CSARCH Lecture Series: Cache Memory: Mapping Function

Sensei RL Uy

College of Computer Studies

De La Salle University

Manila, Philippines

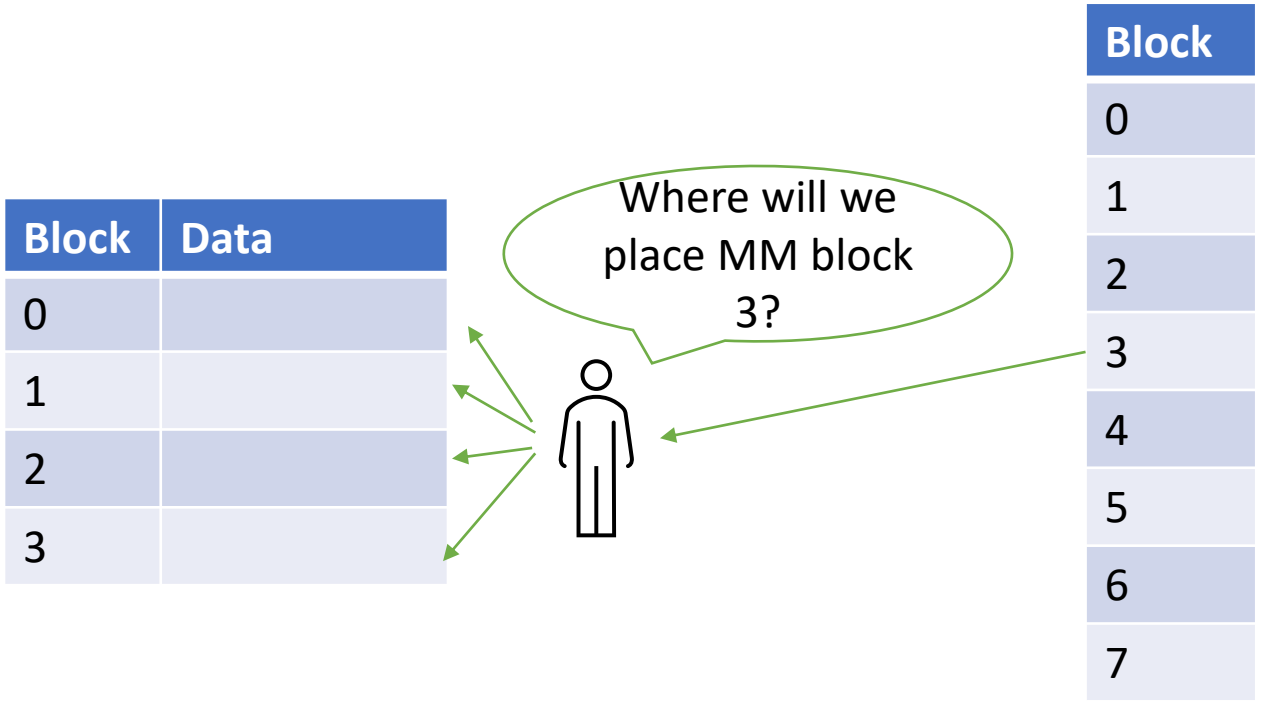
'2110

Copyright Notice

Overview

Reflect on the following question:

- Given a main memory block, where will it be place in the cache memory?



Overview

- This sub-module introduces the concept of cache memory mapping function
- The objectives are as follows:
  - ✓Describe the direct cache memory mapping function
  - ✓Describe the full associative cache memory mapping function
  - ✓Describe the block set associative cache memory mapping function

Cache operation

- In which cache block will the main memory blocks be placed? [Mapping function]
- Which cache block to replace? [replacement algorithm]

Mapping Function

- ***Mapping functions*** are used to identify where the main memory blocks are placed in the cache
- Three types of mapping function:
  - Direct
  - Full Associative
  - Block Set Associative

