Memory Systems

Memory hierarchy, cache memory, virtual memory

Gloria Beraldo (gloria.beraldo@dei.unipd.it)
Department of Information Engineering, University of Padova

Topics:

- Memory hierarchy and access time
- Locality of reference
- Types of cache memories and their usage
- Virtual memory

Book reference:

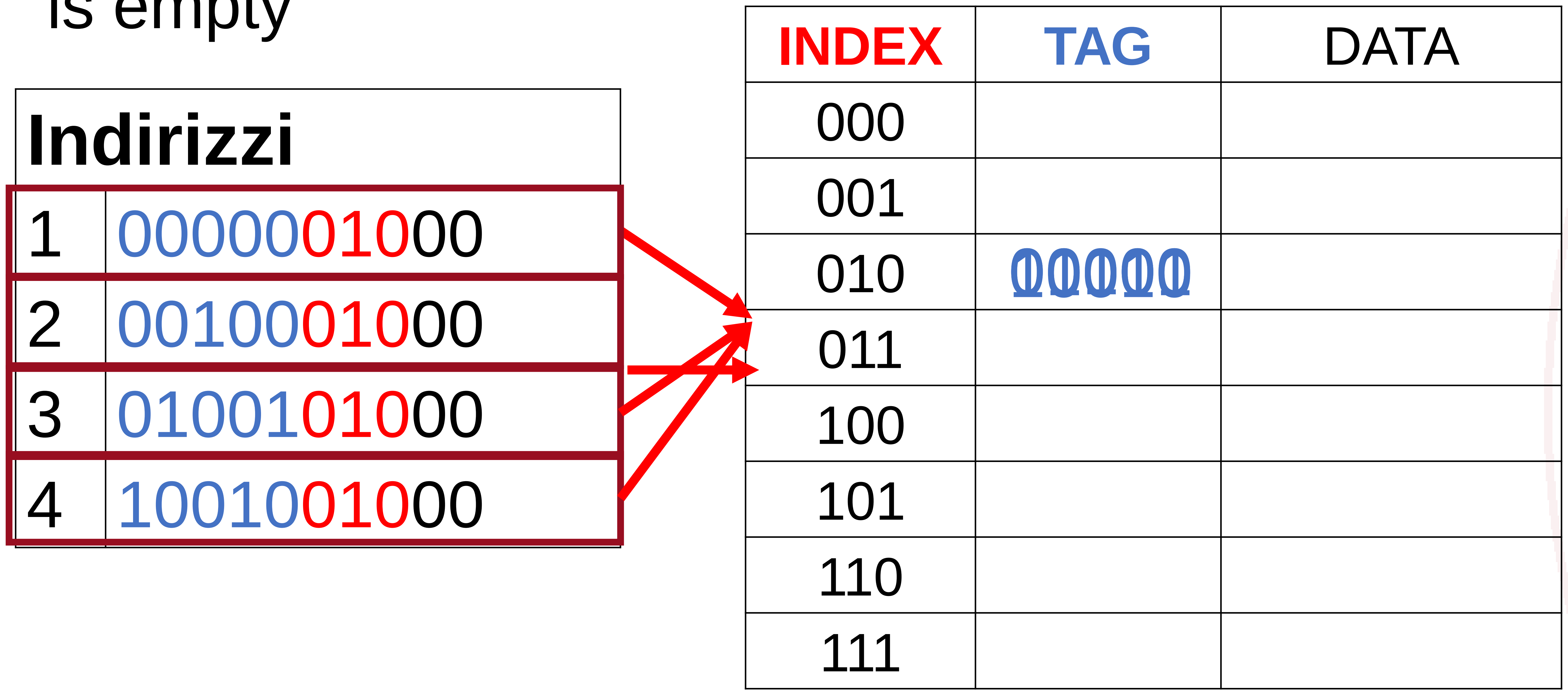
- Chapter 12

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Direct mapped cache: pro and cons

- **Pro**:
  - Simple organization
- **Cons**:
  - The line to be overwritten is predetermined: a block cannot be inserted in a line chosen from those that are probably no longer needed
  - Lowe Hit rate (less efficiency).

Example: The CPU requires the access to the following adressess, at the beginning, the cache is empty



Indirizzi	
1	0000001000
2	0010001000
3	0100101000
4	1001001000

INDEX	TAG	DATA
000		
001		
010	[illegible]	
011		
100		
101		
110		
111		

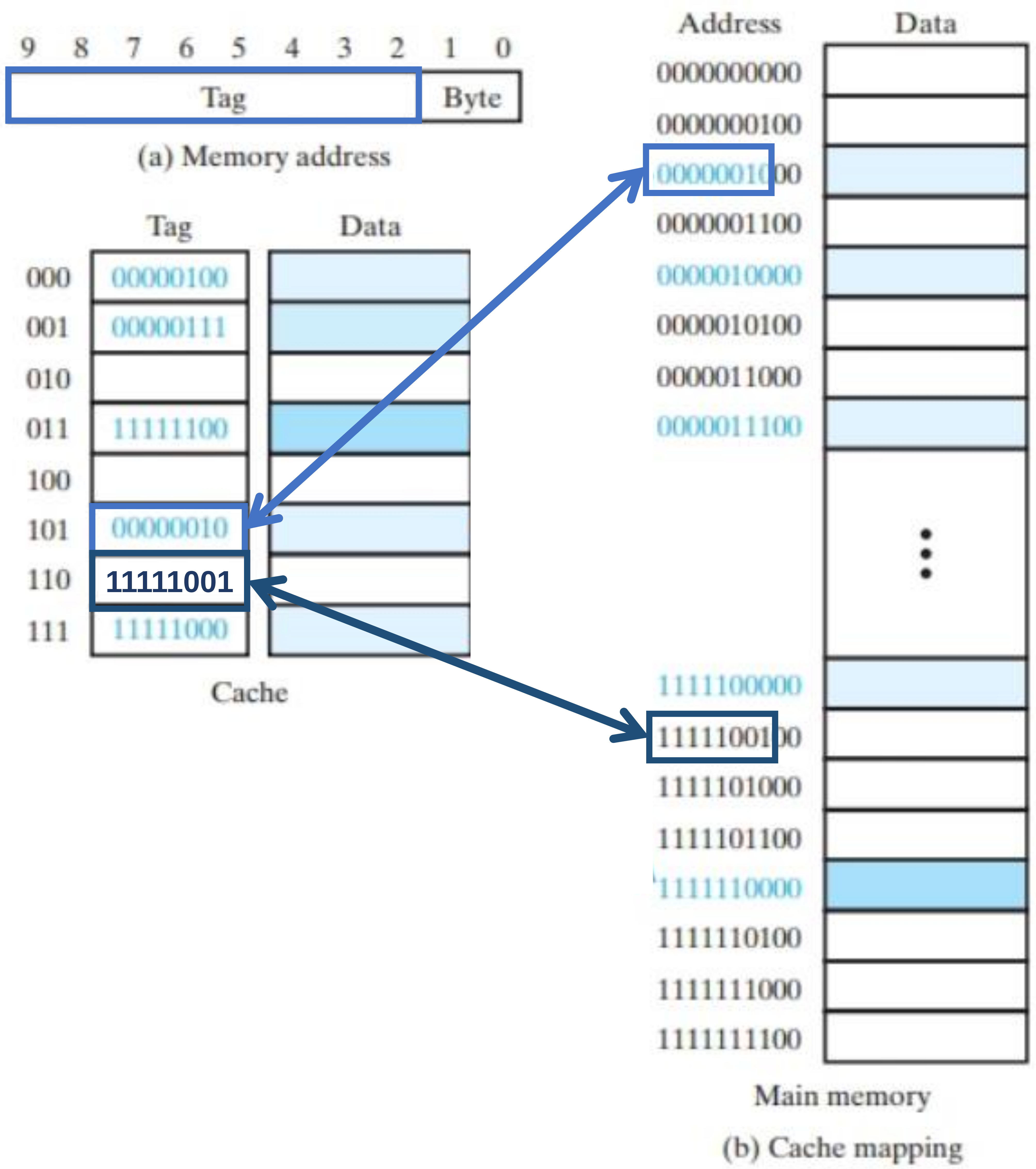
1. Load the data at the index **010**
2. **TAG** different, delete the data at the **index 010**, and load the new data with **TAG 00100** (also with the free lines)
3. **TAG** different, delete the data at the **index 010**, and load the new data with **TAG 01001** (also with the free lines)
4. **TAG** different, delete the data with **index 010**, and load the data with **TAG 10010** (also with the free lines)

Fully Associative Cache

Given a 32 bits **cache memory** with 8 words (4 bytes each) and 1KB (256 words) **RAM memory**

The concept behind the fully associative cache is that each memory address can be associated arbitrarily with each cell in the cache

- The **TAG** is identified by the 8 most significant bits
- If, after the request from the CPU, the **TAG** is found in the cache, the corresponding data is loaded
- If the **TAG** is not found, the data is requested in memory and is also saved in the cache
- The new **TAG** is inserted in an empty line
- If there are no empty lines, an «old» **TAG** is replaced with a replacement policy



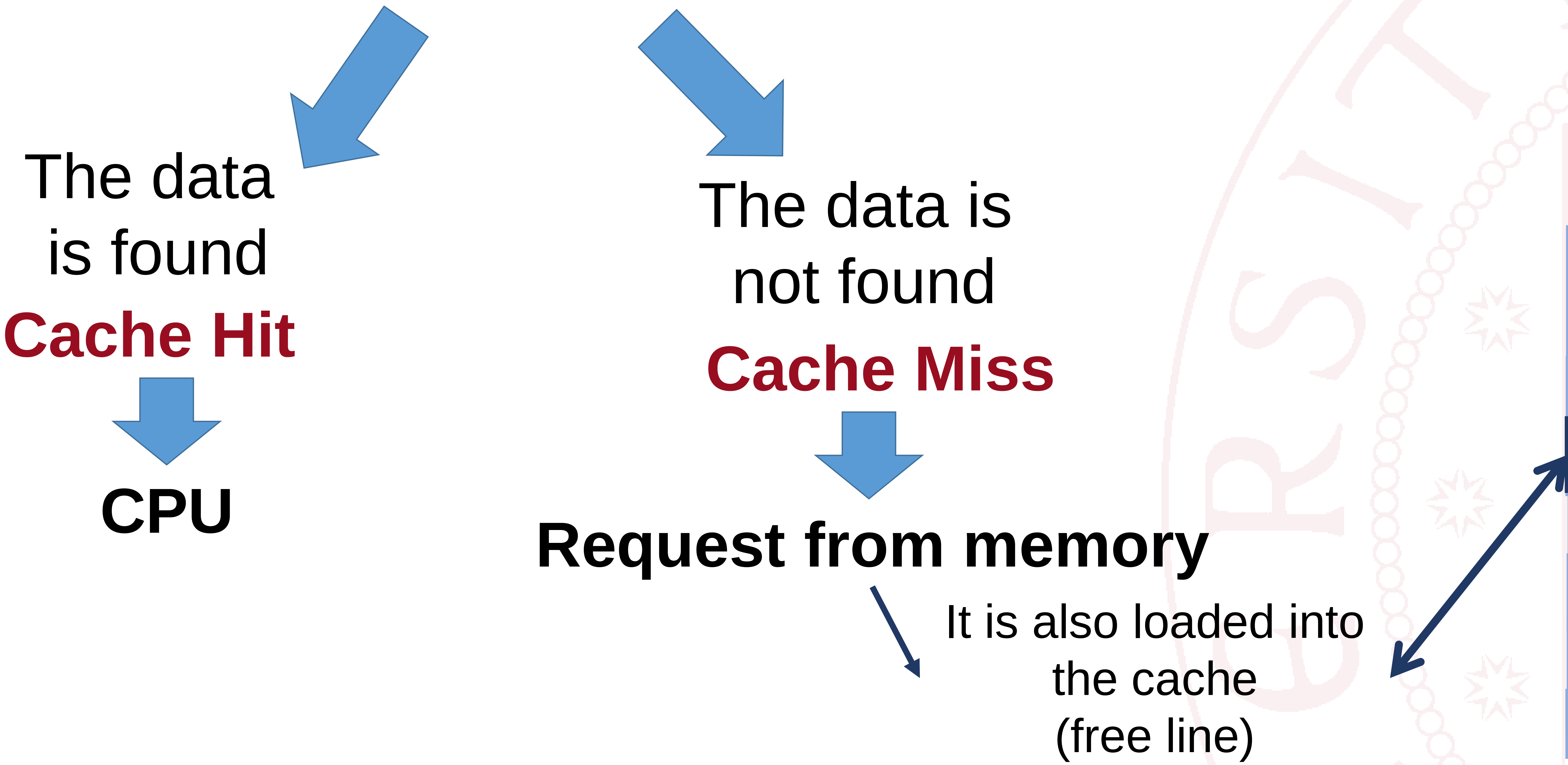
□ **FIGURE 12-4**
Fully Associative Cache

Fully Associative Cache: functioning

- The CPU requires an instruction from the address:

0	0	0	0	0	0	1	1	0	0

- The instruction can be in cache or in memory
- The cache divides the **TAG** and the data
- Look for the **TAG** in the cache



The data is found

Cache Hit

CPU

The data is not found

Cache Miss

Request from memory

It is also loaded into the cache (free line)

Scenario 1

TAG	DATA
01000100	
11000100	
11111100	
00000011	
00000100	

Scenario 2

TAG	DATA
01000100	
11000100	
11111100	
00000011	
00001011	
00000100	

Fully associative cache: replacement methods

If the cache is full and a new item needs to be inserted, replacement methods are applied:

1. **Random replacement**: random replacement of an item currently in the cache
2. **FIFO replacement**: the oldest item in the cache
3. **Least recently used (LRU)**: the least recently used element

Fully associative cache: replacement methods

Suppose the CPU requires the following addresses:

0000001000, 0000001100, 0000010000, 0000001000, 0000001100,

0000011100, 0000001000, 0000011000

TAG	D
00000010	

MISS

TAG	D
00000010	
00000011	

MISS

TAG	D
00000010	
00000011	
00000100	

MISS

TAG	D
00000010	
00000011	
00000100	

HIT

TAG	D
00000010	
00000011	
00000100	

HIT

TAG	D
00000010	
00000011	
00000100	
00000111	

MISS

TAG	D
00000010	
00000011	
00000100	
00000111	

HIT

00000110 causes a miss!!!

Which line to remove?

Fully associative cache: replacement methods

TAG	D
00000010	
00000011	
00000100	
00000111	

HIT

00000110 is a miss!!!

Which line to remove?

