ZHEJIANG UNIVERSITY　COLLEGE OF INFORMATION SCIENCE AND ELECTRONICS ENGINEERING

HOMEWORK #6

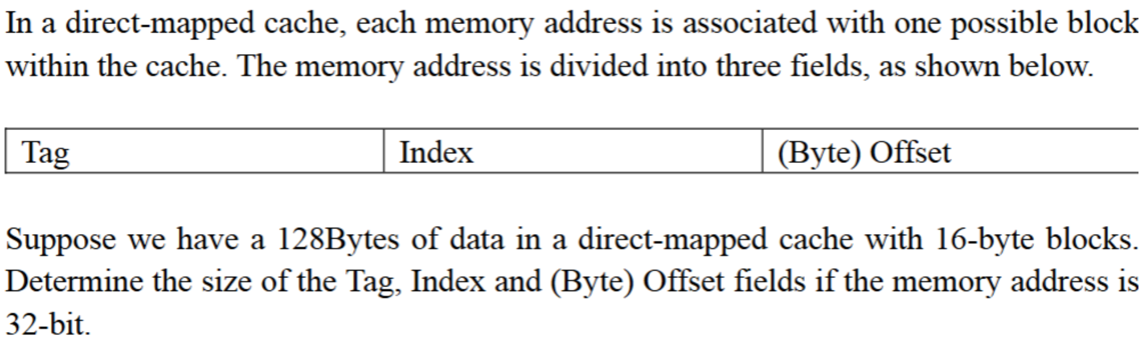
Computer Organization and Design

Name:　　　　　　Student ID:

Major: Electronic Science and Technology

Date: 2024 年 12 月 8 日

Problem 1.



In a direct-mapped cache, each memory address is associated with one possible block within the cache. The memory address is divided into three fields, as shown below.

Tag	Index	(Byte) Offset

Suppose we have a 128Bytes of data in a direct-mapped cache with 16-byte blocks. Determine the size of the Tag, Index and (Byte) Offset fields if the memory address is 32-bit.

Answer：1. Byte Offset

The block size is 16 bytes. Therefore, the number of bits required to address each byte within a block is:

$$\text{Byte Offset size} = \log_2(16) = 4 \text{ bits.}$$

2. Index

The total cache size is 128 bytes, and the block size is 16 bytes. Thus, the cache has:

$$\frac{\text{Cache Size}}{\text{Block Size}} = \frac{128}{16} = 8 \text{ blocks.}$$

The number of bits needed to index these blocks is:

$$\text{Index size} = \log_2(8) = 3 \text{ bits.}$$

3. Tag

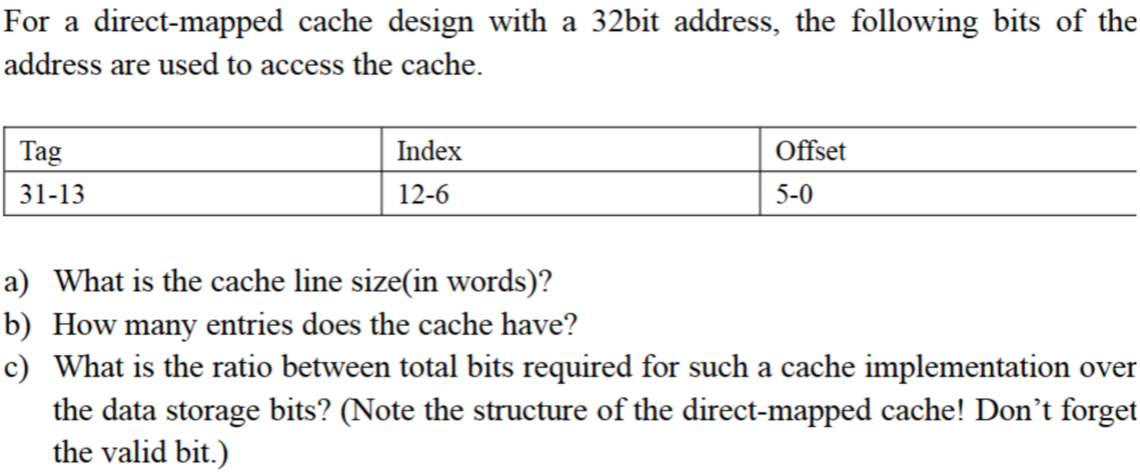
The total memory address size is 32 bits. The remaining bits are used for the tag:

$$\text{Tag size} = 32 - (\text{Index size} + \text{Byte Offset size}) = 32 - (3 + 4) = 25 \text{ bits.}$$

Final Answer

- Tag size: 25 bits
- Index size: 3 bits
- Byte Offset size: 4 bits

Problem 2.



