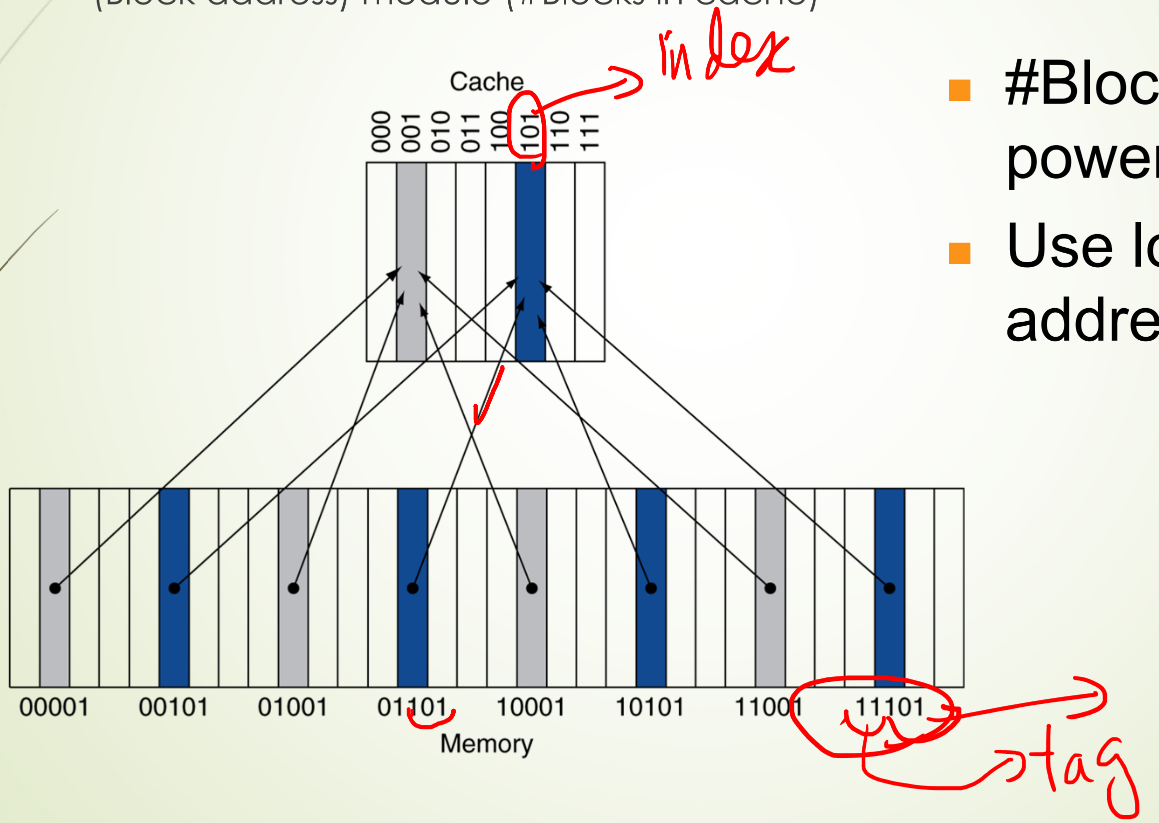
EGC442 Class Notes 4/21/2023

Babak Izadi
Division of Engineering Programs
bai@engr.newpaltz.edu

				Test 1	Test 2
Average				82.5	88.5
Median				85.5	89.0
MAX				92.0	95.0
Minimum				67.0	80.0

Direct Mapped Cache

- Location determined by address
- Direct mapped: only one choice
  - (Block address) modulo (#Blocks in cache)



