TED ÜNİVERSİTESİ

CMPE361 Computer Organization

Department of Computer Engineering
TED University- Fall 2023

Memory Systems 2- Caches

These Slides are mainly based on slides of the text book (downloadable from the book’s website).

Cache

- Highest level in memory hierarchy
- Fast (typically ~ 1 cycle access time)
- Ideally, not practically, supplies most data to processor
- Usually holds most recently accessed data

Cache Design Questions

- What data is held in the cache?
- How is data found?
- What data is replaced?

What data is held in the cache?

- Ideally, cache control should anticipate needed data and put it in cache
  - But impossible to predict future!
- However, we can use past to predict future – make use of temporal and spatial locality:
  - **Temporal locality:** copy newly accessed data into cache
  - **Spatial locality:** copy neighboring data into cache too

Cache Terminology

- **Capacity (*C*):**
  - number of data bytes in cache
- **Block size (*b*):**
  - bytes of data brought into cache at once
  - İn other words, block size is unit of data transport to/from cache
- **Number of blocks in cache (*B = C/b*):**
- **Degree of associativity (*N*):**
  - number of blocks in a set
- **Number of sets in cache (*S = B/N*):**
  - each memory address maps to exactly one cache set

How is data found?

- Each memory address maps to exactly one set
- Caches are categorized by # of blocks in a set:
  - **Direct mapped:** 1 block per set
  - ***N*-way set associative:** *N* blocks per set
  - **Fully associative:** all cache blocks are in 1 set