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

Tag	Index	Offset
31-13	12-6	5-0

a) What is the cache line size(in words)?

b) How many entries does the cache have?

c) What is the ratio between total bits required for such a cache implementation over the data storage bits? (Note the structure of the direct-mapped cache! Don't forget the valid bit.)

Answer : a) Cache Line Size (in words):

- The offset field is 6 bits, so the cache line size is:

$$2^6 = 64 \text{ bytes.}$$

- Assuming each word is 4 bytes, the cache line size in words is:

$$\frac{64}{4} = 16 \text{ words.}$$

b) Number of Cache Entries:

- The index field is 7 bits, so the number of cache entries is:

$$2^7 = 128.$$

c) Ratio of Total Bits to Data Storage Bits:

- Each cache entry contains:

$$\text{Data storage bits} = 64 \times 8 = 512 \text{ bits,}$$

$$\text{Tag bits} = 19 \text{ bits,}$$

$$\text{Valid bit} = 1 \text{ bit.}$$

- Total bits per entry:

$$512 + 19 + 1 = 532 \text{ bits.}$$

- Total data storage bits:

$$128 \times 512 = 65536 \text{ bits.}$$

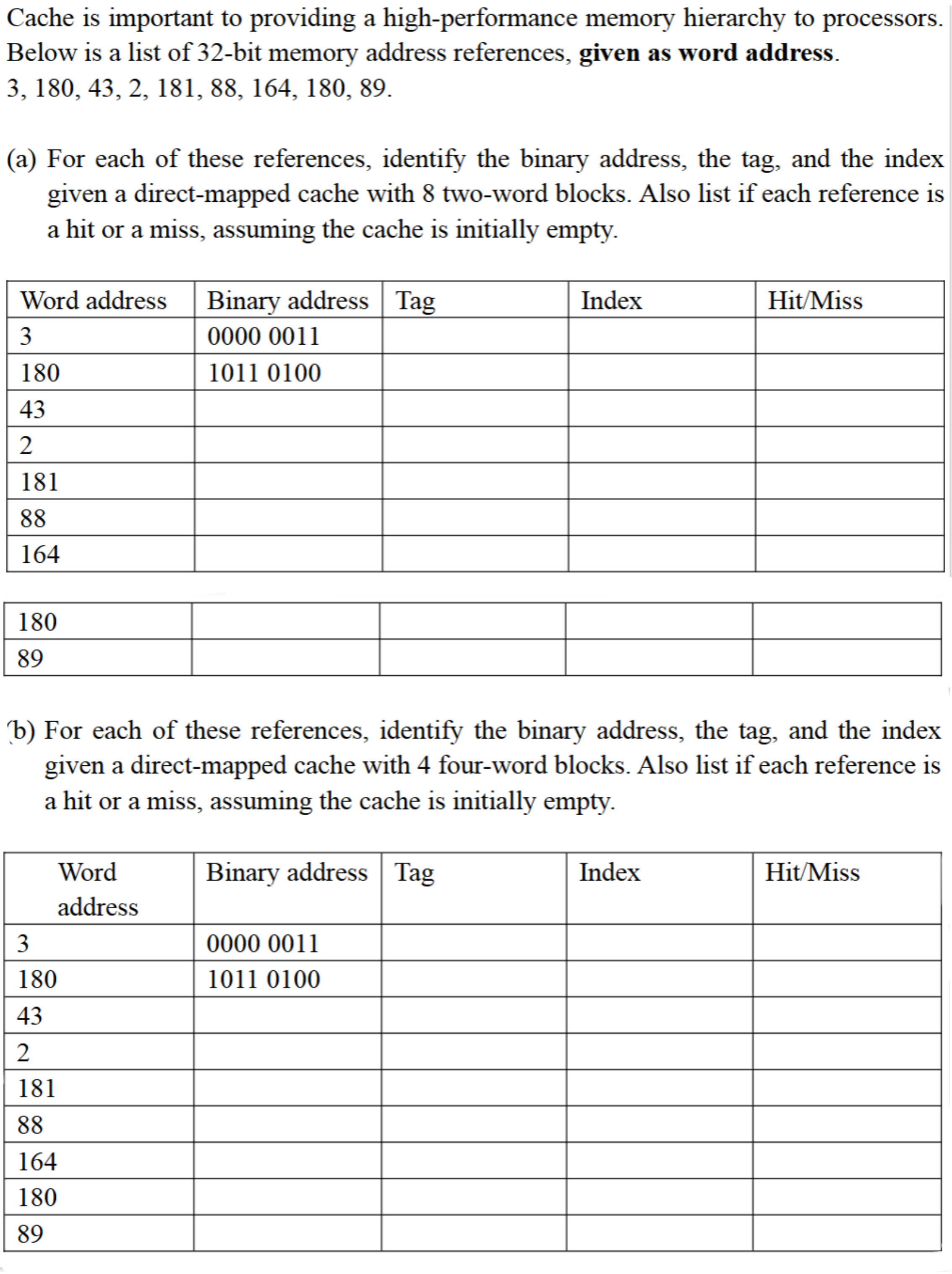
- Total cache implementation bits:

$$128 \times 532 = 68160 \text{ bits.}$$

- Ratio:

$$\text{Ratio} = \frac{68160}{65536} \approx 1.04.$$

Problem 3.



Cache is important to providing a high-performance memory hierarchy to processors. Below is a list of 32-bit memory address references, **given as word address**.
3, 180, 43, 2, 181, 88, 164, 180, 89.

(a) For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 8 two-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

Word address	Binary address	Tag	Index	Hit/Miss
3	0000 0011			
180	1011 0100			
43				
2				
181				
88				
164				
180				
89				

(b) For each of these references, identify the binary address, the tag, and the index given a direct-mapped cache with 4 four-word blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

Word address	Binary address	Tag	Index	Hit/Miss
3	0000 0011			
180	1011 0100			
43				
2				
181				
88				
164				
180				
89				

Answer: (a) 8 Two-Word Blocks Cache

For the 8 two-word blocks cache:

- Each block has 2 words (8 bytes), requiring 3 bits for the block offset.
- The cache has 8 blocks, so 3 bits are required for the index.
- The remaining 26 bits are used for the tag.

Word address	Binary address	Tag	Index	Hit/Miss
3	00000011	00000000000000000000000000	000	Miss
180	10110100	00000000000000000000000010	110	Miss
43	00101011	00000000000000000000000000	101	Miss
2	00000010	00000000000000000000000000	000	Hit
181	10110101	00000000000000000000000010	110	Hit
88	01011000	00000000000000000000000001	011	Miss
164	10100100	00000000000000000000000010	100	Miss
180	10110100	00000000000000000000000010	110	Hit
89	01011001	00000000000000000000000001	011	Hit

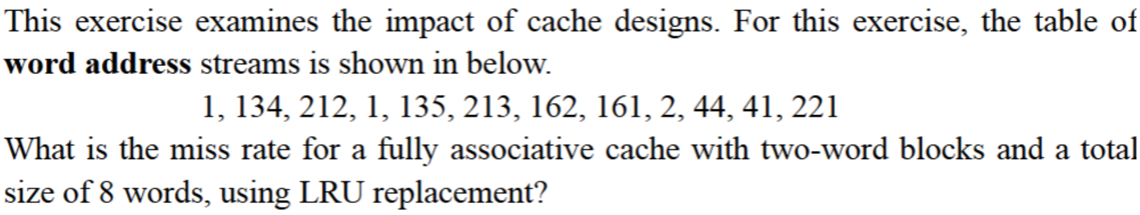
(b) 4 Four-Word Blocks Cache

For the 4 four-word blocks cache:

- Each block has 4 words (16 bytes), requiring 4 bits for the block offset.
- The cache has 4 blocks, so 2 bits are required for the index.
- The remaining 26 bits are used for the tag.