Random replacement:

random replacement of an item currently in the cache

TAG	D
00000010	
00000011	
00000100	
00000110	

MISS

FIFO replacement:

the oldest item in the cache

TAG	D
00000110	
00000011	
00000100	
00000111	

MISS

Least recently used (LRU):

the least recently used element

TAG	D	
00000110		3 times
00000011		2 times
00000110		1 time
00000111		1 time

MISS

Fully associative cache: TAG search

Example: The CPU requests access to an address (cache access time: 2 ns),

$N_{word-cache}$ = 8

- **Best case scenario**
  - The address is in the cache and is located in the first memory location
  - Time to retrieve the data: 2 ns
- **Worst case scenario**
  - The address is not in the cache
  - Time to retrieve the data: $2\,ns \cdot N_{word-cache}$ →$16\,ns$
  - Time comparable with the access to the main memory

Fully associative cache: TAG search

- The search for the **TAG** is speeded up with the use of an associative memory
- **Associative memory** or **CAM** (Content Addressable Memory) is a memory able to carry out, in parallel, the comparison between a searched data and all the data contained therein.
- The memory returns the addresses of the data contained therein and equal to the searched data.

Searched data
xyz

TAG	DATA
01000100	abc
11000100	bcd
11111100	def
00001011	xyz
00000100	hal

Hit Address
00001011

Fully associative cache: CAM

- Reading in fully associative memory is symmetrical with respect to the RAM:
  - la RAM receives in input an address and returns the data contained at the address;
  - the CAM receives a data in input and returns the address (or addresses) of the locations that contain that data

- Consequently, the hardware that makes a CAM is very complex and expensive.

Fully associative cache: pro e cons

- **Pro**:
  - Simple organization
  - Ability to exploit all cache locations
- **Cons**:
  - Very complex to make and very expensive

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains **4 bytes**. The cache uses the LRU as a replacement method.

YOU ARE ASKED TO DESIGN THE CACHE DESCRIBED ABOVE

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.

- $\frac{4*8\ bit}{16\ bit} = 2\ words\ per\ line$
- $2\ lines$

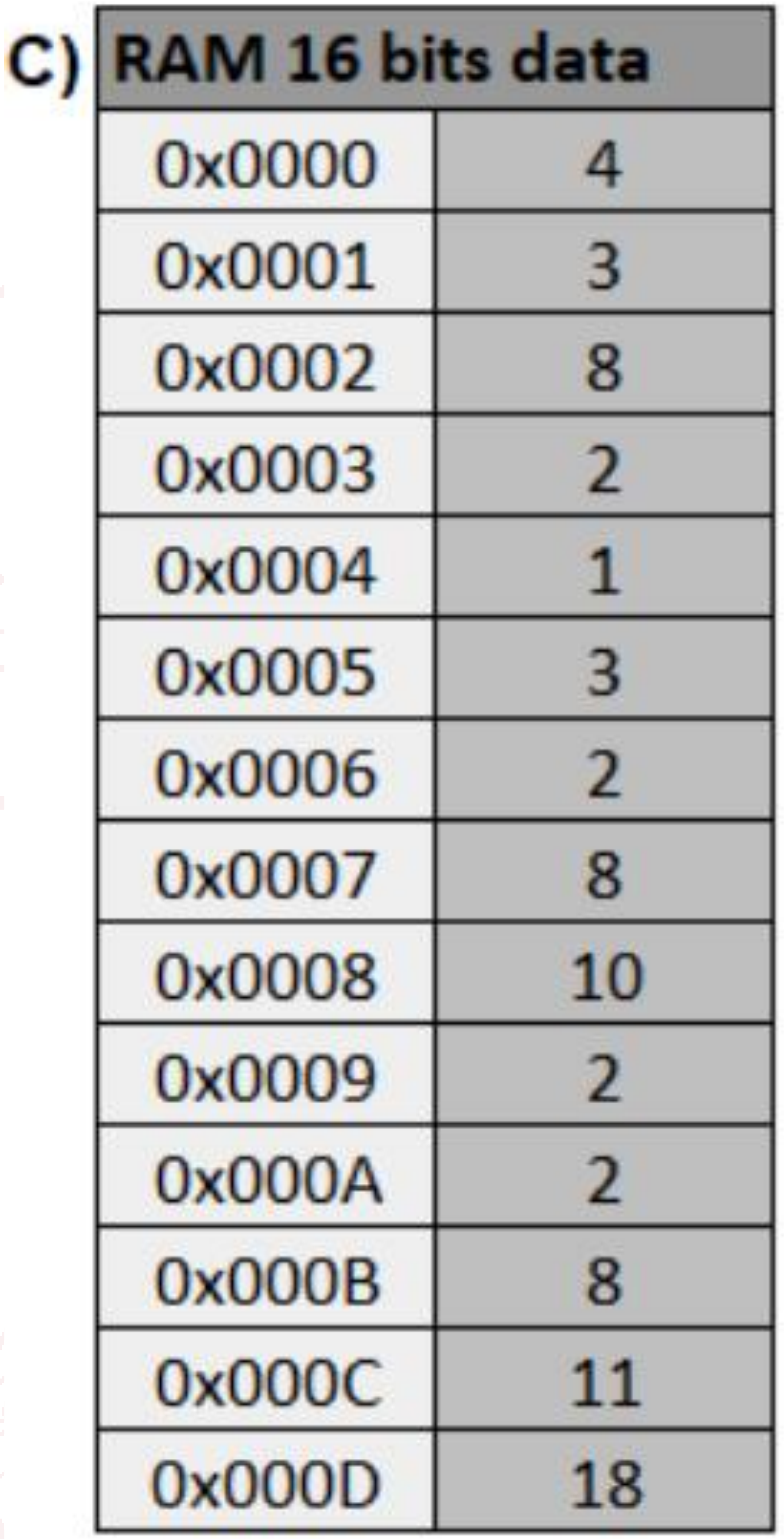
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.

THE DATA IN THE MAIN MEMORY ARE PROVIDED



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

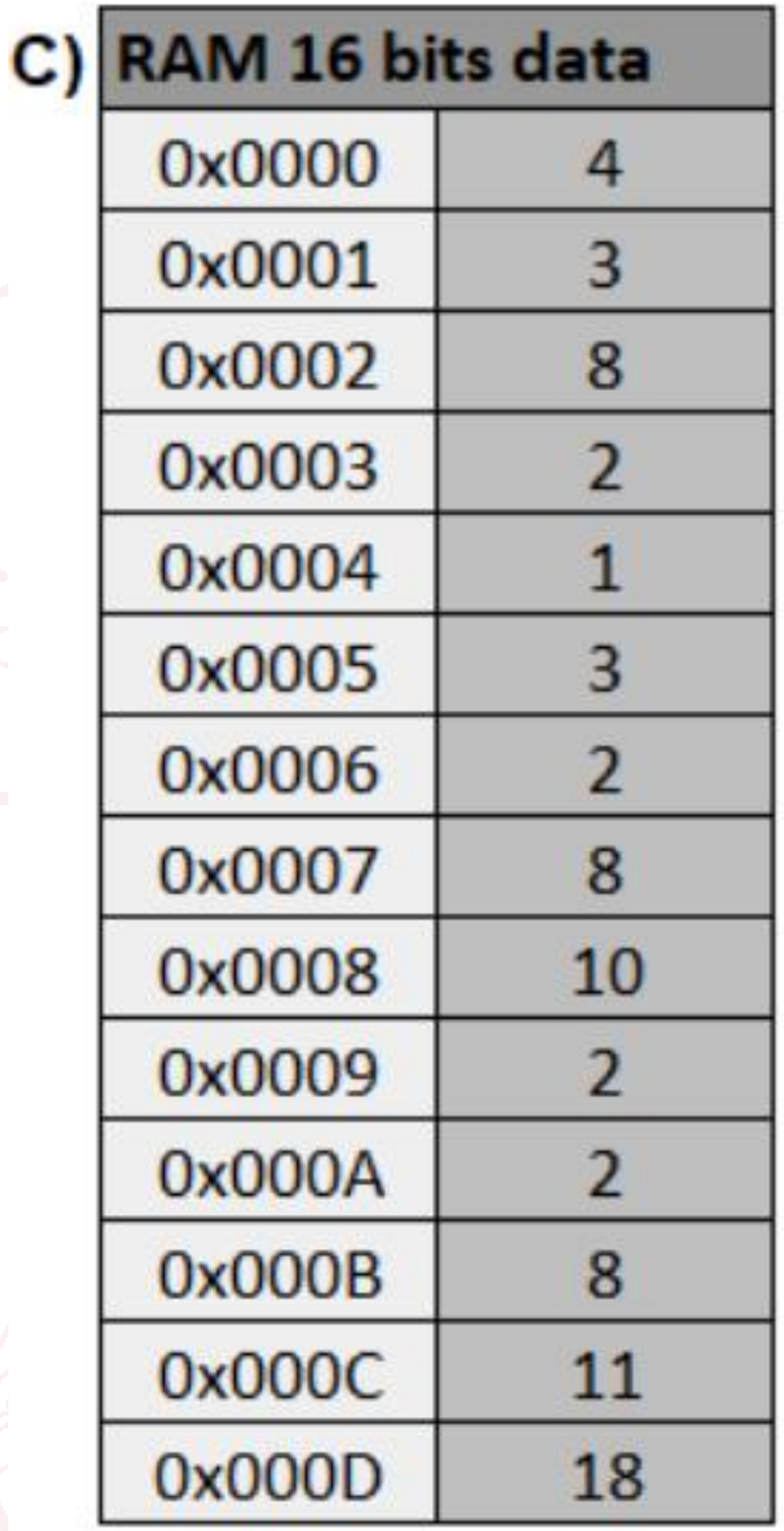
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.

THE DATA IN M[3] HAS TO BE READ



C)

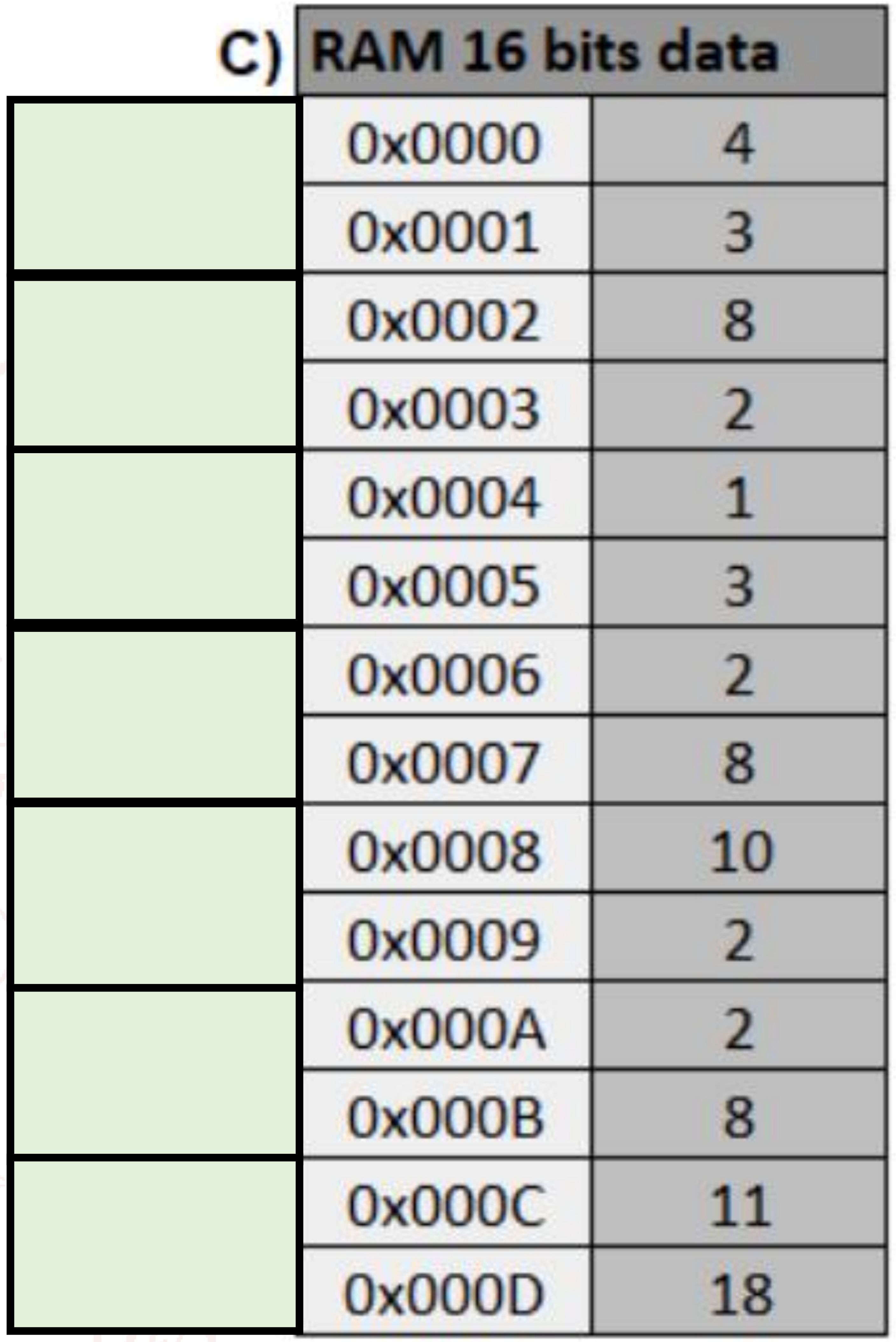
RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.



