EECE 2322: Fundamentals of Digital Design and Computer Organization

Lecture 15_2: Microarchitecture

Xiaolin Xu
Department of ECE
Northeastern University

Cache

- Highest level in memory hierarchy
- Fast (typically ~ 1 cycle access time)
- Ideally 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?

Focus on data loads, but stores follow same principles

What data is held in the cache?

- Ideally, cache anticipates needed data and puts it in cache
- But **impossible to predict future**
- Use past to predict future – 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
- **Number of blocks (*B = C/b*):**
  - number of blocks in cache: $B = C / b$
- **Degree of associativity (*N*):**
  - number of blocks in a set
- **Number of sets (*S = B/N*):**
  - each memory address maps to exactly one cache set

How is data found?

- Cache organized into *S* sets
- Each memory address maps to exactly one set
- Caches 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 in 1 set
- Examine each organization for a cache with:
  - Capacity ($C = 8$ words)
  - Block size ($b = 1$ word)
  - So, number of blocks ($B = 8$)

Example Cache Parameters

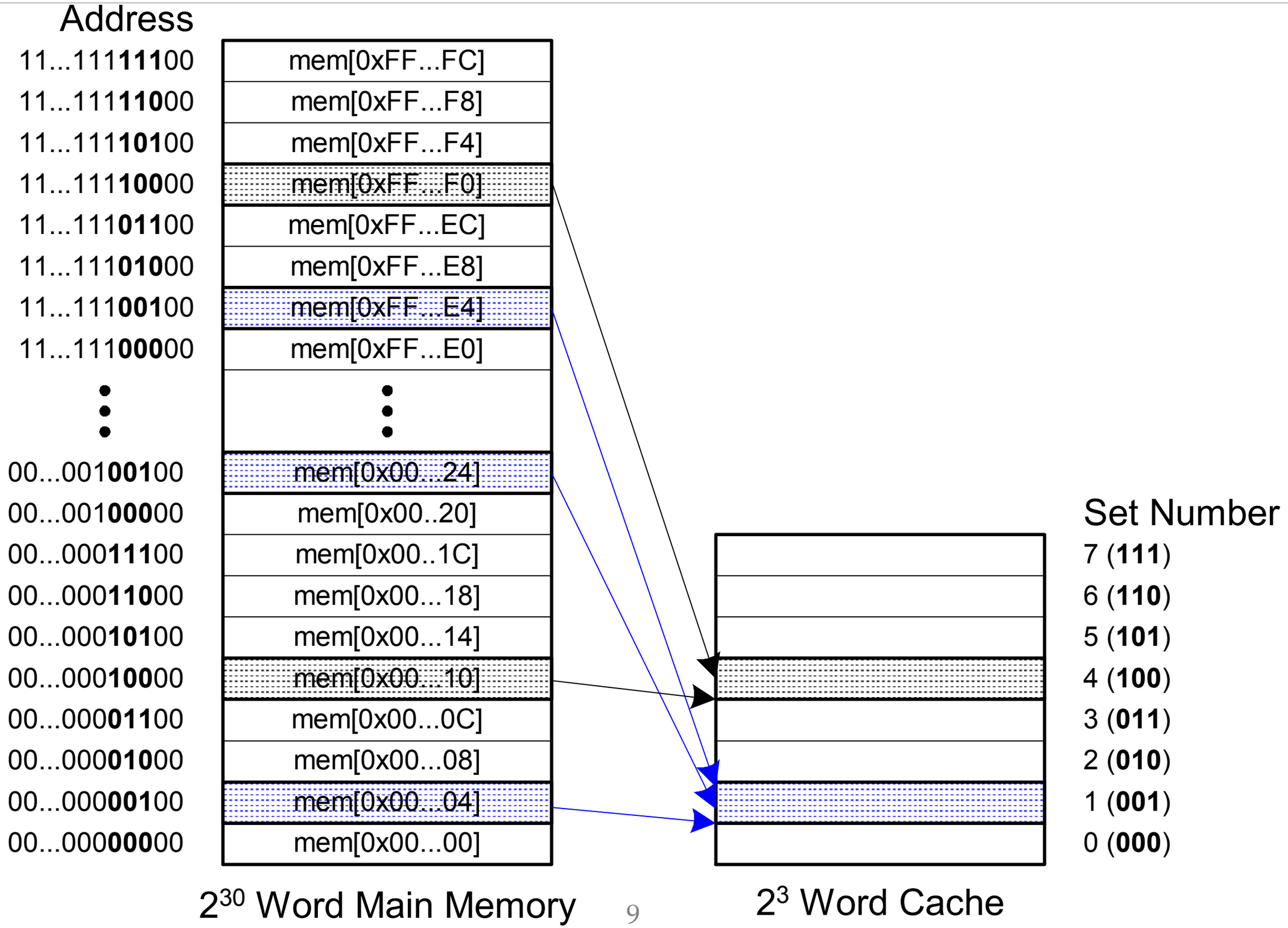
- ***C* = 8** words (capacity)
- ***b* = 1** word (block size)
- So, ***B* = 8** (# of blocks)

