different bytes in the main memory could possibly be mapped to it ?

We have a cache size of: $m \cdot 2^{23} bytes$, there are $\frac{2^{40}}{2^{23}} m = m \cdot 2^{17} bytes$ in the main memory he could be mapped to.

Name: Hugo JONCOUR ID: 40139230

4. [19 marks] Consider a computer with 32-bit registers. The memory is word addressed. There is a direct-mapped cache with 4K cache frames. Cache lines are 16 words. Into which cache frame, and with what tag value, does 32-bit word address '45677def' go? Show your work. Answers in hexadecimal.

We have:

- Cache with 4k frames $4K = 4096 = 2^{12}$ so frame addresses are 12 bytes
- Cache line has 16 words $16 = 2^4$ so word addressability is 4 bits & tag is 16 bits
- 32-bit registers

Converting '45677def' from hexadecimal to binary we get: 10001010110011101111101111011111

(31 bits) (32 bits)

100 0101 0110 0111 0111 1101 1110 1111 = 0100 0101 0110 0111 0111 1101 1110 1111

First 16 bits are the tag: 0100 0101 0110 0111

Next 12 bits are the cache frame: 0111 1101 1110

Last 4 digits are the word offset: 1111

The tag 0100 0101 0110 0111 equals to 4567 in hexadecimal

The binary cache frame 0111 1101 1110 equals to 7DE in hexadecimal

The word offset 1111 is 15 in decimal or "F" in hexadecimal

So, the given address word in 7DE cache frame with tag 4567 and binary cache frame is F.

Name: Hugo JONCOUR ID: 40139230

5. [20 marks] Consider a computer whose memory latency is 800 cycles. Think of this as the time required to traverse a synchronous memory pipeline consisting of i) the network from processor to memory, ii) processing inside the memory, and iii) the network from memory to processor. The peak input bandwidth of this pipeline is 8 bytes/cycle. The processor manages to sustain 720 outstanding (load) memory references to floating-point operands in the pipeline in each and every cycle.

a) Using Little's law, calculate the sustained delivered operand bandwidth in bytes/cycle arriving at the processor. Does the peak input bandwidth limit you? Show your work.

Let:

- x be the operand bandwidth in $words/cycle$
- y the outstanding load memory references in the pipeline
- t the memory latency

From Little's Law we have:

$$x = \frac{y}{t} \text{ where } y = 720 \qquad \text{and } t = 800$$

Or:

$$x = \frac{720}{800} = 0.9 \frac{words}{cycle}$$

The peak input bandwidth does not limit as the sustained delivered operand bandwidth $0.9 < 1$ which is the peak input bandwidth.

b) The processor has a cache, with a 64-byte cache line, whose hit rate is 95%. Assuming that each floating-point arithmetic operation requires one new operand, and that the peak arithmetic performance of the processor is 35 flops/cycle, what is the sustained arithmetic performance of this processor in flops/cycle? Is there a bottleneck? If so, identify it. Show your work.

Let:

- x be the operand bandwidth

The sustained arithmetic performance of this processor is:

$$= \frac{x}{(1 - HitRate)} = \frac{0.9}{(1 - 0.95)} = 18 \frac{flops}{cycle}$$

There is no bottleneck as the sustained arithmetic performance of this processor is 18, and $18 < 35$ which is the peak arithmetic performance.