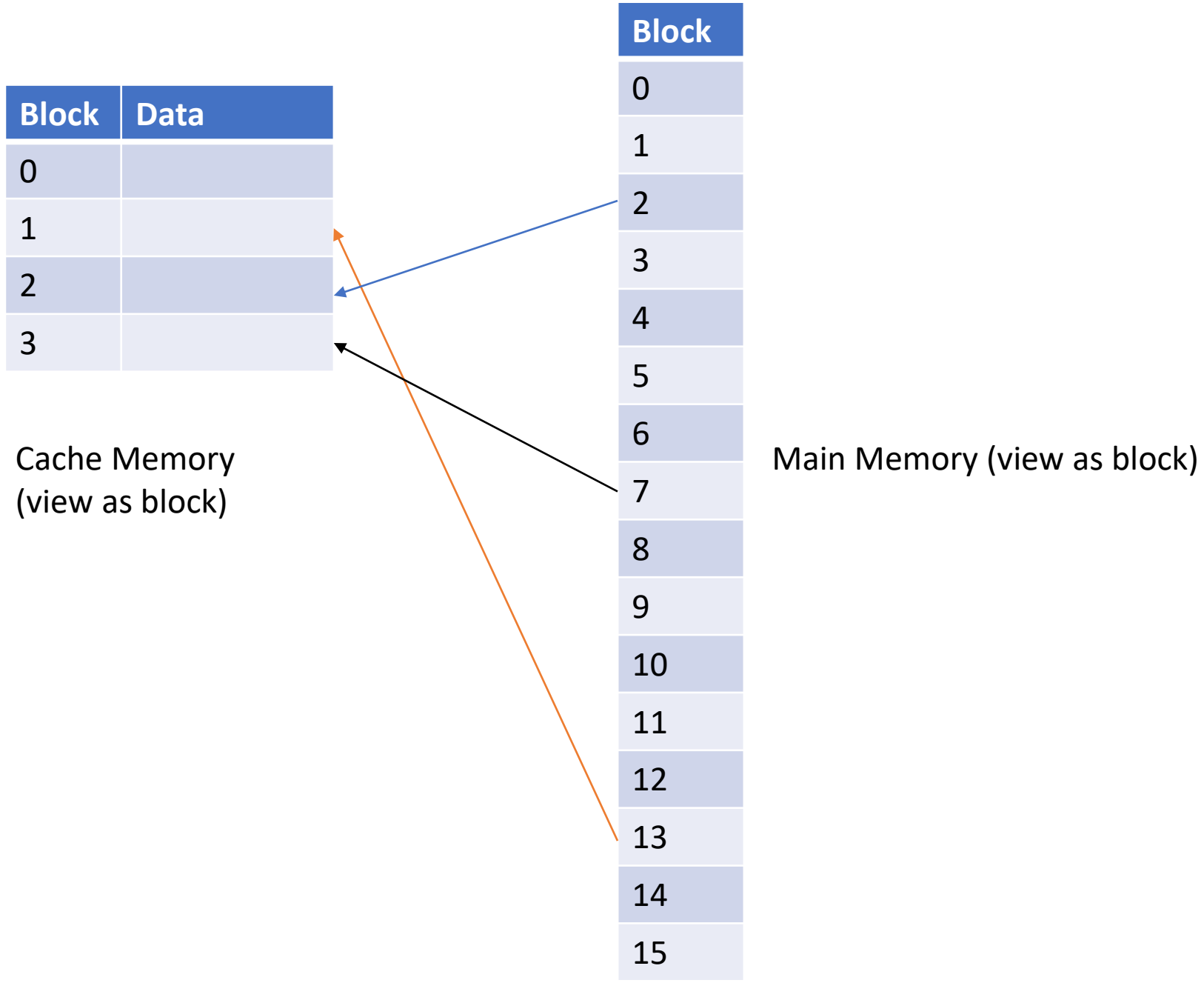
Direct Mapping (main memory block)

