

UNIVERSIDAD CARLOS III DE MADRID
DEPARTAMENTO DE INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA. ESTRUCTURA DE COMPUTADORES
Ejemplos de preguntas de los miniexamen

Ejercicio 1. Se tiene un procesador de 20 bits con un banco de registro de 24 registros, que direcciona una memoria principal a nivel de byte. Conteste correctamente y de forma breve a las siguientes preguntas:

- a) ¿Qué es el contador de programa y cuántos bits almacena?

Es el registro cuyo contenido es la dirección de la siguiente instrucción a ejecutar. 20 bits.

- b) ¿Qué es el registro de instrucciones y cuántos bits almacena?

Es el registro cuyo contenido es la instrucción en curso. 20 bits.

- c) ¿Qué es el registro MBR y cuántos bits almacena?

Es el registro que contiene el valor traído/llevado desde/a memoria. 20 bits.

- d) Si un registro almacena un valor en **complemento a uno**,
¿Qué rango valores puede almacenar?

El rango en Ca1 es $[-2^{20-1}+1 \dots -0,0, \dots 2^{20-1}-1]$

- e) Si un registro almacena un valor en **complemento a dos**,
¿Qué rango valores puede almacenar?

El rango en Ca1 es $[-2^{20-1} \dots -1,0, \dots 2^{20-1}-1]$

- f) ¿Cuántos bits son usados para identificar un registro?

Si hay 24 registros, con 5 bits (logaritmo en base 2 por exceso).

- g) ¿Cuántos bits son almacenados en un registro?

Si es un procesador de 20 bits de palabra (los registros son de 20 bits) entonces 20 bits.

- h) ¿Cuántos bits son usados para direccionar una celda de memoria?

Si es un procesador de 20 bits de palabra (los registros son de 20 bits) entonces 20 bits.

- i) ¿Cuántos bytes son almacenados en una posición de memoria?

Si es una memoria direccionada a nivel de byte entonces 1 byte.

Ejercicio 2. Sume los siguientes números representados en formato IEEE754, indicando brevemente los pasos que ha de seguirse de forma general: 0x41900000 + 0x7F800000

0x41900000

0100 0001 1001 0000 0000 0000 0000 0000
0x 4 1 9 0 0 0 0 0

Signo = 0 (positivo)

Exponente = 10000011 ($131 - 127 = 4$)

Mantisa = 1001 (con bit implícito)

- **Si caso especial**
 - **Trato casos especiales**
- **Si no es caso especial**
 - **Los represento como $1,xxx * 2^y$**
 - **Igualo exponentes**
 - **Sumo mantisas**
 - **Normalizo**

$$1,001_2 \times 2^4 = 10010,0_2 \times 2^0 = 18_{10}$$

0x7F800000

Es un caso especial: positivo, exponente a unos y mantisa a ceros -> + infinito

Como es un caso especial, el resultado es infinito (0x7F800000)

Ejercicio 3. Dada el siguiente código en ensamblador del MIPS 32:

```
.data
    str1: .asciiz "hola mundo"
.text
    .glob main
main:
    li    $v0  0
    li    $t1,  0
    li    $t2  str1
if1:    lw    $t3  $t2
        beqz $t3, fin1
        add  $v0, $v0 1
        add  $t2 $t2 4
        b   if1
fin1: jr  $ra
```

Se pide: indique **todos** los **errores** que contiene el fragmento anterior, si se busca calcular la longitud de la cadena de caracteres y guardar dicha longitud en el registro \$v0.

Los errores son:

```

    li    $v0  0
    li    $t1,  0
    la    $t2  str1
if1:    lb    $t3  ($t2)
        beqz $t3, fin1
        addi $v0, $v0 1
        addi $t2 $t2 1
        b   if1
fin1: jr  $ra
```

Ejercicio 4. Se tiene un procesador con 32 registros de 64 bits cada uno. La memoria se direcciona a nivel de byte y los números se representan en complemento a dos. Conteste correctamente y de forma breve a las siguientes preguntas:

a) ¿Cuántos GB de memoria principal puede direccionar este computador?

Se pueden direccionar 2^{64} bytes = $2^{64} / 2^{30}$ GB = 2^{34} GB

b) ¿Se puede representar el número 2^{63} en este computador, razone su respuesta?

En este computador se emplean 64 bits para representar los números. Con 64 bits el rango de representación en complemento a dos es $[2^{63}, 2^{63} - 1]$, por tanto no se puede representar ese número.

c) ¿Qué es el contador de programa y para qué se utiliza? Si en el contador de programa se almacena un número en binario puro, ¿cuál sería el rango de representación?

