

# Computer Architecture (Lab). Week 7

Vladislav, Artem, Hamza, Manuel

Innopolis University

*v.ostankovich@innopolis.ru*

*a.burmyakov@innopolis.ru*

*m.rodriguez.osuna@innopolis.university*

*h.salem@innopolis.university*

October 15, 2020

- MIPS Arithmetic Operations + MARS

# Topic of the lab

- Comparison of numbers
- Jumps and branches
- Loops
- Functions
- Working with memory

## Comparison: *slt*, *slti*

- *slt* - set 1 if less than (R-type)
- *slti* - set 1 if less than immediate (I-type)
- Usage

```
slt  $d, $s, $t           # $d is one if $s less than $t
slti $t, $s, immediate    # $t is one if $s less than immediate
```

# Jump

- jal: Jump and link
- j: Unconditional jump
- jr: Jump to register

# Branch Instruction

**Branch** - conditional change of the control flow (“conditional jump”).

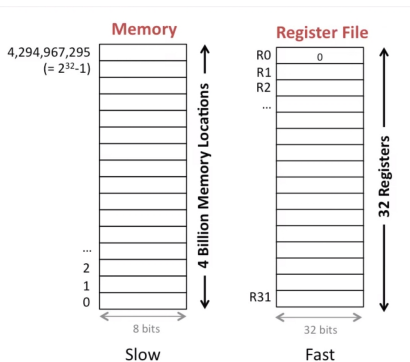
Instructions:

- **BEQ** - Branch on equal
- **BGEZ** - Branch on greater than or equal to zero
- **BGTZ** - Branch on greater than zero
- **BNE** - Branch on not equal
- ...branch and link and others...

# Working with Memory

Main concepts:

- **Load:** load data to register file from memory
- **Store:** store data to memory from register file

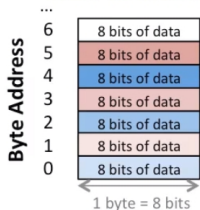


Difference of two types of memory. Figure by David B

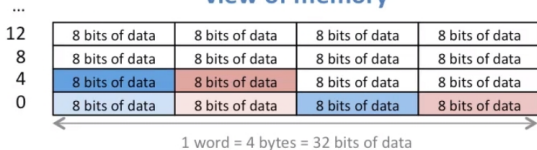
# Byte-aligned vs Word-aligned Memory

- MIPS memory is byte-addressable, but registers are a word-size (4 bytes)
- Word addresses are always divisible by 4

## Byte-addressable view of memory



## Word-aligned view of memory



Difference of two types of memory. Figure by David B



# Memory Instructions

- **lb/lw** = Load byte/word

Usage:

- lw register\_destination, RAM\_source
- lb register\_destination, RAM\_source

- **sb/sw** = Store byte/word

Usage:

- sw register\_source, RAM\_dest
- sb register\_source, RAM\_dest

## Example 1: Infinite Loop

```
.text
main:
    move $t0, $zero        # reset counter

loop:
    addi    $t0, $t0, 1      # Increase counter
    j loop    # jump backwards
```

## Example 2: Integers Subtraction Function

```
addi $t0, $zero, 10      # $t0 = 10
addi $t1, $zero, 20      # $t1 = 20
addi $t2, $zero, 30      # $t2 = 30
addi $t3, $zero, 40      # $t3 = 40

add $a0, $t0, $t1        # $a0 = 30
add $a1, $t2, $t3        # $a1 = 70

jal myFunction           # Jump to myFunction function
add $a0, $zero, $v0      # Save result: $a0 = -40

li $v0, 1                # Print integer
syscall

li $v0, 10               # Exit
syscall

myFunction:
    sub $v0, $a0, $a1    # $v0 = -40
    jr $ra              # Jump to the next instruction after function call
```

## Example 3: Conditional Loops

### Implementing a for-loop

```
.data
    array: .space 10      #allocate 10 consecutive bytes

.text
init:
    li $t3, 10           # Fill in number of desired iterations
    la $t0, array         # Load base address of array
    li $t1, 1             # Prepare constant to fill
fillarray:
    sb $t1, 0($t0)        # Store constant
    addi $t0, $t0, 1      # Shift to next cell of the memory
    addi $t3, $t3, -1     # Decrement counter
    bgtz $t3, fillarray   # Branch if > 0 or exit
```

## Example 4: Update Variable in Memory

```
.data
    var: .word 17    # declare a word-size variable "var"=17

.text
main:
    lw $t0, var      #load from RAM into register $t0: $t0 = var
    li $t1, 16        #Fill in $t1 with a constant value of 16
    sw $t1, var        #Update memory: var = 16
```

## Exercise 1: Signum of User Input

Implement a program that prompts a number from user and prints the result of signum function which is declared as:

$$\text{sgn}(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

Hints:

- Syscall 5 - read integer (\$v0 will contain a result)
- Syscall 1 - print integer

## Exercise 2 \*Optional

Write a program that will count the number of words in a string. By term “word” here is meant the sequence of chars that does not contain space symbols. String can be requested from user or declared in the .data section.

Hint: Space code in ASCII is  $32_{10} = 20_{16}$

## Useful Links

- <https://www.youtube.com/watch?v=rZev35tJaEY>
- <http://logos.cs.uic.edu/366/notes/mips%20quick%20tutorial.htm>
- <http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html>
- <http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html>



# Acknowledgements

- This lab was created and maintained by Vitaly Romanov, Aidar Gabdullin, Munir Makhmutov, Ruzilya Mirgalimova, Muhammad Fahim, Vladislav Ostankovich, Alena Yuryeva and Artem Burmyakov