COEN 210 Computer Architecture: Assignment 3

Name: Inchara Raveendra

SCU ID: 00001653600

5.2 Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 64-bit memory address references, given as word addresses.

0x03, 0xb4, 0x2b, 0x02, 0xbf, 0x58, 0xbe, 0x0e, 0xb5, 0x2c, 0xba, 0xfd

5.2.1 For each of these references, identify the binary word address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list whether each reference is a hit or a miss, assuming the cache is initially empty.

Word Address	Binary Address	Tag	Index	Hit/ Miss
0x03	0000 0011	0	3	M
0xb4	1011 0100	b	4	M
0x2b	0010 1011	2	b	M
0x02	0000 0010	0	2	M
0xbf	1011 1111	b	f	M
0x58	0101 1000	5	8	M
0xbe	1011 1110	b	e	M
0x0e	0000 1110	0	e	M
0xb5	1011 0101	b	5	M
0x2c	0010 1100	2	c	M
0xba	1011 1010	b	a	M
0xfd	1111 1101	f	d	M

5.2.2 For each of these references, identify the binary word address, the tag, the index, and the offset given a direct-mapped cache with two-word blocks and a total size of eight blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

Word Address	Binary Address	Tag	Index	Offset	Hit/ Miss
0x03	0000 0011	0	1	1	M
0xb4	1011 0100	b	2	0	M
0x2b	0010 1011	2	5	1	M
0x02	0000 0010	0	1	0	H
0xbf	1011 1111	b	7	1	M
0x58	0101 1000	5	4	0	M
0xbe	1011 1110	b	7	0	H
0x0e	0000 1110	0	7	0	M
0xb5	1011 0101	b	2	1	H
0x2c	0010 1100	2	6	0	M
0xba	1011 1010	b	5	0	M
0xfd	1111 1101	f	6	1	M

5.2.3 You are asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with a total of eight words of data:

- **C1 has 1-word blocks,**
- **C2 has 2-word blocks, and**
- **C3 has 4-word blocks.**

a) C1

Word Address	Binary Address	Tag	Index	Hit/ Miss
0x03	0000 0011	0x00	3	M
0xb4	1011 0100	0x16	4	M
0x2b	0010 1011	0x05	3	M

0x02	0000 0010	0x00	2	M
0xbf	1011 1111	0x17	7	M
0x58	0101 1000	0x0b	0	M
0xbe	1011 1110	0x17	6	M
0x0e	0000 1110	0x01	6	M
0xb5	1011 0101	0x16	5	M
0x2c	0010 1100	0x05	4	M
0xba	1011 1010	0x17	2	M
0xfd	1111 1101	0x1F	5	M

b) C2

Word Address	Binary Address	Tag	Index	Hit/ Miss
0x03	0000 0011	0x00	1	M
0xb4	1011 0100	0x16	2	M
0x2b	0010 1011	0x05	1	M
0x02	0000 0010	0x00	1	M
0xbf	1011 1111	0x17	3	M
0x58	0101 1000	0x0b	0	M
0xbe	1011 1110	0x17	3	H
0x0e	0000 1110	0x01	3	M
0xb5	1011 0101	0x16	2	H
0x2c	0010 1100	0x05	2	M
0xba	1011 1010	0x17	1	M
0xfd	1111 1101	0x1F	2	M

c) C3

Word Address	Binary Address	Tag	Index	Hit/ Miss

0x03	0000 0011	0x00	0	M
0xb4	1011 0100	0x16	1	M
0x2b	0010 1011	0x05	0	M
0x02	0000 0010	0x00	0	M
0xbf	1011 1111	0x17	1	M
0x58	0101 1000	0x0b	0	M
0xbe	1011 1110	0x17	1	H
0x0e	0000 1110	0x01	1	M
0xb5	1011 0101	0x16	1	M
0x2c	0010 1100	0x05	1	M
0xba	1011 1010	0x17	0	M
0xfd	1111 1101	0x1F	1	M

C1 miss rate = 100%

C2 miss rate = 10/12 = 83%

C3 miss rate = 11/12 = 92%

5.3 By convention, a cache is named according to the amount of data it contains (i.e., a 4 KiB cache can hold 4 KiB of data); however, caches also require SRAM to store metadata such as tags and valid bits. For this exercise, you will examine how a cache's configuration affects the total amount of SRAM needed to implement it as well as the performance of the cache. For all parts, assume that the caches are direct-mapped, byte addressable, and that addresses and the words are 32 bits.