Word address	Binary address	Tag	Index	Hit/Miss
3	00000011	00000000000000000000000000	00	Miss
180	10110100	00000000000000000000000010	11	Miss
43	00101011	00000000000000000000000000	10	Miss
2	00000010	00000000000000000000000000	00	Hit
181	10110101	00000000000000000000000010	11	Hit
88	01011000	00000000000000000000000001	01	Miss
164	10100100	00000000000000000000000010	10	Miss
180	10110100	00000000000000000000000010	11	Hit
89	01011001	00000000000000000000000001	01	Hit

Problem 4.



This exercise examines the impact of cache designs. For this exercise, the table of **word address** streams is shown in below.

1, 134, 212, 1, 135, 213, 162, 161, 2, 44, 41, 221

What is the miss rate for a fully associative cache with two-word blocks and a total size of 8 words, using LRU replacement?

Answer： For the 4 four-word blocks cache:

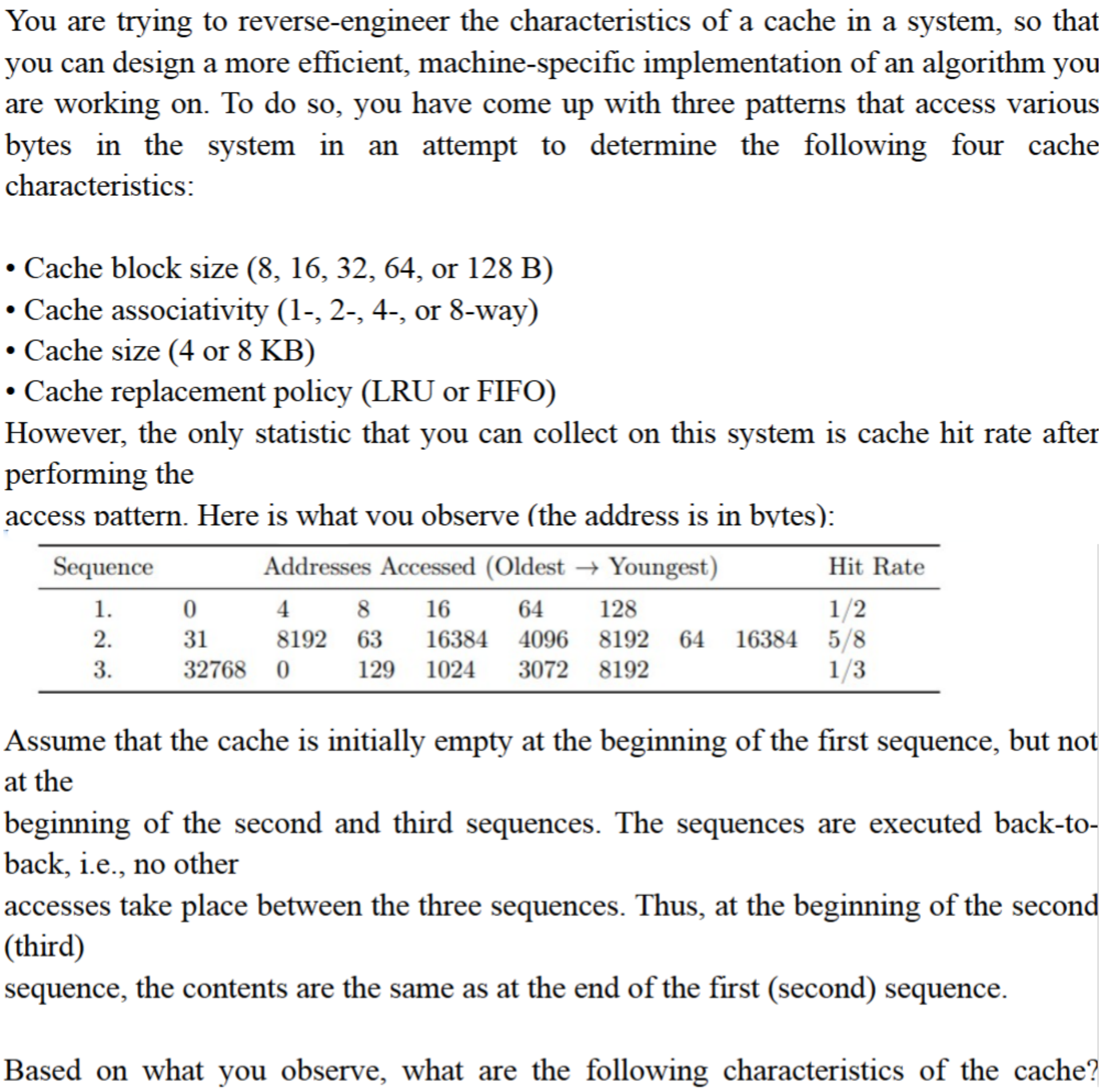
- Each block has 2 words (8 bytes), requiring 3 bits for the block offset.
- The cache is fully associative, so the index is 0 bits.
- The remaining 29 bits are used for the tag.