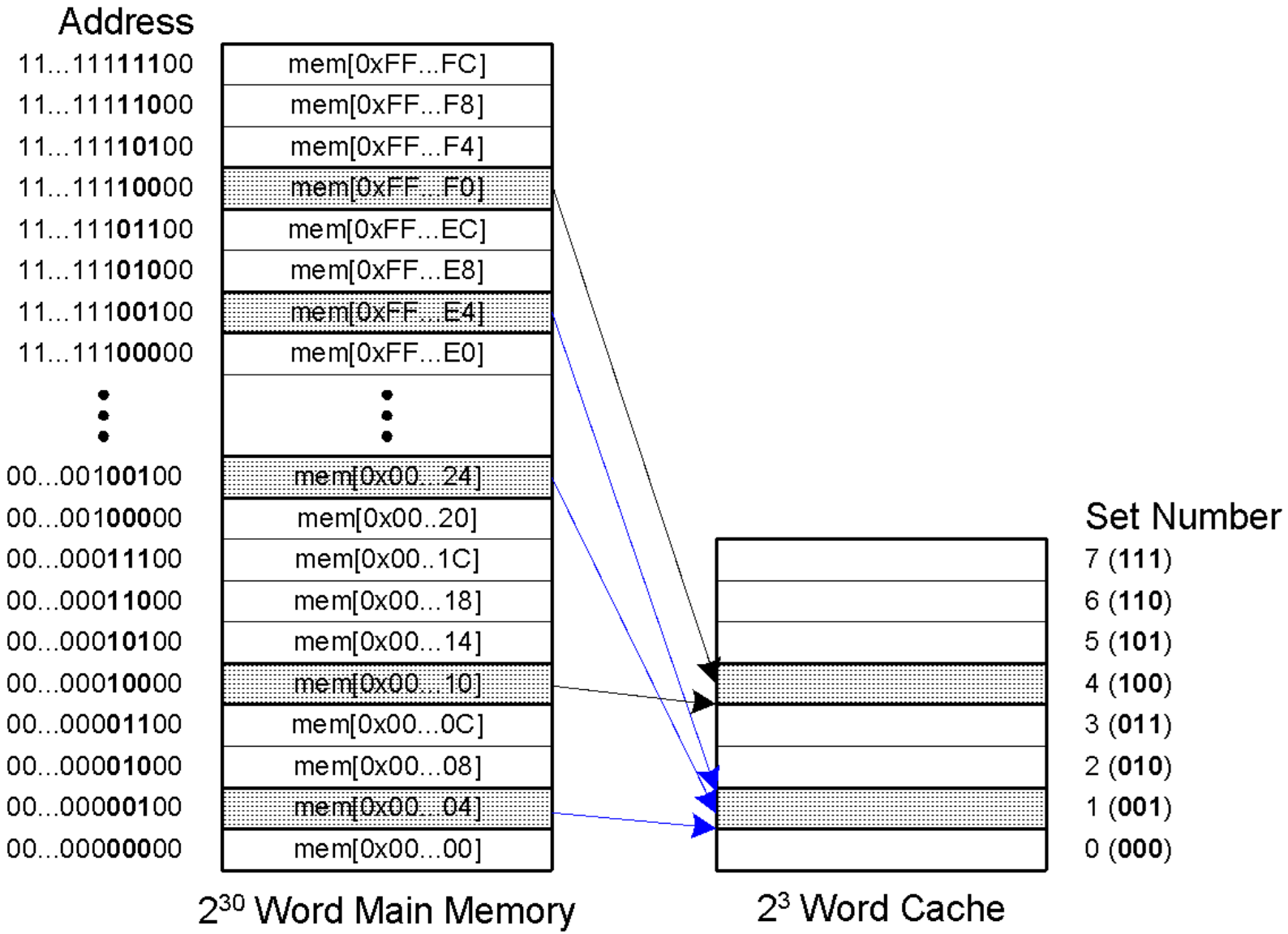
Example Cache

- A cache with:
  - Capacity C = 8 words
  - Block size b = 1 word
  - So, number of blocks B = 8

  is given

(Ridiculously small, but will illustrate organizations)

Direct Mapped Cache



- Each set has only one block.
- Memory is word (=4 bytes) alligned i.e. Valid addresses are multiple of 4.
- 2 LSB are always zero.
- Next $\log_2 S = \log_2 8 = 3$ bits of memory address are used to determine the set of cache to which a block brought from memory is mapped.
- As there are only 8 sets in cache (each one with one block), block 0 and 8 brought from memory map to same set.

Example

- To what cache set does the word at address 0x00000014 map?
  - to set 5
- Name another address that maps to the same set:
  - any adress. With bits 4-2 maches 5 (b’101) will map to the same set.
  - for example, address 0xFFFFFFF4 also maps to set 5.
- Words at addresses 0x34, 0x54, 0x74, . . . , 0xFFFFFFF4 map to set 5.

