

UNIVERSIDAD CARLOS III DE MADRID
Computer Structure
Exam

You have **1:30 hrs** for this exam.

You may **NOT** use any handouts, lecture notes, books, calculators, nor any other external help.

Exercise 1. What is the representation of the following numbers:

- a) -32 in one's complement with 6 bits
- b) -31 in two's complement with 7 bits
- c) -8 in sign-magnitude with 6 bits
- d) 12 in two's complement with 5 bits

Solution

- a) -32 one's complement with 6 bits
The range is $[-31, 31]$, then the number cannot be represented.
- b) -31 in two's complement with 7 bits
31 in binary with 7 bits is 0011111. Complement: 1100000, and add 1, the results is : 1100001
- c) -8 in sign-magnitude with 6 bits
101000
- d) 12 in two's complement with 5 bits
01100

Exercise 2. Represent the number -4.625 using the IEEE 754 single precision format

Solution

$$-4,625_{(10)} = -100,101_{(2)} = -100,101 \times 2^0 = 1,00101 \times 2^2$$

Sign = 1, ngative number

Exponent = $2 + 127 = 129 = 10000001$

Mantissa = 0010100000.... 00000

The number is $11000000100101000000000000000000 = 0xC0940000$

Exercise 3 Describe a MIPS instruction that uses indirect register as addressing mode. Describe this type of addressing mode.

Solution

`lw $t1, ($t2)`

The operand to load in \$t1 is stored in the address stored in register \$t2.

Exercise 4. Consider a 64 bit processor, with 36 registers and 130 machine instructions. Given the following hypothetical instruction: `beqz $t1, n($t2), m($t3)` where `$t1`, `$t2` and `$t3` are registers, and `n` and `m` are numbers that represent displacements. Show a possible format for this instruction assuming that the instruction occupies one word.

Solution

We need 8 bits to represent the operation code.

We need 6 bits to represent a register in this computer.

The format is the following:

CO	t1	n	t2	m	t3
8	6	19	6	19	6

Exercise 5. Consider a function called `func` that receives four integer values as parameters and returns one integer value. Given the following fragment:

```
.data
    a: .word 5
    b: .word 7
    c: .word 9
```

```
.text
```

Write the code needed to invoke the previous function with the parameters: `a`, `b`, `c` and `17`. The value returned by the function must be printed using the appropriate system call.

Solution

```
lw      $a0, a
lw      $a1, b
lw      $a2, c
li      $a3, 17

jal     func

move    $a0, $v0
li      $v0, 1
syscall

move    $a0, $v1
syscall
```

Exercise 6. Given the following definition of an integer array:

```
array: .word 10, 20, 7, 830, 40, 6, 5, 7 #
```

Write an assembly code to print the number of times that the number `7` is found in the array.

```

la      $t0, array
li      $t1, 0                      ; index
li      $t2, 8                      ; number of elements
li      $t3, 7
li      $t4, 0                      ; counter
bucle:  bge $t1, $t2, fin
        lw $t5, ($t0)
        bne $t5, $t3, noContar
        add $t4, $t4, 1
noContar: addi $t0, $t0, 4
        addi $t1, $t1, 1
fin:    li $v0, 1
        move $a0, $t4
        syscall                      ; print

```

The diagram illustrates a computer architecture with the following components and connections:

- Main Memory:** Connected to the **Addresses bus** and **Data bus**. It has read (**R**) and write (**W**) control lines.
- Processor:**
 - Internal bus:** Connects internal components. It has control lines **Td** and **Ta** to the **MAR** and **MBR** respectively.
 - Registers Bank:** Receives **RA**, **RB**, **RC**, and **SC** control signals. It outputs **A** and **B** to **MUX A** and **MUX B**. It also receives **E** from the **ALU**.
 - MUX A and MUX B:** Multiplexers that take inputs from the **Registers Bank** and the **Internal bus** (via **MA** and **MB** control lines) and output to the **ALU**.
 - ALU:** Receives an **OP Cod.** (operation code) and data from **MUX A** and **MUX B**. It outputs to the **Internal bus** and the **PSW** via **T5** and **T6**.
 - PC (Program Counter):** Receives a **4-bit** value from the **Internal bus** (via **T4**) and outputs to the **Addresses bus** (via **C4** and **C5**).
 - IR (Instruction Register):** Receives data from the **Internal bus** (via **T8**) and outputs to the **Control Unit** (via **C6**).
 - RT1, RT2, RT3:** Registers that receive data from the **Internal bus** (via **C9**, **C10**, and **C11** respectively) and output to the **Internal bus** (via **T1**, **T2**, and **T3** respectively).
 - PSW (Program Status Word):** Receives data from the **Internal bus** (via **T7**) and outputs to the **Control Unit** (via **C7** and **C8**).
- Control Unit:** Receives **Control signals** and **Interrupt signals**. It outputs control signals to the **Processor** and the **Main Memory**.

- Write the control signals needed to execute the elemental operation $R8 \leftarrow IR$, where $R8$ is the register “8” of the register bank..
- If the control signals $T5$ and $C9$ are activated in one clock cycle, indicate the elemental operation implemented by this control signals.
- Write the sequence of elemental operations and control signals (fetch included) needed to execute the instruction `li $t0, 20`.

a) RC = 01000, SC y T8

- b) $RT1 \leftarrow RT3$
- c) Cycle 0: $MAR \leftarrow PC$ Signals: T4, C3
Cycle 1: $MBR \leftarrow MP$ Signals: Td, L, C2
 $PC \leftarrow PC + 4$ C4
Cycle 2: $RI \leftarrow MBR$ Signals: T3, C6
Cycle 3: Decoding
Cycle 4: $\$t0 \leftarrow RI(20)$