Ridiculously small, but will illustrate organizations

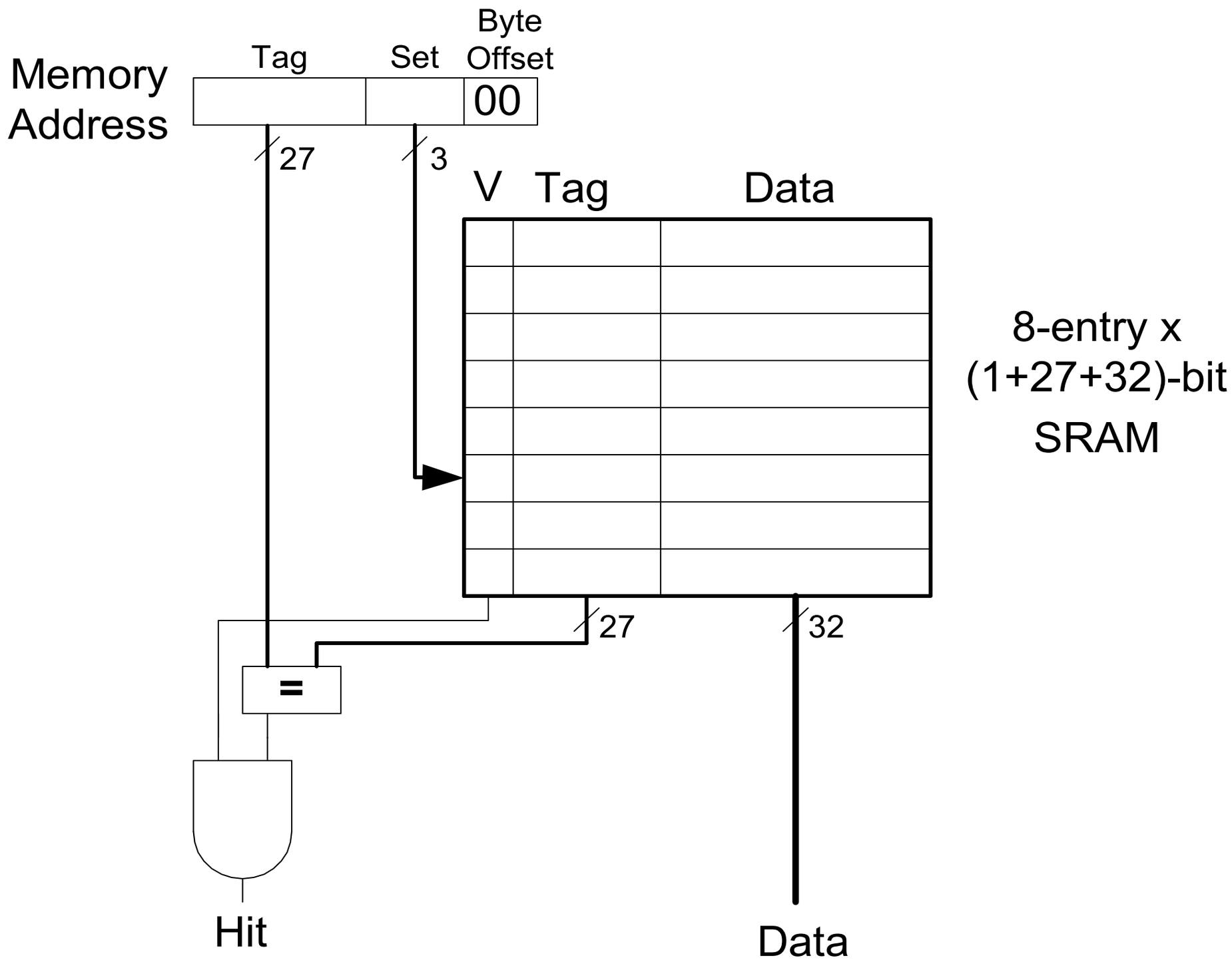
Mapping

- Mapping: the relationship **between the address of data in main memory and the location of that data in the cache**
- Each memory address maps to exactly one set in the cache
  - Some of the address bits are used to determine which cache set contains the data