C)	RAM 16 bits data	
	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
	0x000C	11
	0x000D	18

THE DATA IN M[3] HAS TO BE READ

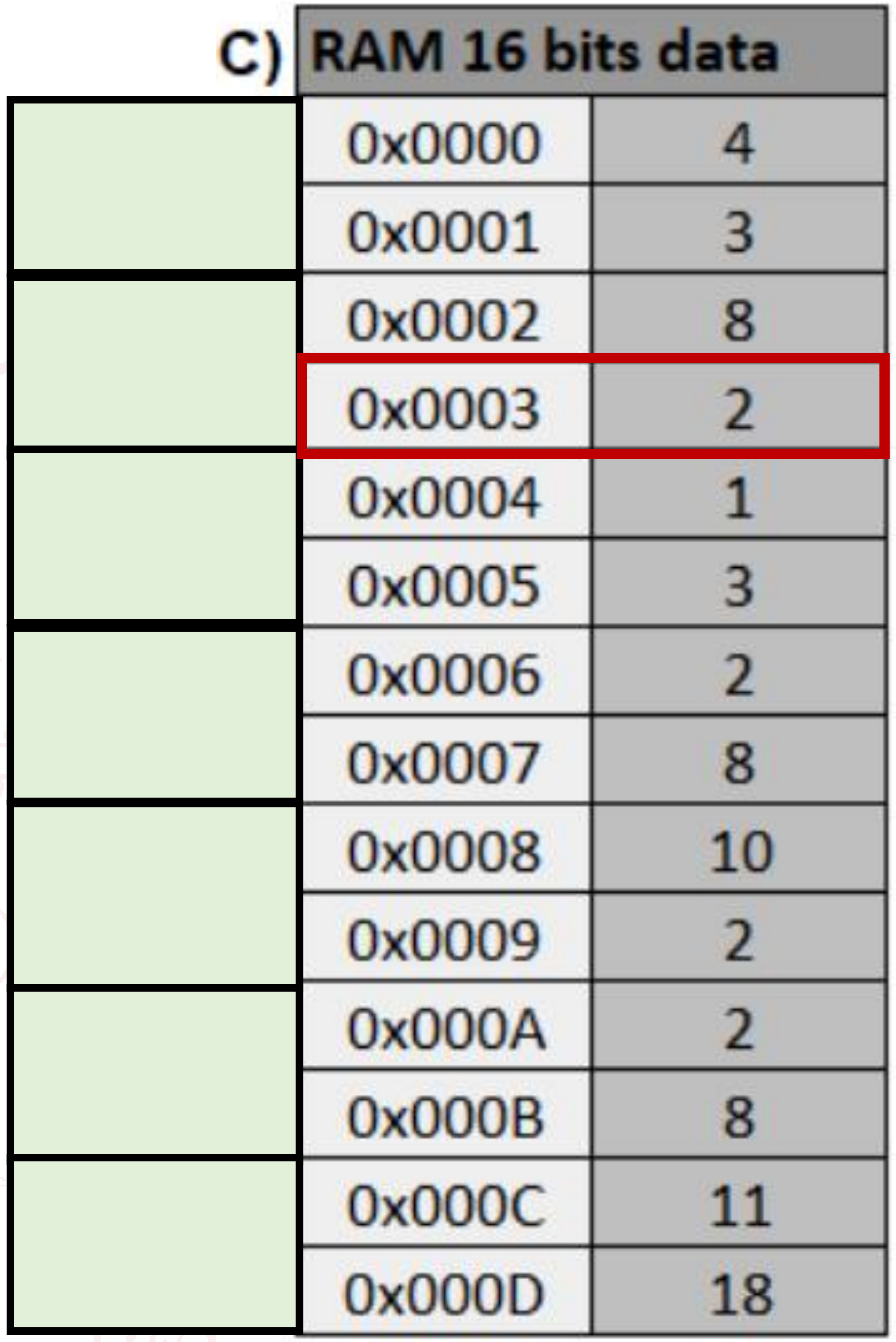
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.

The cache is empty, we have a **MISS, the corresponding block is copied in the cache**



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

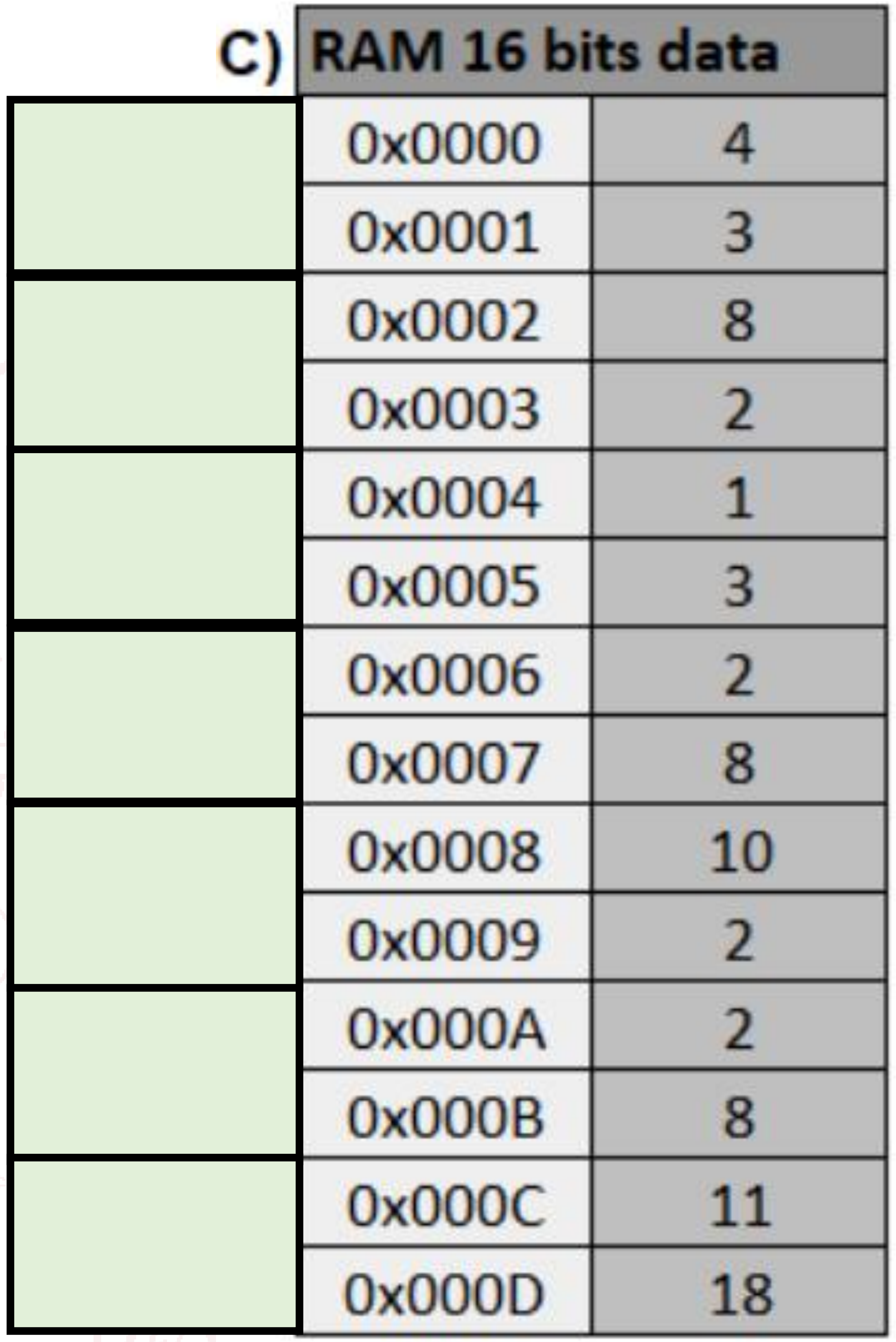
8	2

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-**bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.



C)	RAM 16 bits data	
	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
	0x000C	11
	0x000D	18

THE DATA IN M[2] HAS TO BE READ

Example from previous exam

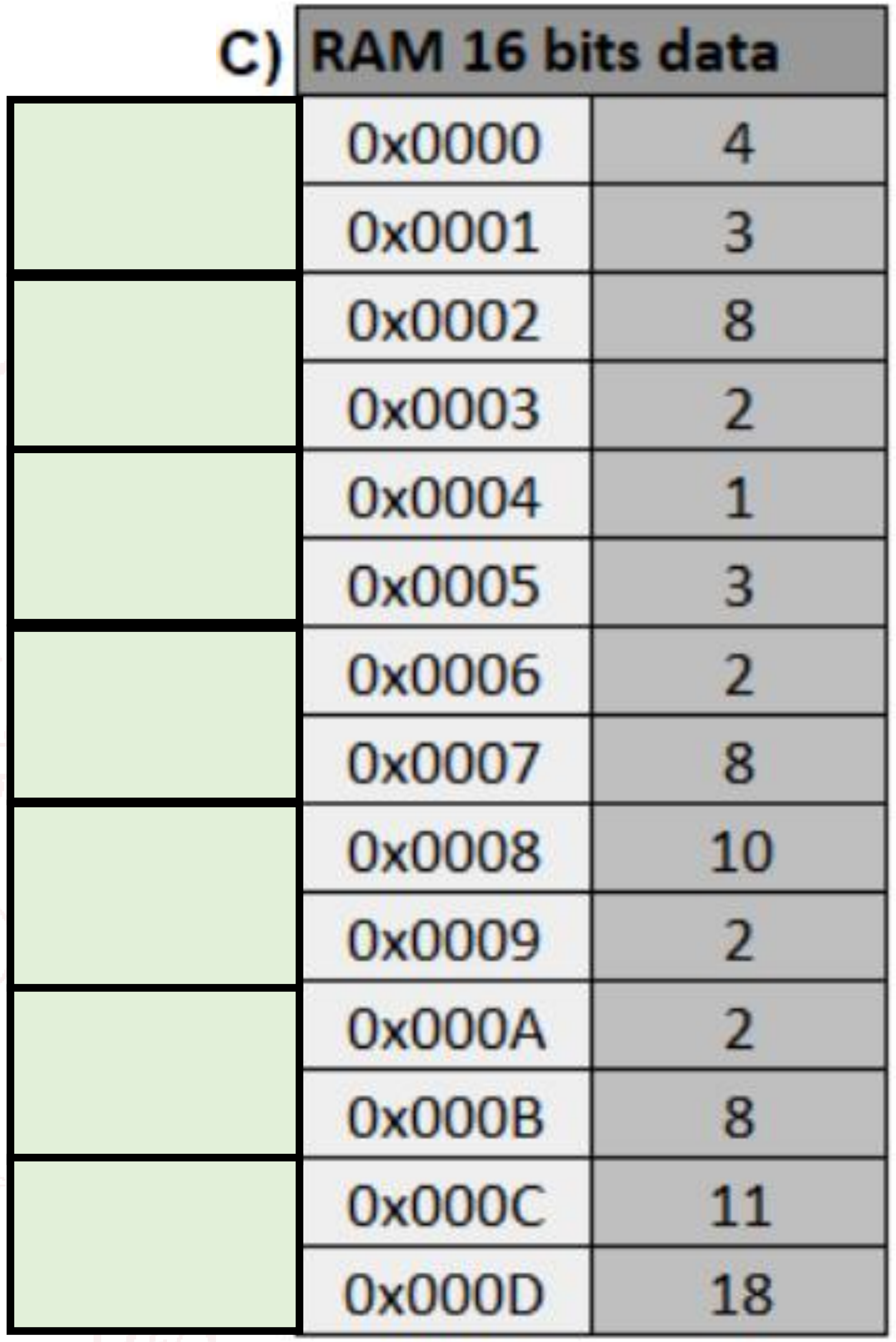
Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.

The cache contains the data we need, we have a **HIT**

8	2



C)

	RAM 16 bits data	
	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
	0x000C	11
	0x000D	18

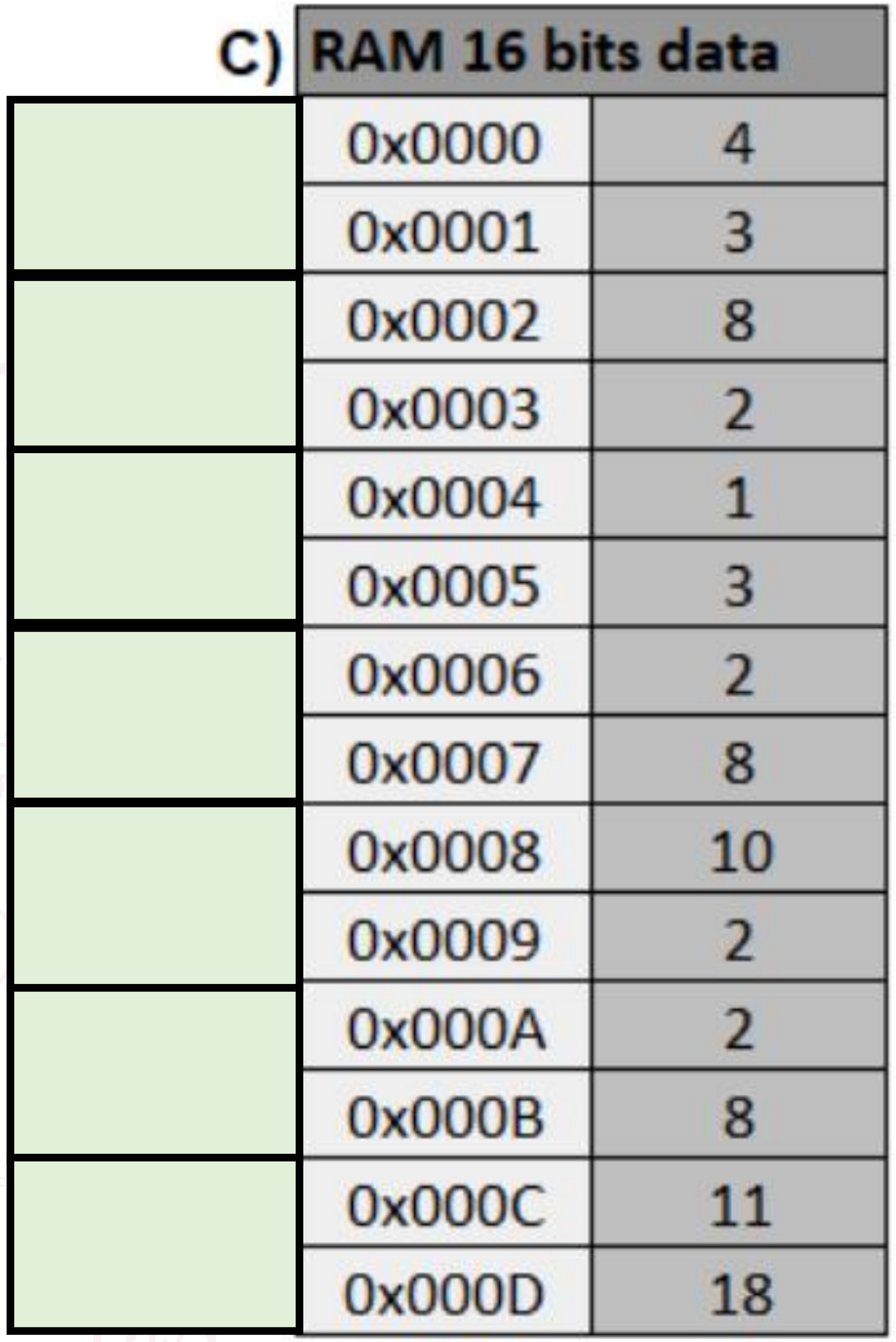
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-**bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.

