UNIVERSIDAD CARLOS III DE MADRID
Computer Structure
Exam

You have **1.5 hrs** for this exam.
You may **NOT** use any handouts, lecture notes, books, calculators, nor any other external help.

Exercise 1. Represent using the IEEE 754 single precision format the value **-36**.

Solution

The number 36 in binary system is 100100. $100100 = 1.00100 \times 2^5$. Then:

- Sign is 1, because the number is negative.
- The exponent is 5, then the exponent stored is 5 + 127 = 132 = 10000100 in binary.
- The mantissa is 001000000 …. 00000

The number -36 is represented as 11000010000100000000000000000000

Exercise 2. Given the following program:

```
.text
                .globl main
 main:
                li   $a0,  5
                jal  function
                move $a0, $v0
                li   $v0, 1
                syscall

                li   $v0, 10
                syscall

function:       li $t0, 10
                bgt $a0, $t0, et1
                li $t0, 10
                add $v0, $t0, $a0
                b et2
     et1:       li $t0, 8
                add $v0, $t0, $a0
     et2:       jr $ra
```

What is the value printed in the first system call ?

Solution

The function receives as argument in register $a0 the value 5. As this value is less than 10, the function adds the value 10 ($t0) to 5, and the results is stored in $v0. This value is the value printed (15).

Exercise 3. Consider a function calle `AddValule`. This function receives three arguments:

- The first one is the init address of an array. The components of this array are integer numbers.
- The second one is an integer value that represents the number of components in the array.
- The third one is an integer value.

Te function modifies the array, adding the value passed in the third argument to all components. Reply the following questions:

a) What are the registers used to pass the arguments to the function?
b) Write the code, using the MIPS32 assembly language, for this function?
c) Given the following fragment:

```
.data
       v:  .word    7, 8, 3, 4, 5, 6
.text
       .globl main

 main:
```

Include in the previous `main` function, the assembly sentences needed to invoke the function `AddValue` implemented in b), in order to add the value 5 to all components of the array `v`.

Solution

a) The init address is passed in $a0, the number of elements in `$a1` y and the value to add in `$a2`.

b)

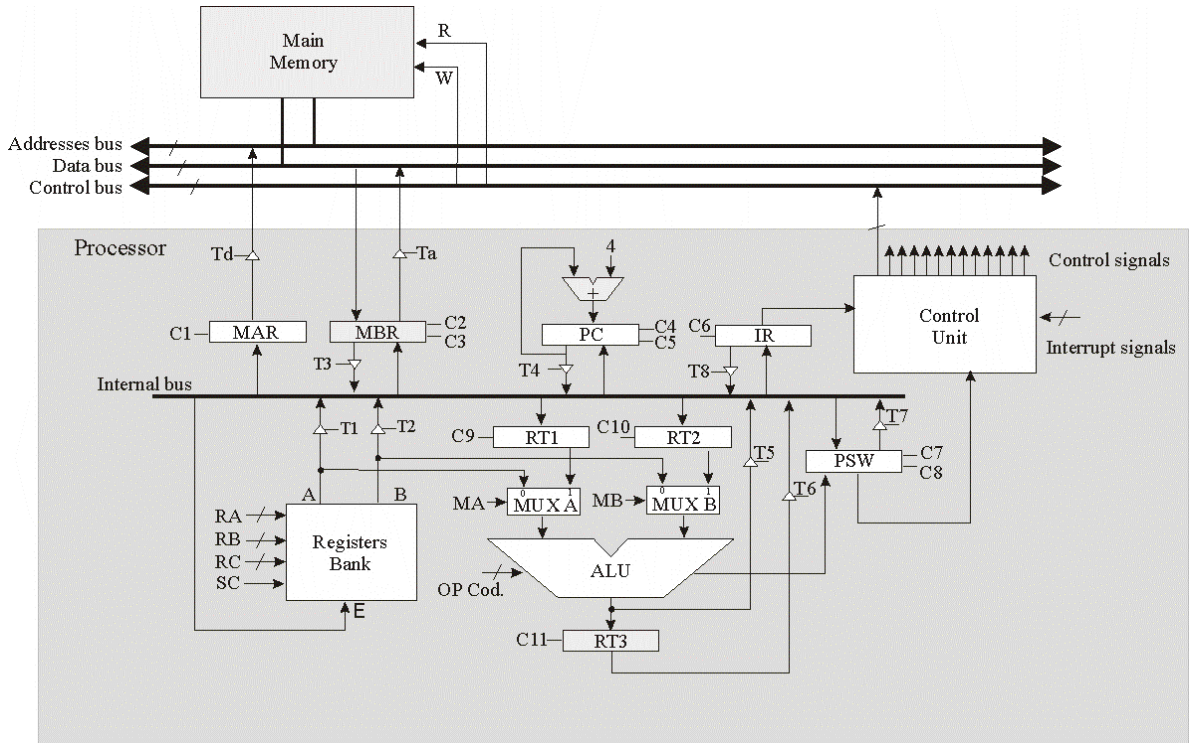
```
AddValue:  li    $t0, 0
                 move  $t1, $a0

     loop: bgt   $t0, $a1, end
                 lw    $t2, ($t1)
                 add   $t2, $t2, $a2
                 sw    $t2,  ($t1)
                 addi  $t0, $t0, 1
                 addi  $t1, $t1, 4
                 b loop
     end:       jr $ra
```

