Microprocessor & Computer Architecture (µpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

Microprocessor & Computer Architecture (µpCA)

Unit 3: Mapping Functions

Dr. D. C. Kiran

Department of Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Syllabus

~~Unit 1: Basic Processor Architecture and Design~~

~~Unit 2: Pipelined Processor and Design~~

Unit 3: Memory

- ~~Memory Hierarchy~~
- ~~Principles of Locality~~
- ~~Cache Design Principles~~

Mapping Functions

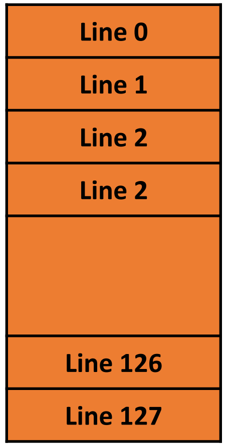
- ~~Direct Mapping~~
- Full Associative Mapping
- Set Associative Mapping

Unit 4: Input/Output Device Design

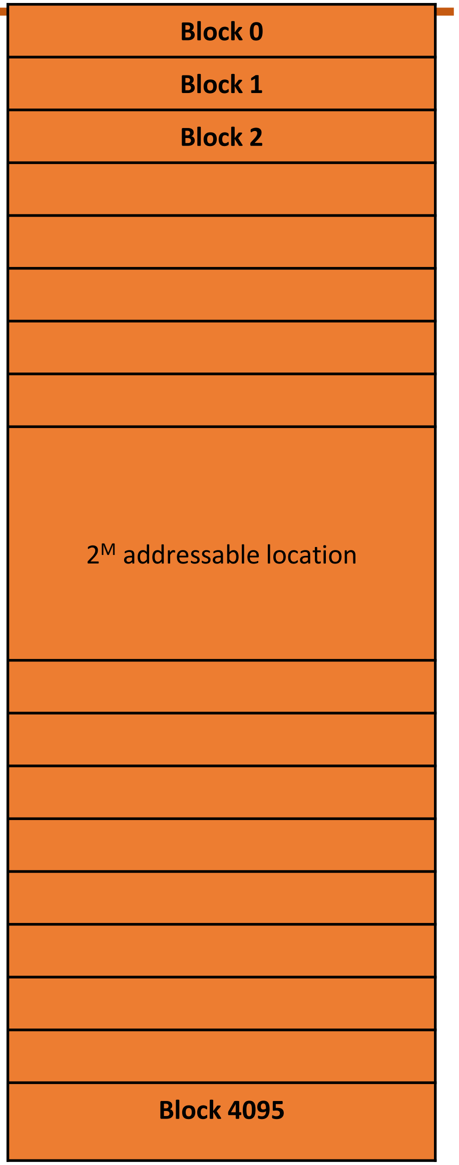
Unit 5: Advanced Architecture

Full Associative Mapping

- **A simple processor example:**
  - Main memory is addressable by a 16-bit address.
  - Main memory has 65536 (64 k) words.
  - Main memory has 4096 Blocks of 16 words each.
  - Consecutive addresses refer to consecutive words.
  - Cache consisting of 128 Lines of 16 words each.
  - Total size of cache is 2048 (4 K) words.
  - 4096 can be placed on any 128 Lines.



128 Block Cache



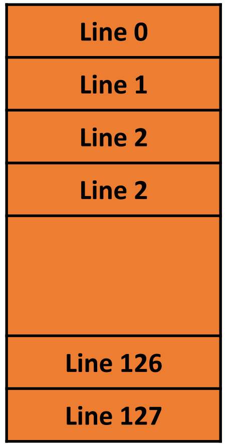
Full Associative Mapping

- *Block Placement:*
  - Main Memory block can be placed into any cache line
- *Block Identification:*
  - Memory Address is divided into Word, and Tag part
  - word is used to identify placement
  - Tag is used to identify validity of the word
  - Ex: How to differentiate between 1st word in Block 0 and 1st word in Block 256
  - Need to search all 128 lines for suitable match
- *Block Replacement:*
  - Replacement algorithms are used to replace an existing block in the cache when cache is full
  - Flexible and use cache space efficiently
  - *Does avoid thrashing*