Word address	Binary address	Tag	Hit/Miss
1	00000001	000000000000000000000000000	Miss
134	10000110	000000000000000000000010000	Miss
212	11010100	000000000000000000000011010	Miss
1	00000001	000000000000000000000000000	Hit
135	10000111	000000000000000000000010000	Hit
213	11010101	000000000000000000000011010	Hit
162	10100010	000000000000000000000010100	Miss
161	10100001	000000000000000000000010100	Hit
2	00000010	000000000000000000000000000	Hit
44	00101100	000000000000000000000000101	Miss
41	00101001	000000000000000000000000101	Hit
221	11011001	000000000000000000000011001	Hit

By LRU replacement policy, the cache process is as follows: 1 miss; 134 miss; 212 miss; 1 hit; 135 hit; 213 hit; 162 miss; 161 hit; 2 hit; 44 miss and replace 134; 41 hit; 221 hit.

so the miss rate is $5/12 = 0.4167$

Problem 5.



You are trying to reverse-engineer the characteristics of a cache in a system, so that you can design a more efficient, machine-specific implementation of an algorithm you are working on. To do so, you have come up with three patterns that access various bytes in the system in an attempt to determine the following four cache characteristics:

- Cache block size (8, 16, 32, 64, or 128 B)
- Cache associativity (1-, 2-, 4-, or 8-way)
- Cache size (4 or 8 KB)
- Cache replacement policy (LRU or FIFO)

However, the only statistic that you can collect on this system is cache hit rate after performing the
access pattern. Here is what you observe (the address is in bytes):

Sequence	Addresses Accessed (Oldest → Youngest)								Hit Rate
1.	0	4	8	16	64	128			1/2
2.	31	8192	63	16384	4096	8192	64	16384	5/8
3.	32768	0	129	1024	3072	8192			1/3

Assume that the cache is initially empty at the beginning of the first sequence, but not at the
beginning of the second and third sequences. The sequences are executed back-to-back, i.e., no other
accesses take place between the three sequences. Thus, at the beginning of the second (third)
sequence, the contents are the same as at the end of the first (second) sequence.

Based on what you observe, what are the following characteristics of the cache?

Answer: (a) Cache block size

From the sequence 1,the block size is either 32 or 64 bytes to make the hit rate =0.5

From the sequence 2, we can make sure that the block size is 64 bytes. If the block size is 32 bytes, the address 63 will miss and the hit rate will not be 5/8

(b) Cache associative

if Cache is 4KB, the number of blocks is 64, if Cache is 8KB, the number of blocks is 128.

no matter the cache size is 4KB or 8KB, the 8192 address will map to the same set as 0 address, as well as 16384 address and 32768 address.

According to the sequence 3, the 32768 address, 1024 address, 3072 address must be miss and the 129 address must be hit.Considering the hit rate is 2/6, 0 address and 8192 address only one of them can be hit.

- if the cache is 1-way or 2way set associative, the 0 address and 8192 address will both be miss.
- if the cache is 8-way set associative, the 0 address and 8192 address will both be hit.
- thus the cache size is 4-way set associative.

(c) Cache size

now we know the cache is 4-way set associative, the block size is 64 bytes.

if the cache size is 4KB,the total number of blocks is 64, index size is 0-15,every set has 4 blocks.

the address 1024 will map to the same set as 0 address. At this case, no matter the Cache replacement policy is LRU or FIFO, the Sequence 3 will not be 1/3 hit rate. So the cache size is 8KB.

