



香港中文大學

The Chinese University of Hong Kong

CSCI2510 Computer Organization

Tutorial 10: Hints for Cache Implementation

Tsun-Yu YANG

yangty@cse.cuhk.edu.hk





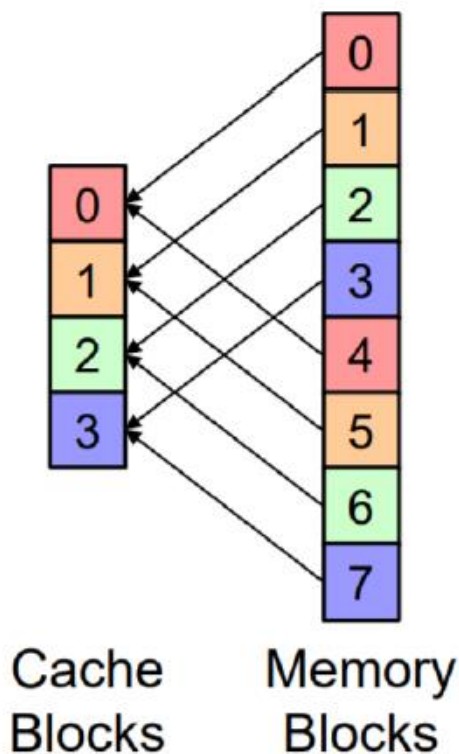
- Review of Direct & Associative Mapping
- Set-Associative Mapping
- Hints for Set-Associative Mapping Implementation
- Hints for Subroutine Implementation

Review of Direct Mapping



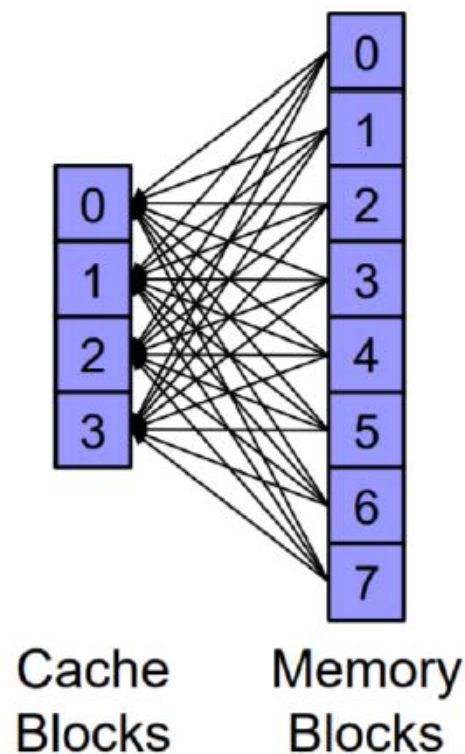
Direct

A Memory Block is directly mapped (%) to a Cache Block.



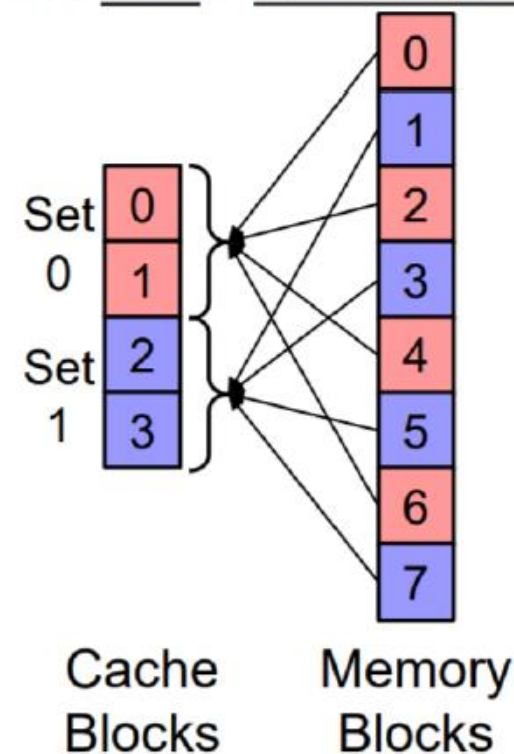
Associative

A Memory Block can be mapped to any Cache Block.
(First come first serve!)