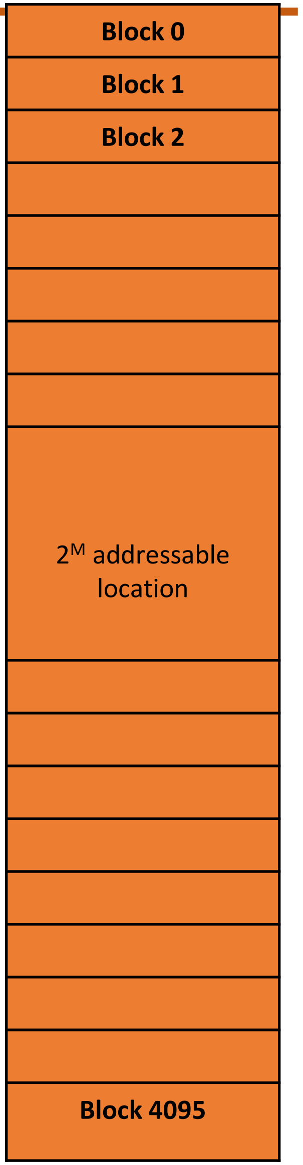
16 bit Address

Tag , 12 bits	**Word,4 bits**

2^4 =16 words **2^{12}** = 4096 blocks can be on any Line

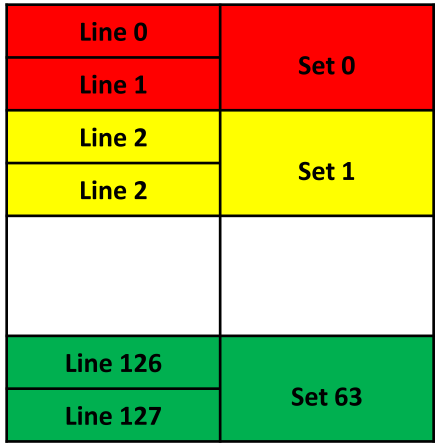


128 Block Cache



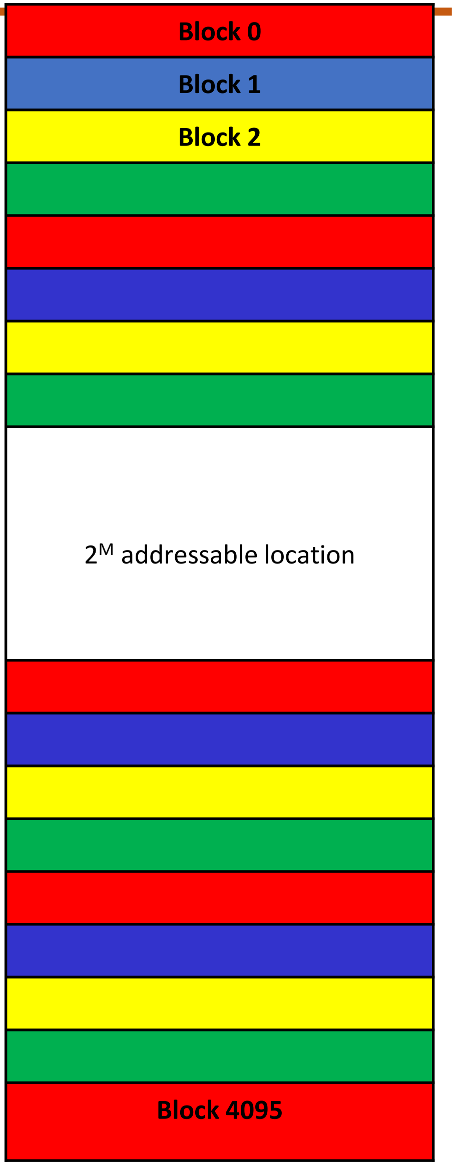
Set Associative Mapping

- **A simple processor example:**
  - Main memory is addressable by a 16-bit address.
  - Main memory has 65536 (64 k) words.
  - Main memory has 4096 Blocks of 16 words each.
  - Cache consisting of 128 Lines of 16 words each.
  - 2 Line in one set. 128/2= 64 Sets
  - 4096/64= 64 possible Blocks for one Set