5.3.1 Calculate the total number of bits required to implement a 32 KiB cache with two-word blocks.

Given memory address as 32 [bits] and cache size as 32 [KiB].

Block size = 2 words = 2 x 4 B = 8 B

Number of blocks = 32 KiB / 8 B = 2^12 blocks

Index bits = log2(number of blocks) = log2(2^12) = 12 bits

Offset bits = 1 word select (log2(2 word/block))+ 2 byte select (log2(8 B/word)) = 3 bits

Tag bits = 32 bit address - 12 index bits - 3 offset bits = 17 bits

Total bits/block = tag bits + data bits + 1 valid bit = 17 + 64 +1 = 82 bits

Total bits in cache = 82 bits/block x 2^12 blocks = 335872 bits

5.3.2 Calculate the total number of bits required to implement a 64 KiB cache with 16-word blocks. How much bigger is this cache than the 32 KiB cache described in Exercise 5.3.1? (Notice that, by changing the block size, we doubled the amount of data without doubling the total size of the cache.)

The memory address and cache size are 32 bits and 64 KiB respectively.

The Block size would be 16 words which corresponds to 16 x 4 B = 64 B

The number of blocks will be 64 KiB / 64 B = 2^10 blocks

The index bits is log2(number of blocks) = log2(2^10) = 10 bits

The offset bits (log2(64 B/word)) = 6 bits

Tag bits = 32 bit address - 10 index bits - 6 offset bits = 16 bits

Total bits/block = tag bits + data bits + 1 valid bit = 16 + 512 +1 = 529 bits

Total bits in cache = 529 bits/block x 2^10 blocks = 541696

541696/335872 = 1.612 large

5.3.3 Explain why this 64 KiB cache, despite its larger data size, might provide slower performance than the first cache.

Typically increased size comes with increased hit latency and larger miss penalties.

Because the block size is larger, it takes longer to select the correct word/byte on a hit.

Likewise, more data needs to be loaded in on a miss.

5.3.4 Generate a series of read requests that have a lower miss rate on a 32 KiB two-way set associative cache than on the cache described in Exercise 5.3.1.

The following shows that the lower miss rate on a 32 KiB two-way set associative cache than on the 32 KiB.

Address	32KiB Block	Hit/ Miss	32KiB Set	Hit/ Miss
0x00000	Block 0	Miss	Set 0 - Way 0	Miss
0x10000	Block 0	Miss	Set 0 - Way 1	Miss
0x00000	Block 0	Miss	Set 0 - Way 0	Hit

5.7 Consider the following program and cache behaviors.

Data Reads per 1000 Instruction	Data Writes per 1000 Instructions	Instruction Cache Miss Rate	Data Cache Miss Rate	Block Size (bytes)
250	100	0.30%	2%	64

5.7.1 Suppose a CPU with a write-through, write-allocate cache achieves a CPI of 2. What are the read and write bandwidths (measured by bytes per cycle) between RAM and the cache? (Assume each miss generates a request for one block.)

CPI = 2, therefore, 0.5 instruction accesses per cycle.

Instruction accesses generate an average read traffic = 0.5 x .003 x 64 = **0.096 bytes/cycle**

25% of instructions generate a read request. 2% of these generate a cache miss;
Therefore, read misses = 0.5 x 0.25 x 0.02 x 64 = **0.16 bytes/cycle of read traffic**

Write misses generate an average of 0.5 x 0.1 x 8 = **0.4 bytes/cycle of write traffic**

10% of instructions generate a write request. 2% of these generate a cache miss.
Therefore, write misses = 0.5 x 0.1 x 0.02 x 64 = **0.064 bytes/cycle of read traffic**

The total read bandwidth = 0.096 + 0.16 + 0.064 = 0.32 bytes/cycle, and
the total write bandwidth is 0.4 bytes/cycle

5.7.2 For a write-back, write-allocate cache, assuming 30% of replaced data cache blocks are dirty, what are the read and write bandwidths needed for a CPI of 2?

8: 4%	16: 3%	32: 2%	64: 1.5%	128: 1%

The data write bandwidth = 0.5 x (0.25 + 0.1) x 0.02 x 0.3 x 64 = 0.0672 bytes/cycle.