Set Associative

A Memory Block is directly mapped (%) to a Cache Set.
In a Set? Associative

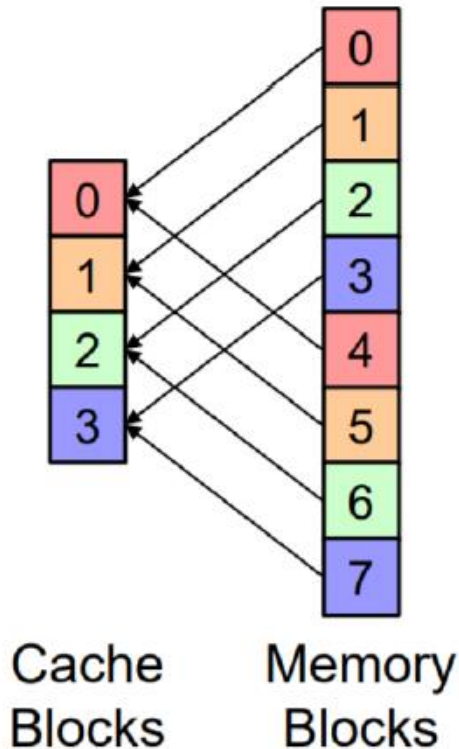


Review of Associative Mapping



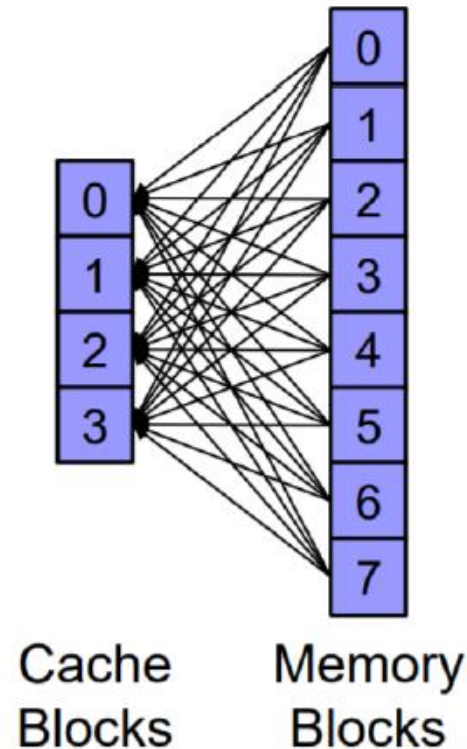
Direct

A Memory Block is directly mapped (%) to a Cache Block.



Associative

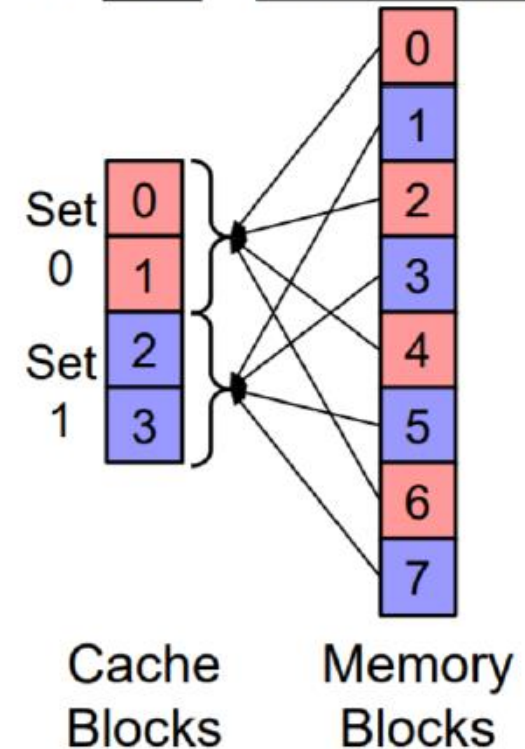
A Memory Block can be mapped to any Cache Block.
(First come first serve!)



Set Associative

A Memory Block is directly mapped (%) to a Cache Set.

