

# Problem 5.2

## 5.2.1

Word Address	Binary Address	Tag	Index	Hit/Miss
3	0000 0011	0	3	MISS
180	1011 0100	11	4	MISS
43	0010 1011	2	11	MISS
2	0000 0000	0	2	MISS
191	1011 1111	11	15	MISS
88	0101 1000	5	8	MISS
190	1011 1110	11	14	MISS
14	0000 1110	0	14	MISS
181	1011 0101	11	5	MISS
44	0010 1100	2	12	MISS
186	1011 1010	11	10	MISS
253	1111 1101	15	13	MISS

## 5.2.2

Word Address	Binary Address	Tag	Index	Hit/Miss
3	0000 0011	0	1	MISS
180	1011 0100	11	2	MISS
43	0010 1011	2	5	MISS
2	0000 0000	0	1	HIT
191	1011 1111	11	7	MISS
88	0101 1000	5	4	MISS
190	1011 1110	11	7	HIT
14	0000 1110	0	7	MISS
181	1011 0101	11	2	HIT
44	0010 1100	2	6	MISS
186	1011 1010	11	5	MISS
253	1111 1101	15	6	MISS

## 5.2.3

Word Address	Binary Address	Tag	Cache 1		Cache2		Cache 3	
			Index	H/M	Index	H/M	Index	H/M
3	0000 0011	0	3	M	1	M	0	M
180	1011 0100	22	4	M	2	M	1	M
43	0010 1011	5	3	M	1	M	0	M
2	0000 0000	0	2	M	1	M	0	M
191	1011 1111	23	7	M	3	M	1	M
88	0101 1000	11	0	M	0	M	0	M
190	1011 1110	23	6	M	3	H	1	H
14	0000 1110	1	6	M	3	M	1	M
181	1011 0101	22	5	M	2	H	1	M
44	0010 1100	5	4	M	2	M	1	M
186	1011 1010	23	2	M	1	M	0	M
253	1111 1101	31	5	M	2	M	1	M

Cache 1 Miss Rate: 12/12

Cache 2 Miss Rate: 10/12

Cache 3 Miss Rate: 11/12

Cache 1 total cycles:  $12 * 25 + 12 * 2 = 324$

Cache 2 total cycles:  $10 * 25 + 12 * 3 = 286$

Cache 3 total cycles:  $11 * 25 + 12 * 5 = 335$

Cache 2 has the best performance

#### 5.2.4

Data\_size = 32 KiB

Block\_size = 2 blocks

Word\_size = 4 bytes

Data\_size = blocks \* block\_size \* word\_size

$(32 \text{ KiB} / 4 \text{ bytes per word}) / 2 \text{ words per block} = 4096 \text{ blocks}$

Tag\_size =  $32 - \log_2(\text{blocks}) - \log_2(\text{block\_size}) - \log_2(\text{word\_size}) = 17 \text{ bits}$

Valid\_bit\_size = 1

Total\_size = data\_size + (valid\_bit\_size + tag\_size) \* blocks = 41984 bytes

Block\_size = 16 blocks

Word\_size = 4 bytes

Tag\_size =  $32 - \log_2(\text{blocks}) - \log_2(\text{block\_size}) - \log_2(\text{word\_size}) = 14 \text{ bits}$

Valid\_bit\_size = 1

$41984 \leq 64 * \text{blocks} + 15 * \text{blocks}$

Blocks = 531

1024-block cache

The larger block size could require an increased hit time and an increased miss penalty than the original cache. The decreased amount of blocks could increase the rate at which conflict misses occur.

#### 5.2.5

Associative caches are supposed to reduce the rate at which conflict misses occur. A series of read requests that has the same 12-bit index field but with a different tag field will cause many misses to occur. This cache sequence would miss on every access. However, a 2-way set associate cache would hit on every access after the first two.

#### 5.2.6

Yes it is possible to use this formula to index the cache. However, since the bits are XOR'd, we lose information about the five bits. Thus, to identify the address in the cache, we must include more tag bits.