THE DATA IN M[4] HAS TO BE READ



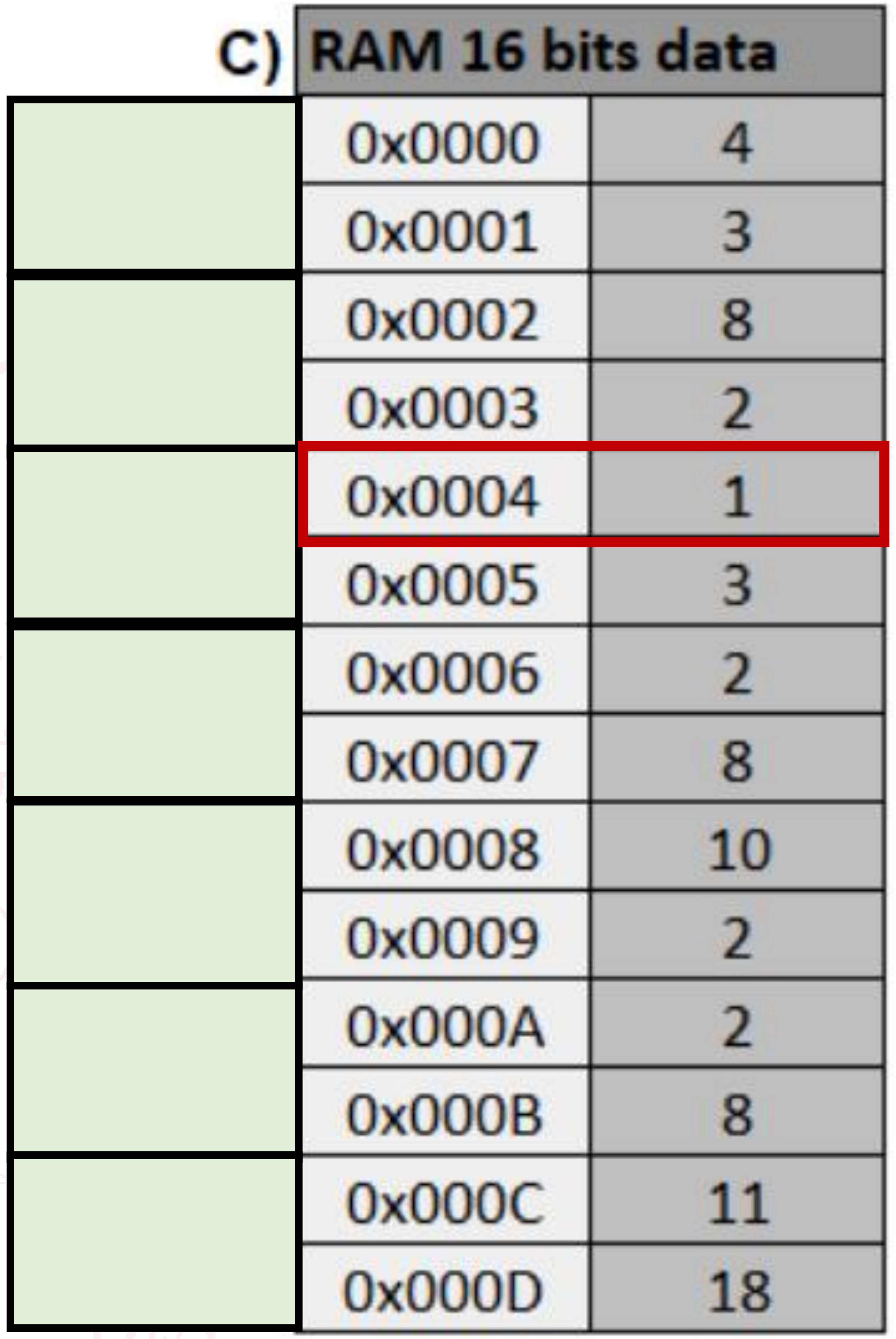
C)	RAM 16 bits data	
	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
	0x000C	11
	0x000D	18

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.



C)	RAM 16 bits data	
	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
	0x000C	11
	0x000D	18

The data is not in the cache, we have a **MISS, the corresponding block is copied in the cache**

8	2
1	3

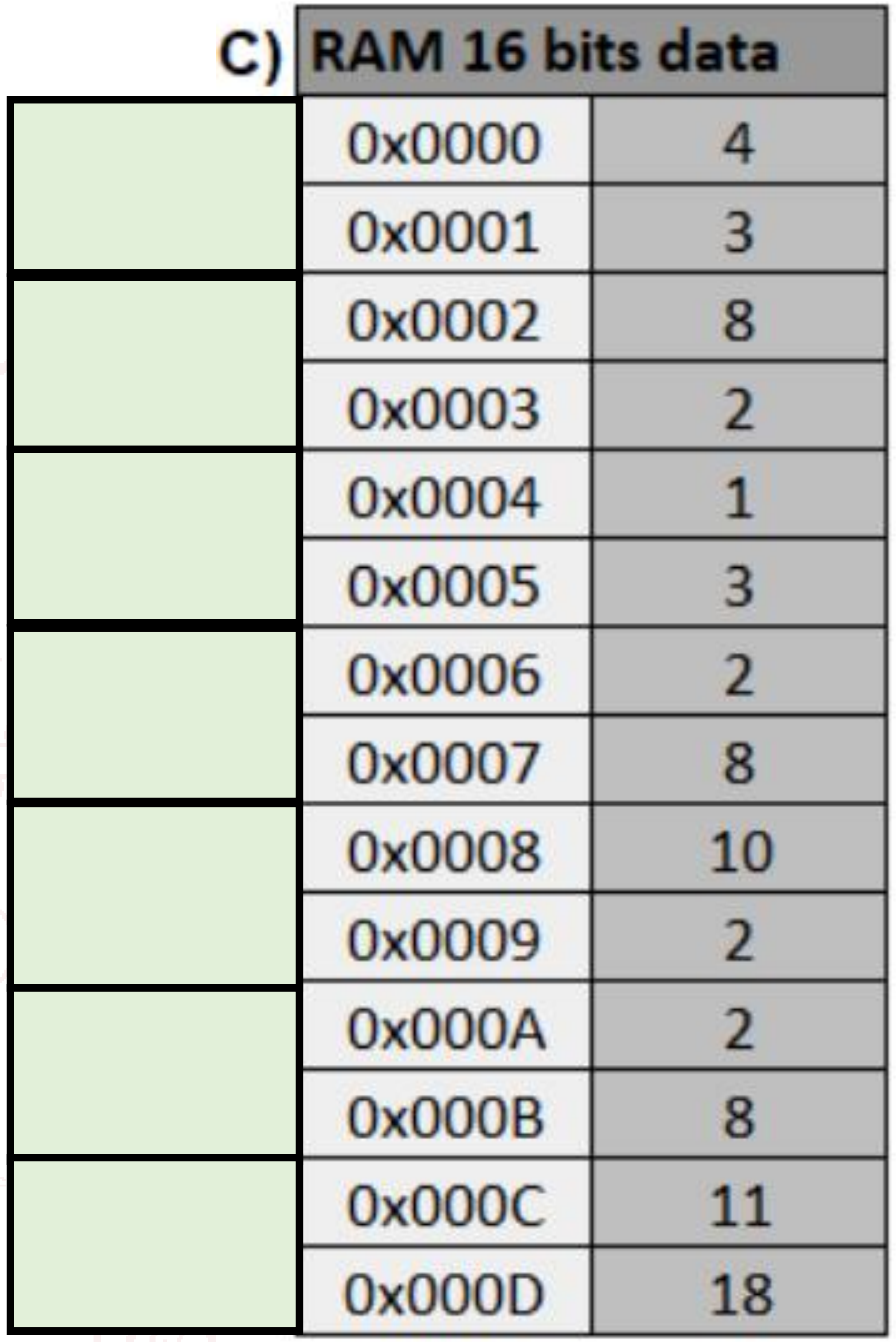
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.

THE DATA IN M[6] HAS TO BE READ

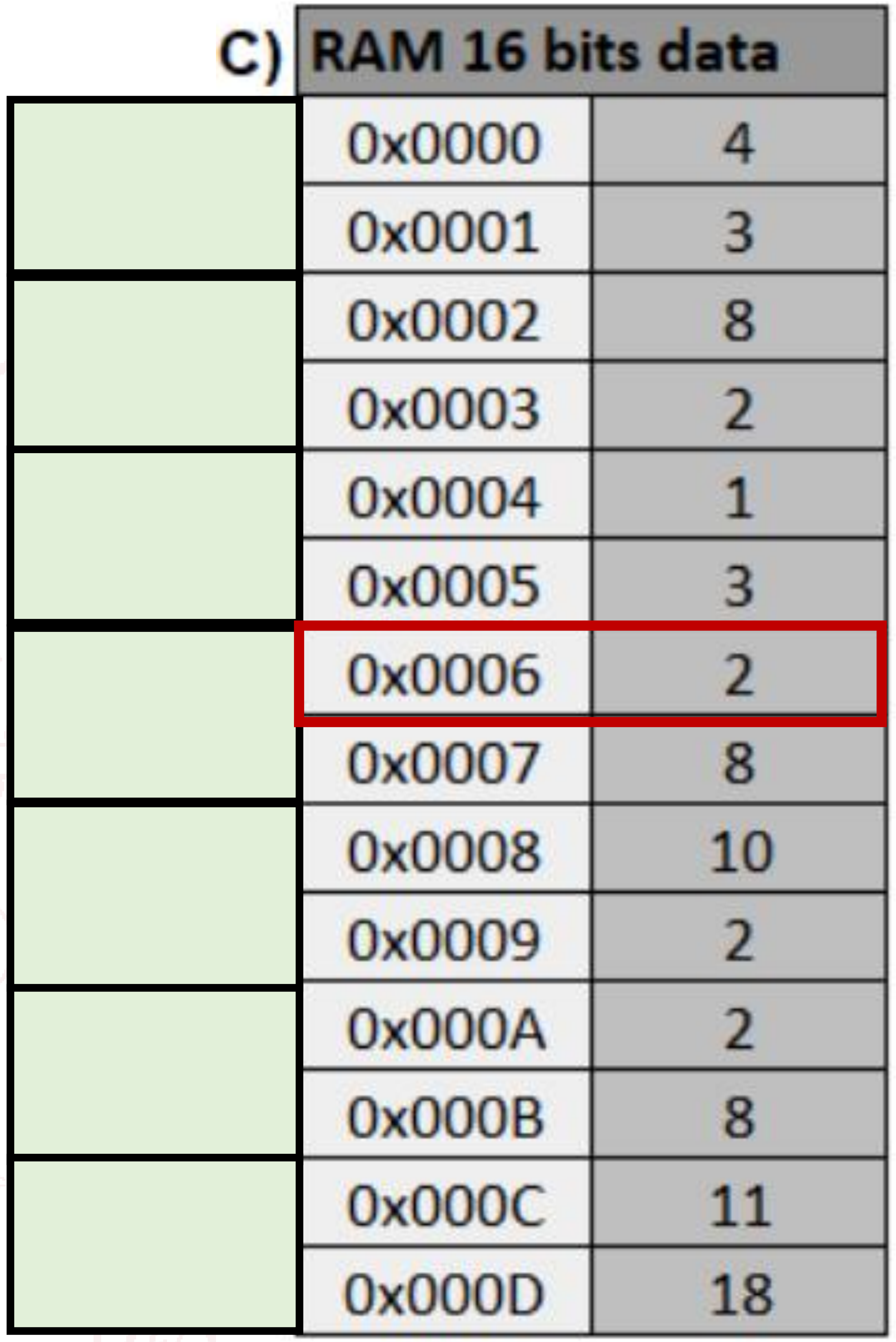


C)	RAM 16 bits data	
	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
	0x000C	11
	0x000D	18

Example from previous exam

Description:

A single-cycle computer has the following specifications:
- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via **16-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the LRU as a replacement method.



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

The data is not in the cache, we have a **REPLACE, the corresponding block is copied in the cache**

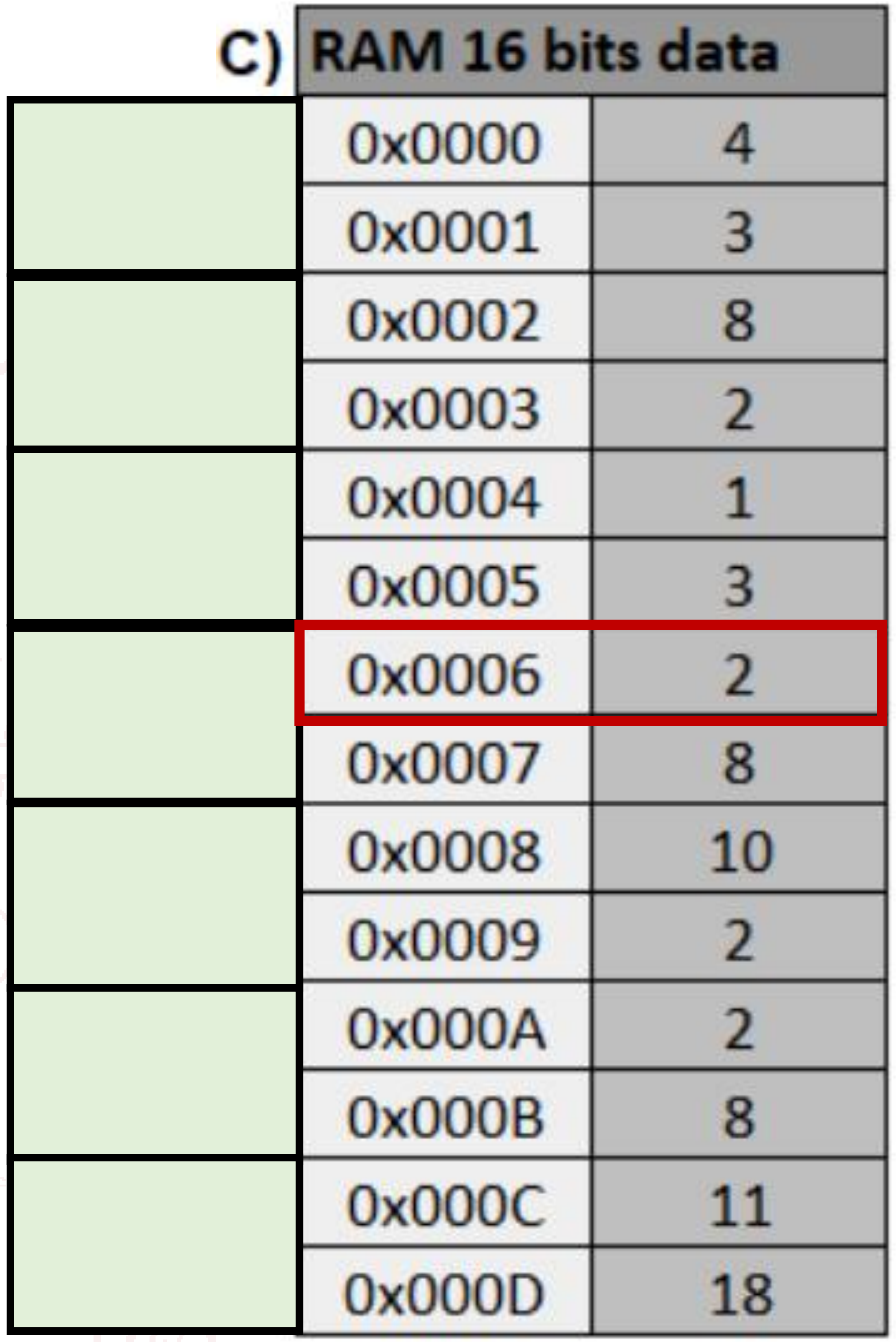
8	2
1	3

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16**-bit words**;
- There is a fully associative cache with **2 lines** connected only to the data memory. Each line contains 4 **bytes**. The cache uses the **LRU as a replacement method**.