(d) Cache replacement policy

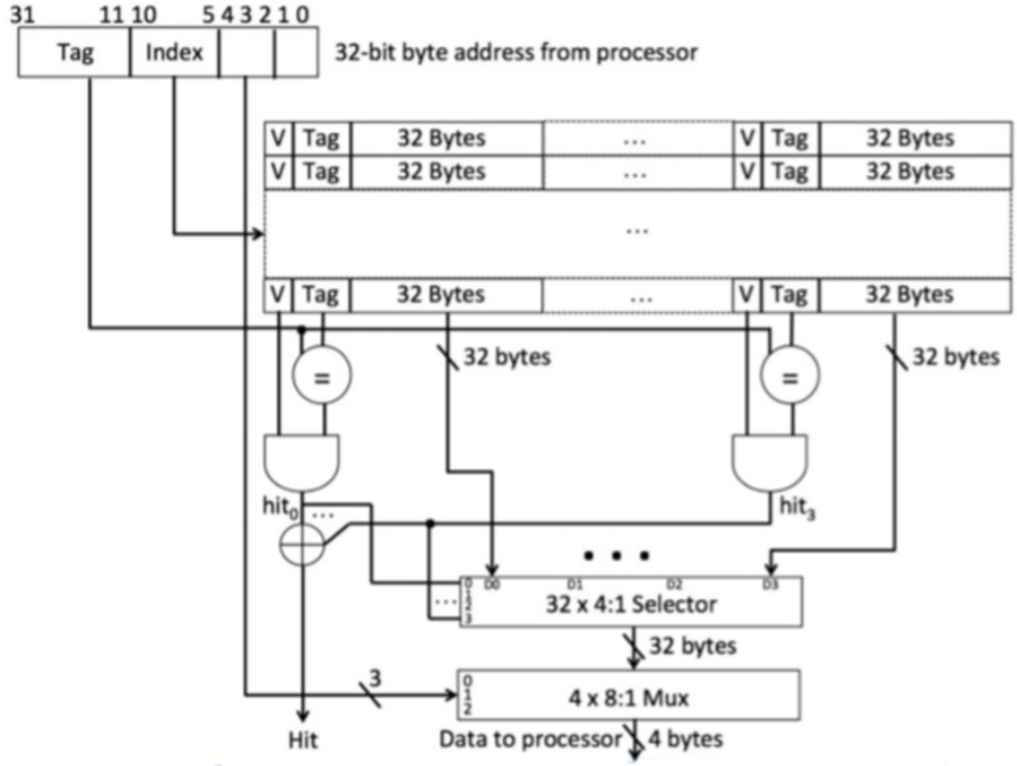
now we know the cache is 4-way set associative, the block size is 64 bytes. the cache size is 8KB. If the cache replacement policy is FIFO, the sequence 3 will be 1/6 hit rate, which is not consistent with the question. If the cache replacement policy is LRU, the sequence 1,2,3 will be 1/2, 5/8, 2/6 hit rate, which is consistent with the question.

(e) Final Answer

the cache block size is 64 bytes, the cache is 4-way set associative, the cache size is 8KB, the cache replacement policy is LRU.

Problem 6.

You are given the sketch of a cache design below:



Note: a 4:1 Selector has four control inputs labeled 0, 1, 2, 3 and four data inputs D0, D1, D2, D3. It connects Di to the output when the hit control signal is true. Assume that at most one control input is true at any time. The selector function is undefined when none of the control inputs are true.

Answer the following questions about the cache above.

a) What is the block size of the cache in bytes?
b) What is the number of blocks in this cache?
c) What is the total data capacity of the cache in bytes?
d) What is the associativity of the cache?
e) Is this a write-through or write-back cache?
f) What is the total number of valid bits in the cache?
g) What is the total number of tag bits in the cache?

