UNIVERSIDAD CARLOS III DE MADRID
Computer Structure
Exam

You have **2 hrs** for this exam.
You may **NOT** use any handouts, lecture notes, books, calculators, nor any other external help.

Exercise 1. Consider the following program:

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

                li   $v0, 10
                syscall
function:
                move $t0, $a0
                li   $t1, 0
         loop: beq  $t0, 0, end
                add  $t1, $t1, $t0
                sub, $t0, $t0, 1
                b    loop
          end:  move $v0, $t1
                jr $ra
```

Reply the following questions, justifying your answer:

a) What is the value printed in the first system call?
b) If register $a0 used for arguments passing, has a value represented in one's complement, what is the range of values that can be stored in this register?

Solution:

a) The function adds the following numbers: 5, 4, 3, 2, and 1, and returns the value in the register $v0. Then the, value (15) is printed.
b) When a n-bit word is used, the representation range for one's complement numbers is
c) $[-2^{n-1}+1, 2^{n-1}-1]$. In MIPS 32, registers have 32 bits, then n = 32, and the range is $[-2^{31}+1, 2^{31}-1]$.

Exercise 2. Consider a function called `Vowels`. This function receives as argument the init address of a string, and calculates the number of occurrences of `'a'` in the string. When then null string is passed, the function returns -1.

a) Write, using the MIPS32 assembly language, the code of this function.
b) What is the register used to pass the argument, and what is the register used to return the result?
c) Given the following program:

```
.data
        cadena:  .asciiz    "Hola"
.text
        .globl main

 main:
```

Include in the above main function, the assembly sentences needed to invoke the function implemented before. Include, also, the assembly sentences needed to print the value returned by the function. The argument that can be passed to the function is `cadena`.

Solution:

a) The string is passed in `$a0` and the result is returned in `$v0`. The code for the function is:

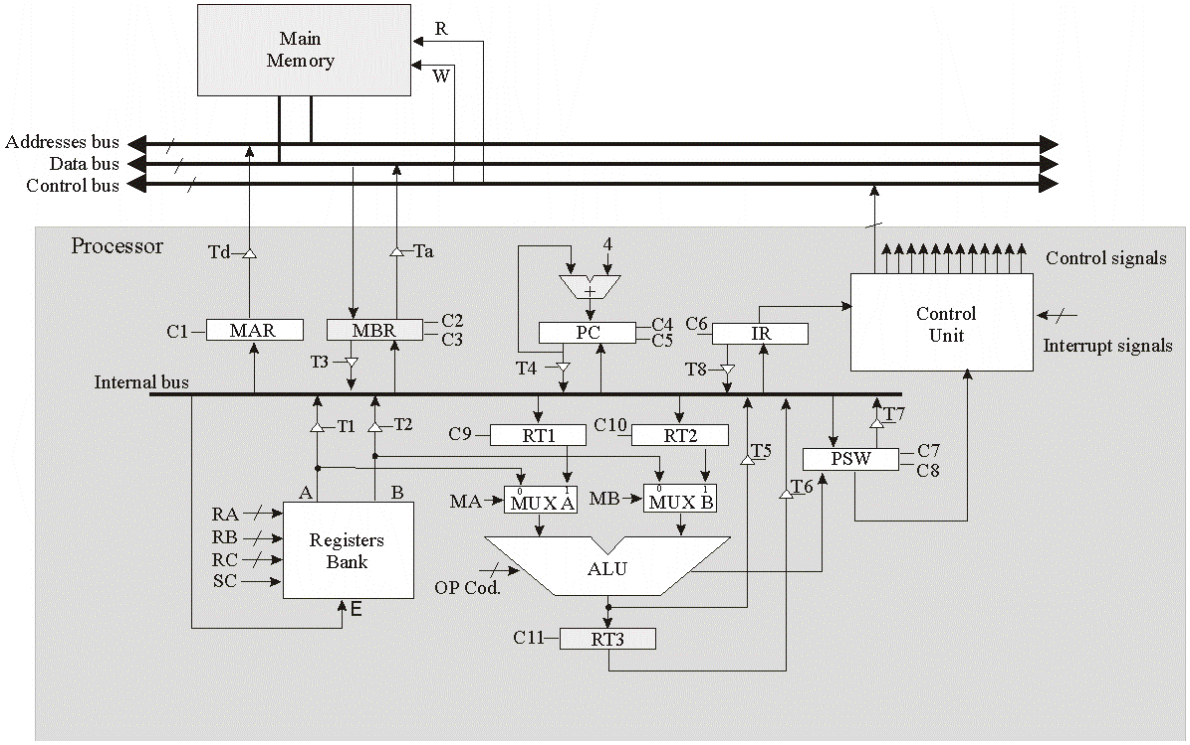
```
Vowels:    li   $t0, -1     // count the number of 'a'
           move $t1, $a0
           beqz $t1, fin
           li   $t0, 0
           li   $t2, 'a'
   loop:   lbu  $t3, ($t1)
           beqz $t3, end
           bneq $t3, $t2, noA
           addi $t0, $t0, 1
      noA: addi $t1, $t1, 1
           b loop