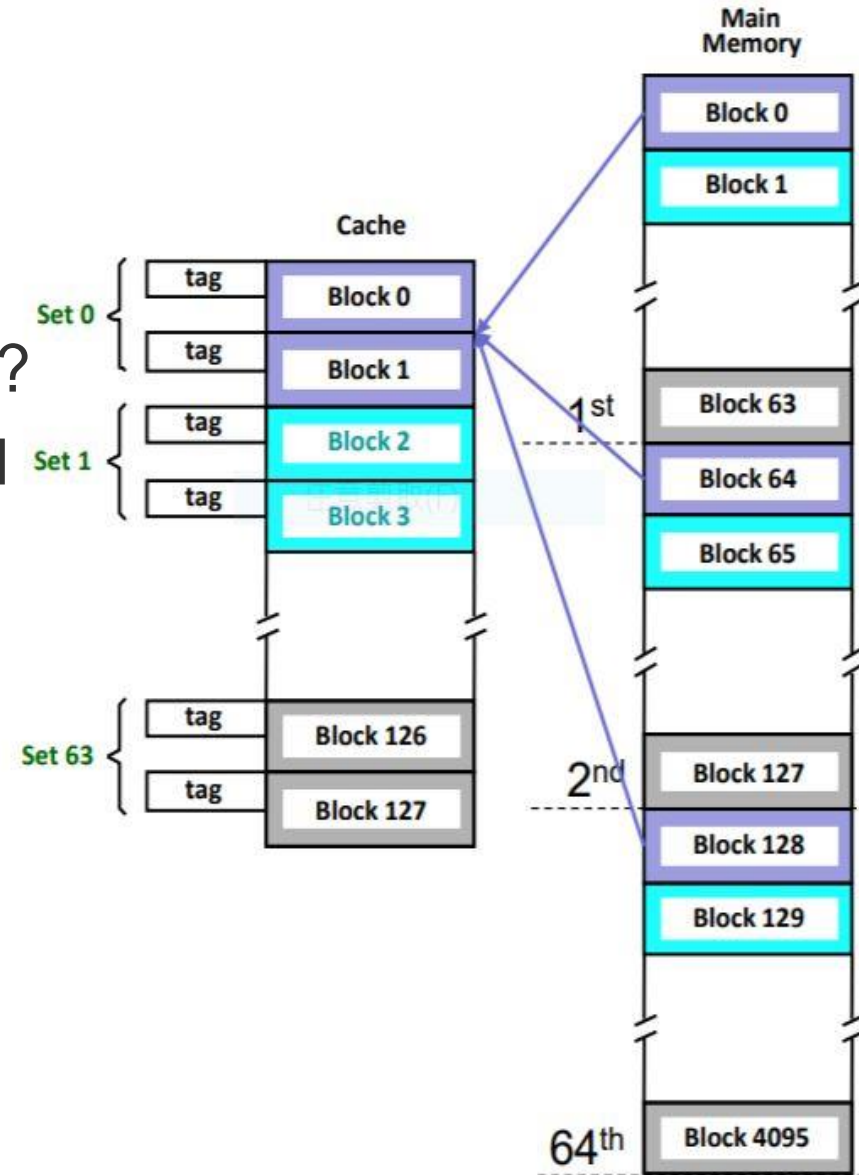
In a Set? Associative



Set-Associative Mapping



- Set-associative mapping is a combination of direct & associative mapping
- How to know Cache Hit or Miss?
 - Given a memory address, find the corresponding Set
 - Search all tags within the Set to find it's Hit or Miss.
- How to do replacement algorithm?
 - Find a block to be replaced within the given Set



HW3 Programming Assignment Ex1



- You are required to complete four main functions in **set_associative.asm**

- SearchSets
 - Direct_Mapping.asm
- SearchTags
 - Associative_Mapping.asm
- CacheHit
- CacheMiss

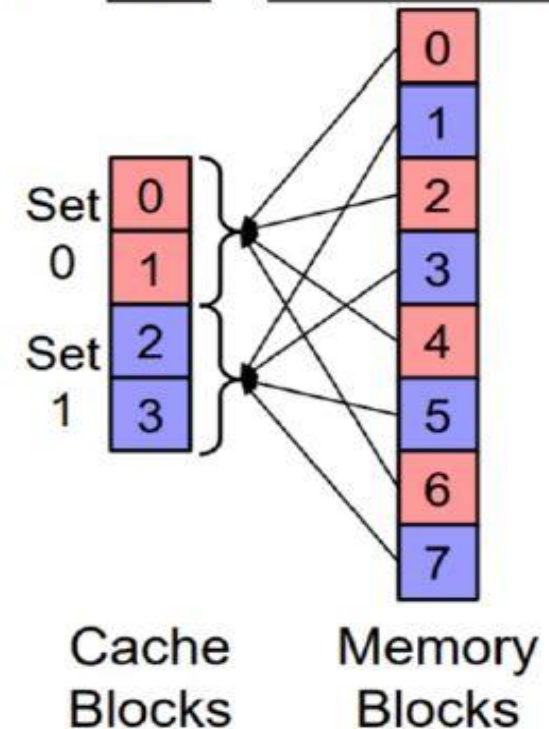


16-bit Main Memory Address

Set Associative

A Memory Block is **directly mapped** (%) to a **Cache Set**.