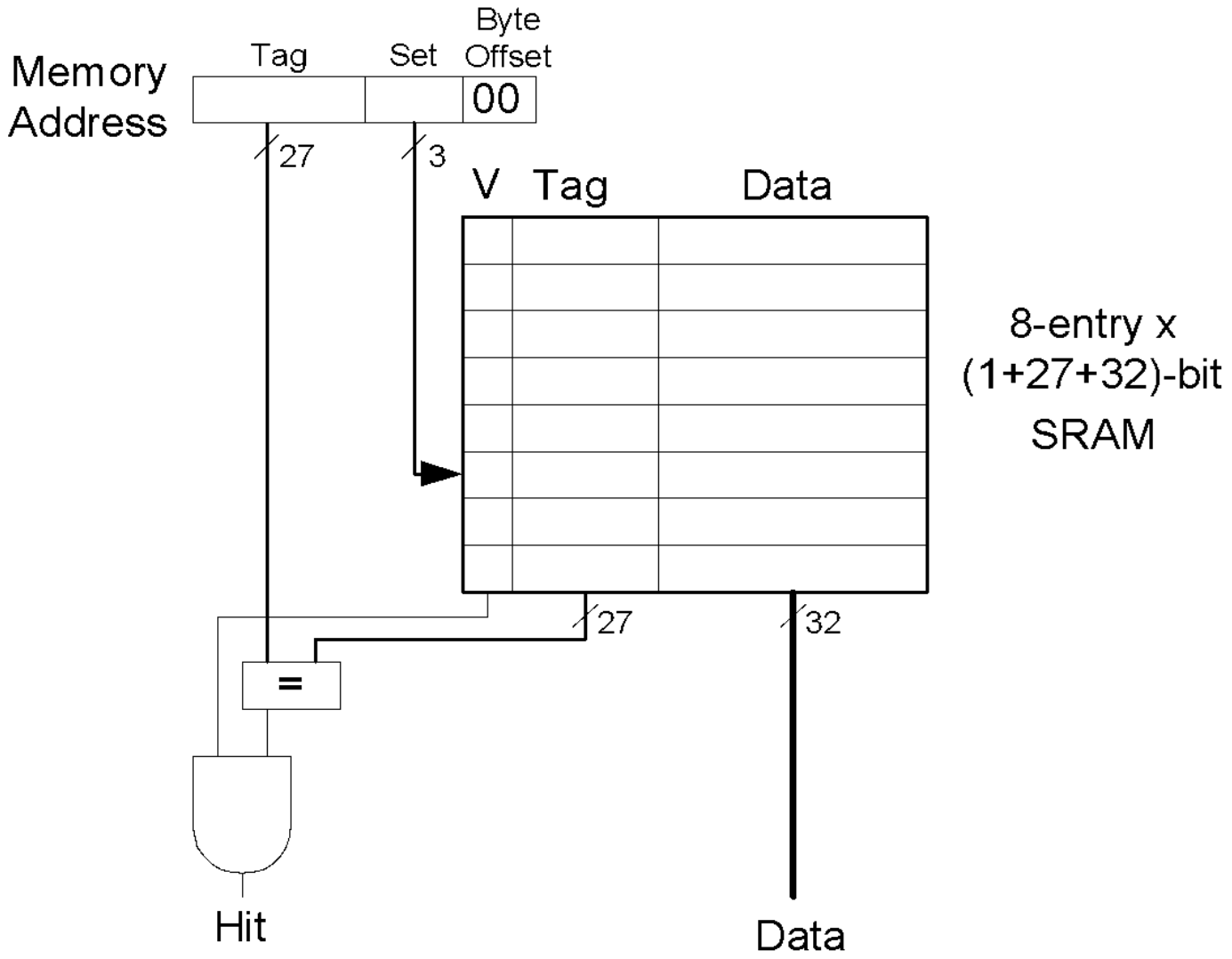
Cache fields, considering the example

- In a Word alligned and byted-addressable memory such as MIPS memory
  - Byte offset: 2 lsb of a memory address
  - Set bits: Next three $\log_2 S$ bits (3), where S=8)
  - Tag bits: remaining bits (27 bits) of memory address.
    - Tage bits indicate which of the many possible addresses is held in that set.

Example 2

- Find the number of set and tag bits for a direct mapped cache with 1024 (2^{10}) sets and a one-word block size, with 32 bits address.
- # Set bits=10
- # Number of sets= 1024
- # Tag bits= 32-10 -2= 20 bits.

Direct Mapped Cache Hardware



How does the cache operates?

- Ignore 2 least significant bits
- Go the the set the memory address mapped, usıng next 3 bits
- Check if the cache entry is valid
- Check if the tag fields in the cache set and memory match,
- If match is true, data is in the cache, retrieve the data; otherwise it is a cache miss; retrieve the data from the memory

Direct Mapped Cache Performance example

```
# MIPS assembly code

        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)
        lw   $t2, 0xC($0)
        lw   $t3, 0x8($0)
        addi $t0, $t0, -1
        j    loop
done:
```

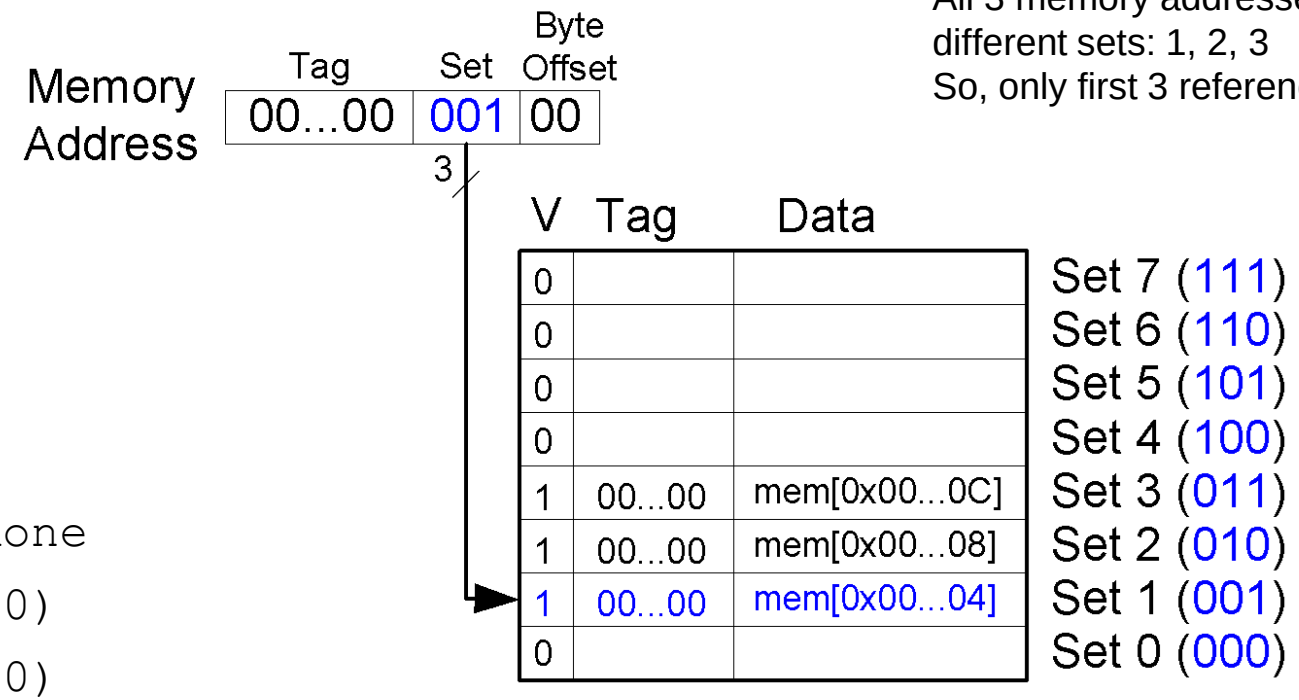
What is Content of the Cache?