5.9 Cache block size (B) can affect both miss rate and miss latency. Assuming a machine with a base CPI of 1, and an average of 1.35 references (both instruction and data) per instruction, find the block size that minimizes the total miss latency given the following miss rates for various block sizes.

5.9.1 What is the optimal block size for a miss latency of 20 × B cycles?

B = 8; 0.04 x 20 x 8 = 6.4

B = 16; 0.03 x 20 x 16 = 9.6

B = 32; 0.02 x 20 x 32 = 12.8

B = 64; 0.015 x 20 x 64 = 19.2

B = 128; 0.01 x 20 x 128 = 25.6

B = 8 is optimal

5.9.2 What is the optimal block size for a miss latency of 24 + B cycles?

B = 8; 0.04 x (24 + 8) = 1.28

B = 16; 0.03 x (24 + 16) = 1.2

B = 32; 0.02 x (24 + 32) = 1.12

B = 64; 0.015 x (24 + 64) = 1.32

B = 128; 0.01 x (24 + 128) = 1.52

B = 32 is optimal

5.9.3 For constant miss latency, what is the optimal block size?

B = 128

5.10 In this exercise, we will look at the different ways capacity affects overall performance. In general, cache access time is proportional to capacity. Assume that main memory accesses take 70 ns and that 36% of all instructions access data memory. The following table shows data for L1 caches attached to each of two processors, P1 and P2.

	L1 Size	L1 Miss Rate	L1 Hit Time
P1	2 KiB	8.0%	0.66 ns
P2	4 KiB	6.0%	0.90 ns

5.10.1 Assuming that the L1 hit time determines the cycle times for P1 and P2, what are their respective clock rates?

P1 Cycle Time = 0.66 ns

f = 1/T = 1/(0.66 x 10^-9) = 1.5151 GHz

P2 Cycle Time = 0.90 ns

f = 1/T = 1/(0.90 x 10^-9) = 1.11 GHz

5.10.2 What is the Average Memory Access Time for P1 and P2 (in cycles)?

AMAT = Hit Time + (Miss Rate) x (Miss Latency)

For P1: AMAT = 0.66 + (0.08)(70) = 6.26 ns

For P2: AMAT = 0.90 + (0.06)(70) = 5.10 ns

5.10.3 Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 and P2? Which processor is faster? (When we say a "base CPI of 1.0", we mean that instructions complete in one cycle, unless either the instruction access or the data access causes a cache miss.)

For the next three problems, we will consider the addition of an L2 cache to P1 (to presumably make up for its limited L1 cache capacity). Use the L1 cache capacities and hit times from the previous table when solving these problems. The L2 miss rate indicated is its local miss rate.

L2 Size	L2 Miss Rate	L2 Hit Time
1 MiB	95%	5.62ns

CPI = Base CPI + L1 Miss Rate x (Instruction + Data Memory Access) x Miss Penalty

For P1: CPI = 1 + (0.08)(1.36)(70) = 8.616 cycles/instruction

For P2: CPI = 1 + (0.06)(1.36)(70) = 6.712 cycles/instruction

5.10.4 What is the AMAT for P1 with the addition of an L2 cache? Is the AMAT better or worse with the L2 cache?

$1 + 0.08\ [9 + 0.95 \times 107] = 9.85$

AMAT is worse with L2 cache

5.10.5 Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache?

$\text{CPI} = \text{AMAT} + \%\text{memory} \times (\text{AMAT}-1)$

$\text{CPI} = 9.85 \times 0.36 + 9.56 = 13.106$

5.10.6 What would the L2 miss rate need to be in order for P1 with an L2 cache to be faster than P1 without an L2 cache?

AMAT with L2 < AMAT with L1 only

$1 + 0.08\ [9 + m \times 107] < 9.56$

Therefore, $m < 0.916$

5.10.7 What would the L2 miss rate need to be in order for P1 with an L2 cache to be faster than P2 without an L2 cache?

$\text{CPI\_P1} \times 0.66 < 6.63$

$\text{CPI\_P1} < 10.05$

$\text{CPI\_P1} = \text{AMAT\_P1} + 0.36\ (\text{AMAT\_P1} – 1)$

$\text{AMAT\_P1} + 0.36\ (\text{AMAT\_P1}-1) < 10.05$

$\text{AMAT\_P1} < 7.65$

$1 + 0.08\ [9 + m \times 107] < 7.65$

$m < 0.693$

Miss rate can be at most 69.3%