C)	RAM 16 bits data	
	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
	0x000C	11
	0x000D	18

We need to apply the LRU policies as a replacement method

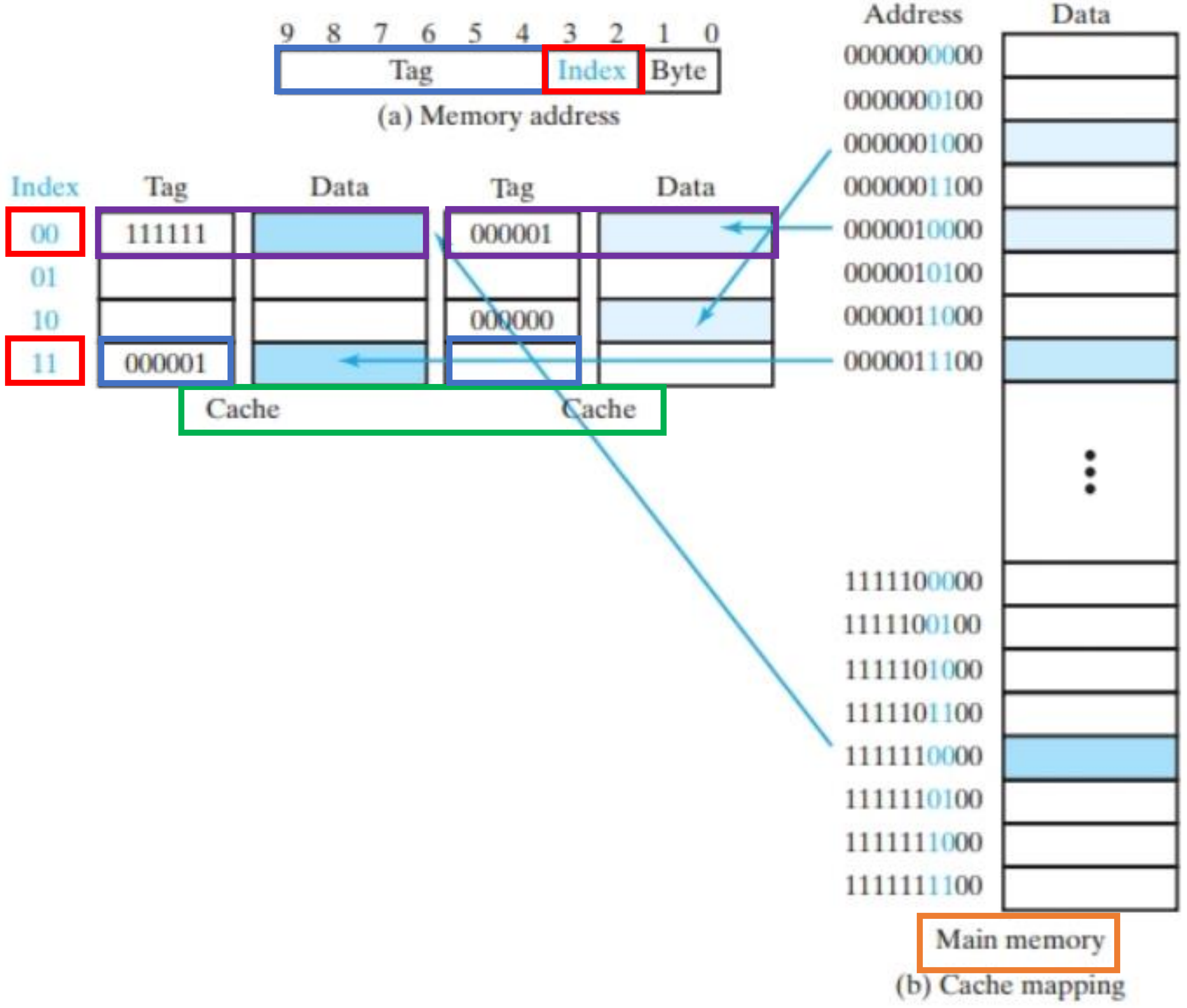
8	2
2	8

K-way set associative cache

Consider a 32 bits **cache memory** with 8 words (4 bytes each) and a 1 KB **RAM memory** (256 words)

The concept behind the k-way set associative cache is to associate an **INDEX** with more locations (< number of locations in the CACHE) in order to speed the **TAG** match up. It is a solution in the middle between the direct mapped cache and the fully associative cache

- For each **INDEX** there is a **SET** s (ways) of locations (e.g., with s=2 → it has 2 ways)
- Cache structure: 4 rows, 2 columns
- For each **INDEX** (row), 2 **TAGs** (columns) are associated
- Once the **INDEX** has been identified, the**TAG** search is made on the number of columns



(a) Memory address

(b) Cache mapping

K-way set associative cache: functioning

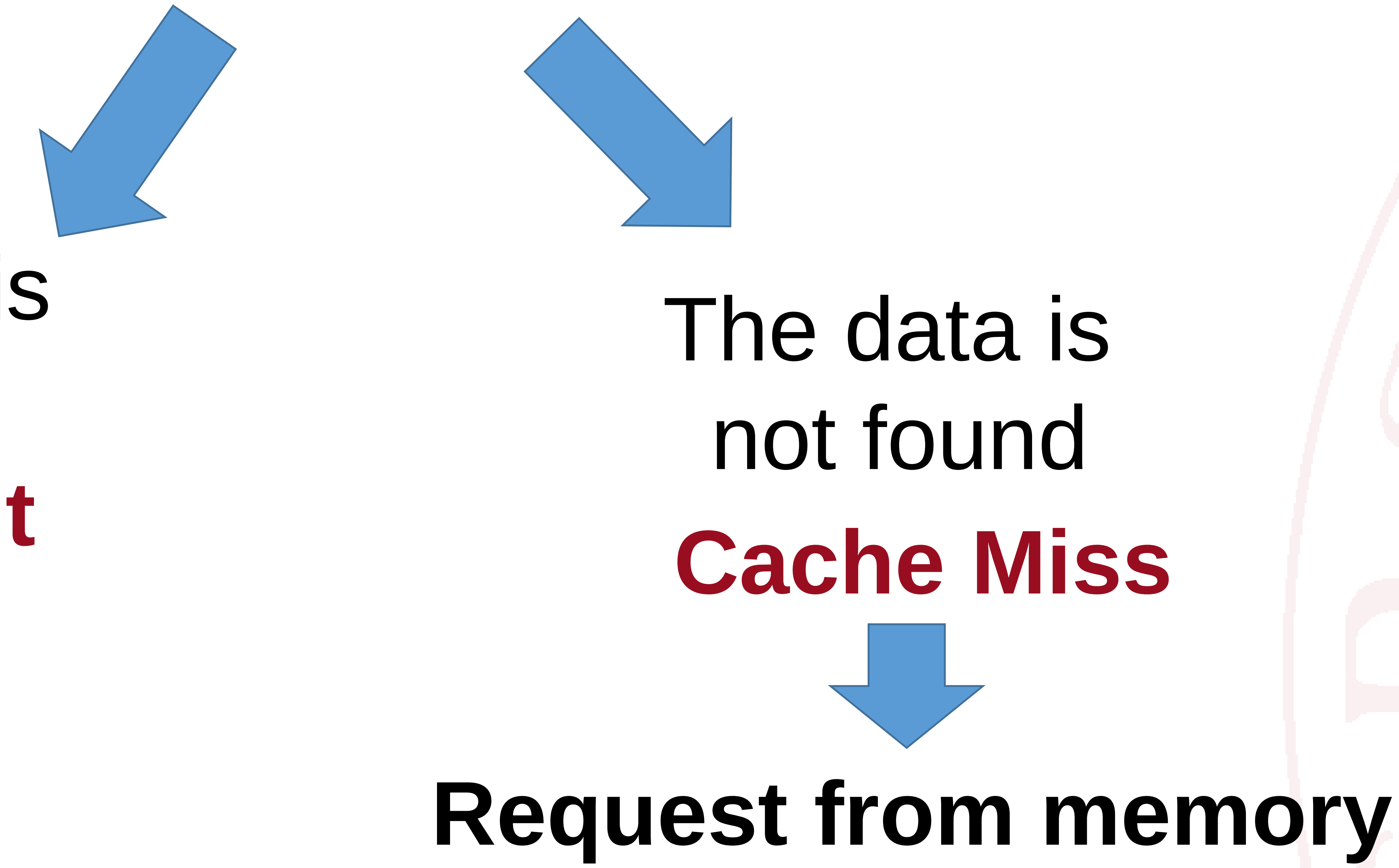
- The CPU requires an instruction from the address:

0	0	0	0	0	0	1	1	0	0

- The cache divides the **TAG** and the **INDEX** (000000 and 11)
- The **TAGs** related to index11 are retrieved
- Compare the **TAGs** with the required instruction

INDEX	TAG	DATA
00		
01		
10		
11	010000 000000	
	000100	

2 ways



The data is found

Cache Hit

CPU

The data is not found

Cache Miss

Request from memory

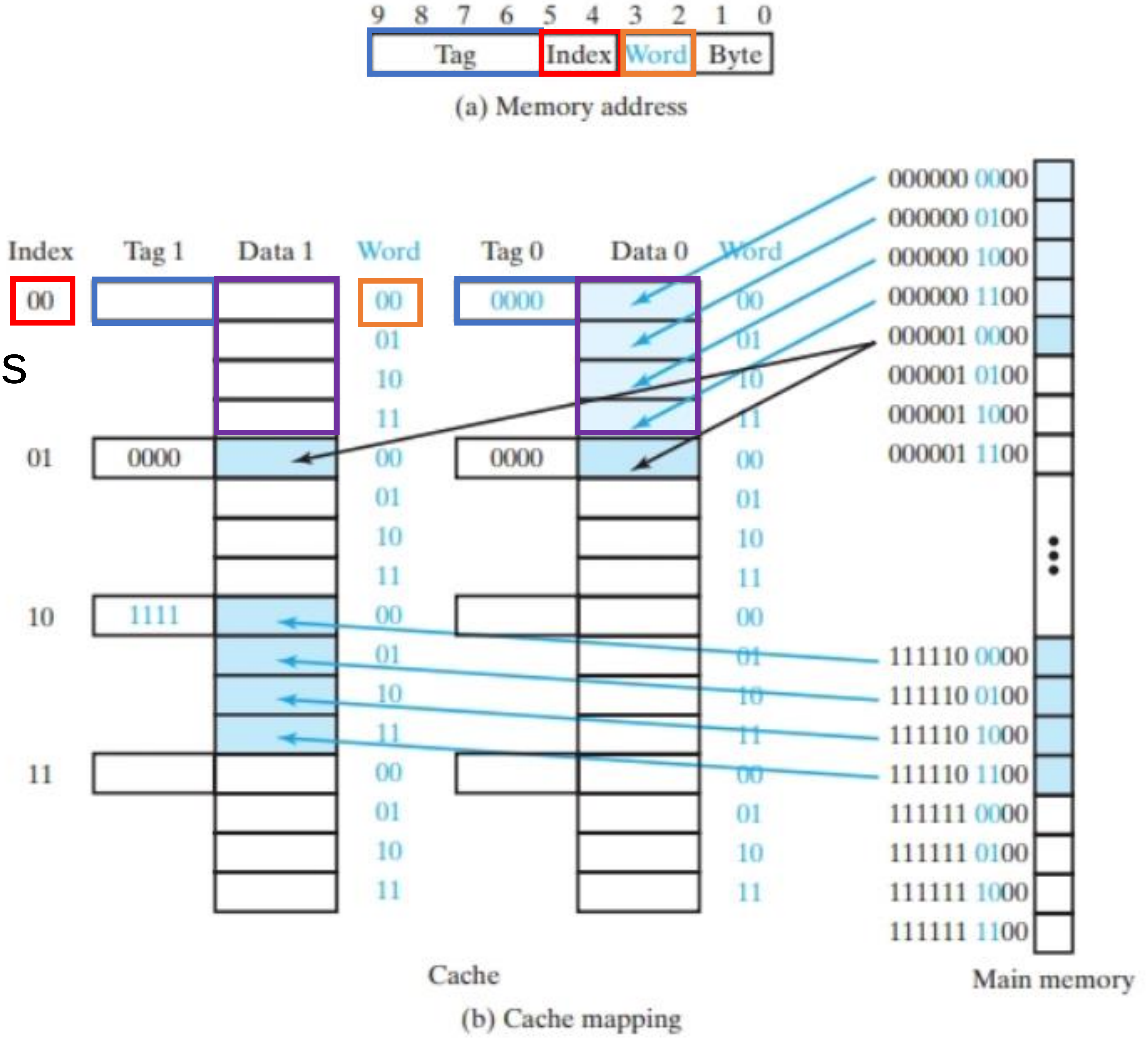
2-way set-associative with 4-word lines

0	0	0	0	0	0	1	1	0	0

- The 10 bits address are divided into the **TAG**, **INDEX**, **WORD**, Data
- For each **TAG**, there is a block (**LINE**) of 4 words identified by the bits 3 and 2
- The **INDEX** is applied to identify the 2 **TAGs**
- The **INDEX** and the **WORD** are applied to read the 2 words associated with the**TAGs**
- The 2 **TAGs** are compared with the required address
- If they **MATCH**, the word is provided to the CPU
- If they don't **MATCH**, the **TAG** and the **WORD** are used to load the **LINE** (4 words) into the cache

▼

Spatial locality



□ **FIGURE 12-8**
Set-Associative Cache with 4-Word Lines

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words lines** connected only to the data memory.