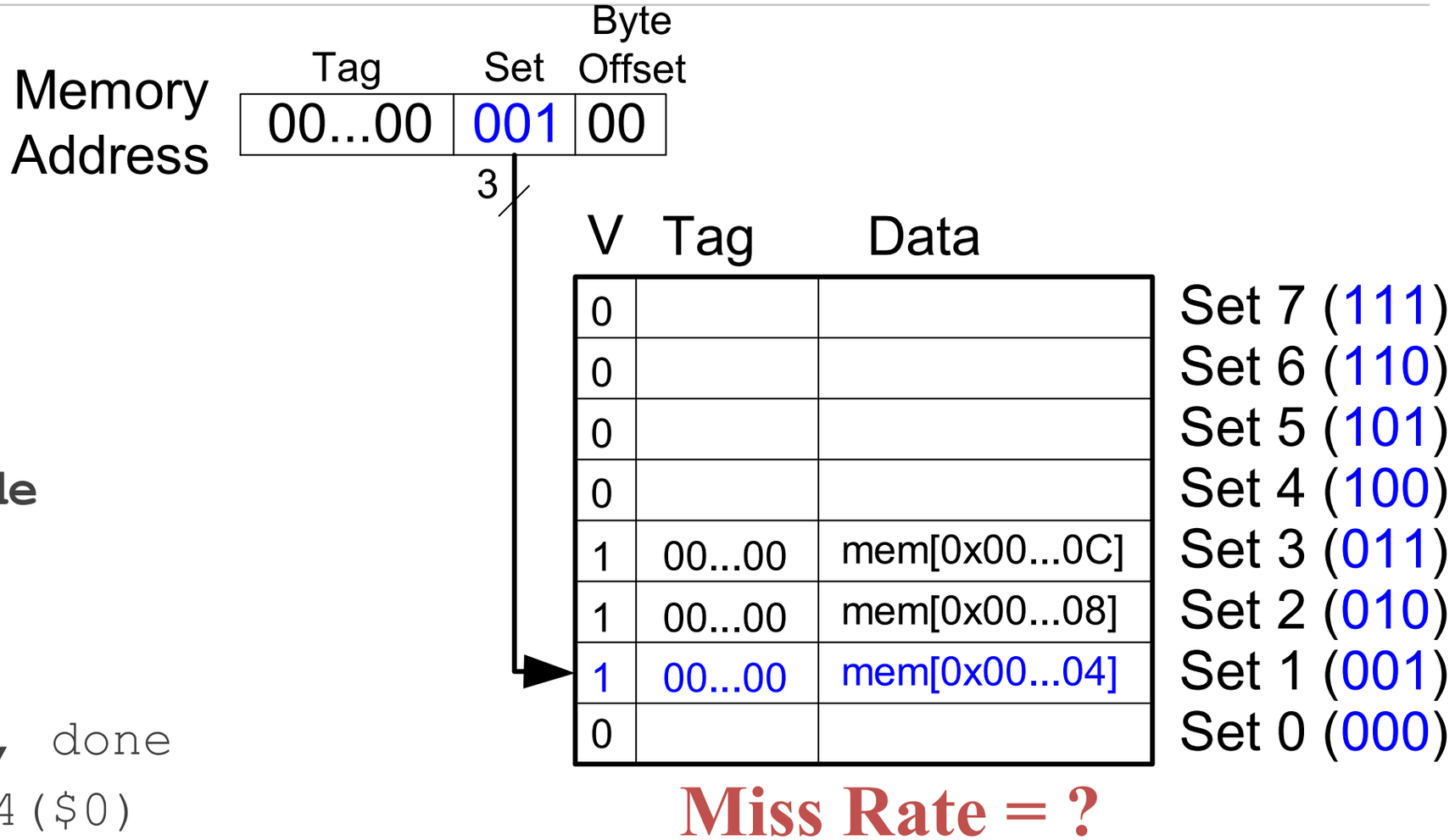
Direct Mapped Cache



Direct Mapped Cache Hardware



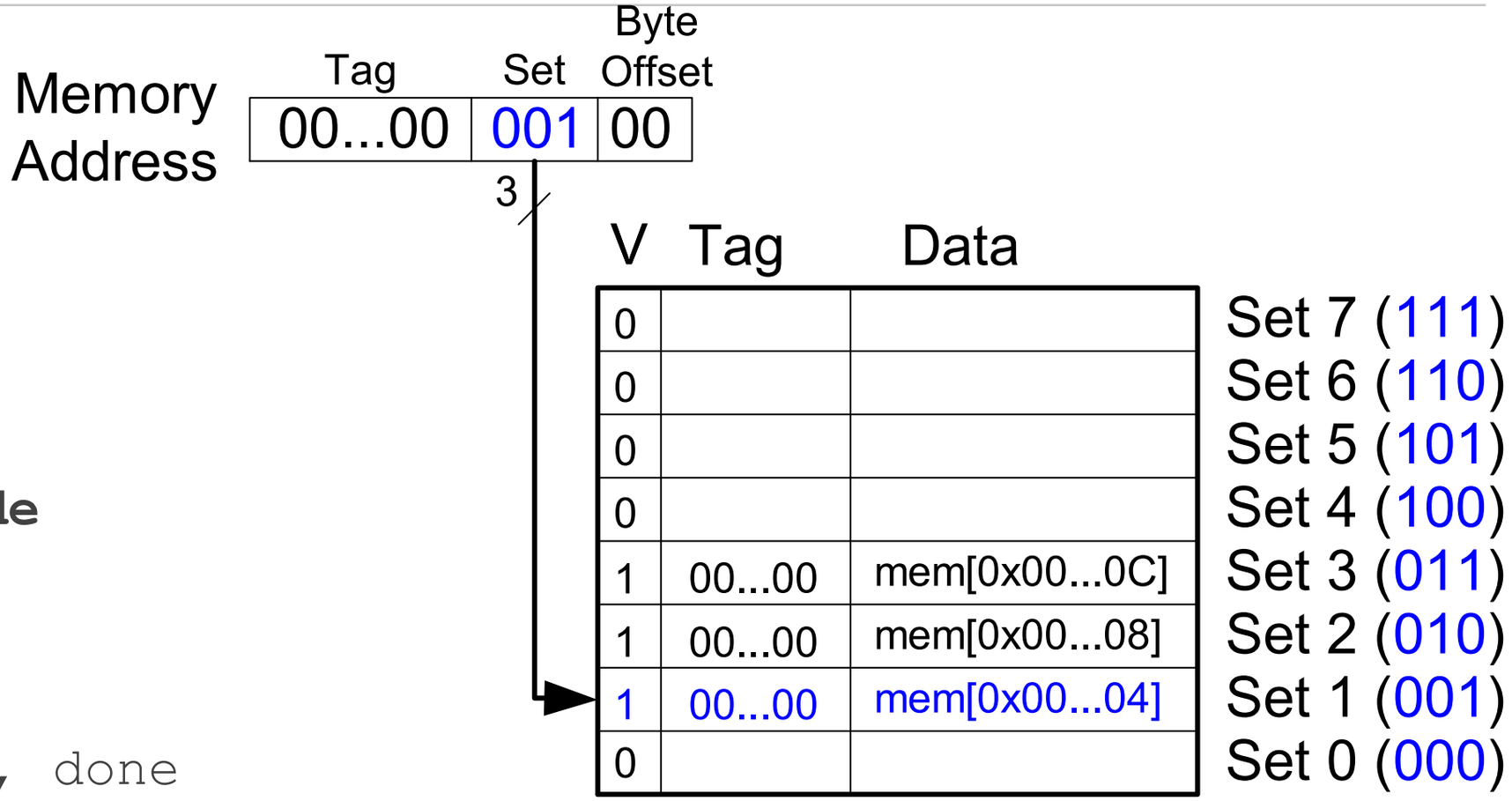
Direct Mapped Cache Performance



Miss Rate = ?

```