- In direct mapping, if main memory (MM) is viewed as block, main memory blocks are mapped onto the cache blocks in **modulo** fashion.
  - (MM block number) modulo (# of blocks in the cache)

Direct Mapping

- Cache has 4 blocks
- Main memory has 16 blocks

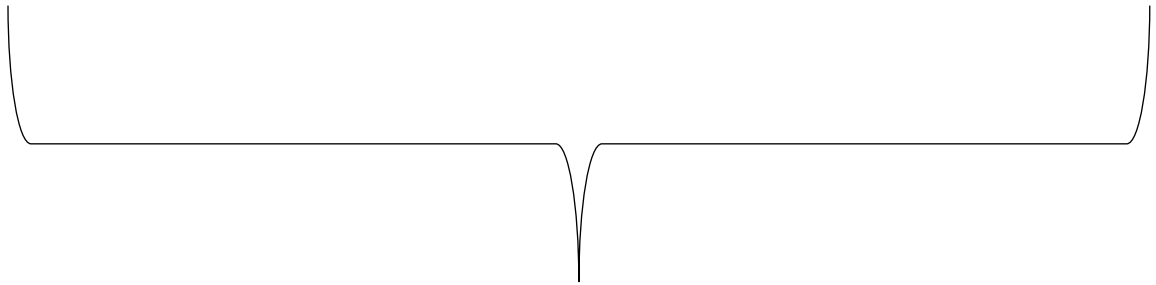
- Main memory block 13 is mapped to (13 mod 4 = 1) cache block 1
- Main memory block 2 is mapped to (2 mod 4 = 2) cache block 2
- How about main memory block 7?
- Main memory block 7 is mapped to (7 mod 4 = 3) cache block 3



Direct Mapping (main memory address)

- In direct mapping, if main memory (MM) is viewed as address, a main memory address is mapped onto the cache blocks by obtaining the "block" partition of the main memory address.

TAG (*n*-*k*-*w* bits)	Block (*k* bits)	Word (*w* bits)
# of memory blocks that are mapped to a particular cache block	# of blocks in the cache (2^k)	# of word in a block (2^w)



main memory address (*n* bits)

Direct Mapping

Example: A direct-mapped cache consists of a total of 4 blocks. Each block is 4 words. The main memory size is 256 words (figure is NOT illustrated to scale due to space constraint).

TAG(T)	Block (B)	Word (W)
4	2	2

---------- 8 bits ----------

MM address size 256 words = 8 bits

Block size (W) = 4 words = 2 bits

# of cache blocks (B) = 4
Block (B) = 2

TAG(T) = 8-2-2 = 4 bits

Valid bit	TAG	Data
0		
0		
0		
0		

Cache Memory

Address (binary)	Data (Hex)	Address (binary)	Data (hex)
00000000		00010000	
00000001		00010001	
00000010		00010010	
00000011		00010011	
00000100		00010100	
00000101		00010101	
00000110		00010110	
00000111		00010111	
00001000		00011000	
00001001		00011001	
00001010		00011010	
00001011		00011011	
00001100		00011100	
00001101		00011101	
00001110		00011110	
00001111		00011111	

Main Memory

Direct Mapping

Example: A direct-mapped cache consists of a total of 4 blocks. Each block is 4 words. The main memory size is 256 words (figure is NOT illustrated to scale due to space constraint).

TAG(T)	Block (B)	Word (W)
4	2	2

---------- 8 bits ----------

MM address 7 (00000111) is mapped to cache block 1

MM address 99 (01100011) is mapped to cache block 0

How about MM address 201?