El contador de programa es un registro del procesador que almacena la dirección de la siguiente instrucción a ejecutar. El rango de representación sería $[0..2^{64}-1]$

Ejercicio 5. Indique el valor decimal correcto correspondiente a los siguientes números representados en formato IEEE754 de forma brevemente razonada.

a) **0xC1340000**

1100 0001 0011 0100 0000 0000 0000 0000
0x 4 1 3 4 0 0 0 0

Signo = 1 (negativo)

Exponente = 10000010 ($130 - 127 = 3$)

Mantisa= 01101 (con bit implícito)

$$1,01101_2 \times 2^3 = 1011,01_2 \times 2^0 = -11,25_{10}$$

Ejercicio 6. Dada el siguiente código en un lenguaje de alto nivel:

```
int vec[100];
int b = 5;

main ()
{
    int i = 0;
    for (i = 0; i < 100; i++)
        vec[i] = b+i;
}
```

Se pide: escriba su equivalente en lenguaje MIPS32 de forma clara, concisa y correcta.

```
.data
    .align 2 #siguiente dato alineado a 4
    vec: .space 400
    b:    .word 5

.text
    .glob main

main:
    li    $t0, 5
    la    $t1, vec
    li    $t2, 0
    li    $t3, 100
    li    $t4, 0
bucle:  bge $t2, $t3, fin
        sw  $t0, vec($t4)
        addi $t2, $t2, 1
        addi $t4, $t4, 4
fin:    b    bucle
```

Ejercicio 7. Conteste correctamente y de forma brevemente razonada al siguiente problema.
 Considere un computador de 15 bits con un juego de 12 instrucciones máquina y un banco de 8 registros.
 Indique un formato de instrucción para poder codificar la instrucción lw de la forma: lw \$t1 8(\$t2)
 Seleccione el mejor direccionamiento a registro base para el programador.

Solución:

4 bits	3 bits	3 bits	5 bits
Código de instrucción	Registro	Registro	relleno

Primera palabra

15 bits
desplazamiento

segunda pa

Ejercicio 8. Un estudiante desea acceder al elemento `matriz[i][j]` un elemento de una matriz de 3 filas por cinco columnas. Se pide que conteste correctamente el trozo de código en ensamblador para realizar dicho acceso si la variable `i` que indica la fila está en `$t1`, y la variable `j` está en `$t2`.

Solución:

```
# (i*5+j)*4
mul $t3 $t1 5
add $t3 $t2
mul $t3 $t3 4

# *( matriz + (i*5+j)*4 )
la $t4 matriz
add $t4 $t4 $t3
lw $t4 ($t4)
```

Ejercicio 9. Un estudiante ha escrito una subrutina, pero no la ha comentado. La subrutina se denomina `X` y se muestra a continuación. Se pide que conteste correctamente qué parámetros tiene dicha subrutina, que resultados devuelve y qué realiza la rutina (se puede usar una expresión matemática en función de los parámetros de entrada o una descripción del resultado obtenido).

```
X:      li    $t5 0x8000
        sl    $t5 $t5 16
        and   $t2 $a0 $t5
        li    $t5 0x7F80
        sl    $t5 $t5 16
        and   $v0 $a0 $t5
        srl   $t2 $t2 31
        srl   $v0 $v0 23

        li    $t7 0x0007
        sl    $t7 $t7 16
        liu   $t0 0xFFFF
        or     $t7 $t7 $t0
        addi  $v0 $v0 -127
        and   $v1 $a0 $t7
        beq   $t2 $0 fin1
fin1:    jr    $ra
```

Nombre de la rutina: `X`

Parámetros:

Resultados:

Solución:

Nombre de la rutina: `X`

Parámetros:

`$a0` -> número

Resultados:

`$v0` -> exponente sin exceso 127

`$v1` -> mantisa sin signo

Ejercicio 10. Dado las siguientes funciones en lenguaje C:

```
int F1 ( int a, int b, int c, int d, char e, char f ) {
    int ret[10] ;
    ret[2] = (a+b+c+d+(int)e+(int)f);
    return ret;
}

int main ( int argc, char *argv[] ) {
    F1(1,2,4,5,'a','b');
}
```

Codifique dichas rutinas en ensamblador de MIPS32 siguiendo el convenio de paso de parámetros visto en clase (y disponible en el material publicado). NOTA: no es necesario usar el registro de marco de pila.

