# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

## Organización y Arquitectura de Computadores

PRIMERA EVALUACIÓN – I TÉRMINO 2011-2012

Nombre:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Matrícula:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## TEMA 1 (45 puntos)

El párrafo a continuación es parte de una entrevista a John Hennessy acerca de los principios del diseño de la arquitectura RISC.

En función de esta porción de la entrevista y/o de su conocimiento previo, responda las siguientes preguntas:

“[We] decided early on that the machine should be pipelined, that it should aim for single cycle execution. We quickly saw all the advantages of single cycle execution in terms of not having to deal with instructions which we get page faults in the middle of them or interrupts in the middle of them.

…

We decided not to implement byte addressing. That was kind of an experimental try to say, “What could you do if you didn’t supporting it? Could you make a machine that really worked well and how much advantage did that gain you in terms of simplicity?” […] And so there were trade-offs like that that got made early on.

…

The RISC machines […] made it easier to think about design in a piece of hardware that was that complicated because things are a lot simpler if the instructions only take one cycle and they only do simple formats, you can actually contemplate something that is logically that complicated and get it right. That’s the ongoing issue: the simplicity of the instruction set, or the care in the design of the instruction set makes it easier to do the implementation. And I think that’s an ongoing advantage that translates into how big the design team needs to be, how long it takes to do the design.”

- John Hennessy, Fundador de MIPS

1. ¿En donde radica la complejidad de diseño de un procesador RISC?
2. ¿Por qué el datapath de un sistema RISC es más simple que el datapath de un sistema CISC?
3. Justifique la facilidad inherente de realizar pipelining sobre un sistema RISC.

## TEMA 2 (55 puntos)

Considere el siguiente ejemplo de compilación de un programa en C hacia MIPS:

void swap(int v[], int i, int j){

int tmp;

tmp = v[i];

v[i] = v[j];

v[j] = tmp;

}

swap:

lw $t0, $a1($a0)

lw $t1, $a2($a0)

sw $t1, $a1($a0)

sw $t0, $a2($a0)

jr $ra

Se intentó de traducir un programa para calcular los cubos de una secuencia de enteros pero el compilador dejó ciertos espacios en blanco. **Complete** la tarea del compilador.

syscall:

1 print\_int

4 print\_string

.data

int cubo(int n){

// se considera que n>0

int sum = 0;

int impar = n\*n-n-1;

for(int i=0; i<n; i++){

sum += impar;

impar += 2;

}

return sum;

}

.text

imprimir\_cubos:

# manejo del stack

# manejo del stack

# manejo del stack

li $s0, 1

addi $s1, $a0, 1

print\_loop:

move $a0, $s0

jal cubo

blt $s0, $s1, print\_loop

# manejo del stack

# manejo del stack

# manejo del stack

jr $ra

void imprimir\_cubos(int n){

// se considera que n>0

for(int i=1; i<n+1; i++)

printf(“%i al cubo = %i\n”,

i, cubo(i));

}

.text

cubo:

li $t0, 0 # $t0 = sum

li $t2, 0

calc\_loop:

blt $t2, \_\_\_\_\_ , calc\_loop

# retornar sum

jr $ra