Answer :

a 32Bytes

b index size = 6bits, number of blocks= $2^6 * 4 = 256$

c tag size = 32 - 6 - 5 = 21bits. total data capacity = 256 * 32 * 4 = 32786Bytes, total capacity = $256 * (21 + 1 + 32 * 8) * 4$=284,672bits=35,584Bytes

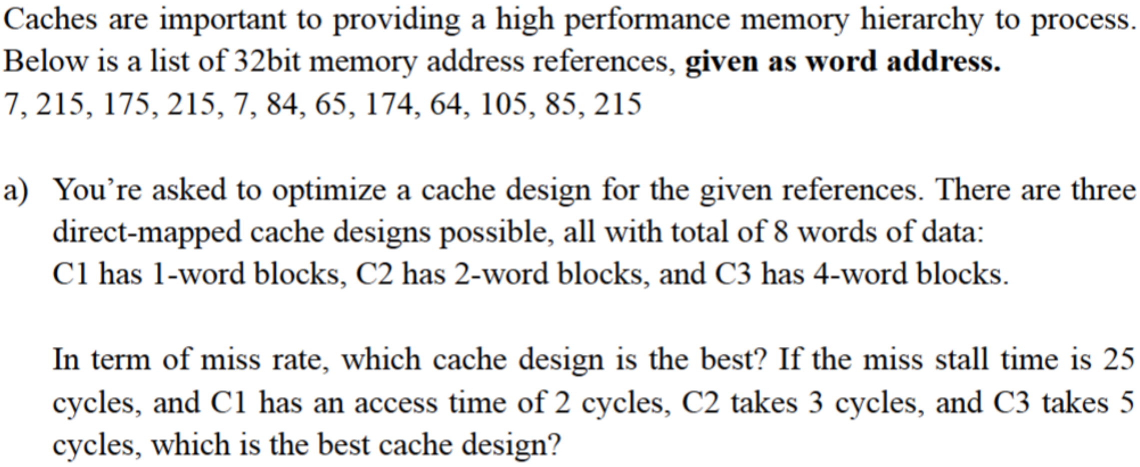
d 4-way set associative

e the cache don't have dirty bits, so the write policy is write-through

f total valid bits = 256 * 4 = 1024bits

g total number of tag bits = 256 * 21 * 4 = 21,504bits

Problem 7.



Caches are important to providing a high performance memory hierarchy to process. Below is a list of 32bit memory address references, **given as word address.**
7, 215, 175, 215, 7, 84, 65, 174, 64, 105, 85, 215

a) You're asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with total of 8 words of data:
C1 has 1-word blocks, C2 has 2-word blocks, and C3 has 4-word blocks.

In term 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?

Answer :

Word address	Binary address
7	00000000000000000000000000000111
215	00000000000000000000000011010111
175	00000000000000000000000010101111
215	00000000000000000000000011010111
7	00000000000000000000000000000111
84	00000000000000000000000001010100
65	00000000000000000000000001000001
174	00000000000000000000000010101110
64	00000000000000000000000001000000
105	00000000000000000000000001101001
85	00000000000000000000000001010101
215	00000000000000000000000011010111

- for C1,the cache has 1-word blocks, so the bit offset is 2, the cache has 8 entries, so the index is 3, the remaining 27 bits are used for the tag. the cache process is as follows: MMMHHMMHHMHM,5hits,7misses
- for C2,the cache has 2-word blocks, so the bit offset is 3, the cache has 4 entries, so the index is 2, the remaining 27 bits are used for the tag. the cache process is as follows: MMMHHMMHHMHM,5hits,7misses
- for C3,the cache has 4-word blocks, so the bit offset is 4, the cache has 2 entries, so the index is 1, the remaining 27 bits are used for the tag. the cache process is as follows: MMMHMMMMMMHM,2hits,10misses

In term of miss rate, the cache C1 and C2 have the same miss rate, 7/12=0.5833 and the cache C3 has a miss rate of 10/12=0.8333 so C1,and C2 are the best.

If the miss stall time is 25cycles, and C1 has an access time of 2 cycles, C2 takes 3 cycles, and C3 takes 5cycles, which is the best cache design?

- C1: 7 misses * 25 cycles + 5 hits * 2 cycles = 175 + 10 = 185 cycles
- C2: 7 misses * 25 cycles + 5 hits * 3 cycles = 175 + 15 = 190 cycles
- C3: 10 misses * 25 cycles + 2 hits * 5 cycles = 250 + 10 = 260 cycles

so the best cache design is C1.