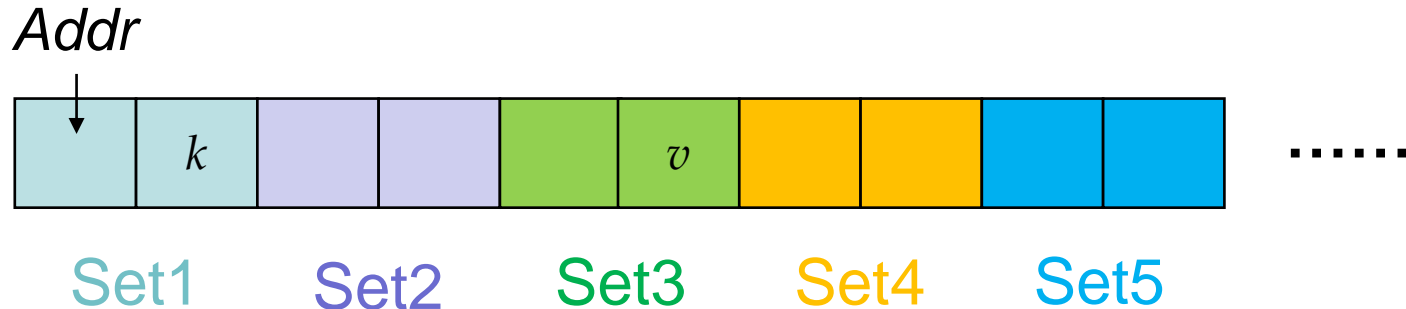
In a **Set?** **Associative**



How to search all tags within a set?



- Suppose each element is 4-byte in the below array:



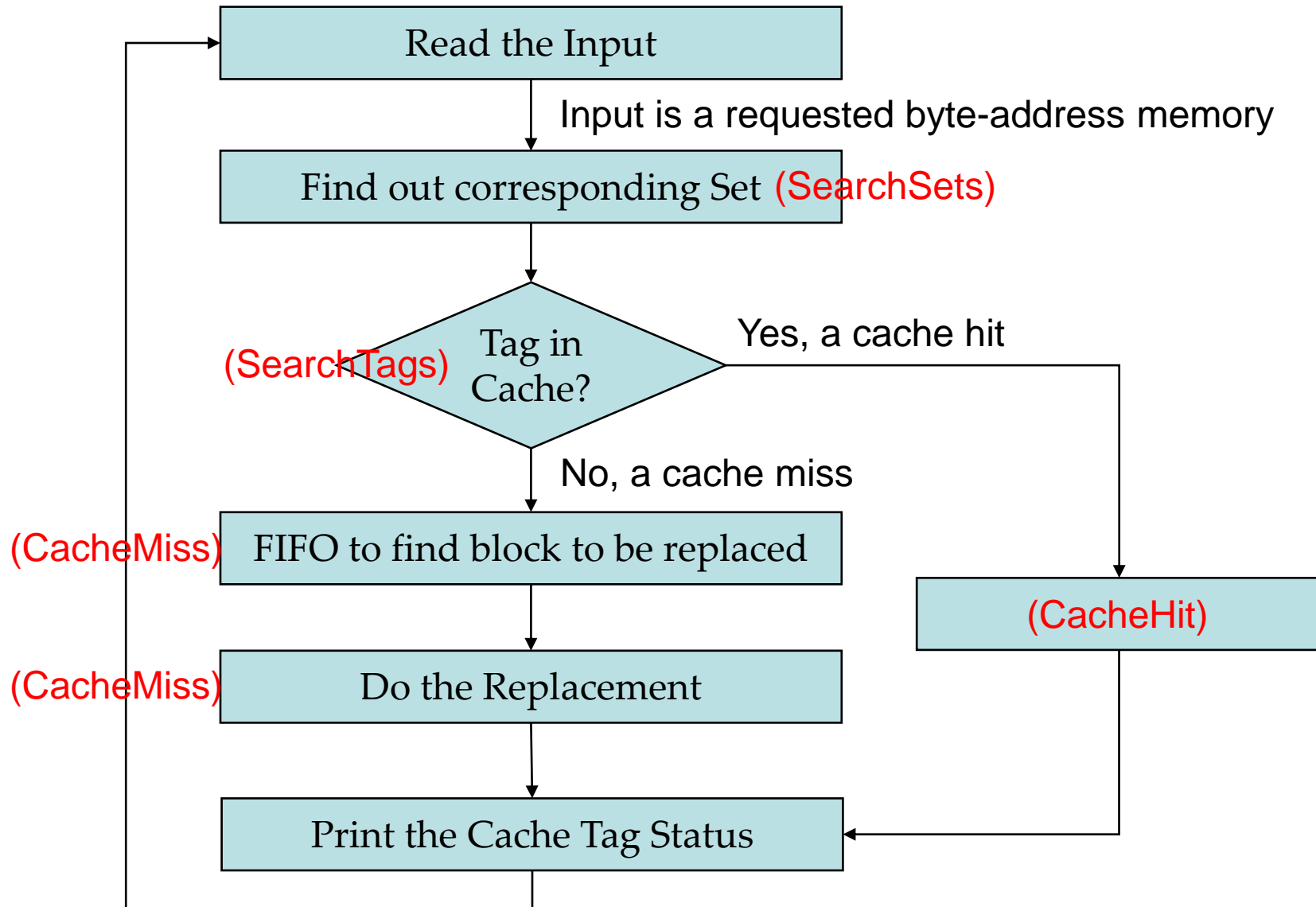
- What's the memory address of k ?
 - Answer: $\text{Addr} + 4 * 1$
- What's the memory address of v ?
 - Answer: $\text{Addr} + 4 * 2 * (3-1) + 4 * 1$
- What's the memory address of i^{th} element in j^{th} Set
 - Answer: $\text{Addr} + 4 * 2 * (j-1) + 4 * (i-1)$
 - ↘ Do this calculation first, then you have the address pointing to the first element of the Set