Line 0	Set 0
Line 1	
Line 2	Set 1
Line 2	
Line 126	Set 63
Line 127	

128 Block Cache



Block 0

Block 1

Block 2

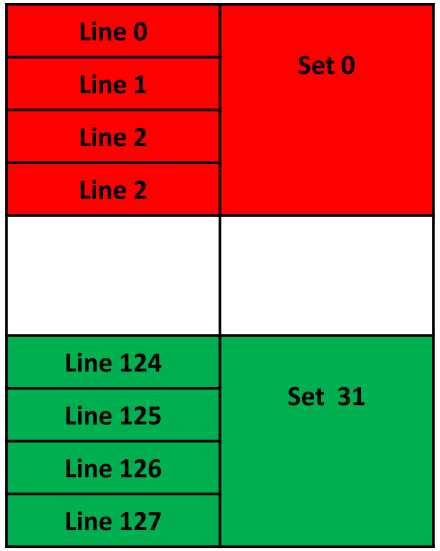
2^M addressable location

Block 4095

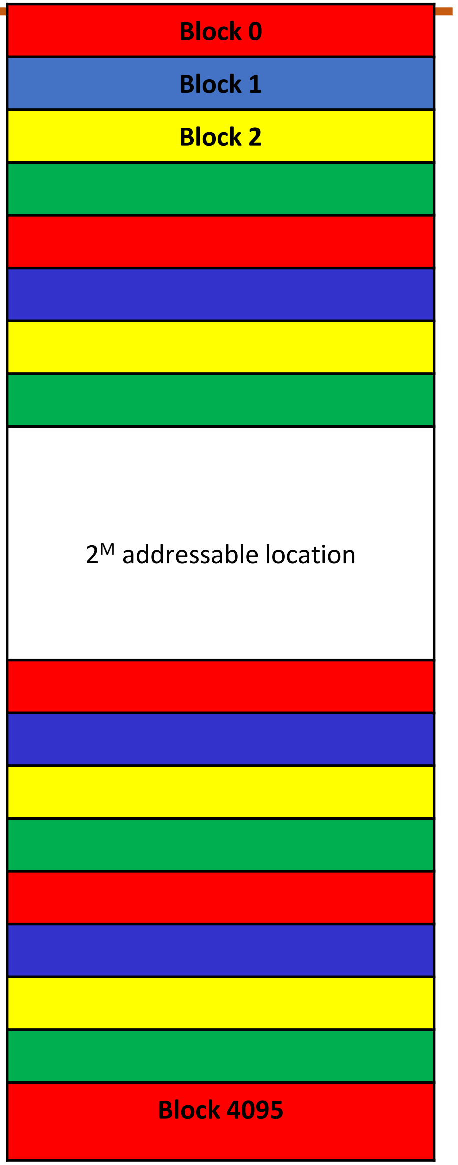
Microprocessor & Computer Architecture (μpCA)

Set Associative Mapping

- **A simple processor example:**
  - Main memory is addressable by a 16-bit address.
  - Main memory has 65536 (64 k) words.
  - Main memory has 4096 Blocks of 16 words each.
  - Cache consisting of 128 Lines of 16 words each.
  - 4 Line in one set. 128/4= 32 Sets
  - 4096/32= 128possible Blocks for one Set