MM address 201 (11001001) is mapped to cache block 2

Valid bit	TAG	Data
0		Block 0
0		Block 1
0		Block 2
0		Block 3

Cache Memory

Address (binary)	Data (Hex)	Address (binary)	Data (hex)
00000000		00010000	
00000001		00010001	
00000010		00010010	
00000011		00010011	
00000100		00010100	
00000101		00010101	
00000110		00010110	
00000111		00010111	
00001000		00011000	
00001001		00011001	
00001010		00011010	
00001011		00011011	
00001100		00011100	
00001101		00011101	
00001110		00011110	
00001111		00011111	

Main Memory

Direct Mapping

- **Example**: A direct-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - How many bits are there in a main memory address?
  - In the main memory address, how many bits are there in each of the TAG, BLOCK, and WORD fields?
  - What is the size of the cache memory (in bits)?
  - What is the size of the main memory (in bits)?

TAG(T)	Block (B)	Word (W)

Direct Mapping

- **Example**: A direct-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - In which cache block will memory block 10 be mapped?
  - In which cache block will memory address 10 be mapped?

Direct Mapping

- **Example**: A direct-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - How many bits are there in a main memory address? 7
  - In the main memory address, how many bits are there in each of the TAG, BLOCK, and WORD fields?
  - What is the size of the cache memory (in bits)? 524
  - What is the size of the main memory (in bits)? 2048

TAG(T)	Block (B)	Word (W)
2	2	3

MM address = 16 blocks*8 words/block = 128 words (2^7) = 7 bits

Block size is 2^3 words (3 bits); # of cache blocks = 2^2 (2 bits)

Each cache block = 2^3 words * 2^4 bits/word + 2 bits(tag) + 1 bit (valid) = 131 bits

cache memory size = 131 bits * 4 = 524 bits

Main memory size = 2^4 blocks * 2^3 words * 2^4 bits/word = 2^{11} or 2048 bits

Direct Mapping

- **Example**: A direct-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - In which cache block will memory block 10 be mapped? 2
  - In which cache block will memory address 10 be mapped? 1

memory block 10 modulo 4 = 2

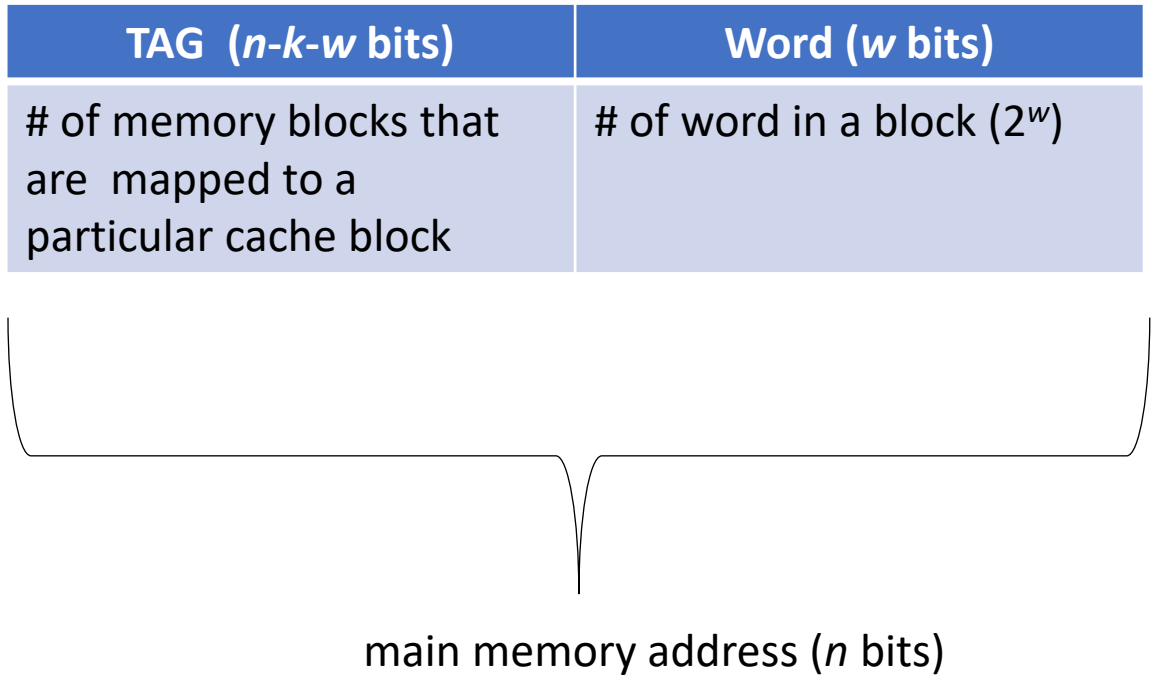
memory address 10 (00**01**010) mapped to cache block 1

