

UNIVERSIDAD CARLOS III DE MADRID. DEPARTAMENTO DE INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA. DOBLE GRADO ADE E INGENIERÍA INFORMÁTICA
ESTRUCTURA DE COMPUTADORES

20 de junio de 2019

Examen final

Para la realización del presente examen se dispondrá de **2:00 horas**.

Lea con atención el enunciado del examen. No se responderán preguntas después de transcurridos 30 minutos.

NO se podrán utilizar libros, apuntes **ni** calculadoras (o dispositivos electrónicos) de ningún tipo.

Ejercicio 1 (1puntos). En relación al estándar IEEE 754 de simple precisión se pide:

- a) Represente el valor 28,25.
- b) Indique el valor decimal del número representable que se encuentra inmediatamente a continuación y es mayor del anterior (28,25).

Ejercicio 2 (3 puntos). Considere la rutina `MayorQue`. Esta rutina acepta **cuatro** parámetros de entrada:

- La dirección de inicio de una matriz (se almacena por filas) de números de tipo `int`, de dimensión $M \times N$
- El número de filas de la matriz (M).
- El número de columnas de la matriz (N).
- Un valor entero A

La función devuelve el número de elementos de la matriz cuyo valor es mayor que el valor A .

Se pide:

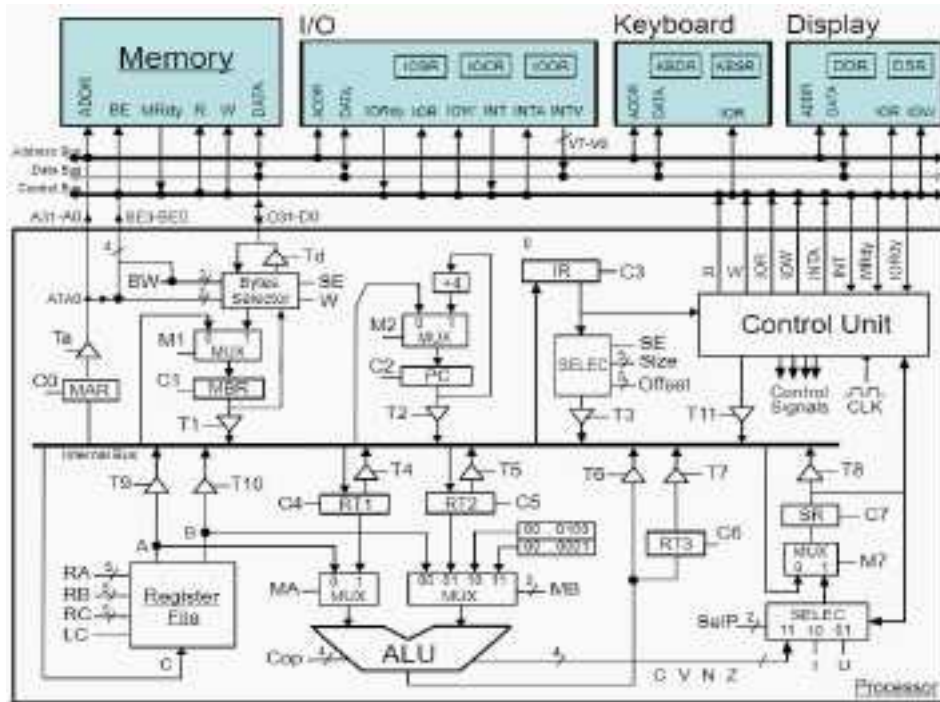
- a) Codifique correctamente la rutina `MayorQue` anteriormente descrita. Ha de seguirse estrictamente el convenio de paso de parámetros del MIPS visto en clase. Se valorará positivamente que dicha función invoque a otras funciones.
- b) Dado el siguiente segmento de datos:

```
.data
Matriz: .word    8, 4, 3, 3, 5,
                2, 3, 0, 4, 5,
                0, 0 ,1, 2, 3

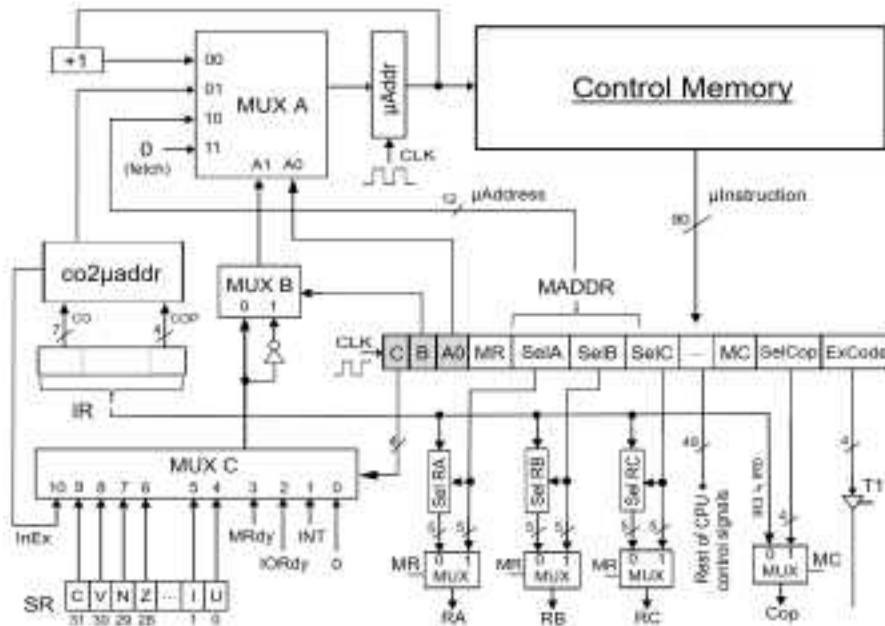
M: .word 3
N: .word 5
A: .word 7
```