## Problem 5.7

## 5.7.1

Word Address	Binary Address	Tag	Index	Hit/Miss	Way 0	Way 1	Way 2
3	0000 0011	0	1	MISS	T(1) = 0		
180	1011 0100	11	2	MISS	T(1) = 0 T(2) = 11		
43	0010 1011	2	5	MISS	T(1) = 0 T(2) = 11 T(5) = 2		
2	0000 0000	0	1	MISS	T(1) = 0 T(2) = 11 T(5) = 2	T(1) = 0	
191	1011 1111	11	7	MISS	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11	T(1) = 0	
88	0101 1000	5	4	MISS	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11 T(4) = 5	T(1) = 0	
190	1011 1110	11	7	HIT	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11 T(4) = 5	T(1) = 0	
14	0000 1110	0	7	MISS	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11 T(4) = 5	T(1) = 0 T(7) = 0	
181	1011 0101	11	2	HIT	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11 T(4) = 5	T(1) = 0 T(7) = 0	
44	0010 1100	2	6	MISS	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11 T(4) = 5 T(6) = 2	T(1) = 0 T(7) = 0	
186	1011 1010	11	5	MISS	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11 T(4) = 5 T(6) = 2	T(1) = 0 T(7) = 0 T(5) = 11	
253	1111 1101	15	6	MISS	T(1) = 0 T(2) = 11 T(5) = 2 T(7) = 11 T(4) = 5 T(6) = 2	T(1) = 0 T(7) = 0 T(5) = 11 T(6) = 15	

## 5.7.2

Word Address/Tag	Hit/Miss	Contents
3	M	3
180	M	3, 180
43	M	3, 180, 43
2	M	3, 180, 43, 2
191	M	3, 180, 43, 2, 191
88	M	3, 180, 43, 2, 191, 88
190	M	3, 180, 43, 2, 191, 88, 190
14	M	3, 180, 43, 2, 191, 88, 190, 14
181	M	181, 180, 43, 2, 191, 88, 190, 14
44	M	181, 44, 43, 2, 191, 88, 190, 14
186	M	181, 44, 186, 2, 191, 88, 190, 14
253	M	181, 44, 186, 253, 191, 88, 190, 14

## 5.7.3

Word Address	Tag	Hit/Miss	Contents
3	1	M	1
180	90	M	1, 90
43	21	M	1, 90, 21
2	1	H	1, 90, 21
191	95	M	1, 90, 21, 95
88	44	M	1, 90, 21, 95, 44
190	95	H	1, 90, 21, 95, 44
14	7	M	1, 90, 21, 95, 44, 7
181	90	H	1, 90, 21, 95, 44, 7
44	22	M	1, 90, 21, 95, 44, 7, 22
186	143	M	1, 90, 21, 95, 44, 7, 22, 143
253	126	M	1, 90, 21, 95, 44, 7, 22, 143, 126

The miss rate using MRU is 9/12. This is the best possible miss rate. There were no misses on any block that had been removed from the cache.

## 5.7.4

L1 only:

$$0.07 * 100 = 7\text{ns}$$

$$\text{CPI} = 7\text{ns}/0.5\text{ns} = 14$$

Direct Mapped L2:

$$0.07 * (12 + 0.035 * 100) = 1.1\text{ns}$$

$$\text{CPI} = \text{ceil}(1.1\text{ns}/0.5\text{ns}) = 3$$

8-way set associated set L2:

$$0.07 * (28 + 0.015 * 100) = 2.1 \text{ ns}$$

$$\text{CPI} = \text{ceil}(2.1\text{ns}/0.5\text{ns}) = 5$$

Double Memory Access time, L1 only:

$$0.07 * 200 = 14\text{ns}$$

$$\text{CPI} = 14\text{ns}/0.5\text{ns} = 28$$

Double Memory Access time, Direct Mapped L2:

$$0.07 * (12 + 0.035 * 200) = 1.3\text{ns}$$

$$\text{CPI} = \text{ceil}(1.3\text{ns}/0.5\text{ns}) = 3$$

Double Memory Access time, 8-way set associated set L2:

$$0.07 * (28 + 0.015 * 200) = 2.2 \text{ ns}$$

$$\text{CPI} = \text{ceil}(2.2\text{ns}/0.5\text{ns}) = 5$$

Halved Memory Access time, L1 only:

$$0.07 * 50 = 3.5\text{ns}$$

$$\text{CPI} = 3.5\text{ns}/0.5\text{ns} = 7$$

Halved Memory Access time, Direct Mapped L2:

$$0.07 * (12 + 0.035 * 50) = 1\text{ns}$$

$$\text{CPI} = 1\text{ns}/0.5\text{ns} = 2$$

Halved Memory Access time, 8-way set associated set L2:

$$0.07 * (28 + 0.015 * 50) = 2.1 \text{ ns}$$

$$\text{CPI} = \text{ceil}(2.1\text{ns}/0.5\text{ns}) = 5$$

#### 5.7.5

$$0.07 * (12 + 0.035 * (50 + 0.013 * 100)) = 1\text{ns}$$

Adding the L3 cache reduces the overall memory access time. However, it also takes away space for having other types of resources.

#### 5.7.6

A 50ns access time will always result in  $AMAT = 0.07 * 50 = 3.5$  ns regardless of how infrequent the L2 cache miss rate is. The miss rate could be 0 and it would still be 3.5ns. 3.5ns is greater than 1.1ns and 2.1ns which is given by the on-chip L2 caches. No size will achieve the performance goal.