Try

- **Example**: A direct-mapped cache consists of a total of 64 blocks. The main memory contains 4096 blocks, each consisting of 128 words. Each word is 16 bits.
  - How many bits are there in a main memory address?
  - In the main memory address, how many bits are there in each of the TAG, BLOCK, and WORD fields?
  - What is the size of the cache memory (bits)?
  - What is the size of the main memory (bits)?
  - In which cache block will memory block 1000 blocks be mapped?
  - In which cache block will memory address 1000 be mapped?

Full associative Mapping

- In full associative mapping, all main memory blocks can be mapped onto any cache block.



Full Associative Mapping

- Cache has 4 blocks
- Main memory has 16 blocks

- Main memory block 13 can be mapped any empty cache block

- Main memory block 2 can be mapped to any empty cache block

- How about main memory block 7?

- Main memory block 7 can be mapped to any empty cache block

Block	Data
0	
1	
2	
3	

Cache Memory
(view as block)

Block
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Main Memory (view as block)

Full Associative mapping

Example: A fully associative cache consists of a total of 4 blocks. Each block is 4 words. The main memory size is 256 words (figure is NOT illustrated to scale due to space constraint).

TAG(T)	Word (W)
6	2

---------- 8 bits ----------

MM address size 256 words = 8 bits

Block size (W) = 4 words = 2 bits

TAG(T) = 8-2 = 6 bits

MM address 7 can be mapped to any empty cache block

MM address 99 can be mapped to any empty cache block

Valid bit	TAG	Data
0		
0		
0		
0		

Cache Memory

Address (binary)	Data (Hex)	Address (binary)	Data (hex)
00000000		00010000	
00000001		00010001	
00000010		00010010	
00000011		00010011	
00000100		00010100	
00000101		00010101	
00000110		00010110	
00000111		00010111	
00001000		00011000	
00001001		00011001	
00001010		00011010	
00001011		00011011	
00001100		00011100	
00001101		00011101	
00001110		00011110	
00001111		00011111	

Main Memory

Associative Mapping

- **Example**: An associative-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - How many bits are there in a main memory address?
  - In the main memory address, how many bits are there in each of the TAG and WORD fields?
  - What is the size of the cache memory (in bits)?
  - What is the size of the main memory (in bits)?

TAG(T)	Word (W)

Associative Mapping

- **Example**: An associative-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16-bits.
  - In which cache block will memory block 10 be mapped?
  - In which cache block will memory address 10 be mapped?

Associative Mapping

- **Example**: An associative-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - How many bits are there in a main memory address? 7
  - In the main memory address, how many bits are there in each of the TAG and WORD fields?