What is the Miss Rate ?

Direct Mapped Cache Performance

```
# MIPS assembly code

        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)
        lw   $t2, 0xC($0)
        lw   $t3, 0x8($0)
        addi $t0, $t0, -1
        j    loop
done:
```



All 3 memory addresses matche
different sets: 1, 2, 3
So, only first 3 references cause misi

No miss after the first load:
Miss Rate = 3/15 = 20%

Compulsory Misses only!

Example 3: Direct Mapped Cache: Conflict

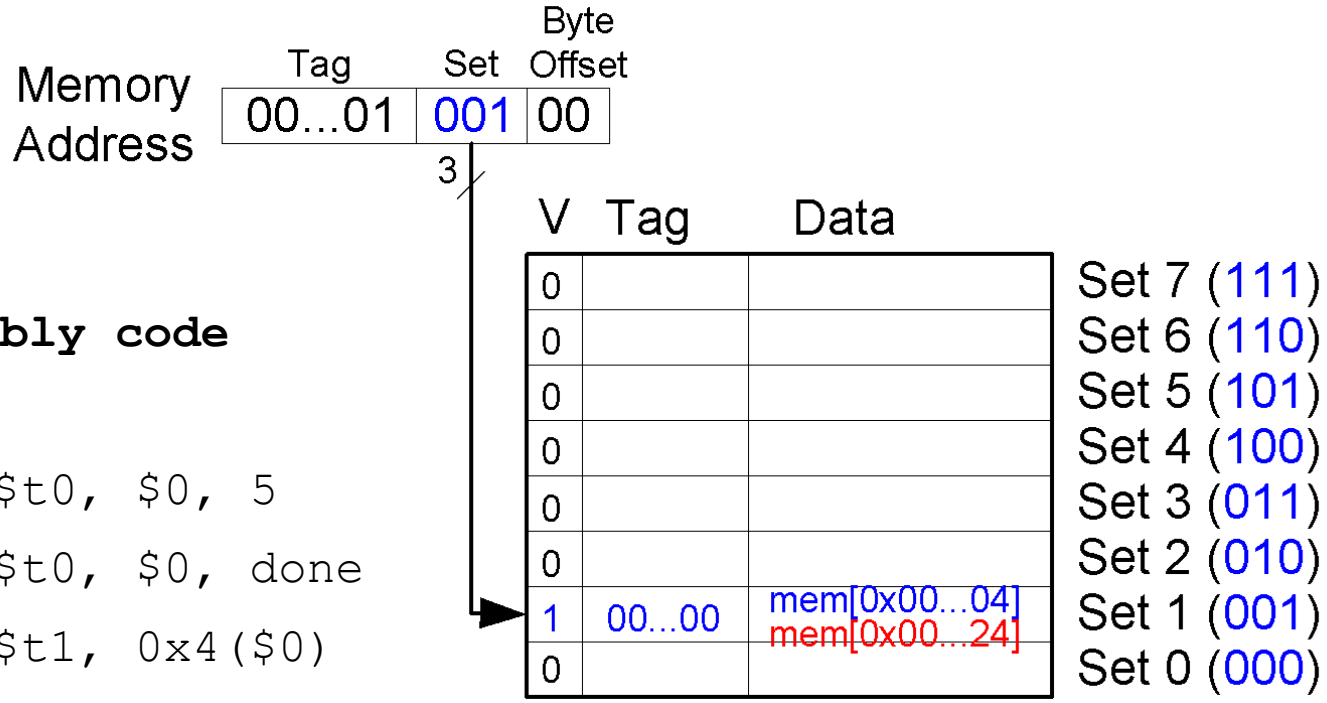
```
# MIPS assembly code

        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)
        lw   $t2, 0x24($0)
        addi $t0, $t0, -1
        j    loop
done:
```

Content of Cache=?

Miss Rate = ?

Direct Mapped Cache: Conflict




```
# MIPS assembly code

        addi $t0, $0, 5
loop:   beq  $t0, $0, done
        lw   $t1, 0x4($0)
        lw   $t2, 0x24($0)
        addi $t0, $t0, -1
        j    loop
done:
```

Miss Rate = 10/10

= 100%

Conflict Misses

- How to reduce conflicts?