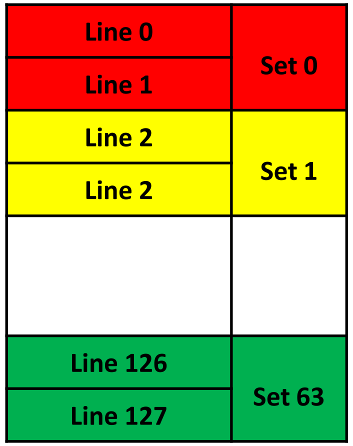


128 Block Cache

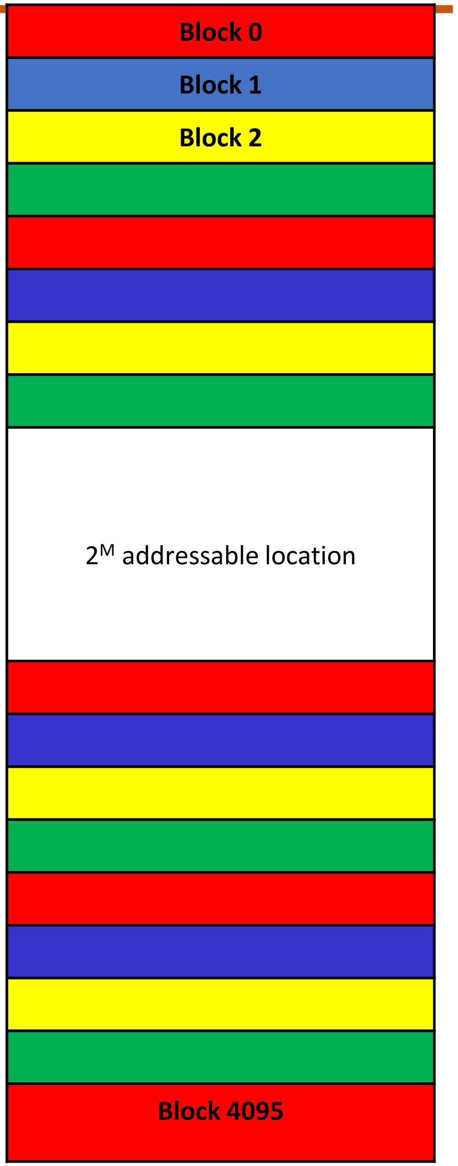


2 Way Set Associative Mapping: Block Placement

- Lines are grouped into sets.
- Mapping function allows a Block of the main memory to reside in any block of a specific set.
- Divide the cache into 64 sets, with two blocks per set. Memory Block 0, 64, 128 etc. map to Set 0, and they can occupy either of the two positions.



Line 0	Set 0
Line 1	
Line 2	Set 1
Line 2	
Line 126	Set 63
Line 127	



2 Way Set Associative Mapping: Block Placement

- Memory address is divided into three fields:

  - 6 bit field determines the set number.

  - High order 6 bit fields are compared to the tag fields of the two blocks in a set.

- Set-associative mapping combination of direct and associative mapping.

Line 0	Set 0
Line 1	
Line 2	Set 1
Line 2	
Line 126	Set 63
Line 127	

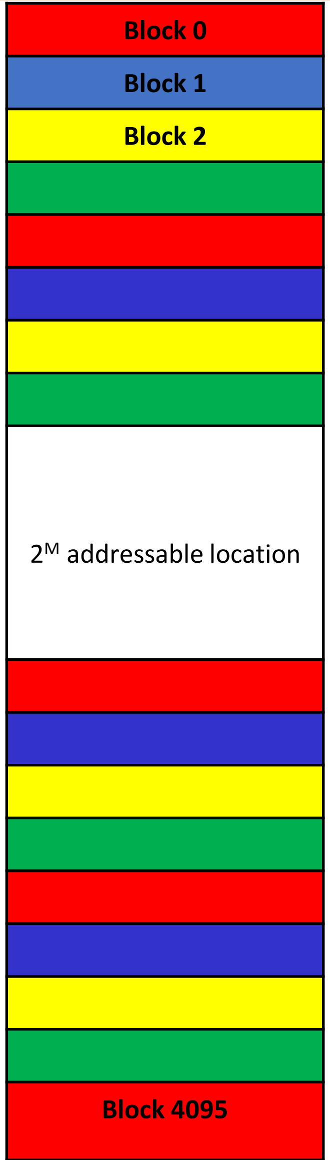
16 bit Address

Tag 6 bits	Set 6 bits	Word 4 bits

2^4 =16 words

2^6 = 64 Sets

2^6 = Which block out of 64 Blocks



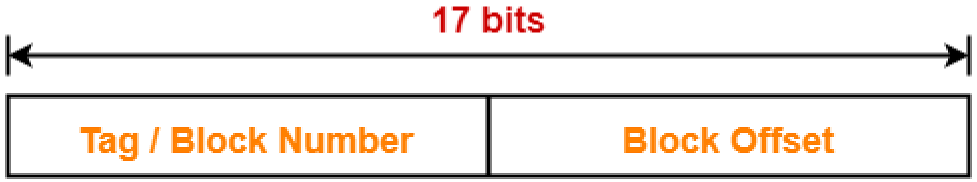
Exercise 1

Consider a fully associative mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find Number of bits in tag