- #Blocks is a power of 2
- Use low-order address bits

Address Subdivision

Cach block = 32 bits

4 bytes

1K x 4 bytes

4K cache bytes

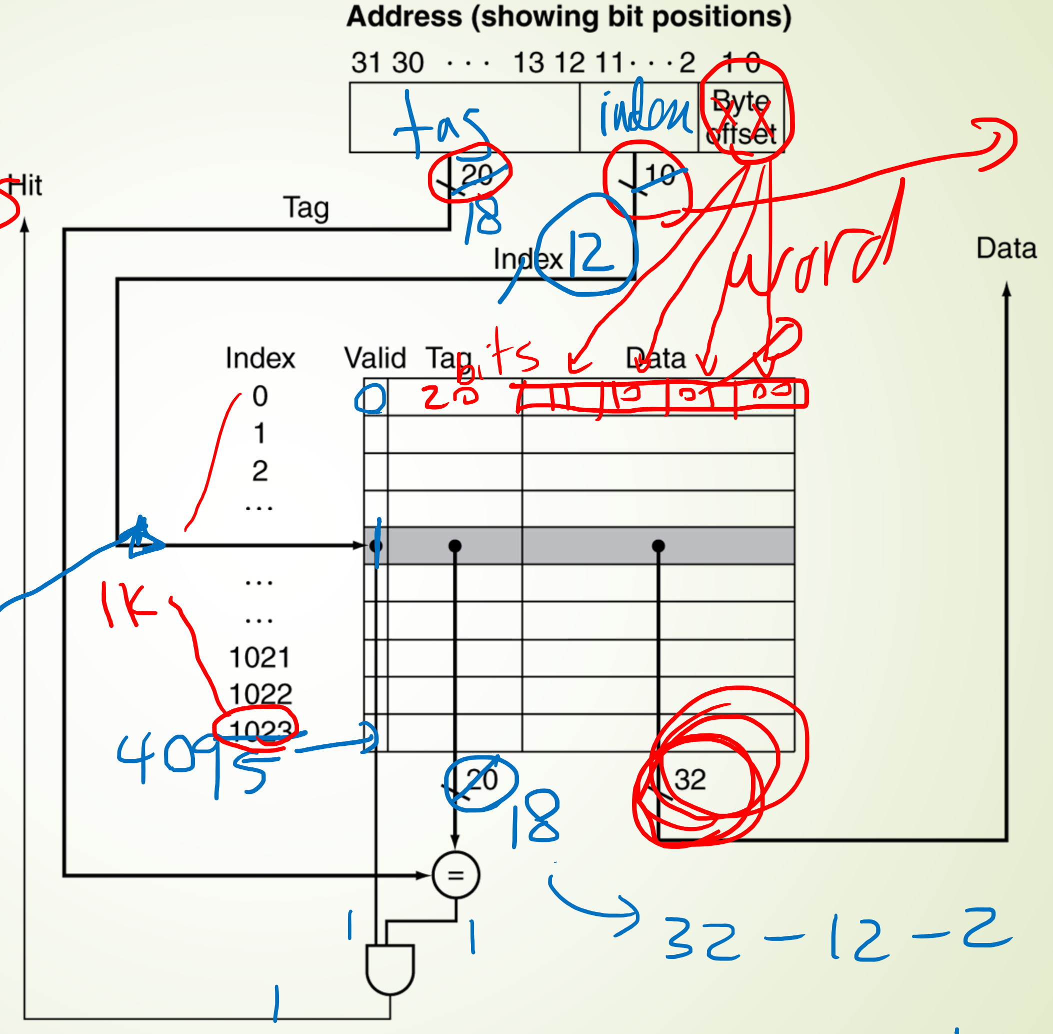
2^{12}

4Kx4

Cache 16K bytes

4K rows x 4 bytes

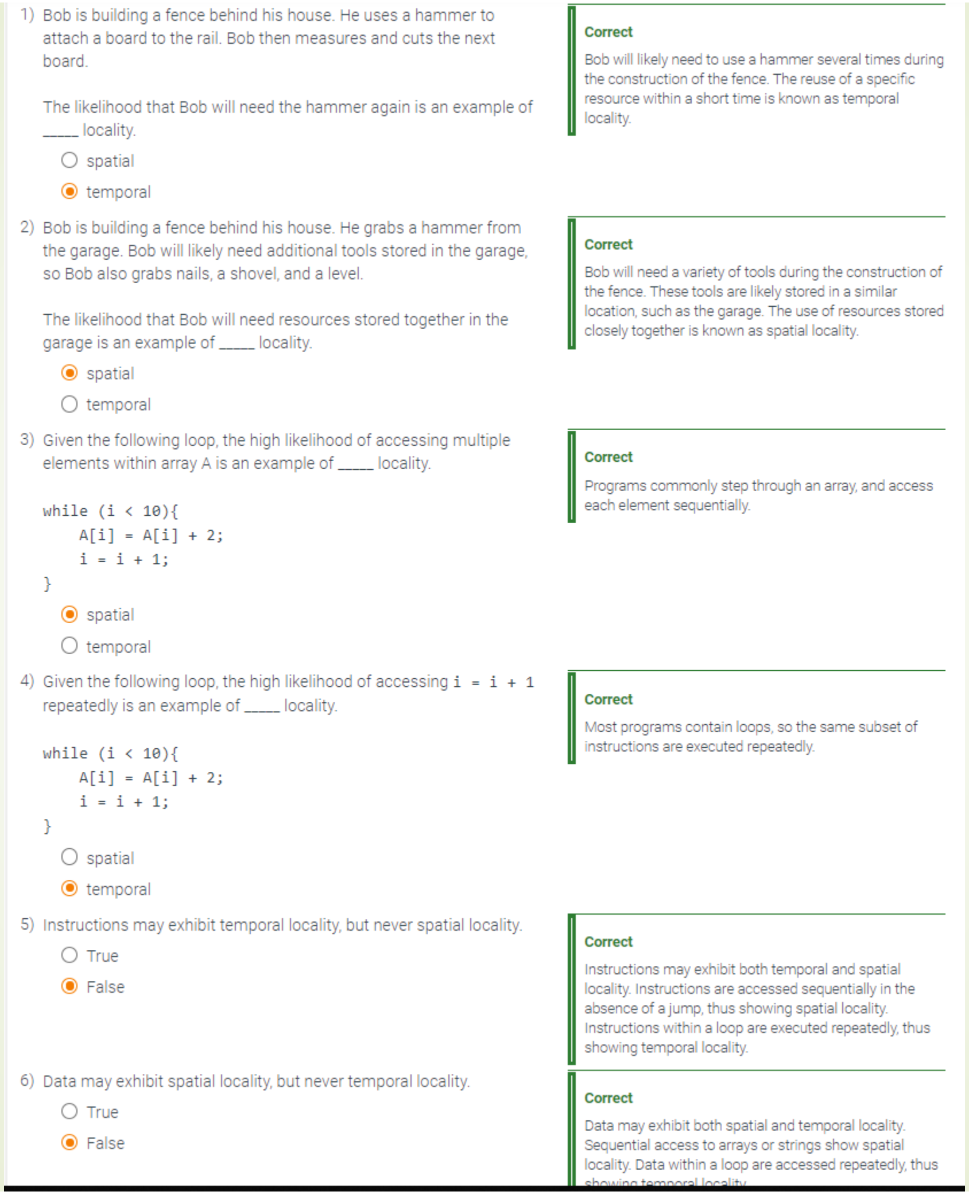
$4096 = 2^{12}$



$\overset{10}{2} = 1024$ rows of cache

$32 - 12 - 2$

What kind of locality are we taking advantage of? temp.



1) Bob is building a fence behind his house. He uses a hammer to attach a board to the rail. Bob then measures and cuts the next board.

The likelihood that Bob will need the hammer again is an example of _____ locality.

○ spatial

● temporal

Correct

Bob will likely need to use a hammer several times during the construction of the fence. The reuse of a specific resource within a short time is known as temporal locality.

2) Bob is building a fence behind his house. He grabs a hammer from the garage. Bob will likely need additional tools stored in the garage, so Bob also grabs nails, a shovel, and a level.

The likelihood that Bob will need resources stored together in the garage is an example of _____ locality.

● spatial

○ temporal

Correct

Bob will need a variety of tools during the construction of the fence. These tools are likely stored in a similar location, such as the garage. The use of resources stored closely together is known as spatial locality.

3) Given the following loop, the high likelihood of accessing multiple elements within array A is an example of _____ locality.

```
while (i < 10){
    A[i] = A[i] + 2;
    i = i + 1;
}
```

● spatial

○ temporal

Correct

Programs commonly step through an array, and access each element sequentially.

4) Given the following loop, the high likelihood of accessing `i = i + 1` repeatedly is an example of _____ locality.

```
while (i < 10){
    A[i] = A[i] + 2;
    i = i + 1;
}
```

○ spatial

● temporal

Correct

Most programs contain loops, so the same subset of instructions are executed repeatedly.

5) Instructions may exhibit temporal locality, but never spatial locality.

○ True

● False

Correct

Instructions may exhibit both temporal and spatial locality. Instructions are accessed sequentially in the absence of a jump, thus showing spatial locality. Instructions within a loop are executed repeatedly, thus showing temporal locality.