YOU ARE ASKED TO DESIGN THE CACHE DESCRIBED ABOVE

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

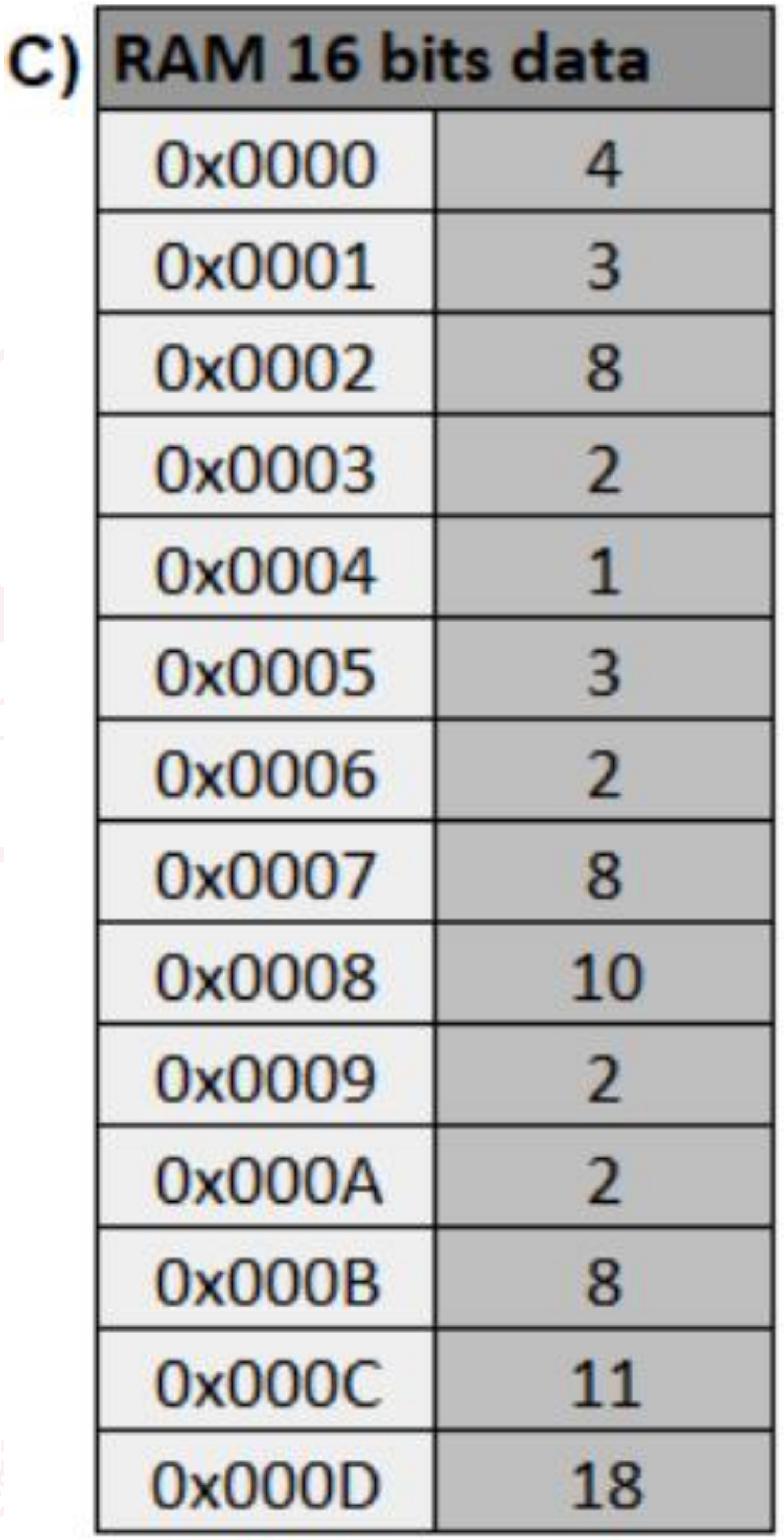
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

THE DATA IN THE MAIN MEMORY ARE PROVIDED



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

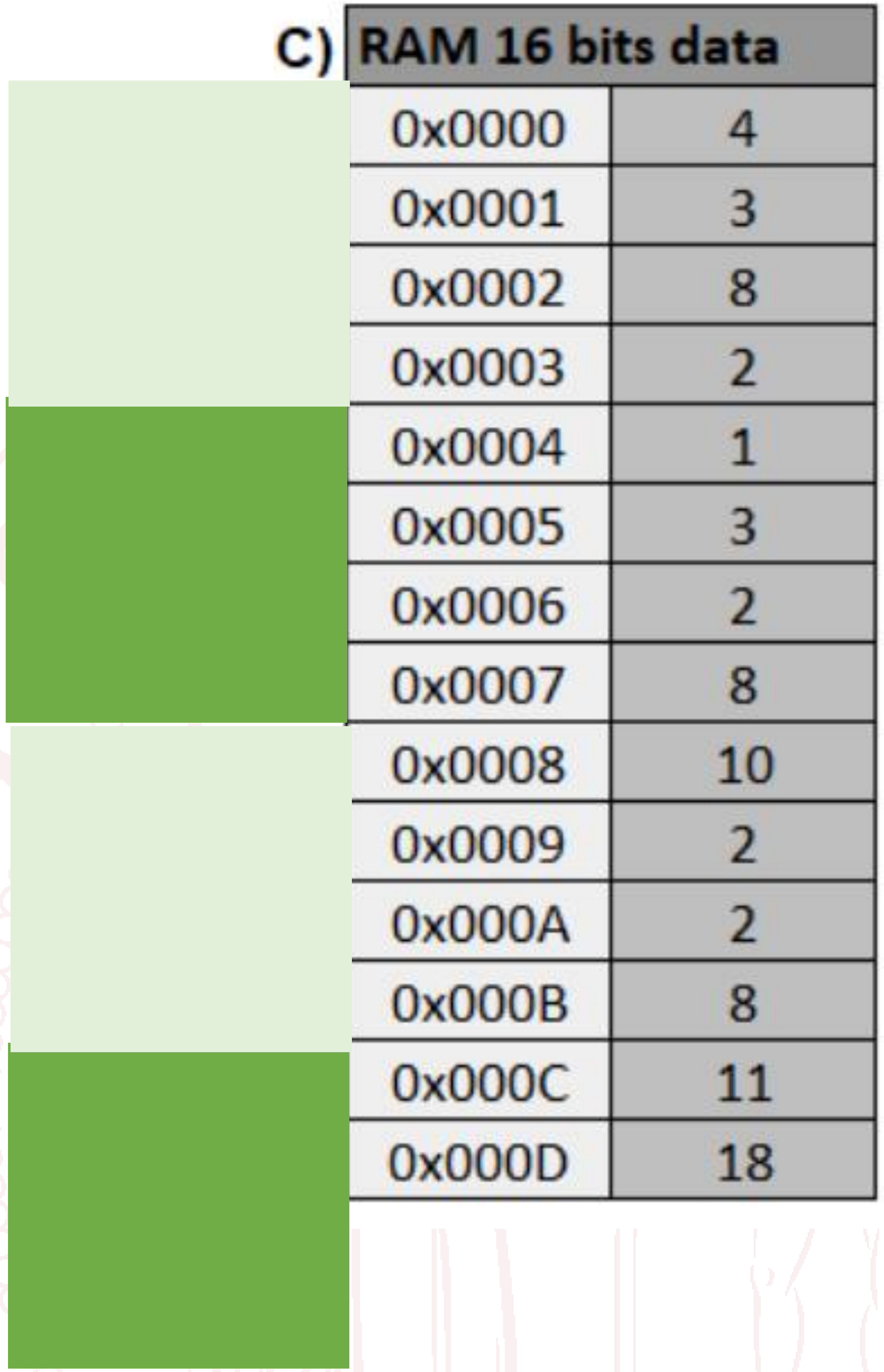
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

The cache described above is composed of **2 way**.
Each line contains **4 words**, we can divide the memory into the following **blocks**:



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

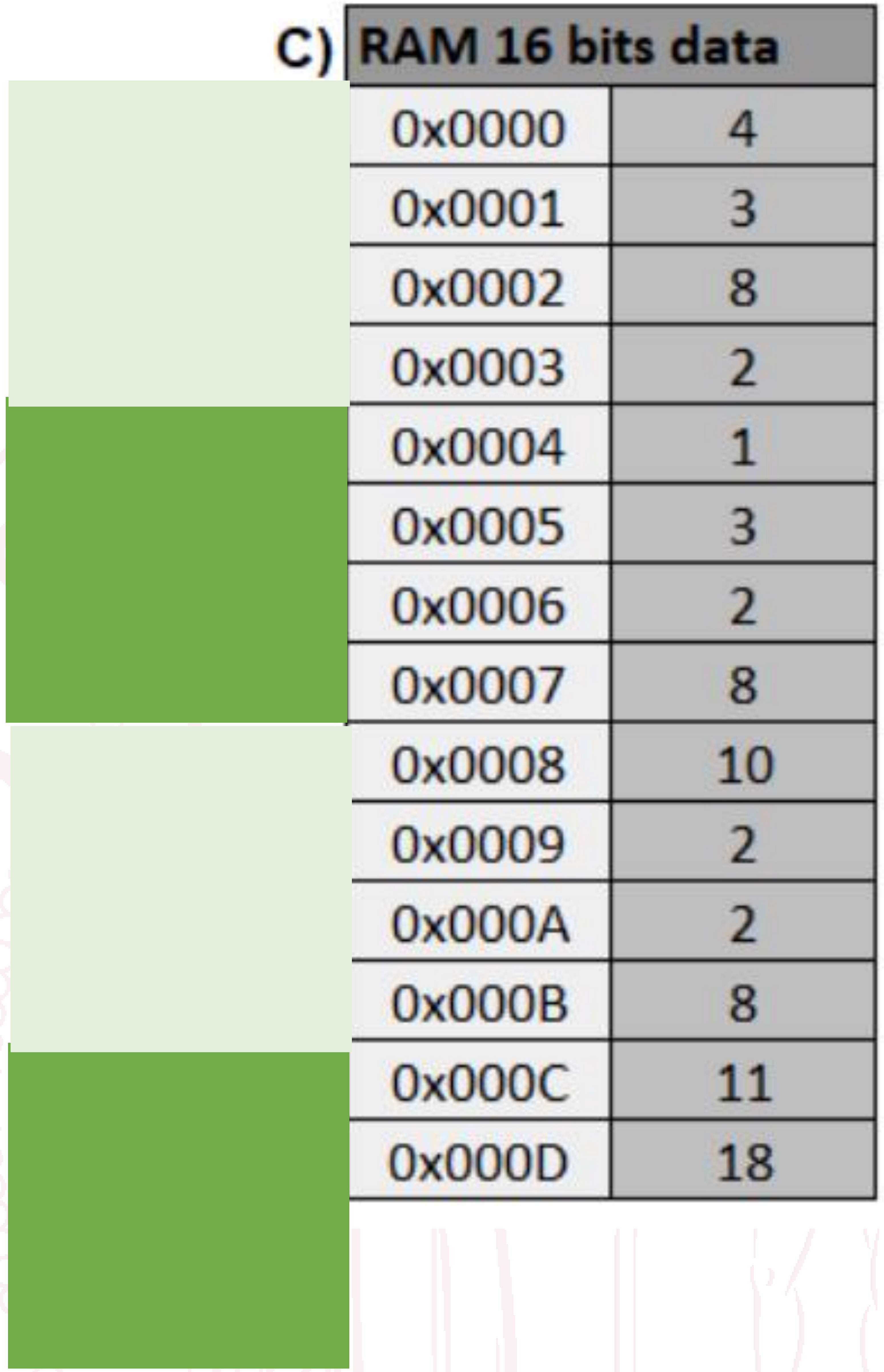
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

THE DATA IN M[3] HAS TO BE READ



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

Example from previous exam

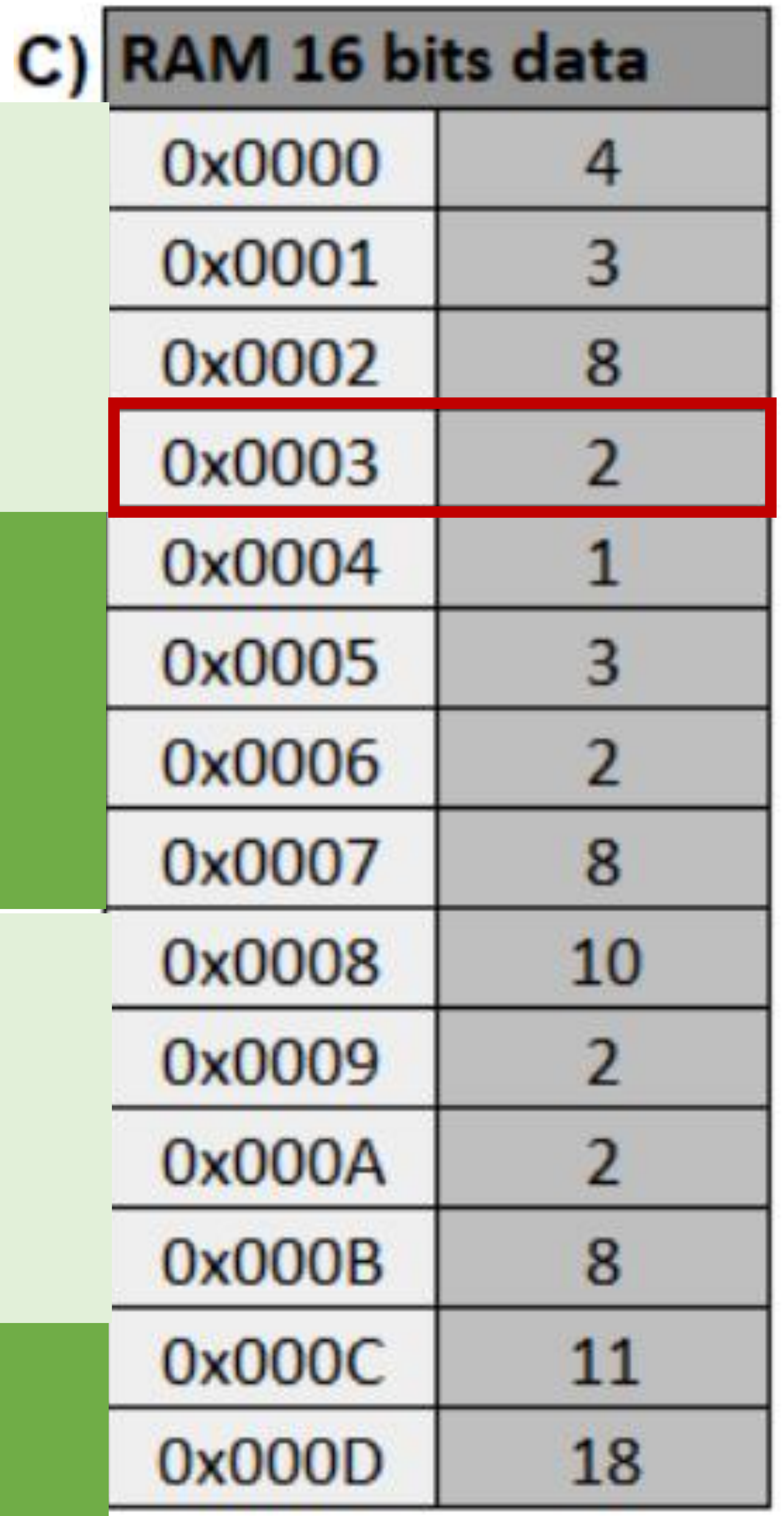
Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