Given-

- Cache memory size = 16 KB
- Block size = Frame size = Line size = 256 bytes
- Main memory size = 128 KB = 2^{17} bytes

Thus, Number of bits in physical address = 17 bits

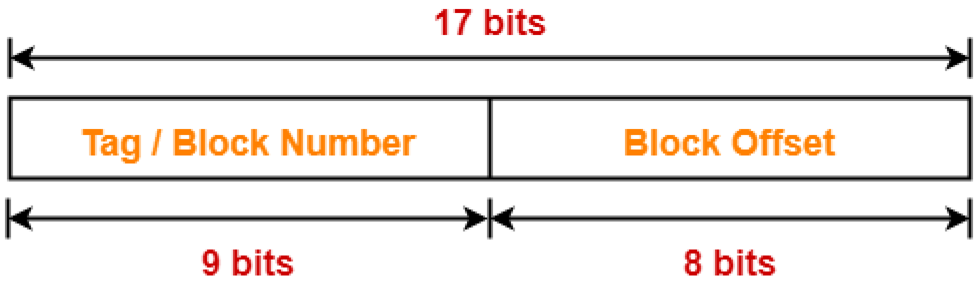


Exercise 1

Block size
= 256 bytes
= 2^8 bytes
Thus, Number of bits in block offset = 8 bits

Number of bits in tag = Num of bits in PA – Num of bits in block offset
= 17 bits – 8 bits = 9 bits

Thus, Number of bits in tag = 9 bits



Exercise 2

Consider a 2-way set associative cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find Number of bits in tag

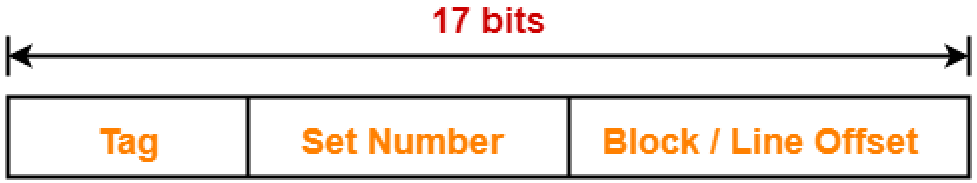
Given-

Cache memory size = 16 KB

Block size = Frame size = Line size = 256 bytes

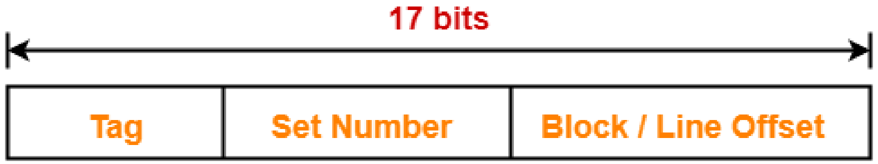
Main memory size = 128 KB = 2^17

Size of Memory Address: 17 bits

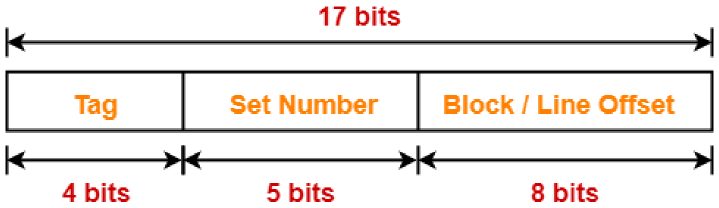


Microprocessor & Computer Architecture (μpCA)

Exercise 2



TAG	Set Number	Block / Line Offset
17-(8+5)= 4 bits	Total number of Blocks or Lines = Cache size / Block size = 16kb/256 = 2^14 bytes / 2^8 bytes = 2^6 Blocks or Lines **Number of Set** =2^6 Blocks or Lines/2 = 2^5	Block Size = 256 = 2^8 **Block Offset = 8 bits**



Exercise 3

A computer system uses 16-bit memory addresses. It has a 2K-byte cache organized in a 2-way set associative manner with 64 bytes per cache block. Assume that the size of each memory word is 1 byte.