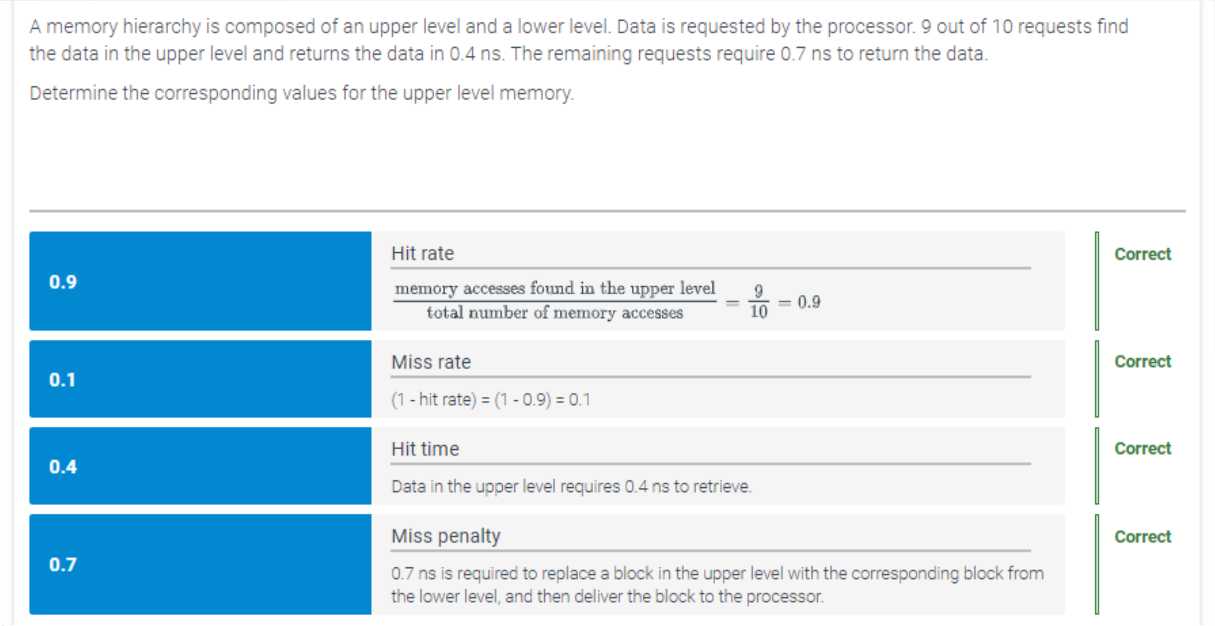
6) Data may exhibit spatial locality, but never temporal locality.

○ True

● False

Correct

Data may exhibit both spatial and temporal locality. Sequential access to arrays or strings show spatial locality. Data within a loop are accessed repeatedly, thus showing temporal locality.



A memory hierarchy is composed of an upper level and a lower level. Data is requested by the processor. 9 out of 10 requests find the data in the upper level and returns the data in 0.4 ns. The remaining requests require 0.7 ns to return the data.

Determine the corresponding values for the upper level memory.

0.9

Hit rate

$$\frac{\text{memory accesses found in the upper level}}{\text{total number of memory accesses}} = \frac{9}{10} = 0.9$$

Correct

0.1

Miss rate

(1 - hit rate) = (1 - 0.9) = 0.1

Correct

0.4

Hit time

Data in the upper level requires 0.4 ns to retrieve.

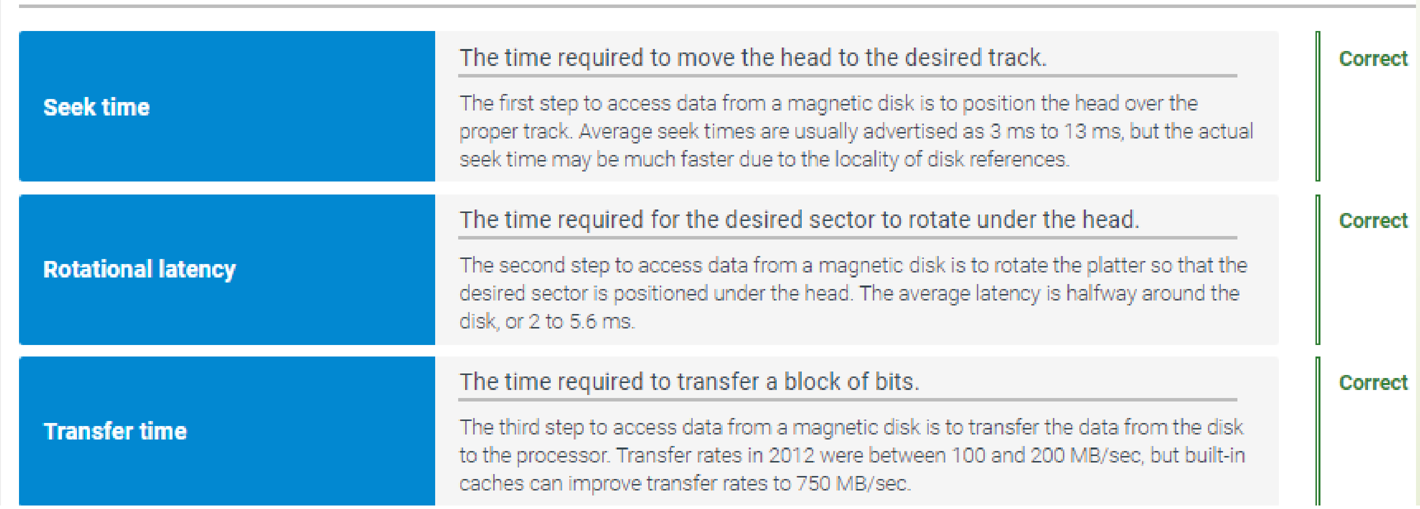
Correct

0.7

Miss penalty

0.7 ns is required to replace a block in the upper level with the corresponding block from the lower level, and then deliver the block to the processor.

Correct



Seek time

The time required to move the head to the desired track.

The first step to access data from a magnetic disk is to position the head over the proper track. Average seek times are usually advertised as 3 ms to 13 ms, but the actual seek time may be much faster due to the locality of disk references.

Correct

Rotational latency

The time required for the desired sector to rotate under the head.

The second step to access data from a magnetic disk is to rotate the platter so that the desired sector is positioned under the head. The average latency is halfway around the disk, or 2 to 5.6 ms.

