VE370 RC7

YUXIN & TIANZE

Lecture Slides

Local/Spatial Locality

Block, word, byte address

- What does what addressable mean?

Hit vs. Miss

Direct Mapped, set associative, fully associative

Write back, write through

Exercise 5.2

Exercise 5.2

In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously.

a.	`for (I=0; I<8; I++)`
	`  for (J=0; J<8000; J++)`
	`  A[I][J]=B[I][0]+A[J][I];`

b.	`for (J=0; J<8000; J++)`
	`  for (I=0; I<8; I++)`
	`  A[I][J]=B[I][0]+A[J][I];`

5.2.1 [5] <5.1> How many 32-bit integers can be stored in a 16-byte cache line?

5.2.2 [5] <5.1> References to which variables exhibit temporal locality?

5.2.3 [5] <5.1> References to which variables exhibit spatial locality?

5.2.1
- 4 integers

5.2.2
- (a). I, J, B[I][0]
- (b). I, J

5.2.3
- (a). A[I][J]
- (b). A[J][I]

Exercise 5.3.1

Exercise 5.3

Caches are important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, given as word addresses.

a.	3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253
b.	21, 166, 201, 143, 61, 166, 62, 133, 111, 143, 144, 61

5.3.1 [10] <5.2> For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

5.3.2 [10] <5.2> For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with two-word blocks and a total size of 8 blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

5.3.3 [20] <5.2, 5.3> 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 8 words of data: C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks. In terms of miss rate, which cache design is the best? If the miss stall time is 25 cycles, and C1 has an access time of 2 cycles, C2 takes 3 cycles, and C3 takes 5 cycles, which is the best cache design?

tag	Word Index
	11
10110	100
10	1011
	10
10111	111
101	1000
10111	110
	1110
10110	101
10	1100
10111	010
11111	101

	Binary address	tag	index	h/m
3	11	0	0011	miss
180	10110100	1011	0100	miss
43	101011	10	1011	miss
2	10	0	0010	miss
191	10111111	1011	1111	miss
88	1011000	101	1000	miss
190	10111110	1011	1110	miss
14	1110	0	1110	miss
181	10110101	1011	0101	miss
44	101100	10	1100	miss
186	10111010	1011	1010	miss
253	11111101	1111	1101	miss

Index	
0000	
0001	
0010	2
0011	3
0100	180
0101	181
0110	
0111	
1000	88
1001	
1010	186
1011	43
1100	44
1101	253
1110	190→14
1111	191

	Binary address	tag	index	h/m
21	10101	1	0101	miss
166	10100110	1010	0110	miss
201	11001001	1100	1001	miss
143	10001111	1000	1111	miss
61	111101	11	1101	miss
166	10100110	1010	0110	hit
62	111110	11	1110	miss
133	10000101	1000	0101	miss
111	1101111	110	1111	miss
143	10001111	1000	1111	miss
144	10010000	1001	0000	miss
61	111101	11	1101	hit

Index	
0000	144
0001	
0010	
0011	
0100	
0101	21→133
0110	166
0111	
1000	
1001	201
1010	
1011	
1100	
1101	61
1110	62
1111	143→111→143

	Binary address	tag	index	h/m
3	11	0	001	miss
180	10110100	1011	010	miss
43	101011	10	101	miss
2	10	0	001	hit
191	10111111	1011	111	miss
88	1011000	101	100	miss
190	10111110	1011	111	hit
14	1110	0	111	miss
181	10110101	1011	010	hit
44	101100	10	110	miss
186	10111010	1011	101	miss
253	11111101	1111	110	miss

Index		
000		
001	2	3
010	180	181
011		
100	88	89
101	42→186	43→187
110	44→252	45→253
111	190→14	191→15

	Binary address	tag	index	h/m
21	10101	1	010	miss
166	10100110	1010	011	miss
201	11001001	1100	100	miss
143	10001111	1000	111	miss
61	111101	11	110	miss
166	10100110	1010	011	hit
62	111110	11	111	miss
133	10000101	1000	010	miss
111	1101111	110	111	miss
143	10001111	1000	111	miss
144	10010000	1001	000	miss
61	111101	11	110	hit

Index		
000	144	145
001		
010	20→132	21→133
011	166	167
100	200	201
101		
110	60	61
111	142→62→110→142	143→63→111→143

Exercise 5.3.3

- C1: 0 hit stall time=27*12
- C2:2 hits stall time=25*10+3*12
- C3: 1 hit stall time=25*11+5*12

	Binary address	Index for C1		Index for C2		Index for C3	
3	11	011	miss	01	miss	0	miss
180	10110100	100	miss	10	miss	1	miss
43	101011	011	miss	01	miss	0	miss
2	10	010	miss	01	miss	0	miss
191	10111111	111	miss	11	miss	1	miss
88	1011000	000	miss	00	miss	0	miss
190	10111110	110	miss	11	hit	1	hit
14	1110	110	miss	11	miss	1	miss
181	10110101	101	miss	10	hit	1	miss
44	101100	100	miss	10	miss	1	miss
186	10111010	010	miss	01	miss	0	miss
253	11111101	101	miss	10	miss	1	miss

Exercise 5.3.4

There are many different design parameters that are important to a cache's overall performance. The table below lists parameters for different direct-mapped cache designs.