HW3 Programming Assignment Ex1



- You are required to complete four main functions in **set_associative.asm**
 - SearchSets
 - SearchTags
 - CacheHit
 - CacheMiss
 - Associative_Mapping.asm
- Related to the replacement algorithm (FIFO in this assignment)
- After the SearchTags, you will know it's a Cache Hit or Miss:
 - What should we do in CacheMiss if the replacement policy is FIFO? (Reference Tut09)

Flowchart of Set-Associative Mapping



HW3 Programming Assignment Ex2



- Convert the four main functions in **set_associative.asm** into subroutine

input:

```
mov EAX, CurrentTimeStamp
inc EAX
mov CurrentTimeStamp, EAX ; update the time stamp
mov EBP, offset CacheTags ; Hint: EBP could be updated in the process, so initialize here
mov ESI, offset CacheTimes ; Hint: ESI could be updated in the process, so initialize here
invoke crt_printf, addr InputStatement
invoke crt_scanf, addr MemoryAddressFormat, addr MemoryAddress
mov EAX, MemoryAddress ; EAX stores the input, which is the requested memory address
; Passing the parameters to SearchSets
call SearchSets
; Getting the results from SearchSets
; Passing the parameters to SearchTags
call SearchTags
; Getting the results from SearchTags
; if cache hit
call CacheHit
; if cache miss
call CacheMiss
jmp PrintCacheHits
```

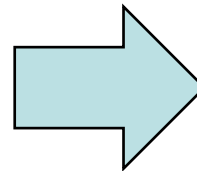
HW3 Programming Assignment Ex2



- An example to show how to use subroutine:

Use **registers** to pass parameters

```
mov EAX, 6  
mov EBX, 10  
jmp addition
```



```
mov EAX, 6  
mov EBX, 10  
call addition  
; ECX stores the result  
call next_function
```

```
addition:  
...  
; ECX stores the result  
jmp next_function
```

```
addition proc  
...  
ret  
addition endp
```

HW3 Programming Assignment Ex2



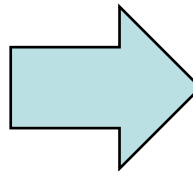
- An example to show how to use subroutine:

Use **processor stack** to pass parameters

```
mov EAX, 6  
mov EBX, 10  
jmp addition
```

addition:

```
...  
; ECX stores the result  
jmp next_function
```



```
push 6  
push 10  
call addition  
; ECX stores the result  
add ESP, 8  
call next_function
```

```
addition proc  
...  
ret  
addition endp
```

Summary



- Review of Direct & Associative Mapping
- Set-Associative Mapping
- Hints for Set-Associative Mapping Implementation
- Hints for Subroutine Implementation