Codifique el fragmento de código que permite invocar correctamente a la función `MayorQue` para la matriz `Matriz` de dimensión $M \times N$ y el valor A . A continuación, se debe imprimir el valor que devuelve la función.

Ejercicio 3 (4 puntos). Dado el procesador WepSIM con la siguiente estructura:



Este procesador dispone de una Unidad de control representada por la siguiente figura:



Considere la instrucción máquina MAYOR Ra (Rb) (Rc). Esta instrucción compara el contenido de las posiciones de memoria almacenadas en los registros Rb y Rc y almacena en el registro Ra el valor mayor almacenado en esas dos posiciones. Se pide:

- Considerando que se dispone de un juego de 130 instrucciones máquina, indique el formato de la instrucción anterior.
- Especifique las operaciones elementales y señales de control (incluyendo el ciclo de *fetch*) necesarias para ejecutar la instrucción anterior, teniendo en cuenta el formato especificado en el apartado a).

NOTA: Asuma que R29 actúa como puntero de pila, que el puntero de pila apunta a cima de pila y que la pila crece hacia direcciones decrecientes de memoria. Considere también que el registro R0 es de solo lectura y almacena el valor 0.

Ejercicio 4 (2 puntos). Considere el siguiente fragmento de código:

```
int      A[8192];
int      B[8192];
double   C[8192];

for (i=0; i < 8192; i++){
    B[i] = A[i] + B[i];
    C[i] = (double) B[i];
}
```

Dicho código se ejecuta en una arquitectura con un ancho de palabra de 32 bits, que incluye una memoria caché de datos de 256 KB, asociativa por conjuntos de 8 vías. La caché tiene líneas de 128 bytes. Asumiendo que la caché está inicialmente vacía. Se pide:

- Indique el número de líneas y conjuntos de la memoria caché de datos.
- Indique el número de accesos a memoria (de datos) que se produce en la ejecución del fragmento anterior.
- Indique la tasa de aciertos en la memoria caché de datos que se produce en la ejecución del fragmento de código anterior.
- Modifique el fragmento de código anterior para reducir el número de accesos a memoria. ¿Afecta el nuevo fragmento de código a la tasa de aciertos a la memoria caché de datos? Razone su respuesta.

Soluciones

Ejercicio 1.

a) $28,25_{(10)} = 11100,01_{(2)} = 1,110001_{(2)} \times 2^4$

El bit de signo es 0

El exponente almacenado = $4 + 127 = 131 = 10000011$

La mantisa almacenada (sin bit implícito) = 110001000000000000000000

Su representación en el estándar IEEE 754 es:

01000001111000100000000000000000

b) El número inmediatamente superior al anterior es:

01000001111000100000000000000001

El valor decimal de este número es $= 1,110001000000000000000001 \times 2^4 =$
 $= 11100,01000000000000000001 = 28,25 + 2^{-19}$

Ejercicio 2.