# 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:
```

Direct Mapped Cache Performance



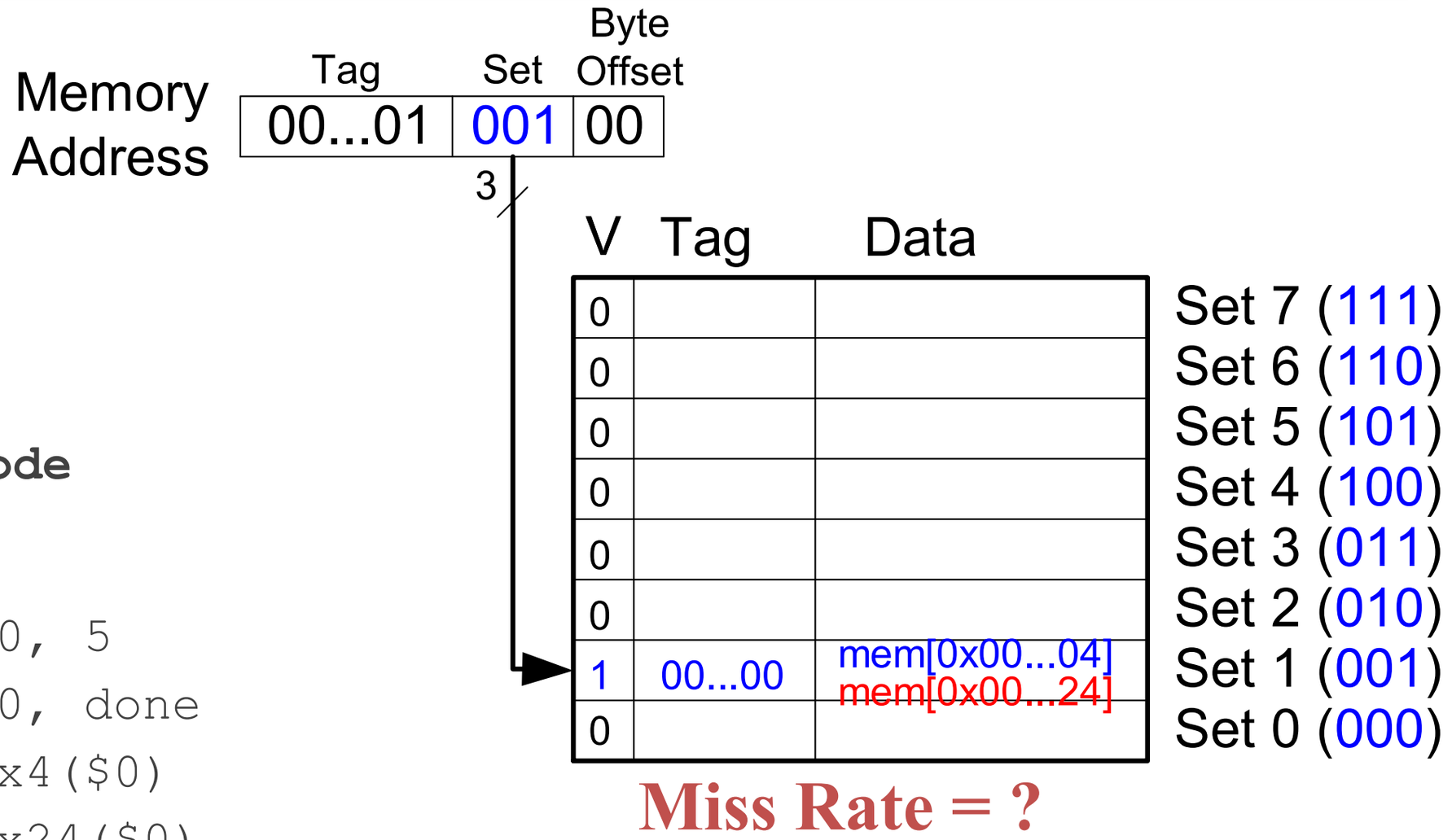

```
# MIPS assembly code


    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:
```