b) The `main` code is:

```
.data
       v: .word  7, 8, 3, 4, 5, 6
.text
       .globl main

 main:     sub   $sp, $sp, 24
           sw    $ra, 20($sp)
           sw    $a0, 4($sp)
           sw    $a1, 8($sp)
           sw    $a2,  12($sp)

           la    $a0, v
           li    $a1, 6
           li    $a2, 5
           jal   AddValue


           li    $v0, 1
           li    $t0, 0
           move $t1, $a0
```

```
loop :     bgt   $t0, $a1, end
           lw    $a0, ($t1)
           syscall
           addi $t0, $t0, 1
           addi $t1, $t1, 4
           b bucle


      end: lw    $ra, 20($sp)
           lw    $a0, 4($sp)
           lw    $a1, 8($sp)
           lw    $a2, 12($sp)
           addi  $sp, $sp, 20

           li    $v0, 10
           syscall
      jr $ra
```

Exercise 4. Consider the following 32-bit computer. The processor has 32 registers and uses one cycle for decoding instructions. The main memory needs one cycle in reading and writing memory operations.



Write the sequence of elemental operations and control signals (fetch included) needed to execute the instruction `addi $t0, $t1, 10`.

Solution

cycle	Elemental operation	Control signals
C1	MAR ← PC	T4, C1
C2	MBR ← MP, PC ← PC + 4	L, Td, C2, C4

C3	RI ← MBR	T3, C6
C4	Decodificación	
C5	TR2 ← RI(10)	T8, C10
C6	$t0 ← $t1 + TR2	RA = <dir de $t1> MA = 0 MB = 1 Cod op = SUMAR T5 RC = <dir de $t0> SC

Exercise 5. Consider a 32 bit computer with a cache memory of 256 KB, lines of 64 bytes, and an access time of 5 ns. The cache is 4-way associative and uses LRU. Reply:

a) What is the number of lines of this cache?
b) What is the number of sets of this cache?
c) What is the size of the blocks transferred between the cache and the main memory?
d) If the time used to transfer a block from main memory to cache is 200 ns, what is the hit ratio needed to obtain an average access time of 20 ns?

Solution

a) The cache size is 256 KB = 2^{18} bytes. Each line has 2^6 bytes, the number of lines is 2^{18} bytes / 2^6 bytes = 2^{12} lines = 4096 lines. The cache is 4-way associative, each set has 4 lines, then the number of sets is 4096 / 4 = 1024 sets..
b) This block is a line, and then the size is 64 bytes.
c) The average access time is given by:

$tm = tc \cdot Pa + (1-Pa) \cdot tf$

where tc = 5 ns, tm = 20 ns and tf = 205 (200 + 5). Then:

$20 = 5 \cdot Pa + (1-Pa) \cdot 205$

Pa = 185 / 200 = 0,92. The hit ratio must be 92 %.