TAG(T)	Word (W)
4	3

  - What is the size of the cache memory (in bits)? 532
  - What is the size of the main memory (in bits)? 2048

MM address = 16 blocks*8 words/block = 128 words (2^7) = 7 bits

Block size is 2^3 words (3 bits)

Each cache block = 2^3 words * 2^4 bits/word + 4 bits(tag) + 1 bit (valid) = 133 bits

cache memory size = 133 bits * 4 = 532 bits

Main memory size = 2^4 blocks * 2^3 words * 2^4 bits/word = 2^{11} or 2048 bits

Associative Mapping

- **Example**: An associative-mapped cache consists of a total of 4 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16-bits.
  - In which cache block will memory block 10 be mapped? anywhere
  - In which cache block will memory address 10 be mapped? anywhere

Try

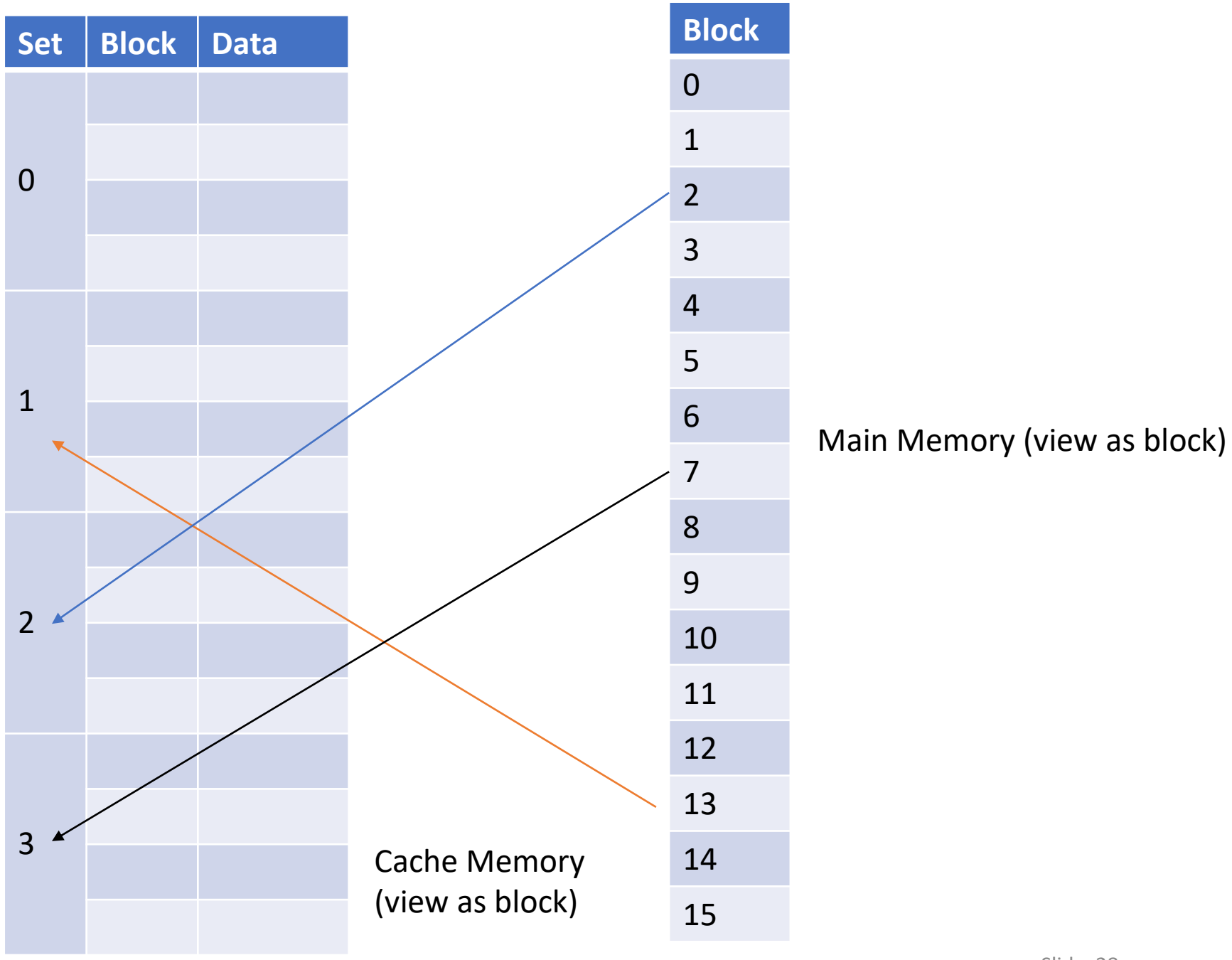
- **Example**: An associative-mapped cache consists of a total of 64 blocks. The main memory contains 4096 blocks, each consisting of 128 words. Each word is 16 bits.
  - How many bits are there in a main memory address?
  - In the main memory address, how many bits are there in each of the TAG and WORD fields?
  - What is the size of the cache memory (bits)?
  - What is the size of the main memory (bits)?
  - In which cache block will memory block 1000 be mapped?
  - In which cache block will memory address 1000 be mapped?