Miss Rate = 3/15
= 20%

Temporal Locality
Compulsory Misses

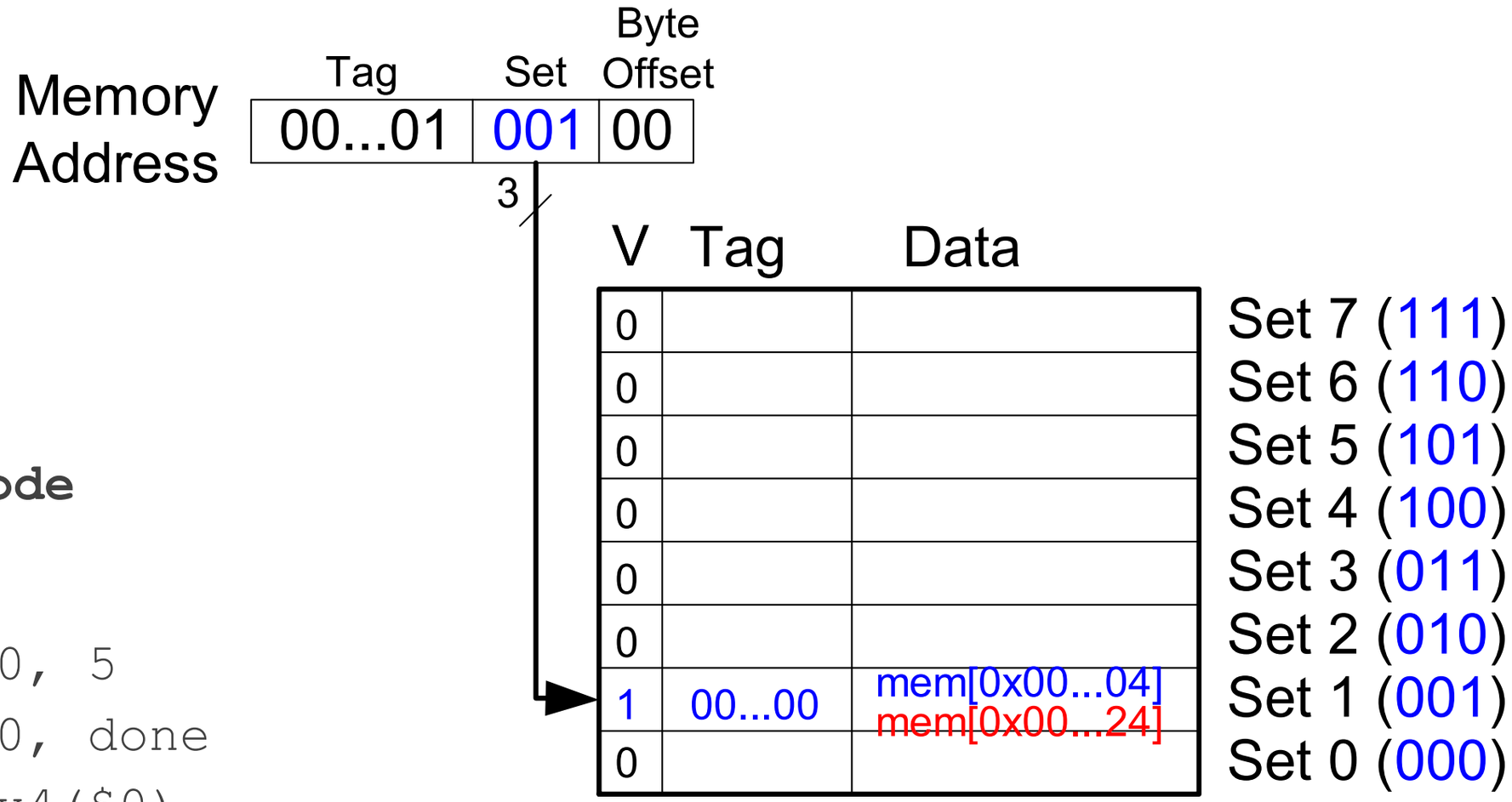
Direct Mapped Cache: Conflict

```
# 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 = ?

Direct Mapped Cache: Conflict



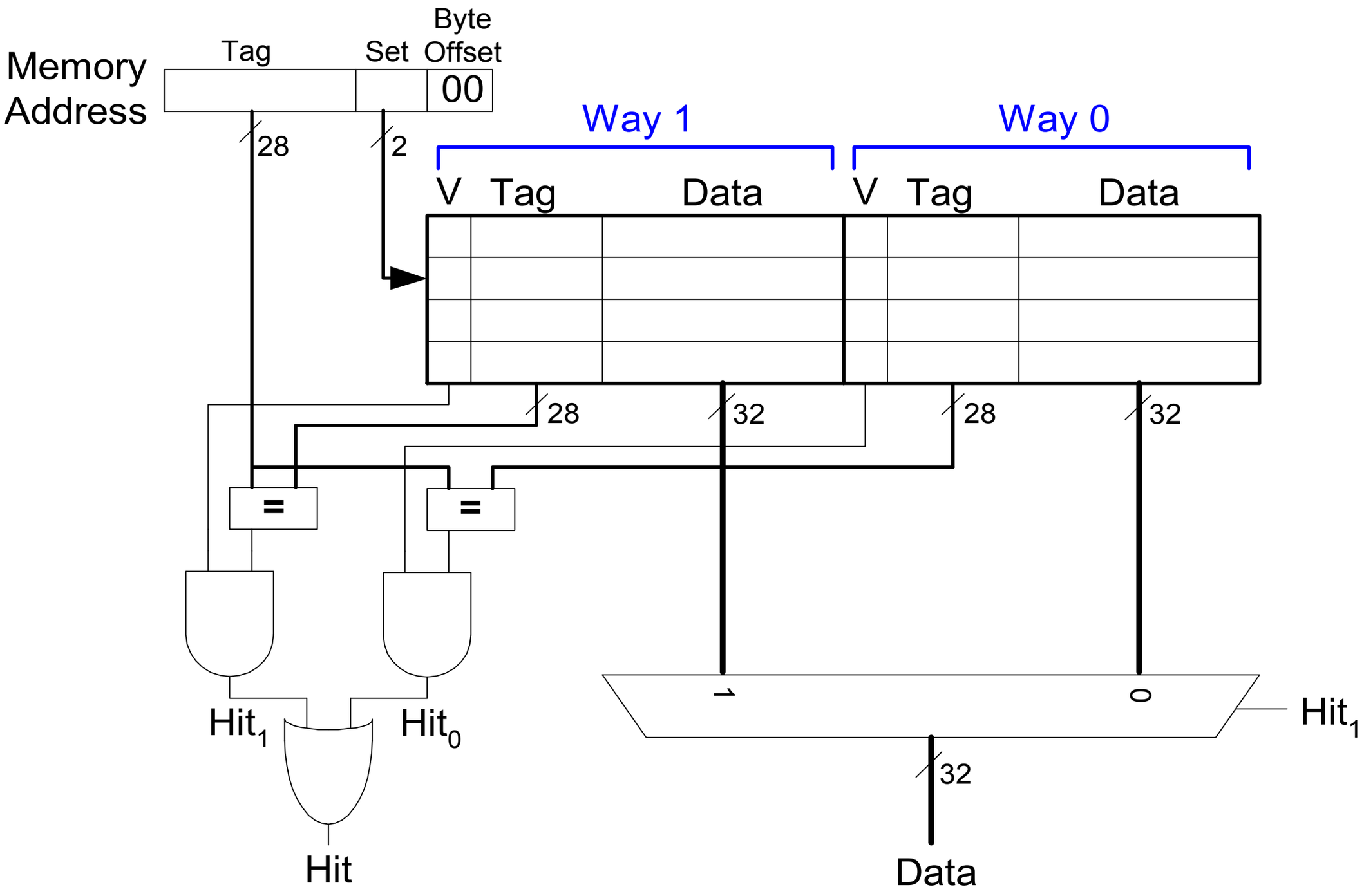