Solución:

```
main:                                     move $v0 $sp
    # push $ra
    subu $sp $sp 4                       # pila
    sw $ra ($sp)                         addu $sp $sp 40

    # F1(1,2,4,5,'a','b')
    li $a0 1                             # return
    li $a1 2                             jr $ra
    li $a2 4
    li $a3 5
    subu $sp $sp 8
    li $t0 'b'
    sw $t0 4($sp)
    li $t0 'a'
    sw $t0 0($sp)
    jal F1
    addu $sp $sp 8

    # pop $ra
    lw $ra ($sp)
    addu $sp $sp 4

    # return
    jr $ra

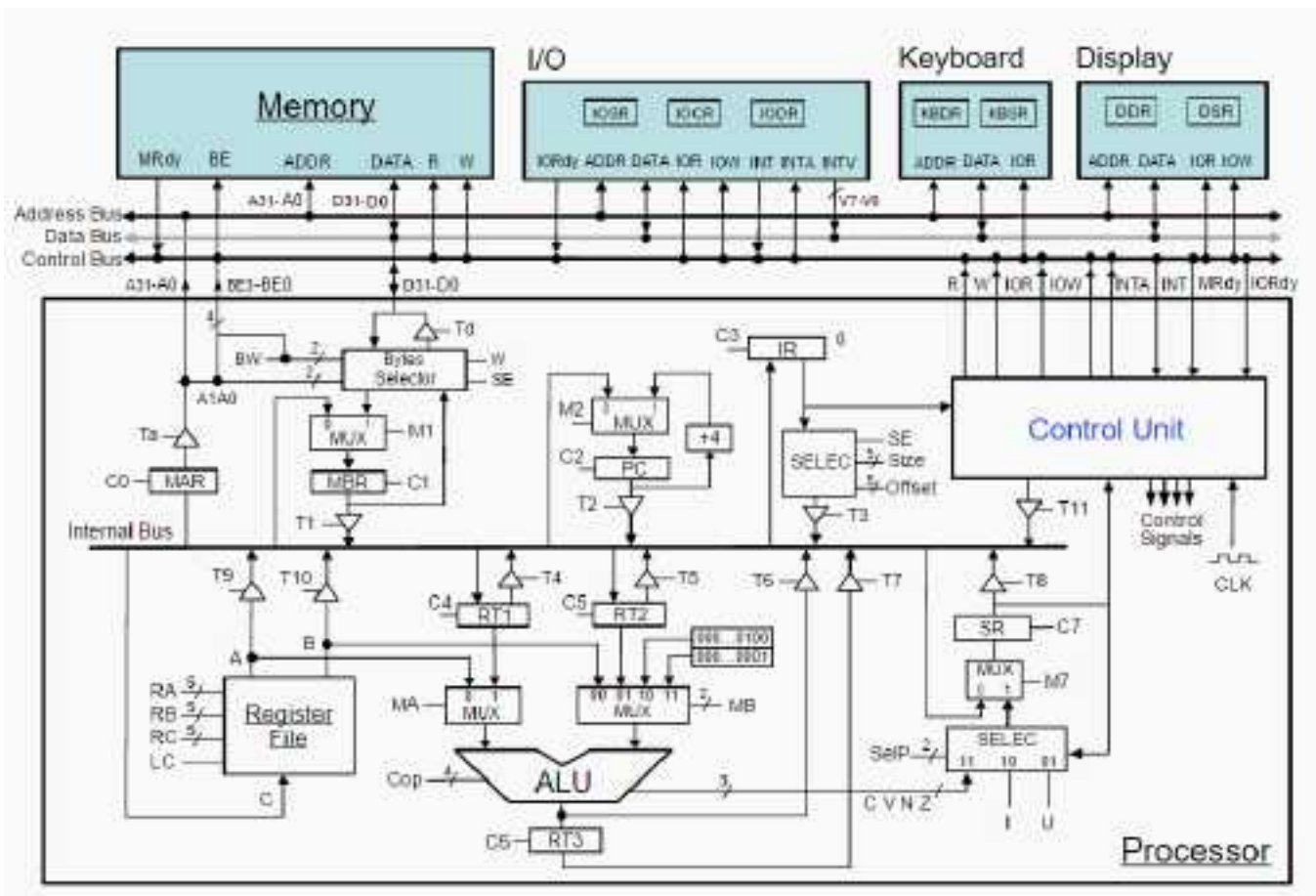
F1:
    # pila <- int ret
    subu $sp $sp 40

    add $t0 $a0 $a1
    add $t0 $v0 $a2
    add $t0 $v0 $a3

    lw $t1 4($sp)
    add $t0 $t0 $t1
    lw $t1 8($sp)
    add $t0 $t0 $t1

    # ret
    sw $t1 8($sp)
```

Ejercicio 11. Dada el procesador con la siguiente estructura:



Especifique las operaciones elementales y señales de control necesarias para ejecutar la instrucción máquina `push R1`. Esta instrucción almacena el valor contenido en el registro R1 en la cima de la pila, reajustando el registro de puntero de pila adecuadamente. Incluya el ciclo de *fetch*.

NOTA: Asuma que R29 es el puntero de pila (apunta a cima de pila y crece hacia direcciones decrecientes de memoria) y que la memoria se direcciona por byte (y tarda 1 ciclo en operar).