Block Set Associative (main memory block)

- In block-set-associative mapping, cache blocks are grouped into sets (i.e., 4-way block set associative means set size is 4 blocks).
- Main memory blocks are then allowed to map onto any cache blocks of the corresponding set.
- This method compromises between the speed of direct mapping and utilization of associative mapping.
- If main memory (MM) is viewed as block, main memory blocks are mapped onto the cache set in **modulo** fashion.
  - (MM block number) modulo (# of sets in the cache)

Block Set Associative

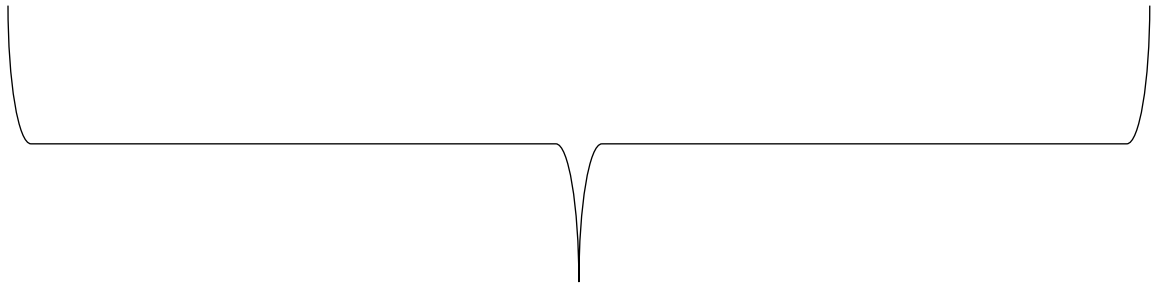
- Cache has 16 blocks and set size of 4 blocks = 4 sets
- Main memory has 16 blocks
- Main memory block 13 is mapped to (13 mod 4 = 1) cache set 1
- Main memory block 2 is mapped to (2 mod 4 = 2) cache set 2
- How about main memory block 7?
- Main memory block 7 is mapped to (7 mod 4 = 3) cache set 3



Block Set Associative (main memory address)

- In block set associative mapping, if main memory (MM) is viewed as address, a main memory address is mapped onto the cache set by obtaining the “set” partition of the main memory address.

TAG (*n*-*k*-*w* bits)	Set (*k* bits)	Word (*w* bits)
# of memory blocks that are mapped to a particular cache set	# of sets in the cache (2^k)	# of word in a block (2^w)



main memory address (*n* bits)

Block Set Associative

Example: A block set associative cache consists of a total of 16 blocks with set size of 4 blocks. Each block is 4 words. The main memory size is 256 words (figure is NOT illustrated to scale due to space constraint).

TAG(T)	Set (Set)	Word(W)
4	2	2

---------- 8 bits ----------

MM address size 256 words = 8 bits

Set size (S) = 16 blocks/4 = 4 sets = 2 bits

TAG(T) = 8-2-2 = 4 bits

Set	Block	Data
0		
1		
2		
3		

Cache Memory

Address (binary)	Data (Hex)	Address (binary)	Data (hex)
00000000		00010000	
00000001		00010001	
00000010		00010010	
00000011		00010011	
00000100		00010100	
00000101		00010101	
00000110		00010110	
00000111		00010111	
00001000		00011000	
00001001		00011001	
00001010		00011010	
00001011		00011011	
00001100		00011100	
00001101		00011101	
00001110		00011110	
00001111		00011111	

Main Memory

Block Set Associative

Example: A block set associative cache consists of a total of 16 blocks with set size of 4 blocks. Each block is 4 words. The main memory size is 256 words (figure is NOT illustrated to scale due to space constraint).