	Cache Data Size	Cache Block Size	Cache Access Time
a.	32 KB	2 words	1 cycle
b.	32 KB	4 words	2 cycle

5.3.4 [15] <5.2> Calculate the total number of bits required for the cache listed in the table, assuming a 32-bit address. Given that total size, find the total size of the closest direct-mapped cache with 16-word blocks of equal size or greater. Explain why the second cache, despite its larger data size, might provide slower performance than the first cache.

Exercise 5.3.4

Given

- 32-bit byte address
- 2^n blocks in cache
- 2^m words per block, 2^{m+2} bytes

Size of tag field = $32 - (n + m + 2)$

- n bits to index blocks in cache
- m bits used to select words in a block
- 2 bits used to select the 4 bytes in a word
- Tag field decreases when n and m increase

Cache size = $2^n \times$ (block size + tag size + valid field size)

Exercise 5.3.4

32 kb = 32* 2^(10) bytes

Index = $\# \; of \; blocks = \frac{32 * 2^{10} \; bytes}{2 \; word * 4 \frac{bytes}{word}} = 2^{12}$

Tag field = 32 – (n+m+2) = 32 – (12 + 1 + 2) = 17

Total size of the cache: $2^{12} * \left(2 * 4 * 8 \frac{bits}{byte} + 17 + 1\right) = 335872\text{bits} = 328kb$

The reason for slower performance is, the larger cache may have a longer access time which will lead to a lower performance.

Exercise 5.4.1-5.4.3

Exercise 5.4

For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

	Tag	Index	Offset
a.	31–10	9–5	4–0
b.	31–12	11–6	5–0

5.4.1 [5] <5.2> What is the cache line size (in words)?

5.4.2 [5] <5.2> How many entries does the cache have?

5.4.3 [5] <5.2> What is the ratio between total bits required for such a cache implementation over the data storage bits?

(a)

- 5.4.1: $2^3 = 8$
- 5.4.2: $2^5 = 32$
- 5.4.3: 1+22/8/32=1.086

Each tag corresponds to: # of blocks* # of words/ block

(b)

- 5.4.1: $2^4 = 16$
- 5.4.2: $2^6 = 64$
- 5.4.3: 1+20/16/32=1.039

Exercise 5.4.4-5.4.5

Starting from power on, the following byte-addressed cache references are recorded.

Address	0	4	16	132	232	160	1024	30	140	3100	180	2180

5.4.4 [10] <5.2> How many blocks are replaced?

5.4.5 [10] <5.2> What is the hit ratio?

5.4.6 [20] <5.2> List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

5.4.4: 3

5.4.5: $\frac{4}{12} = 0.33$

	0	4	16	132	232	160	1024	30	140	3100	180	2180
Index	0	0	0	100	111	101	0	0	100	0	101	100
H/M	M	H	H	M	M	M	M	M	H	M	H	M
Replace	N	N	N	N	N	N	Y	Y	N	Y	N	Y

Exercise 5.4.6

Starting from power on, the following byte-addressed cache references are recorded.

Address	0	4	16	132	232	160	1024	30	140	3100	180	2180

5.4.4 [10] <5.2> How many blocks are replaced?

5.4.5 [10] <5.2> What is the hit ratio?

5.4.6 [20] <5.2> List the final state of the cache, with each valid entry represented as a record of <index, tag, data>.

Index	Tag	Data
00000	11	Mem[3072]
00100	10	Mem[2176]
00101	0	Mem[160]
00111	0	Mem[224]

	0	4	16	132	232	160	1024	30	140	3100	180	2180
Index	0	0	0	100	111	101	0	0	100	0	101	100
H/M	M	H	H	M	M	M	M	M	H	M	H	M
Replace	N	N	N	N	N	N	Y	Y	N	Y	N	Y

Exercise 5.6

Exercise 5.6

Media applications that play audio or video files are part of a class of workloads called "streaming" workloads; i.e., they bring in large amounts of data but do not reuse much of it. Consider a video streaming workload that accesses a 512 KB working set sequentially with the following address stream:

0, 2, 4, 6, 8, 10, 12, 14, 16, ... byte address

5.6.1 [5] <5.5, 5.3> Assume a 64 KB direct-mapped cache with a 32-byte line. What is the miss rate for the address stream above? How is this miss rate sensitive to the size of the cache or the working set? How would you categorize the misses this workload is experiencing, based on the 3C model?

5.6.2 [5] <5.5, 5.1> Re-compute the miss rate when the cache line size is 16 bytes, 64 bytes, and 128 bytes. What kind of locality is this workload exploiting?

0	2	4	6	8	10	…	30	32
M	H	H	H	H	H	H	H	M

5.6.1:

$\frac{1}{16}$ miss rate. It does not have any relationship with the size of the cache or the working set. They are cold misses, which cannot be avoided.

0	2	4	6	8	10	12	14	16
M	H	H	H	H	H	H	H	M

5.6.2:

16 bytes: $\frac{1}{8}$ 64 bytes: $\frac{1}{32}$ 128 bytes: $\frac{1}{64}$

- Spatial locality