Solución:

Operaciones elementales	Señales de control
MAR <- PC	T2, C0
M[MAR] -> MBR, PC <- PC+4	Ta, R, BW=3, M1=1, C1
MBR -> RI	T1, C3
Decodificar	Deco
Salto a Co	Salto CO
MAR <- R29	RA=R29, T9, C0
MBR <- R1	RA=R1, T9, M1=0, C1
M[MAR] <- MBR	Ta, Td, BW=3, W
R29 <- R29 + 4	RA=29, MA=0, MB=4, Cop=+, T6, RC=R29, LC
Salto a fetch	Salto Fetch

Ejercicio 12. Conteste correctamente y de forma razonada al siguiente problema.

Codifique una rutina en ensamblador MIPS32 llamada **contar** a la que se le pasa dos parámetros: la dirección de una matriz cuadrada de enteros **m**, el número **n** de elementos, y devuelve el número de números pares que hay en la matriz. Ha de usar un doble bucle, así como el convenio de paso de parámetros y retorno vista en clase.

Solución:

```
contar:
    # resultado = 0
    li    $v0 0

    # for (i=0; i<$a1; i++)
    #     for (j=0; j<$a1; j++)
    li    $t1 0
b1: bge   $t1 $a1 f1
    li    $t2 0
b2: bge   $t2 $a1 f2

    # t3 = *($a0+(i*$a1+j)*4)
    mul   $t3 $t1 $a1
    add   $t3 $t3 $t2
    mul   $t3 $t3 4
    add   $t3 $t3 $a0
    lw    $t3 ($t3)

    # if ($t3 %2) == 0
    #     resultado++
    rem   $t3 $t3 2
    beqz  $t3 impar
    addi  $v0 $v0 1
impar:

    addi  $t2 $t2 1
    b     b2
f2: addi  $t1 $t1 1
    b     b1
f1: jr    $ra                # return
```

Ejercicio 13. Responda brevemente (en el espacio dado) a las siguientes preguntas:

a) ¿Qué es una memoria caché?

Una memoria de menor capacidad que se ofrece más velocidad de acceso (menos latencia). Está colocada entre la CPU y la memoria principal existente de forma que en memoria caché solo está la parte del programa que se está usando en un momento dado.

b) ¿Qué es un módulo de E/S?

Es un elemento colocado entre el periférico y la CPU de forma que se encarga de ofrecer una interfaz a la CPU que oculte las particularidades del periférico (distintos tamaños de palabra, temporización, control de errores, etc.).

Ejercicio 14. Sea una CPU que accede a una memoria caché de 10 ns de tiempo de acceso en caso de acierto y 100 ns adicionales de transferencia en caso de fallo, ¿Cuál sería el tiempo medio de acceso si el *hit ratio* es de 90%?

$$\begin{aligned} t_m &= t_{ca} * h + (t_{ca} + t_p) * (1 - h) \\ t_m &= 10 * 0.9 + (10 + 100) * (1 - 0.9) = 9 + 110 * 0.1 = 9 + 11 = 20 \text{ ns} \\ t_m &= 20 \text{ ns} \end{aligned}$$

Ejercicio 15. Imagine que la empresa para la que ha desarrollado las instrucciones `adv_2` y `seqv_2` tiene un nuevo proyecto de un sensor de posicionamiento de la cabeza de una persona.

Esta empresa para la que trabaja desarrolla el dispositivo y el módulo de E/S asociado usando una arquitectura MIPS 32 con mapa de entrada y salida separado y técnica de entrada y salida programada.

El módulo de entrada y salida dispone de tres registros de 32 bits:

- **Registro de datos** (con dirección 0x108). Almacena la posición leída.
- **Registro de control** (con dirección 0x104). Cuando se escribe en el registro el valor 2 se inicializa el sensor.
- **Registro de estado** (con dirección 0x100). Si el valor del registro es 0, no se ha realizado ninguna lectura de posición de cabeza. Si el valor es 1, se ha realizado una lectura y el controlador dispone en el registro de datos de la posición. Si el valor es -1, se ha producido un error y ha de volver a inicializar el lector (solo hay que inicializar una vez el lector, salvo error).

Dicha empresa le pide a usted que desarrolle una rutina en ensamblador que se encargue de iniciar el lector y realizar de forma indefinida la lectura de códigos.

El valor del código leído se imprime por pantalla (usando el servicio `syscall` con código 1).

```
# inicializar el hardware
ini:  li    $t0 2
      out $t0 0x104

      # leer registro de control hasta que sea distinto de cero
bwait: in    $a0 0x100
       beqz $a0 bwait

      # si hay error, reiniciar el hardware
       li    $t0 -1
       beq $a0 $t0 ini

      # imprime el valor leído y volver a leer
       in $a0 0x108
       li $v0 1
       syscall

       b bwait
```