TAG(T)	Set (Set)	Word(W)
4	2	2

---------- 8 bits ----------

MM address 7 (00000111) is mapped to cache set 1

MM address 99 (01100011) is mapped to cache set 0

How about MM address 201?

MM address 201 (11001001) is mapped to cache set 2

Set	Block	Data
0		
1		
2		
3		

Cache Memory

Address (binary)	Data (Hex)	Address (binary)	Data (hex)
00000000		00010000	
00000001		00010001	
00000010		00010010	
00000011		00010011	
00000100		00010100	
00000101		00010101	
00000110		00010110	
00000111		00010111	
00001000		00011000	
00001001		00011001	
00001010		00011010	
00001011		00011011	
00001100		00011100	
00001101		00011101	
00001110		00011110	
00001111		00011111	

Main Memory

Block Set Associative Mapping

- **Example**: A 2-way block set associative mapping cache consists of a total of 8 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - How many bits are there in a main memory address?
  - In the main memory address, how many bits are there in each of the TAG, SET, and WORD fields?
  - What is the size of the cache memory (in bits)?
  - What is the size of the main memory (in bits)?

TAG(T)	Set (S)	Word(W)

Block Set Associative Mapping

- **Example**: A 2-way block set associative mapped cache consists of a total of 8 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bit.
  - In which cache set will memory block 10 be mapped?
  - In which cache set will memory address 10 be mapped?

Block Set Associative Mapping

- **Example**: A 2-way block set associative mapping cache consists of a total of 8 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bits.
  - How many bits are there in a main memory address? 7
  - In the main memory address, how many bits are there in each of the TAG, SET, and WORD fields?
  - What is the size of the cache memory (in bits)? 1056
  - What is the size of the main memory (in bits)? 2048

TAG(T)	Set (S)	Word(W)
2	2	3

MM address = 16 blocks*8 words/block = 128 words (2^7) = 7 bits

Block size is 2^3 words (3 bits); # of cache sets = 8 blocks/2blocks/set (2 bits)

Each cache block = 2^3 words * 2^4 bits/word + 2 bits(tag) + 1 bit (valid) = 131 bits

cache memory size = 131 bits * 8 = 1048 bits

Main memory size = 2^4 blocks * 2^3 words * 2^4 bits/word = 2^{11} or 2048 bits

Block Set Associative Mapping

- **Example**: A 2-way block set associative mapped cache consists of a total of 8 blocks. The main memory contains 16 blocks, each consisting of 8 words. Each word is 16 bit.
  - In which cache set will memory block 10 be mapped? 2
  - In which cache set will memory address 10 be mapped? 2

memory block 10 modulo 4 = 2

memory address 10 (0001010) mapped to cache set 2

Try

- **Example**: A 4-way block set associative mapping cache consists of a total of 64 blocks. The main memory contains 4096 blocks, each consisting of 128 words. Each word is 16-bit.
  - How many bits are there in a main memory address?
  - In the main memory address, how many bits are there in each of the TAG, SET, and WORD fields?
  - What is the size of the cache memory (bits)?
  - What is the size of the main memory (bits)?
  - In which cache set will memory block 1000 be mapped?
  - In which cache set will memory address 1000 be mapped?

To recall …

- What have we learned:
  - ✓Describe the direct cache memory mapping function
  - ✓Describe the full associative cache memory mapping function
  - ✓Describe the block set associative cache memory mapping function