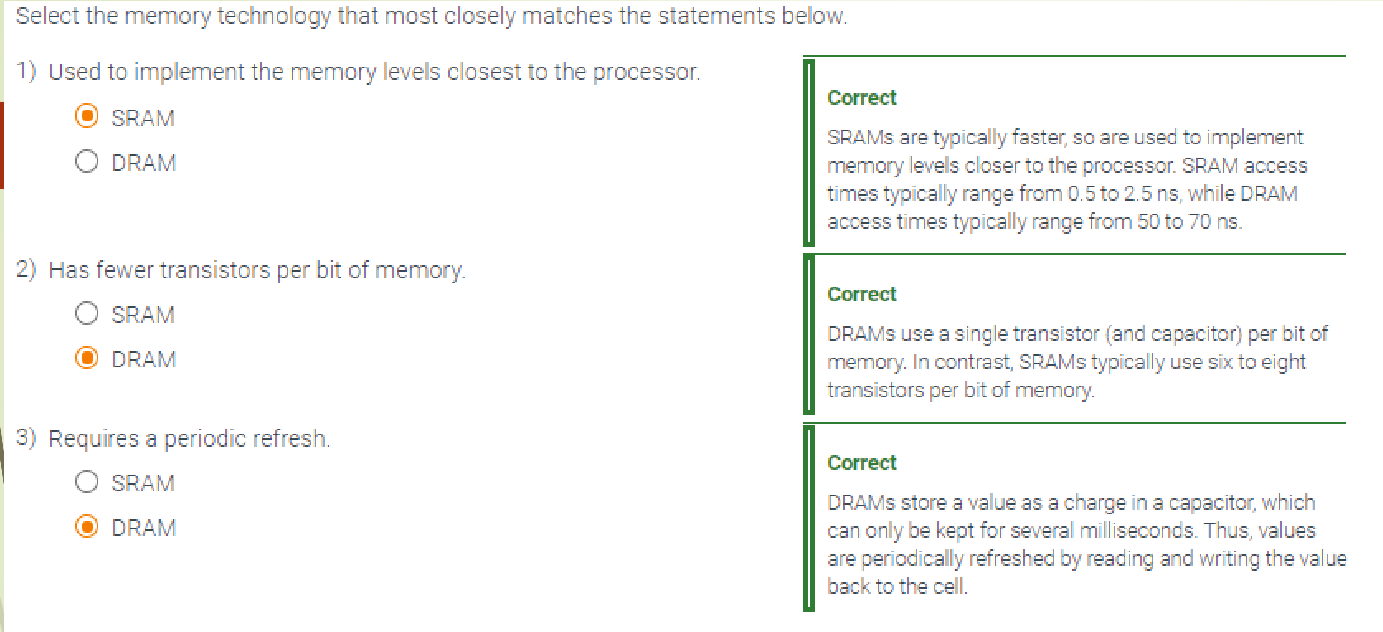
Correct

Transfer time

The time required to transfer a block of bits.

The third step to access data from a magnetic disk is to transfer the data from the disk to the processor. Transfer rates in 2012 were between 100 and 200 MB/sec, but built-in caches can improve transfer rates to 750 MB/sec.

Correct



Select the memory technology that most closely matches the statements below.

1) Used to implement the memory levels closest to the processor.

● SRAM
○ DRAM

Correct

SRAMs are typically faster, so are used to implement memory levels closer to the processor. SRAM access times typically range from 0.5 to 2.5 ns, while DRAM access times typically range from 50 to 70 ns.

2) Has fewer transistors per bit of memory.

○ SRAM
● DRAM

Correct

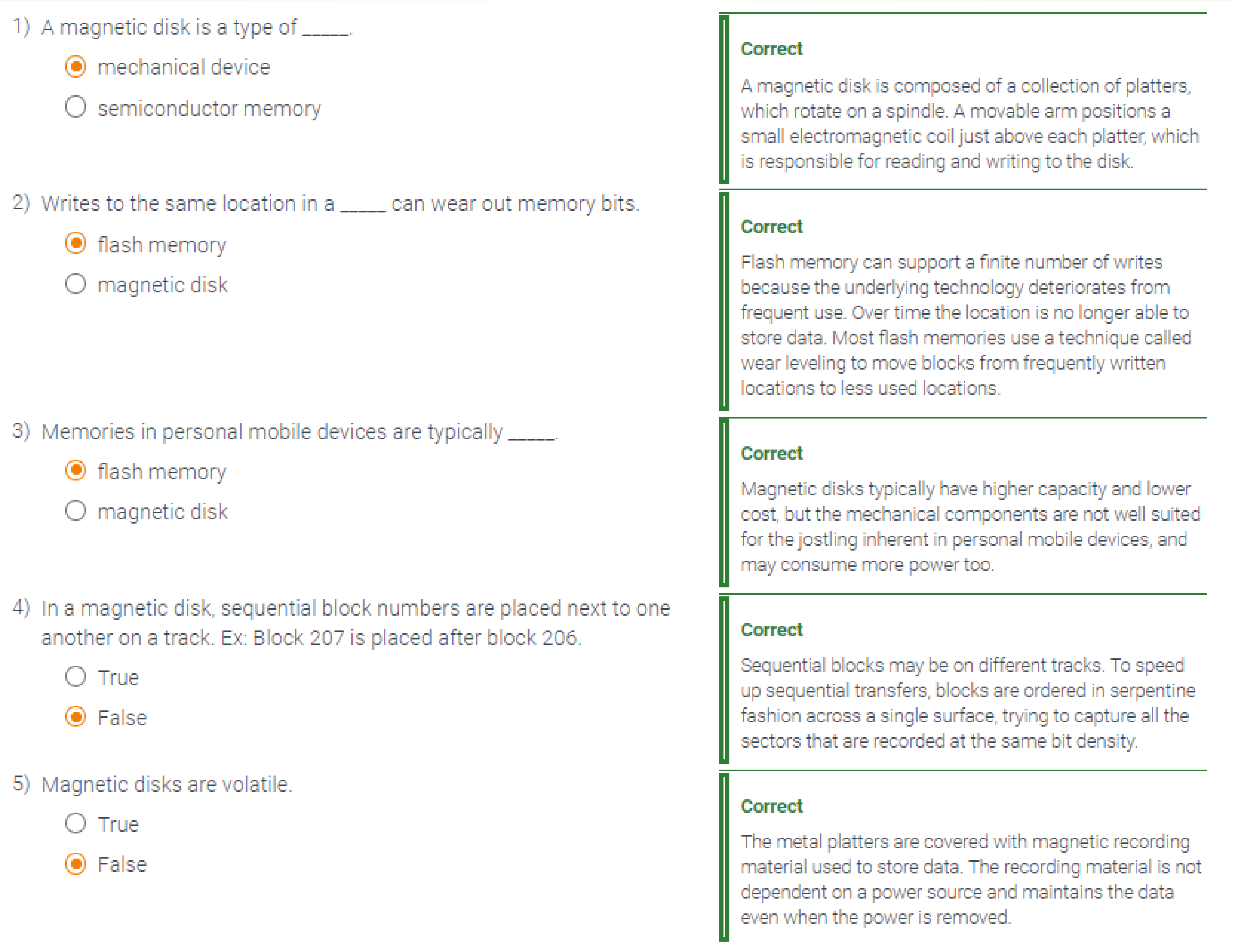
DRAMs use a single transistor (and capacitor) per bit of memory. In contrast, SRAMs typically use six to eight transistors per bit of memory.

3) Requires a periodic refresh.