- a) Esta función recibe 4 argumentos, que se pasan en \$a0, \$a1, \$a2 y \$a3. Se utiliza la función esMayor que determina si un número es mayor que otro. En esta función si \$a0 >= \$a1 se devuelve 1 sino 0.

```
esMayor:    beq    $a0, $a1, mayor
            li     $v0, 0
            jr     $ra
mayor:      li     $v0, 1
            jr     $ra

MayorQue:   addi   $sp, $sp, -16
            sw     $ra, 0($sp)
            sw     $s0, 4($sp)
            sw     $s1, 8($sp)
            sw     $s2, 12($sp)
```

```

        ; Se recorren los MxN elementos de la matriz ($s0)
mul     $s0, $a1, $a2
li      $s1, 0      ; contador de elementos
move    $s2, $a0    ; dirección de inicio de la matriz
move    $a1, $a3
li      $v1, 0      ; Contador de elementos mayor que $a3

bucle:   beg     $s1, $s0, fin
        lw      $a0, ($s2)
        jal     esMayor
        add     $v1, $v1, $v0
        addi    $s1, $s1, 1
        addi    $s2, $s2, 4
        b       bucle

        move    $v0, $v1

        lw      $s2, 12($sp)
        lw      $s1, 8($sp)
        lw      $s0, 4($sp)
        lw      $ra, ($sp)

        addi    $sp, $sp, 16
        jr      $ra

```

b) Para invocar a la función se necesita el siguiente fragmento de código:

```

la      $a0, Matriz
lw      $a1, M
lw      $a2, N
lw      $a3, A

jal     MayorQue

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

```

Ejercicio 3.

- a) La instrucción ocupa una palabra:
- Código de operación: bits 31-25
 - Registro Ra: bits 24-20
 - Registro Rb: bits 19-15:
 - Registro Rc: bits 14- 10
 - Sin uso: 9-0

b)

Fetch:

Ciclo	Operaciones elementales	Señales de control
C1	$MAR \leftarrow PC$	T2, C0
C2	$MBR \leftarrow MP[MAR]$ $PC \leftarrow PC+4$	Ta, R, BW=11, M1=1, C1, M2=1, C2
C3	$IR \leftarrow MBR$	T1, C3
C4	Decodificar y salto a Código de operación.	C=0, B=0, A0=1

Código del microprograma pedido. Para la ejecución de esta instrucción se va a usar el registro de estado. Se va a guardar previamente en RT3 para no modificarlo

Ciclo	Operaciones elementales	Señales de control
C1	$RT1 \leftarrow SR$	T8, C4
C2	$RT3 \leftarrow RT1 + R0$	Cop = suma, SelB=00000, MR = 1 MA, C6
C3	$MAR \leftarrow Rb$	C0, T11, MR=0, SelA=15 (01111)
C4	$MBR \leftarrow MP[MAR]$	Ta, R, BW=11, M1=1, C1
C5	$RT1 \leftarrow MBR$	T1, C4
C6	$MAR \leftarrow Rc$	C0, T11, MR=0, SelA=10 (01010)
C7	$MBR \leftarrow MP[MAR]$	Ta, R, BW=11, M1=1, C1
C8	$RT2 \leftarrow MBR$	T1, C5
C9	$RT1 - RT2$ Actualizar SR	MA=1, MB=1, Cop=RESTA, SelP=11, M7, C7
C10	Si N goto C14	MADDR = C14, C=111
C11	$Ra \leftarrow RT1$	T4, LC, SelA=20 (10100), MR
C12	$SR \leftarrow RT3$	T7, C7
C13	Salto a fetch	C=0, B=1, A0=0, MADDR=fetch
C14	$Ra \leftarrow RT2$	T45 LC, SelA=20 (10100), MR
C15	$SR \leftarrow RT3$	T7, C7
C16	Salto a fetch	C=0, B=1, A0=0, MADDR=fetch

Ejercicio 4.

- a) La caché de datos tiene un tamaño de $256KB = 2^{18}$ bytes. Las líneas son de 128 bytes.
El número de líneas es de $2^{18} / 2^7 = 2^{11} = 2048$.
El número de conjuntos es de $2^{11} / 2^3 = 2^8 = 256$

- b) En cada vuelta del bucle se producen los siguientes accesos:

- Lectura de A
- Lectura de B
- Escritura en B
- Lectura de B
- Escritura en C

Se producen 5 accesos. El número total de accesos es de $5 \times 8192 = 40960$

- c) En una línea de 128 bytes se almacenan $128/4 = 32$ enteros y 16 elementos de tipo double.
Cada 32 iteraciones se produce 1 fallo en A, 1 fallo en B y 2 fallos en C. En total 4 fallos. El número total de fallos es de $4 \times 8192/32 = 1024$
La tasa de aciertos = $(40960-1024)/40960 = 0.975$

- d) Una forma de reducir el número de accesos es reescribir el código de la siguiente manera:

```
for (i=0; i < 8192; i++){
    tmp = (A[i] + B[i]);
    B[i] = tmp;
    C[i] = (double)tmp;
```

En este caso el número de accesos total es de $4 \times 8192 = 32768$ accesos

El número de fallos es el mismo, y la nueva tasa de aciertos es $(32768-1024)/32768 = 0,968$