```
# MIPS assembly code


        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

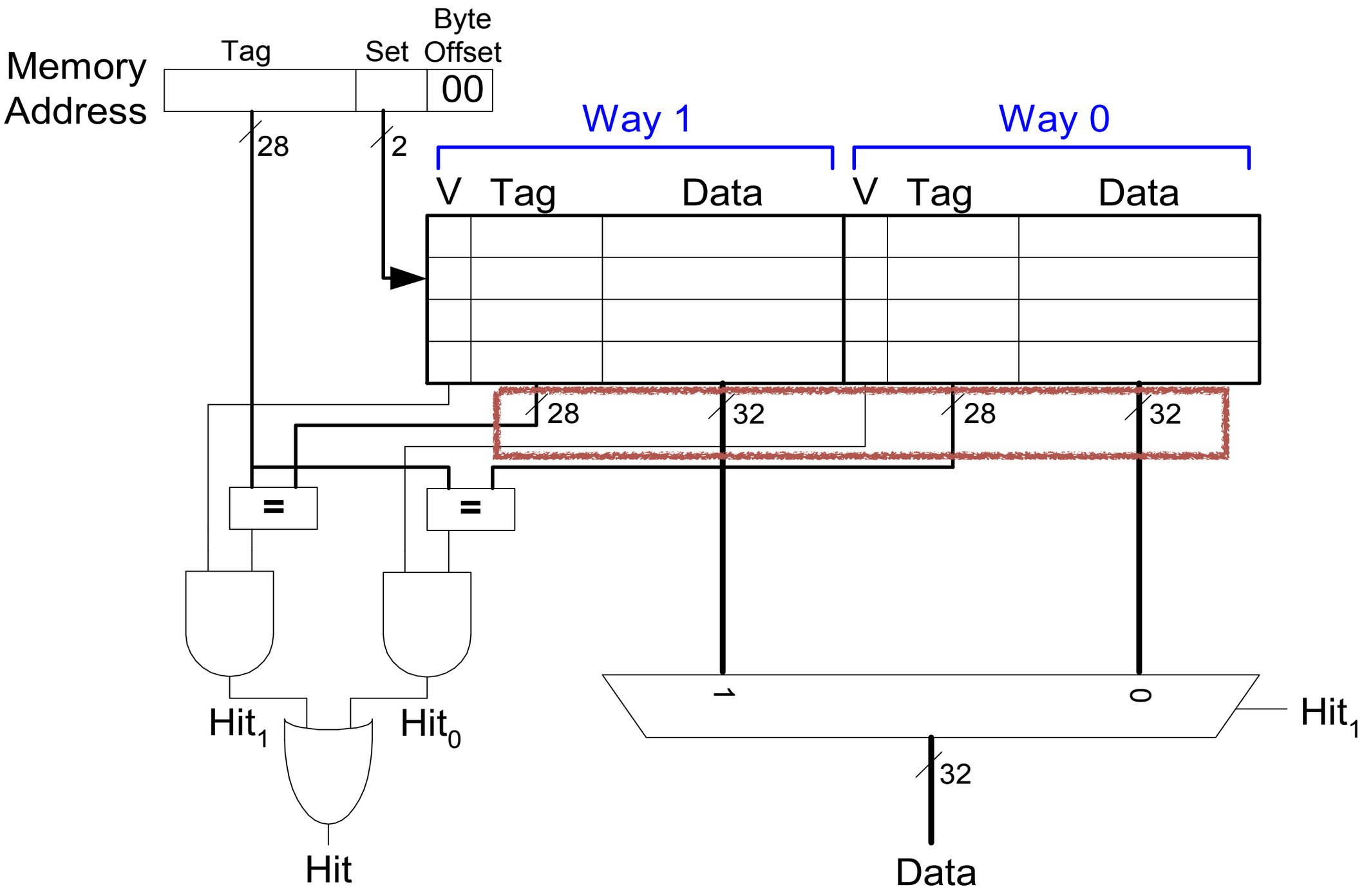
= 100%

Conflict Misses

N-Way Set Associative Cache



N-Way Set Associative Cache

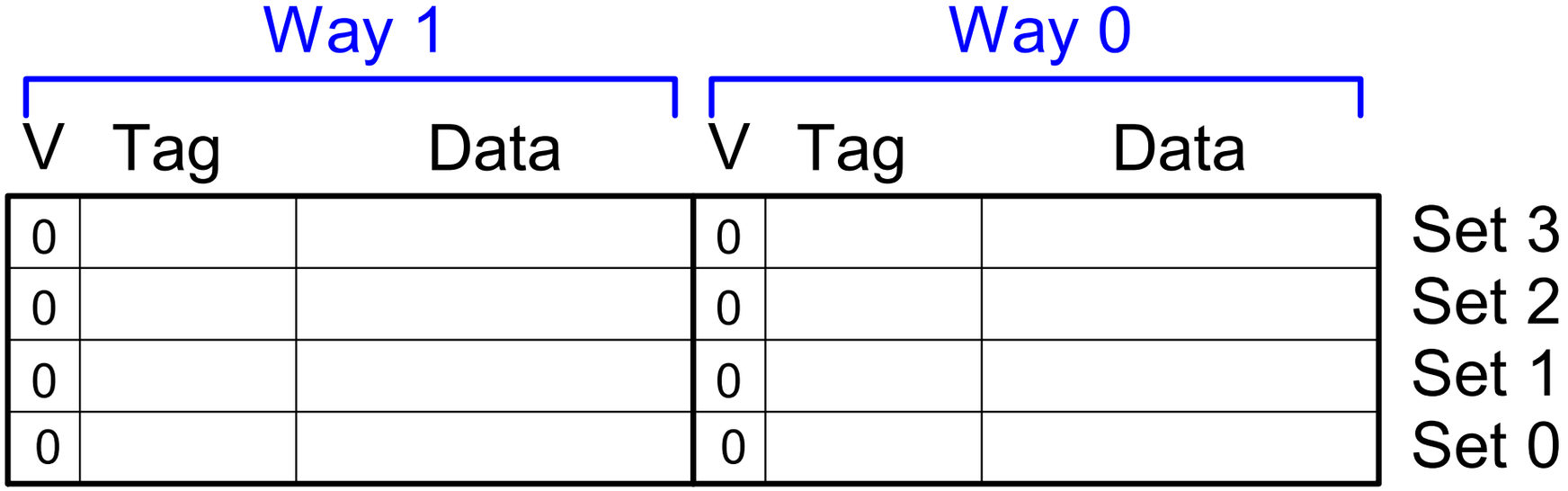


N-Way Set Associative Performance

```
# MIPS assembly code
```

Miss Rate = ?

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



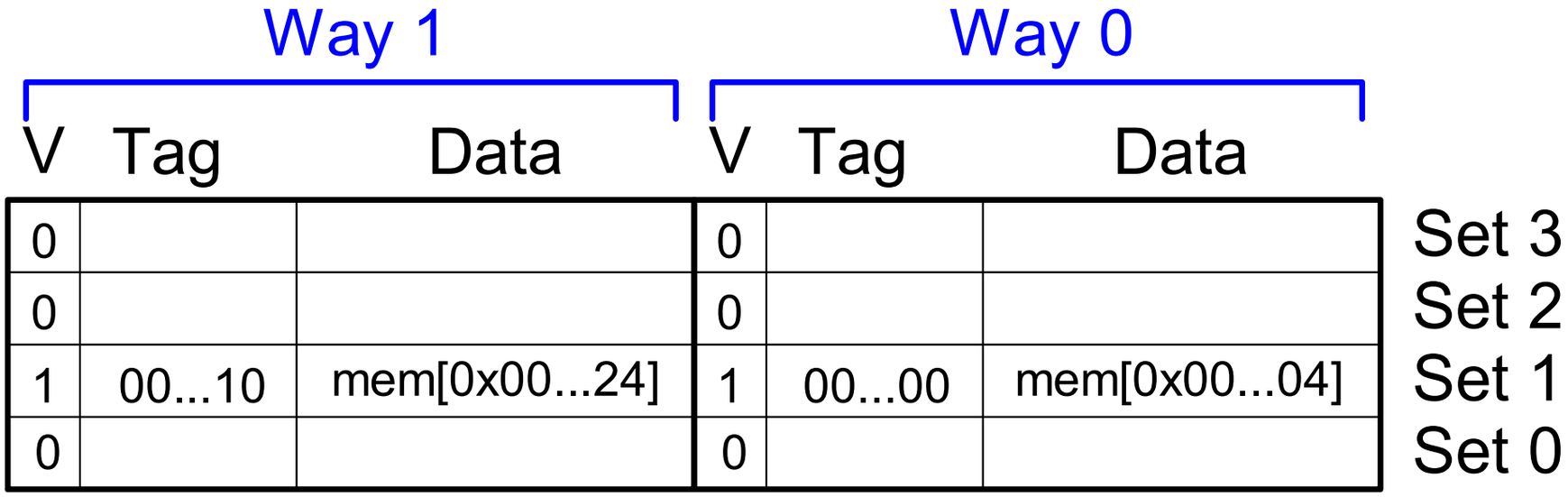
N-Way Set Associative Performance

```
# 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:
```