○ SRAM
● DRAM

Correct

DRAMs store a value as a charge in a capacitor, which can only be kept for several milliseconds. Thus, values are periodically refreshed by reading and writing the value back to the cell.



1) A magnetic disk is a type of _____.

● mechanical device

○ semiconductor memory

Correct

A magnetic disk is composed of a collection of platters, which rotate on a spindle. A movable arm positions a small electromagnetic coil just above each platter, which is responsible for reading and writing to the disk.

2) Writes to the same location in a _____ can wear out memory bits.

● flash memory

○ magnetic disk

Correct

Flash memory can support a finite number of writes because the underlying technology deteriorates from frequent use. Over time the location is no longer able to store data. Most flash memories use a technique called wear leveling to move blocks from frequently written locations to less used locations.

3) Memories in personal mobile devices are typically _____.

● flash memory

○ magnetic disk

Correct

Magnetic disks typically have higher capacity and lower cost, but the mechanical components are not well suited for the jostling inherent in personal mobile devices, and may consume more power too.

4) In a magnetic disk, sequential block numbers are placed next to one another on a track. Ex: Block 207 is placed after block 206.

○ True

● False

Correct

Sequential blocks may be on different tracks. To speed up sequential transfers, blocks are ordered in serpentine fashion across a single surface, trying to capture all the sectors that are recorded at the same bit density.

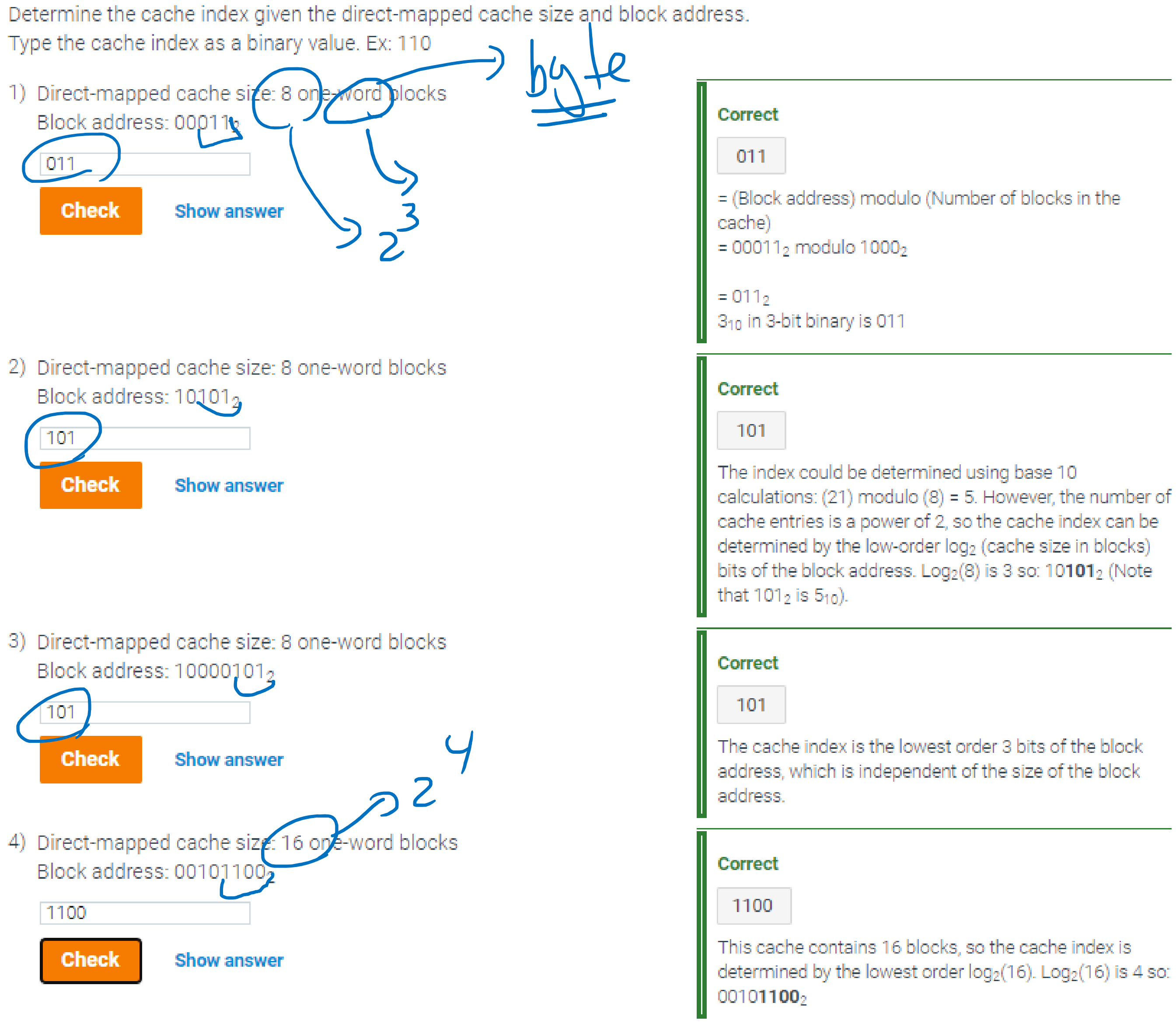
5) Magnetic disks are volatile.

○ True

● False

Correct

The metal platters are covered with magnetic recording material used to store data. The recording material is not dependent on a power source and maintains the data even when the power is removed.



Determine the cache index given the direct-mapped cache size and block address.
Type the cache index as a binary value. Ex: 110

1) Direct-mapped cache size: 8 one-word blocks → byte
Block address: 00011_2

011

2^3

Correct

011

= (Block address) modulo (Number of blocks in the cache)
= 00011_2 modulo 1000_2

= 011_2
3_{10} in 3-bit binary is 011

2) Direct-mapped cache size: 8 one-word blocks
Block address: 10101_2

101

Correct

101

The index could be determined using base 10 calculations: (21) modulo (8) = 5. However, the number of cache entries is a power of 2, so the cache index can be determined by the low-order $\log_2$ (cache size in blocks) bits of the block address. $\text{Log}_2(8)$ is 3 so: $10\mathbf{101}_2$ (Note that 101_2 is 5_{10}).

3) Direct-mapped cache size: 8 one-word blocks
Block address: 10000101_2

101

Correct

101

The cache index is the lowest order 3 bits of the block address, which is independent of the size of the block address.