The cache is empty, we have a **MISS, the corresponding block is copied in the cache**

🟩	4	3	8	2
🟩				



C) RAM 16 bits data

🟩	0x0000	4
	0x0001	3
	0x0002	8
	0x0003	2
🟩	0x0004	1
	0x0005	3
	0x0006	2
	0x0007	8
🟩	0x0008	10
	0x0009	2
	0x000A	2
	0x000B	8
🟩	0x000C	11
	0x000D	18

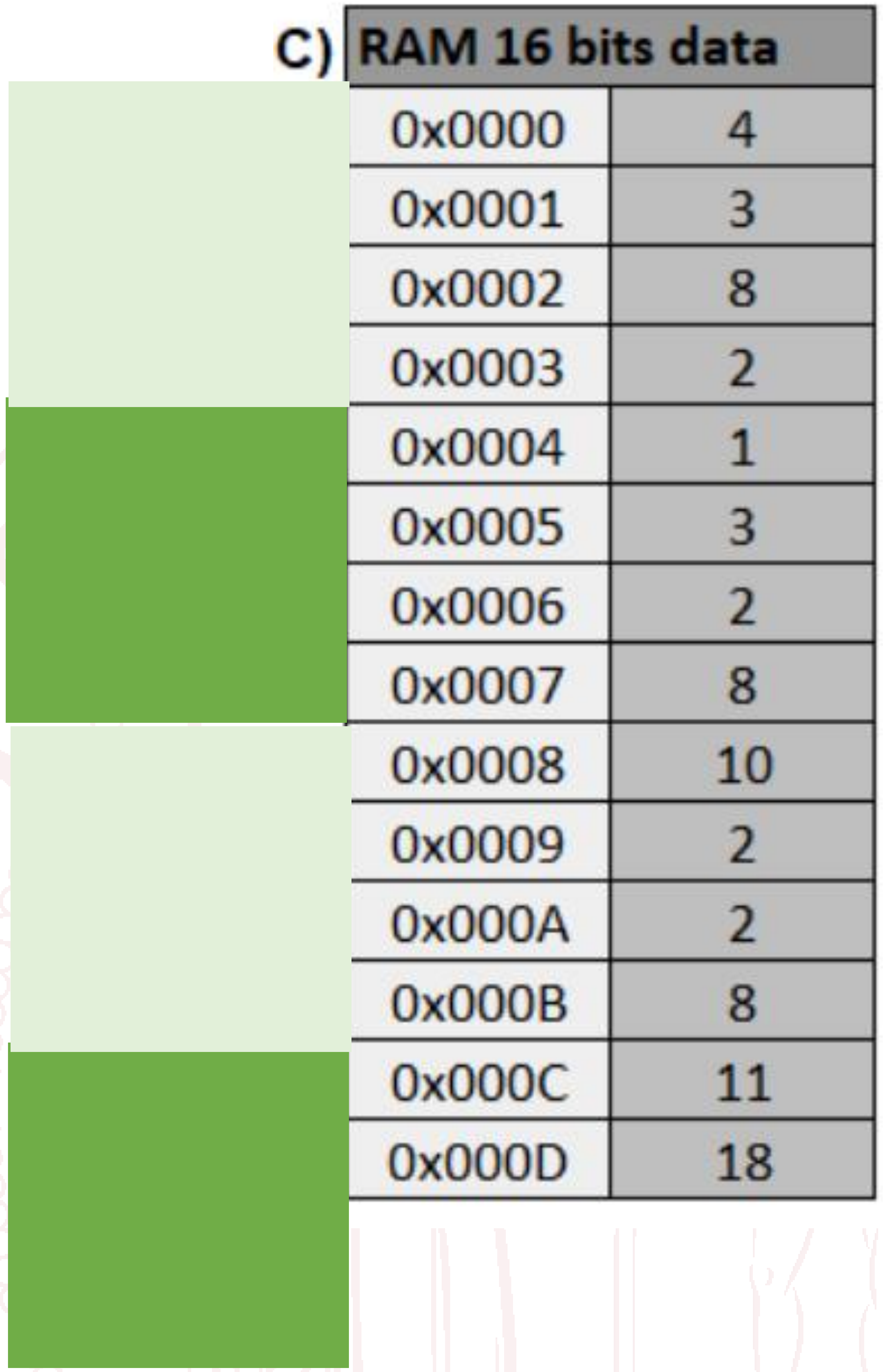
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

THE DATA IN M[1] HAS TO BE READ



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

Example from previous exam

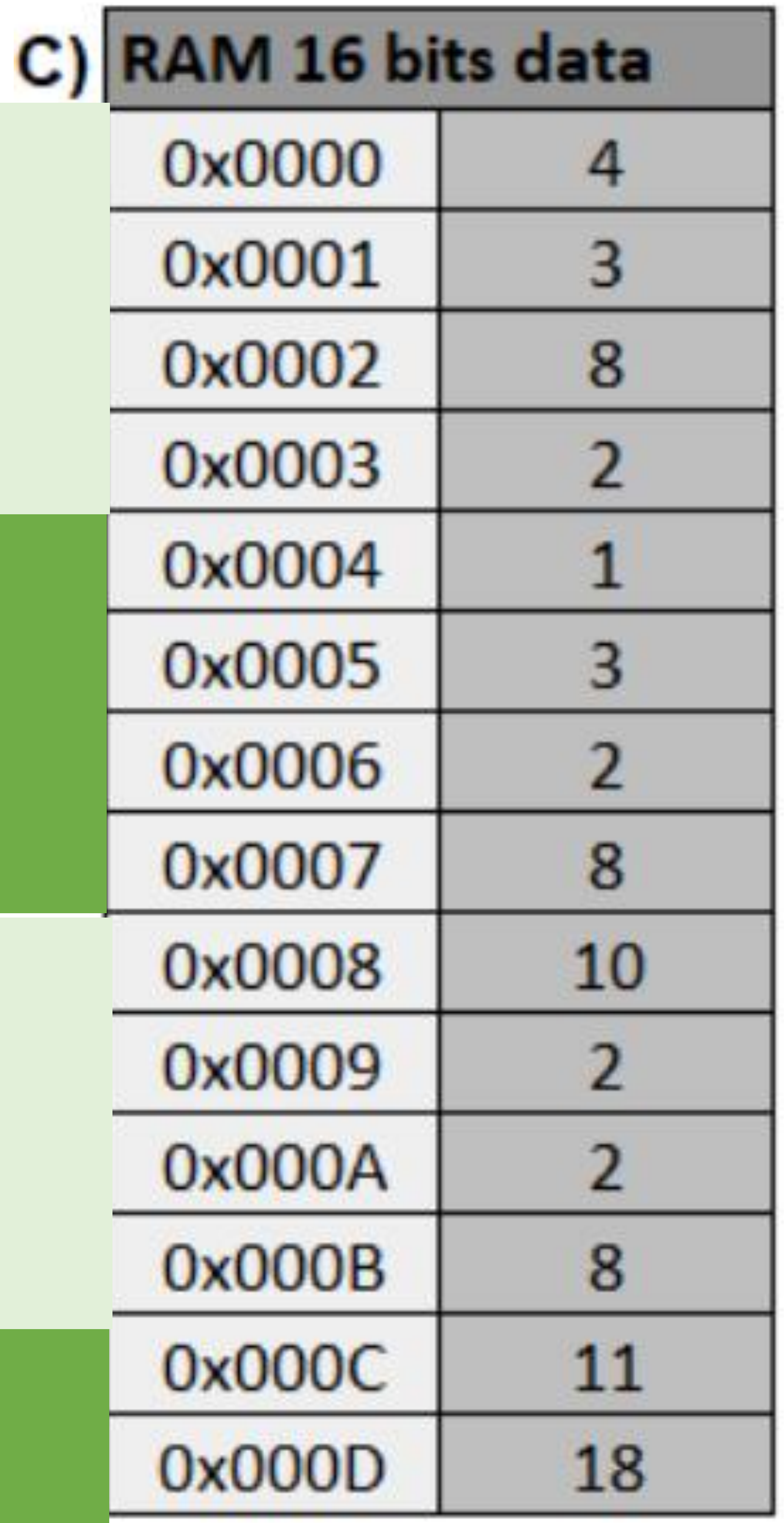
Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

The cache contains the data we need, we have a **HIT**

🟩	4	3	8	2
🟩				
🟩				
🟩				



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

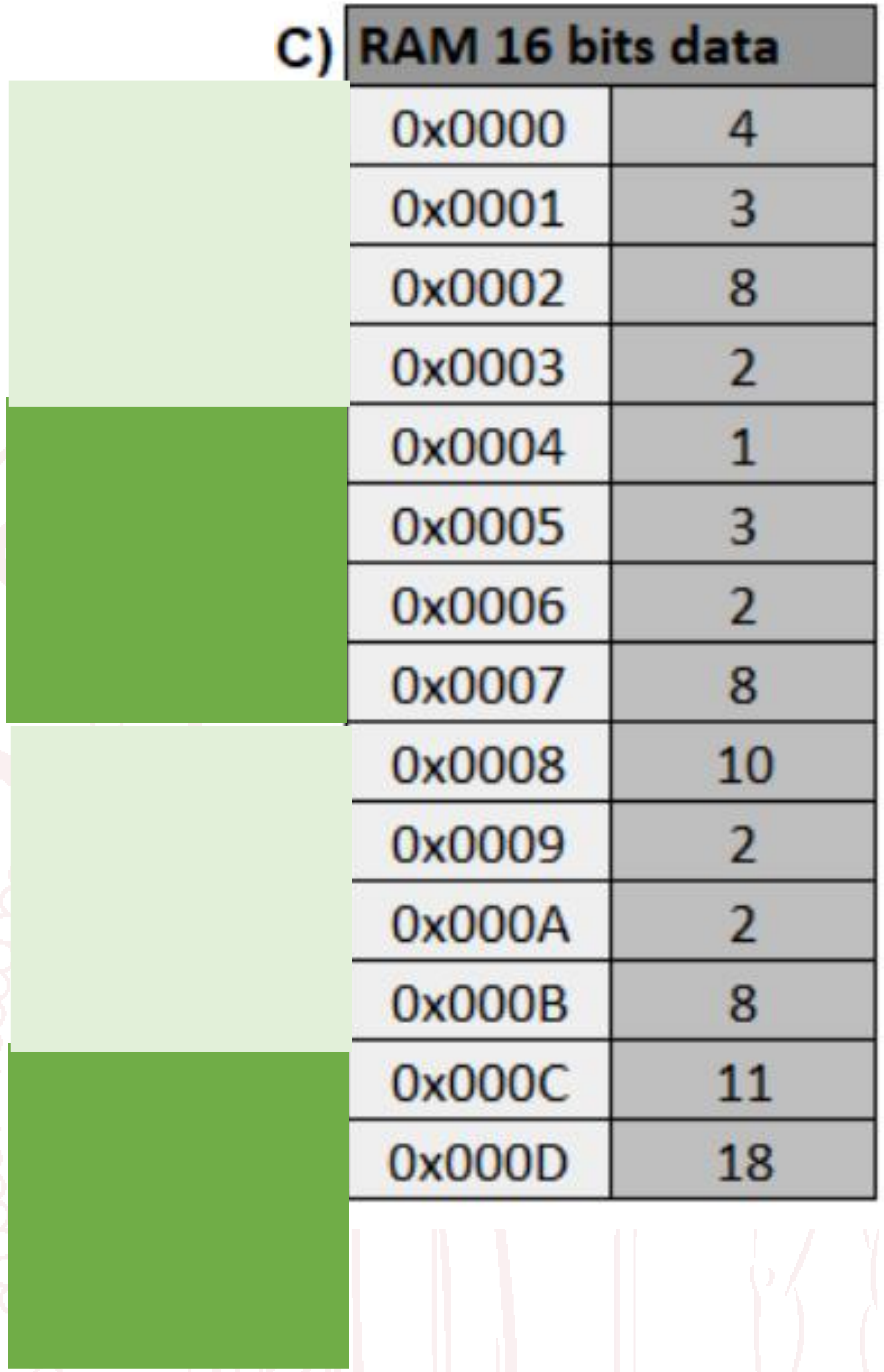
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

THE DATA IN M[4] HAS TO BE READ



C)

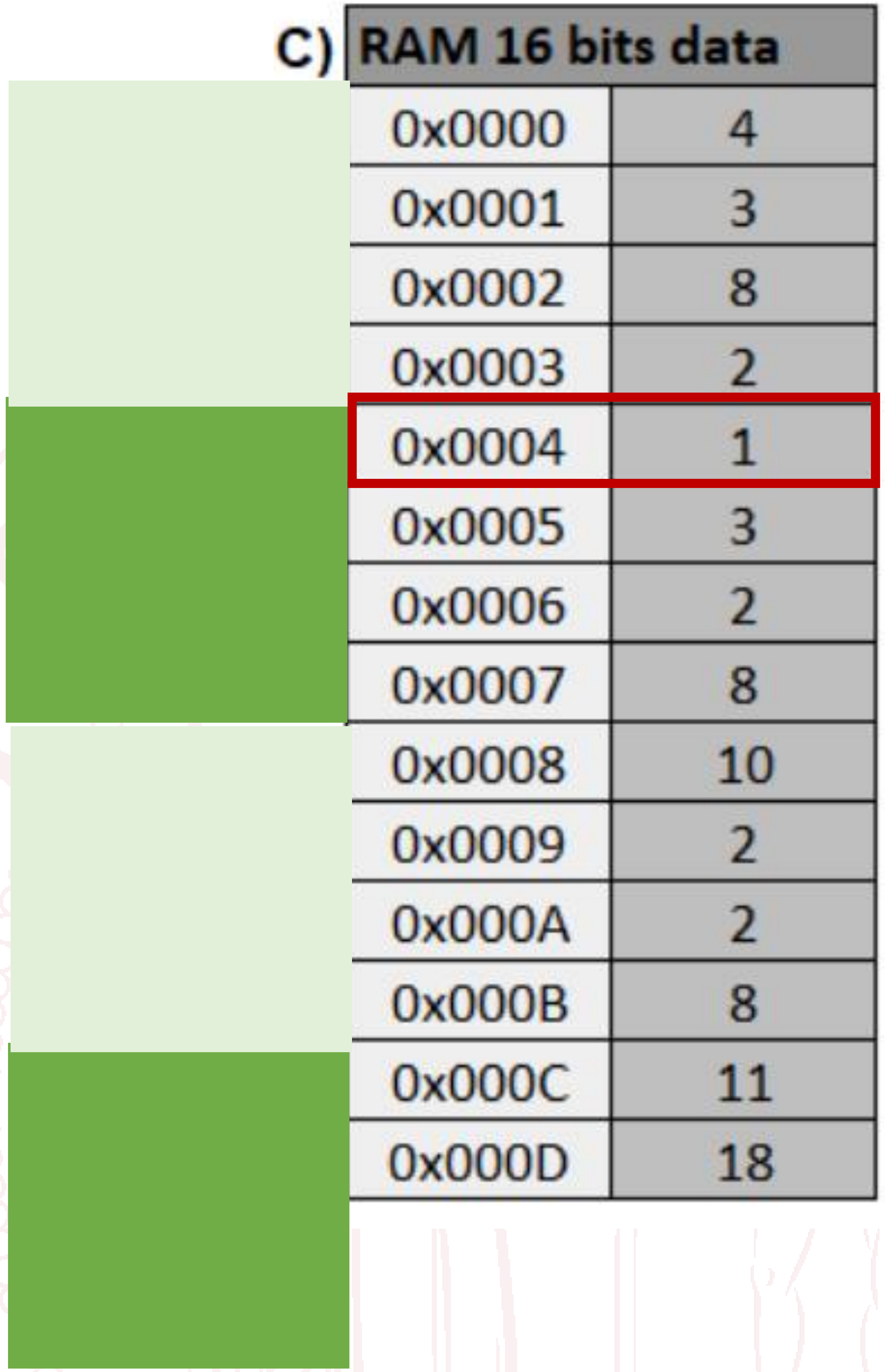
RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