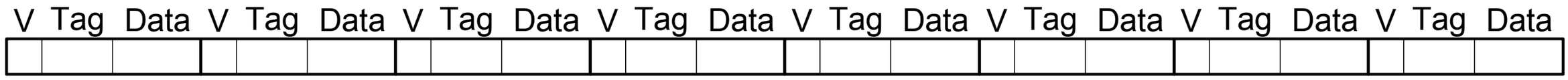
Miss Rate = 2/10

= 20%

Associativity reduces conflict misses



Fully Associative Cache

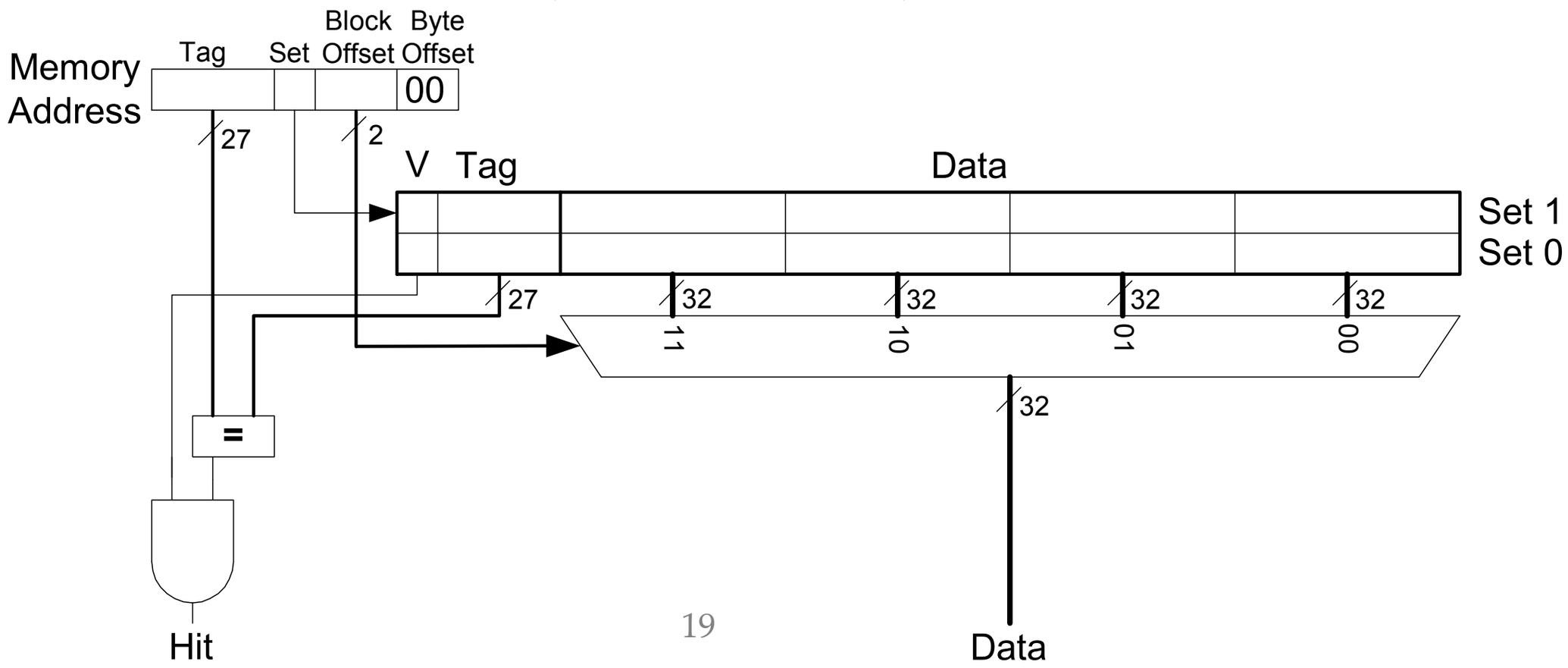


Reduces conflict misses

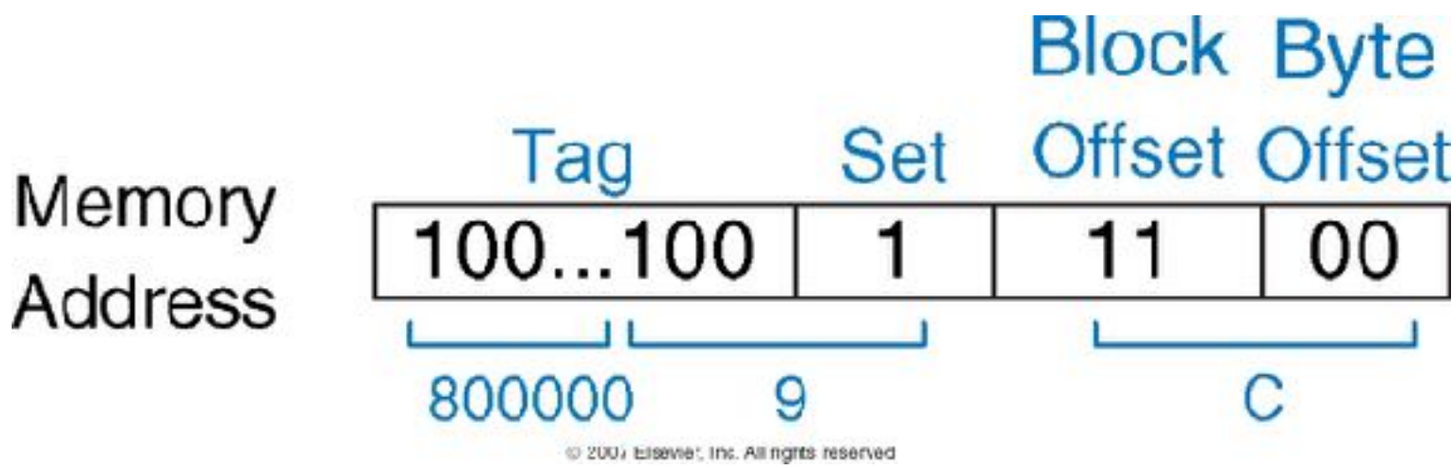
Expensive to build

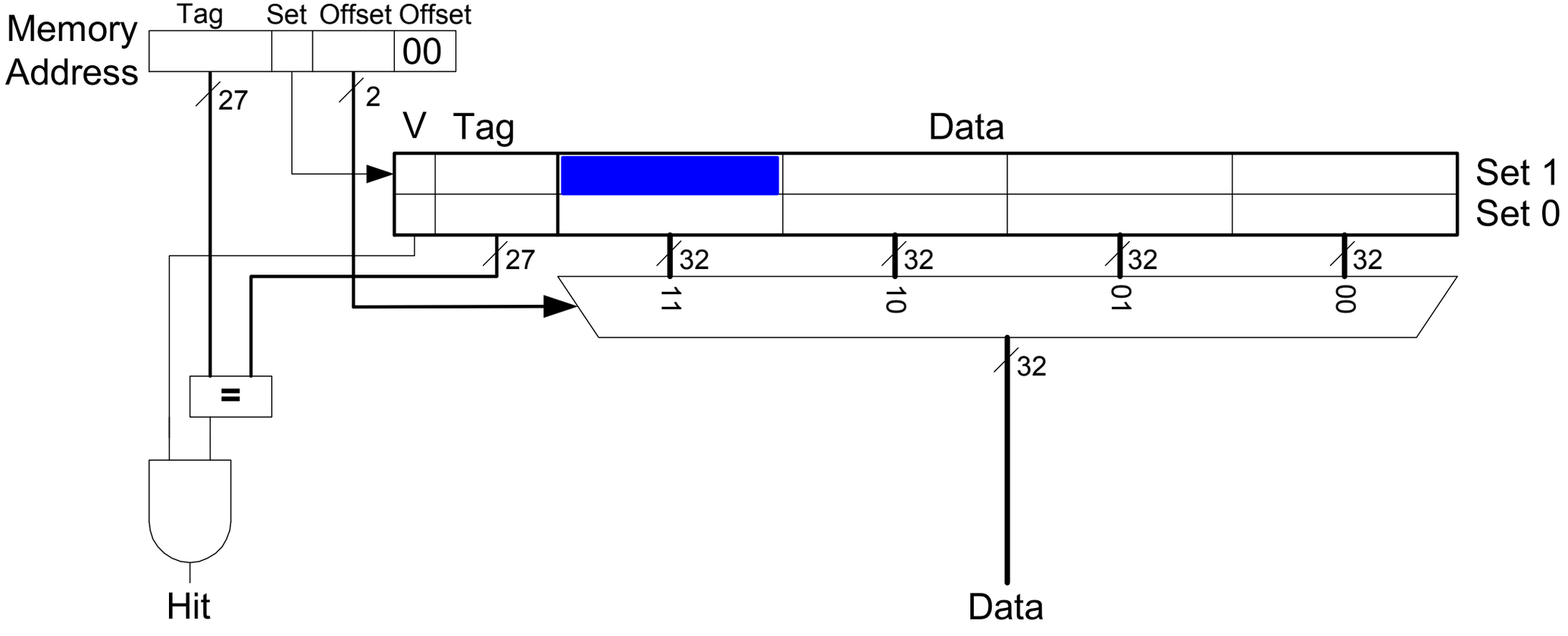
Spatial Locality?