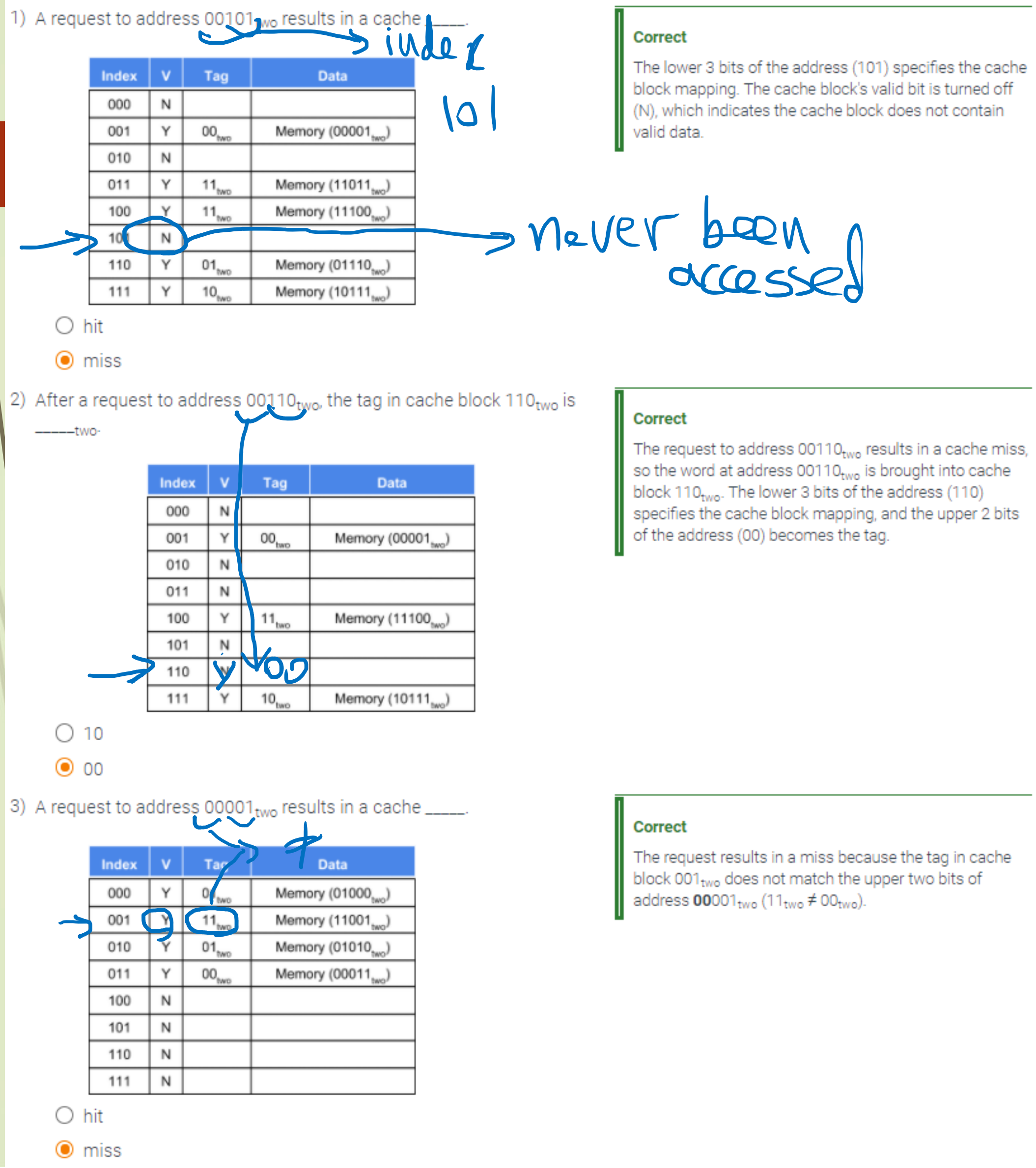
4) Direct-mapped cache size: 16 one-word blocks → 2^4
Block address: 00101100_2

1100

Correct

1100

This cache contains 16 blocks, so the cache index is determined by the lowest order $\log_2(16)$. $\text{Log}_2(16)$ is 4 so: $0010\mathbf{1100}_2$



1) A request to address 00101_{two} results in a cache _____.

index

101

Index	V	Tag	Data
000	N		
001	Y	00_{two}	Memory (00001_{two})
010	N		
011	Y	11_{two}	Memory (11011_{two})
100	Y	11_{two}	Memory (11100_{two})
101	N		
110	Y	01_{two}	Memory (01110_{two})
111	Y	10_{two}	Memory (10111_{two})

never been accessed

○ hit

● miss

Correct

The lower 3 bits of the address (101) specifies the cache block mapping. The cache block's valid bit is turned off (N), which indicates the cache block does not contain valid data.

2) After a request to address 00110_{two}, the tag in cache block 110_{two} is _____two.

Index	V	Tag	Data
000	N		
001	Y	00_{two}	Memory (00001_{two})
010	N		
011	N		
100	Y	11_{two}	Memory (11100_{two})
101	N		
110	N y	00	
111	Y	10_{two}	Memory (10111_{two})

○ 10

● 00

Correct

The request to address 00110_{two} results in a cache miss, so the word at address 00110_{two} is brought into cache block 110_{two}. The lower 3 bits of the address (110) specifies the cache block mapping, and the upper 2 bits of the address (00) becomes the tag.

3) A request to address 00001_{two} results in a cache _____.