The data is not in the cache, we have a **MISS, the corresponding block is copied in the cache**

4	3	8	2
1	3	2	8

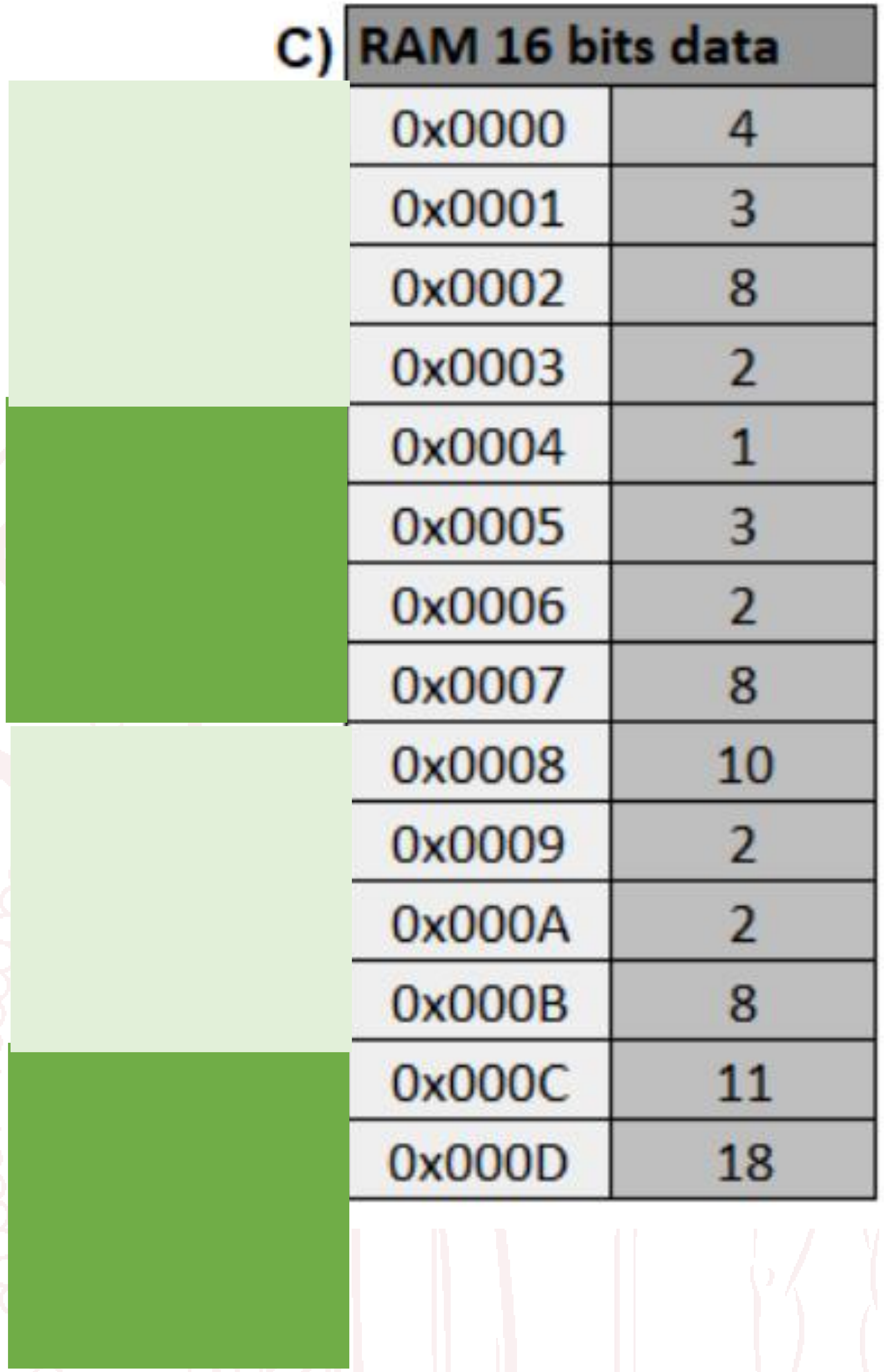
Example from previous exam

Description:

A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

THE DATA IN M[9] HAS TO BE READ



C)

RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

Example from previous exam

Description:

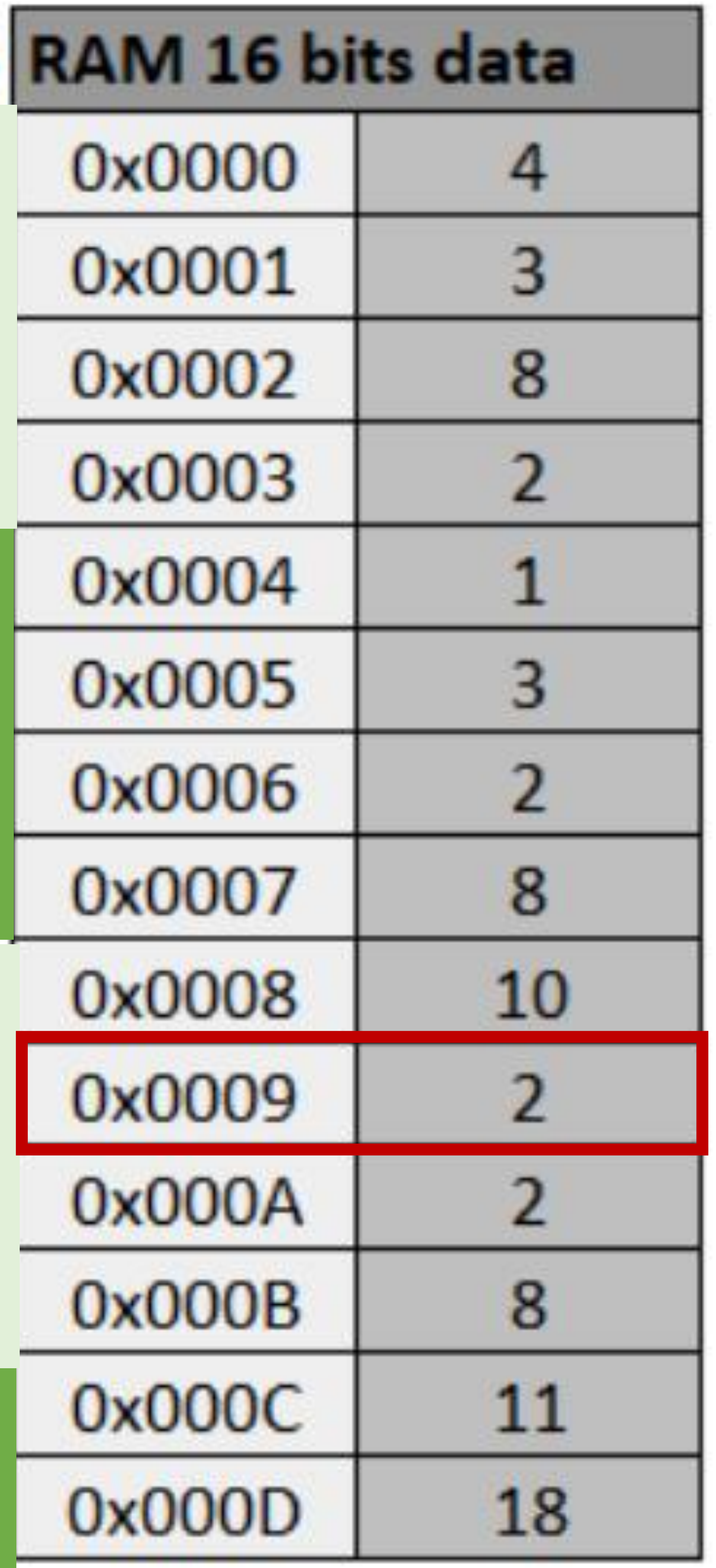
A single-cycle computer has the following specifications:

- Clock equal to 2 MHz;
- 16 bits address lines;
- A 16 KB SDRAM for instructions;
- A 32 KB SDRAM for data;
- Memories are addressed via 16-bit words;
- There is a **2-way set-associative with 4-words per line** connected only to the data memory.

The data is not in the cache, we have a **MISS, the corresponding block is copied in the cache**

4	3	8	2
10	2	2	2
1	3	2	8

C)



RAM 16 bits data	
0x0000	4
0x0001	3
0x0002	8
0x0003	2
0x0004	1
0x0005	3
0x0006	2
0x0007	8
0x0008	10
0x0009	2
0x000A	2
0x000B	8
0x000C	11
0x000D	18

Valid bit

- Let’s assume that the cache is empty, the values inside it are undefined
- After the request from the CPU, a **Cache Hit** can be happened
- However in this case, the word provided to the CPU would not be valid
- Thus, a **valid bit** is used to specify if the LINE of the cache is valid
- The **valid bit** is read together with the TAG

Writing methods in the cache

- So far we have considered the words in the cache as copies of those in main memory used to speed up the reading access.
- However, the matter changes if we also want to write a word that obviously cannot be written only in the cache

- **Write-through method:**
  - The result is always written to both the cache and the main memory
  - Symmetry between cache and central memory
  - It can slow down operations

- **Copy-back method:**
  - The result is written in the cache in case of Cache Hit
  - The result is written in the memory in case of Cache Miss
  - No symmetry between cache ←→ memory, some locations in the cache may no be longer valid

Dirty bit to control the writing

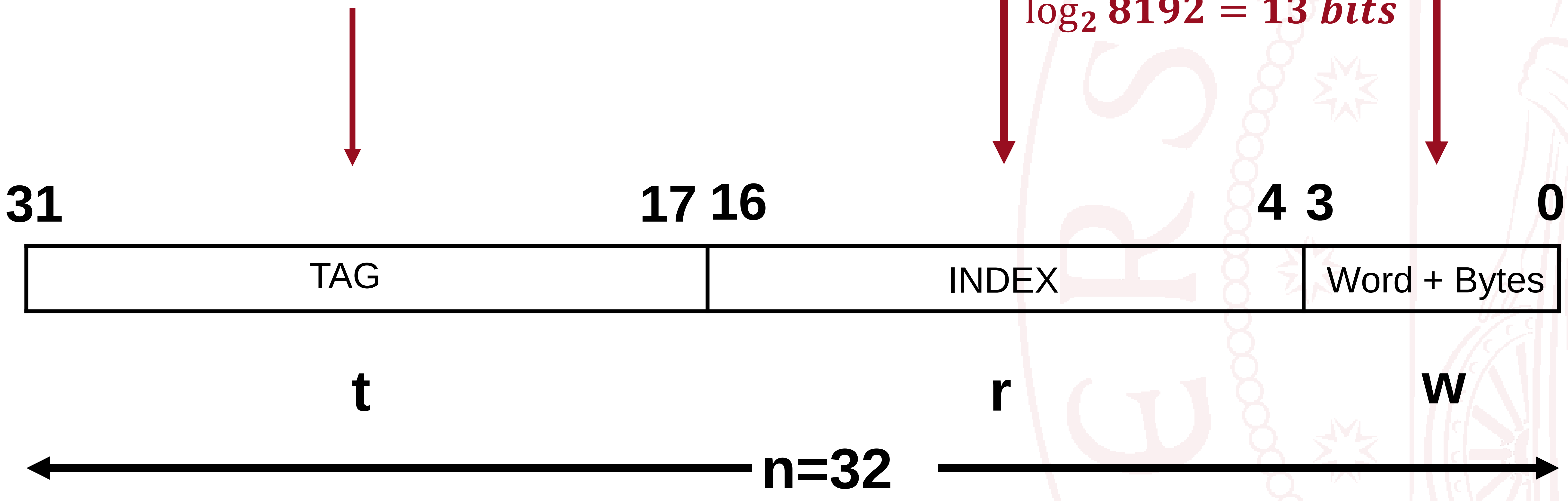
- In the case of the copy-back method, it is neccesary to keep track whether an item has been written or not in the cache
- Why?
  - To write in the memory only when it is necessary and not every time there is a Cache Miss
- A control bit, called **dirty bit,** is used:
  - If the dirty bit is equal to 1, it means that the LINE in cache has been written (and must also be written in memory)
  - If the dirty bit is equal to 0, it means you don’t need to write in the memory

Example of 2-way set-associative with 4-word lines: write-through method

- A 256 KB cache memory, 4 words for the lines and the address is composed of 32 bits
- $N_{bytes} = 256 \cdot 1024 = 262144\ bytes$
  - $N_{bytes \times word} = 4\ bytes$ → each word contains 4 bytes
- $N_{words} = \dfrac{N_{bytes}}{N_{bytes \times word}} = \dfrac{262144}{4} = 65536\ words$
- $N_{bytes \times line} = 16 \rightarrow N_{words \times line} = 4$
- $N_{lines} = \dfrac{N_{words}}{N_{words \times line}} = \dfrac{65536}{4} = 16384$
- $N_{ways} = 2 \rightarrow N_{lines \times way} = 8192$
- $n - w - r = t = 32 - 4 - 13 = 15$

To identify the16 bytes, we need:
$\log_2 16 = 4\ bits$

To identify the 8192 lines, we need:
$\log_2 8192 = 13\ bits$



Disclaimer

Figures from *Logic and Computer Design Fundamentals*, Fifth Edition, GE Mano | Kime | Martin

Questions