- Increase block size:
  - Block size, ***b* = 4 words**
  - *C* = 8 words
  - Direct mapped (1 block per set)
  - Number of blocks, ***B* = 2** ($C/b = 8/4 = 2$)



Cache with Larger Block Size





Direct Mapped Cache Performance

Miss Rate = ?

```
    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:
```

Direct Mapped Cache Performance

```
    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:
```

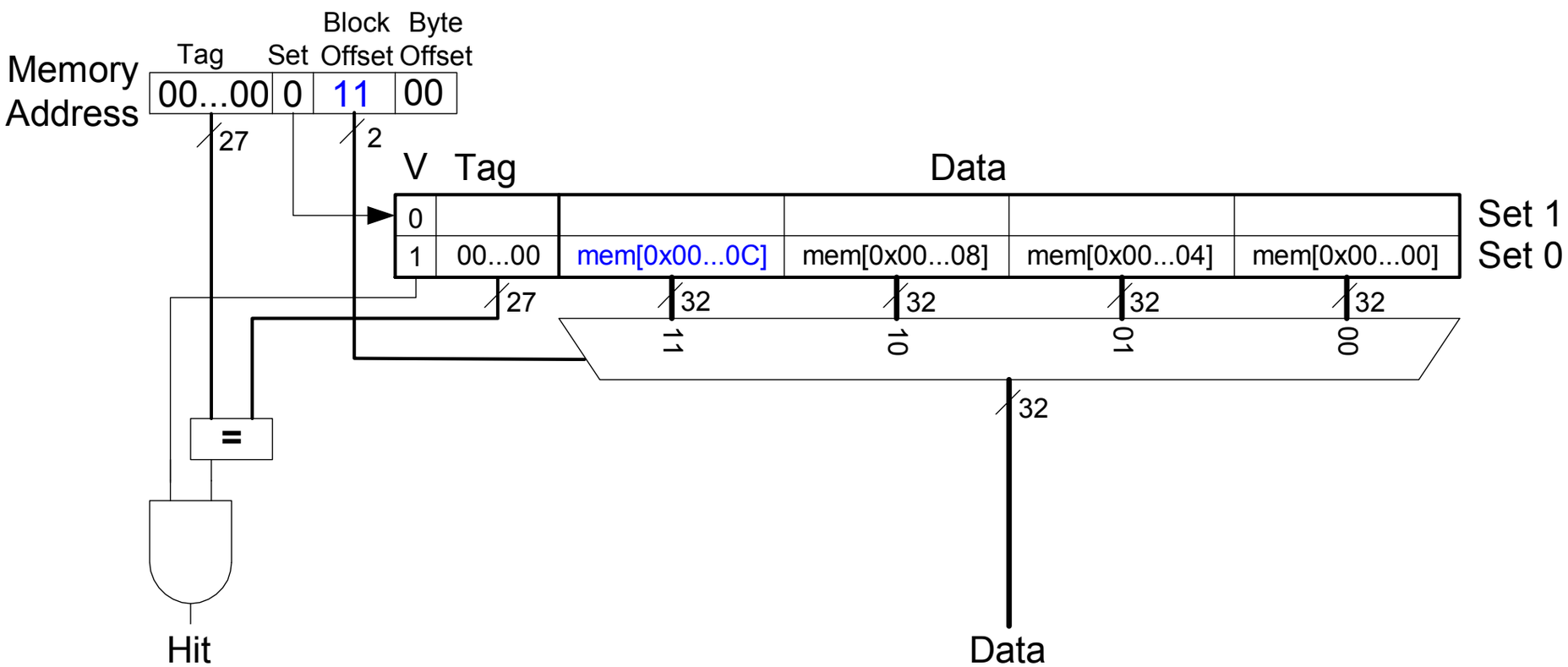
Miss Rate = 1/15

= 6.67%

Larger blocks

reduce compulsory misses

through spatial locality



EECE 2322: Fundamentals of Digital Design and Computer Organization

Lecture 15_2: About the Final Exam

Xiaolin Xu
Department of ECE
Northeastern University

About the Final Exam

- Q1: Numbers and Digital Gates

About the Final Exam

- Q1: Numbers and Digital Gates
  - You will be asked to do the conversion in between different data format, such as binary, hexadecimal, etc

About the Final Exam

- Q1: Numbers and Digital Gates
  - You will be asked to do the conversion in between different data format, such as binary, hexadecimal, etc
  - You will be asked to formulate the function of a given circuit, and express it using Boolean expression

About the Final Exam

- Q2: FSM — The timing diagram and function of an FSM will be provided.

About the Final Exam

- Q2: FSM — The timing diagram and function of an FSM will be provided.
  - You will be asked to draw the **state transition diagram** of this FSM

About the Final Exam

- Q2: FSM — The timing diagram and function of an FSM will be provided.
  - You will be asked to draw the **state transition diagram** of this FSM
  - You will be asked to design this FSM using Verilog — make sure to show the **state encoding**

About the Final Exam

- Q2: FSM — The timing diagram and function of an FSM will be provided.
  - You will be asked to draw the **state transition diagram** of this FSM
  - You will be asked to design this FSM using Verilog — make sure to show the **state encoding**
    - Verilog coding will be evaluated here!

About the Final Exam

- Q3: Basic concepts of MIPS ISA

About the Final Exam

- Q3: Basic concepts of MIPS ISA
  - Instruction types: R, I, J

About the Final Exam

- Q3: Basic concepts of MIPS ISA
  - Instruction types: R, I, J
    - The composition of each instruction and how they interact with the RF and memory

About the Final Exam

- Q4: MIPS Hazard Handling

About the Final Exam

- Q4: MIPS Hazard Handling
  - Data hazard

About the Final Exam

- Q4: MIPS Hazard Handling
  - Data hazard
  - Control Hazard

About the Final Exam

- Q4: MIPS Hazard Handling
  - Data hazard
  - Control Hazard
- How many methods have been introduced to handle, you should be familiar

About the Final Exam

- Q4: MIPS Hazard Handling
  - Data hazard
  - Control Hazard
- How many methods have been introduced to handle, you should be familiar
  - What behind is understanding the pipeline for different instructions, and their dependency!

About the Final Exam

- Q4: MIPS Hazard Handling
  - Data hazard
  - Control Hazard
  - How many methods have been introduced to handle, you should be familiar
    - What behind is understanding the pipeline for different instructions, and their dependency!
- **Good news — No need to draw any schematic!**

About the Final Exam

- Q5: MIPS and Program Language Translation

About the Final Exam

- Q5: MIPS and Program Language Translation
  - Given a code snippet, translate it into MIPS assembly language

About the Final Exam

- Q5: MIPS and Program Language Translation
  - Given a code snippet, translate it into MIPS assembly language
  - **Should be careful, while there is need to stack some registers!**

About the Final Exam

- Q5: MIPS and Program Language Translation
  - Given a code snippet, translate it into MIPS assembly language
  - **Should be careful, while there is need to stack some registers!**
  - **What will happen if some mechanisms fail, say stack?**

About the Final Exam

- Q6: Microarchitecture
  - Understand the schematic and hardware compositions of single cycle, multi-cycle, pipeline
    - RF, Memory, ALU
    - Encoding, Decoding, etc
  - Under stand the performance parameters, and be able to calculate

Thanks!

- A very good, happy, and productive semester!
- Wish everyone the best in your future courses, finals, and career!