(a) Calculate the number of bits in each of the Tag, Block, and Word fields of the memory address.

(b) When a program is executed, the processor reads data sequentially from the following word addresses: **128, 144, 2176, 2180, 128, 2176**

All the above addresses are shown in decimal values. Assume that the cache is initially empty.

For each of the above addresses, indicate whether the cache access will result in a hit or a miss.

Exercise 3

Block size = 64 bytes = 2^6 bytes

Therefore, **Number of bits in the *Word* field = 6**

Cache size = 2K-byte = 2^{11} bytes

Number of cache blocks = Cache size / (Block size x Number of blocks per set)

$= 2^{11}/(2^6 \text{ x } 2)= 2^4$

Therefore, **Number of bits in the *set* field = 4**

Total number of address bits = 16

Therefore, **Number of bits in the *Tag* field = 16 – (6+4) = 6**

For a given 16-bit address, the 6 most significant bits, represent the *Tag*, the next 4 bits represent the *Block*, and the 6 least significant bits represent the *Word*.

Microprocessor & Computer Architecture (μpCA)

Exercise 3

Address = $(128)_{10} = (0000000010000000)_2$

Address = $(144)_{10} = (0000000010010000)_2$

Address = $(2176)_{10} = (0000100010000000)_2$

Address = $(2180)_{10} = (0000100010000100)_2$

Exercise 3

Access # 1:

Address = $(128)_{10}$ = $(0000000010000000)_2$

For this address, *Tag* = **000000**, *Set* = **0010**, *Word* = 000000

Since the cache is empty before this access, this will be a cache **miss**

After this access, **Tag field for the first block in set 0010 is set to 000000**

Access # 2:

Address = $(144)_{10}$ = $(0000000010010000)_2$

For this address, *Tag* = **000000**, *Set* = **0010**, *Word* = 010000

The tag field for this address matches the tag field for the first block in set 0010. Therefore, this access will be a cache **hit**.

Exercise 3

Access # 3:

Address = $(2176)_{10}$ = $(0000100010000000)_2$

For this address, *Tag* = **000010**, *Set*= **0010**, *Word* = 000000

The tag field for this address does not match the tag field for the first block in set 0010.

The second block in set 0010 is empty. Therefore, this access will be a cache **miss**.

After this access, **Tag field for the second block in set 0010 is set to 000010**

Access # 4:

Address = $(2180)_{10}$ = $(0000100010000100)_2$

For this address, *Tag* = **000010**, *Set* = **0010**, *Word* = 000100

The tag field for this address matches the tag field for the second block in set **0010**. Therefore, this access will be a cache **hit**.

Exercise 3

Access # 5:
Address = $(128)_{10}$ = $(0000000010000000)_2$
For this address, *Tag* = **000000**, *Set* = **0010**, *Word* = 000000
The tag field for this address matches the tag field for the **first block in set 0010**.
Therefore, this access will be a cache **hit**.

Access # 6:
Address = $(2176)_{10}$ = $(0000100010000000)_2$
For this address, *Tag* = **000010**, *Set* = **0010**, *Word* = 000000
The tag field for this address matches the tag field for the **second block in set 0010**.
Therefore, this access will be a cache **hit**.

Exercise 3

Cache hit rate = Number of hits / Number of accesses = 4/6 = 0.666

Compare with direct mapping

Cache hit rate = Number of hits / Number of accesses
= 2/6
= 0.333

Next Session

Replacement Policies

THANK YOU

Dr. D. C. Kiran

Department of Computer Science and Engineering

dckiran@pes.edu

9829935135