     end: move $v0, $t0
           jr   $ra
```

b) Arguments are passed in registers `$aX`, and the results in `$vX`. In this case, the string is passed in `$a0` and the result is returned in `$v0`.

c) The `main` is:

```
.data
       cadena:  .asciiz    "Hola"
.text
       .globl main

 main:     sub  $sp, $sp, 24
           sw   $ra, 20($sp)
           sw   $a0, 4($sp)

           la   $a0, cadena
           jal  Vowels

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

           lw   $ra, 20($sp)
           lw   $a0, 4($sp)
           addi $sp, $sp, 24

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

Exercise 3. 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 `lw $t1, ($t2)`.

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	decoding	
C5	MAR ← $t2,	RA = <dir de $t2> T1, C1
C6	MBR ← MP	Td, L, C2
C7	$t1 ← MBR	T3, SC, RA = <dir de $t1>

Exercise 4. Consider a 32 bit computer with a data cache memory of 32 KB and lines of 64 bytes. The cache is 2-way associative and uses LRU. Given the following code:

```
int v[262144]

for (i = 0; i < 262144; i = i + 2)
```

```
v[i] = 9;
```

Reply:

a) What is the number of lines and the number of sets in this cache?
b) Calculate, justifying your answer, the hit ratio obtained in the above loop.

Solution:

a) The size cache is 32 KB = 2^{15} bytes. Each line has 2^6 bytes. Then, the number of lines is 2^{15} bytes / 2^6 bytes = 2^9 lines = 512 lines. The cache is 2-way associative, then, each set has two lines. The number of sets is 512 / 2 = 256 sets.
b) The access pattern is:

   `v[0], v[2], v[4], v[6], v[8] …`

   The cache is organized in lines of 64 bytes, and each integer (`int`) occupies 4 byes. Then, each line allows to store 16 integers elements. The array is accessed in sequential way. Each 16 elements, only 8 are really accessed. Then the miss ratio is 1/8.

Exercise 5. Consider a computer with a paged virtual memory that uses addresses of 20 bits and pages of 1 KB. The physical memory has 256 KB. Reply the following questions briefly, justifying your answer:

a) What is the virtual address format?
b) What is the maximum number of entries in the page table (one level)?
c) How many frames does the memory have?
d) Describe the fields included in the page table.

Solution

a) Pages have 1 KB = 2^{10} bytes. The virtual address has 20 bits. 20 -10 = 10 bits are used for the page number. The format uses the upper 10 bits to represent the page number and the lower 10 bits to represent the displacement inside the page.
b) The maximum number of entries in the page table is 2^{10} = 1024 entries.
c) Then number of frames is 256 KB / 1 KB = 256 frames.
d) Each entry includes:
   a. Present bit.
   b. Modified bit.
   c. Valid bit.
   d. Permissible access.
   e. Frame.