≠

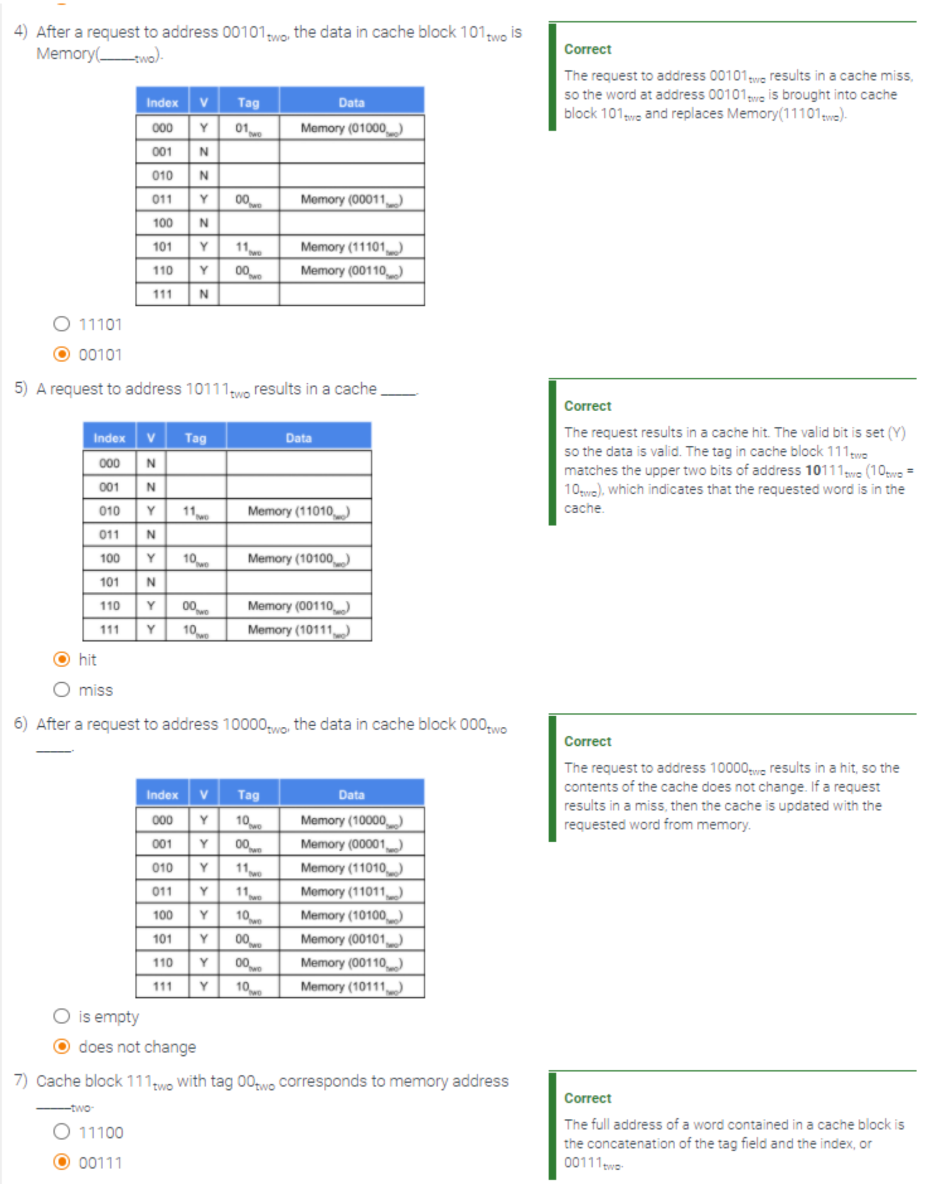
Index	V	Tag	Data
000	Y	01_{two}	Memory (01000_{two})
001	Y	11_{two}	Memory (11001_{two})
010	Y	01_{two}	Memory (01010_{two})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	N		
110	N		
111	N		

○ hit

● miss

Correct

The request results in a miss because the tag in cache block 001_{two} does not match the upper two bits of address **00**001_{two} ($11_{two} \neq 00_{two}$).



4) After a request to address 00101_{two}, the data in cache block 101_{two} is Memory(_____two).

Index	V	Tag	Data
000	Y	01_{two}	Memory (01000_{two})
001	N		
010	N		
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	Y	11_{two}	Memory (11101_{two})
110	Y	00_{two}	Memory (00110_{two})
111	N		

○ 11101

● 00101

Correct

The request to address 00101_{two} results in a cache miss, so the word at address 00101_{two} is brought into cache block 101_{two} and replaces Memory(11101_{two}).

5) A request to address 10111_{two} results in a cache _____.

Index	V	Tag	Data
000	N		
001	N		
010	Y	11_{two}	Memory (11010_{two})
011	N		
100	Y	10_{two}	Memory (10100_{two})
101	N		
110	Y	00_{two}	Memory (00110_{two})
111	Y	10_{two}	Memory (10111_{two})

● hit

○ miss

Correct

The request results in a cache hit. The valid bit is set (Y) so the data is valid. The tag in cache block 111_{two} matches the upper two bits of address **10**111_{two} (10_{two} = 10_{two}), which indicates that the requested word is in the cache.

6) After a request to address 10000_{two}, the data in cache block 000_{two} _____.

Index	V	Tag	Data
000	Y	10_{two}	Memory (10000_{two})
001	Y	00_{two}	Memory (00001_{two})
010	Y	11_{two}	Memory (11010_{two})
011	Y	11_{two}	Memory (11011_{two})
100	Y	10_{two}	Memory (10100_{two})
101	Y	00_{two}	Memory (00101_{two})
110	Y	00_{two}	Memory (00110_{two})
111	Y	10_{two}	Memory (10111_{two})

○ is empty

● does not change

Correct

The request to address 10000_{two} results in a hit, so the contents of the cache does not change. If a request results in a miss, then the cache is updated with the requested word from memory.

7) Cache block 111_{two} with tag 00_{two} corresponds to memory address _____two.

○ 11100

● 00111

Correct

The full address of a word contained in a cache block is the concatenation of the tag field and the index, or 00111_{two}.

$11_{10} \rightarrow$

16 8 4 2 1

0 1 | 0 1 1

tag | index

3 2 1 0

Index	V	Tag	Data
000	Y	01_{two}	Memory (01000_{two})
001	N		
010	~~N~~ Y	01	Memory(10_{10})
011	Y	00_{two}	Memory (00011_{two})
100	N		
101	Y	11_{two}	Memory (11101_{two})
110	Y	00_{two}	Memory (00110_{two})
111	N		

10_{10} 0 1 | 0 1 0

Mem[11]$_{10}$ → Miss

Mem[10_{10}] =

27) Design a direct-mapped cache with the following parameters:

- Address size: 32 bits
- Cache data size: 2 KB
- Cache block: 1 word

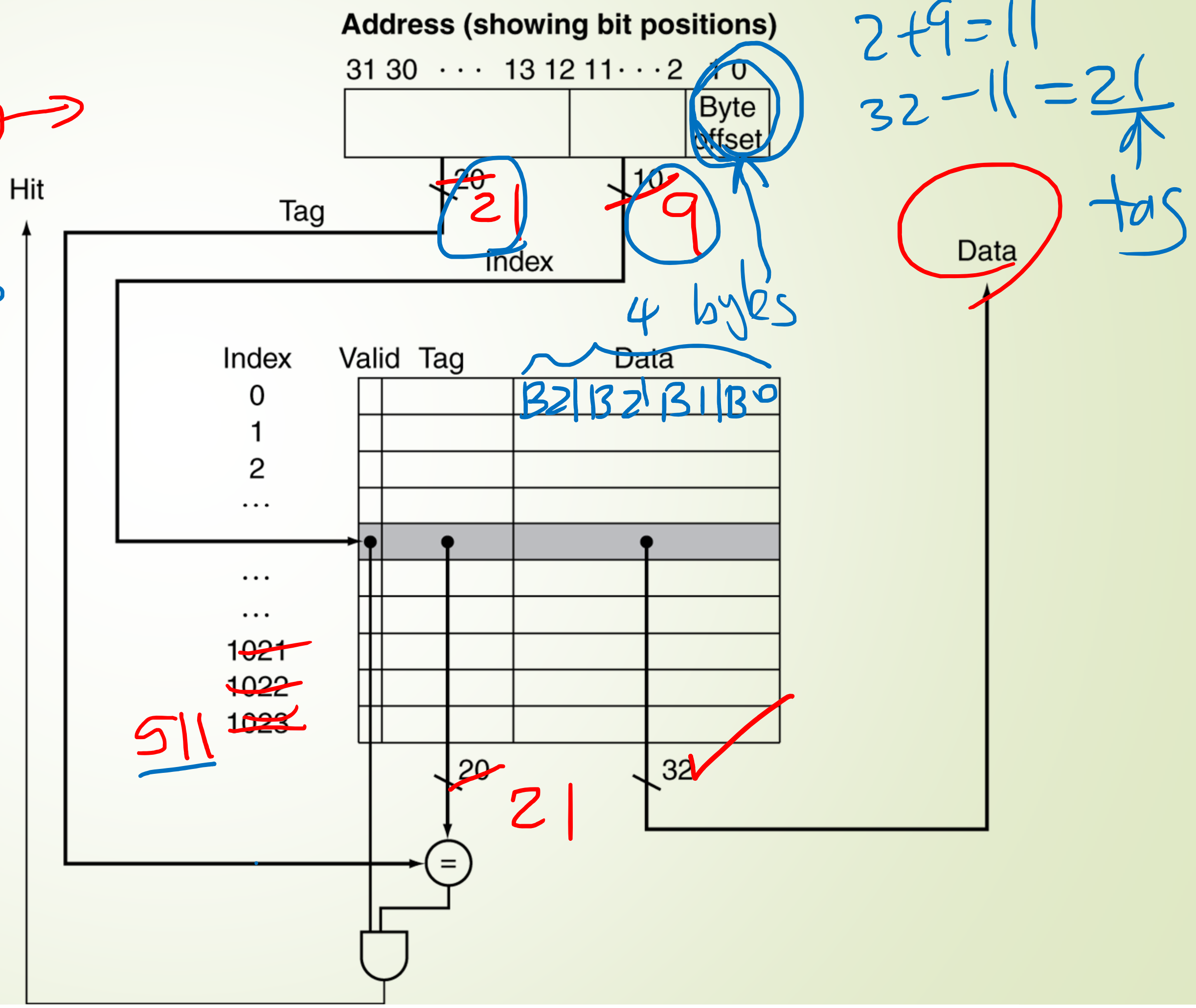
4 bytes

$2KB \div 4 = 512$

$.5k = 512$

$2^x = 512$

$x = 9$ index



$2+9=11$

$32-11=21$

tag

28) The following is a series of address references given as word addresses: 9, 4, 20, 4, 8, 15, 5, 19, 4, 20, 4, 22, 7, 17, 10. Assume direct map with a word size of 1, and a total size of 8 words. Show the hits and misses and final cache contents. Show the final cache content.

Byte

tag	index

$2^3 = 8$

index 3

Location	Hit/Miss?
9	M
4	M
20	M
4	M
8	M
15	M
5	M
19	M
4	H
20	M
4	M
22	M
7	M
17	M
10	M

16 8 4 2 1

0 1 0 0 1

0 0 1 0 0

1 0 1 0 0

0 0 1 0 0

0 1 0 0 0

0 1 1 1 1

0 0 1 0 1

1 0 0 1 1

1 0 1 0 0

1 0 1 1 0

0 0 1 1 1

1 0 0 0 1

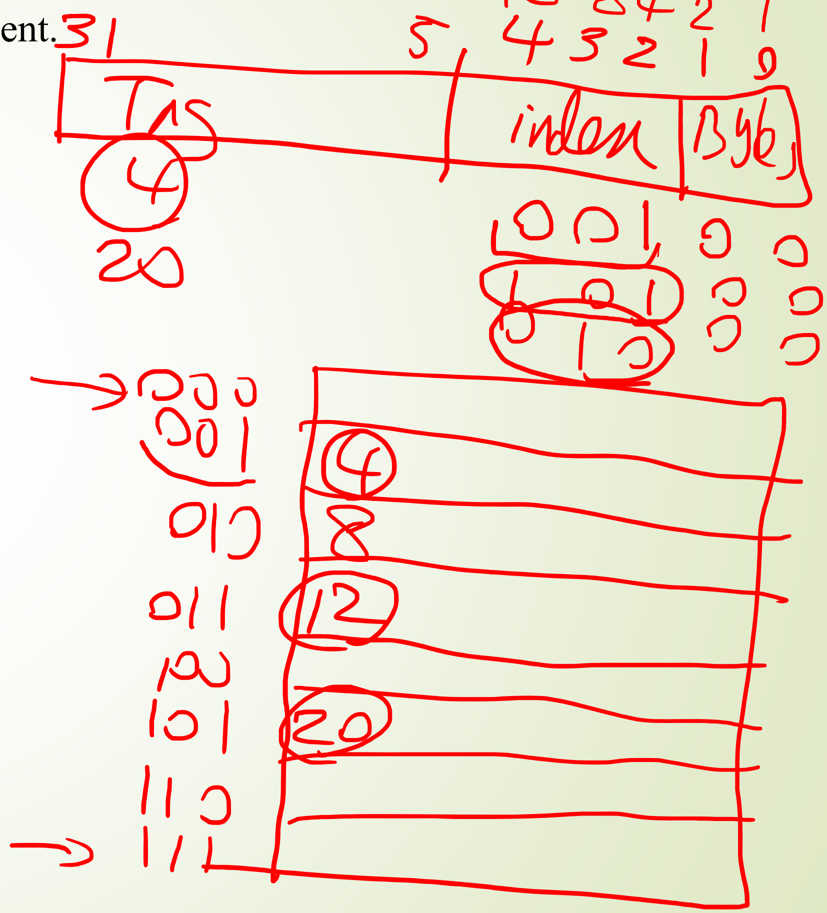
0 1 0 1 0

000	8
001	~~9~~ 17
010	10
011	19
100	~~4~~ ~~20~~ ~~4~~ ~~20~~ ~~4~~ 20
101	5
110	22
111	~~15~~ 7

28) The following is a series of address references given as word addresses: 9, 4, 20, 4, 8, 15, 5, 19, 4, 20, 4, 22, 7, 17, 10. Assume direct map with a word size of 4 bytes and a total size of 8 words. Show the hits and misses and final cache contents. Show the final cache content.

Location	Hit/Miss?
4	M
20	M
4	H
8	M
12	M
4	H
20	H
4	H
7	
17	
10	

01100



31
5
16 8 4 2 1
4 3 2 1 0
Tag
index
Byte
4
20
001 0 0
101 0 0
0 10 0 0
000
001
010
011
100
101
110
111
4
8
12
20