

## Test de Teoría (3.0p)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
c	a	b	c	d	a	b	c	d	a	b	c	d	d	a	a	b	c	d	d	d	c	b	d	b	d	a	d	d	c

## Test de Prácticas (4.0p)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
d	b	d	a	d	a	c	b	b	c	b	a	d	b	a	c	b	c	b	b

## Examen de Problemas (3.0p)

### 1. Ensamblador (0.75 puntos).

Se puntúa **0.05p** por celda (las tres celdas de modo dir. registro cuentan como una, total **0.05 x 15 = 0.75p**)

Instrucción	Fuente			Destino		
	Modo de direccionamiento	Dirección de memoria	Contenido o valor	Modo de direccionamiento	Contenido o valor antes de ejecutar la instrucción	Contenido o valor después de ejecutar la instrucción
<code>mov (%rcx),%al</code>	Indirecto (a través de registro)	0x60 1037	0x10	Registro	0	0x10
<code>shl \$0xc,%rax</code>	Inmediato (constante literal)	N/A	0x0c	Registro	0x10 (=16)	0x1 0000 (=65536)
<code>mov (%rbx,%rax),%rbx</code>	Indexado respecto a base	0x61 1038 + 0x01 0000 = 0x61 1038	0x10	Registro	0x60 1038	0x1010 1010 1010 1010

### 2. Ensamblador (0.75 puntos).

Se supone convención cdecl, a falta de más información sobre cómo se pasa el argumento puntero. Infinitas soluciones válidas entre 10-15 líneas, se valora brevedad, claridad, simplicidad... tal vez eficiencia.

Si programa correcto de 10-15 líneas, nota completa. Pasar de 15 líneas puntúa negativamente.

Correspondería a aproximadamente **0.05p** por línea/instrucción (**0.05 x 15 = 0.75p**)

**little2big:**

```
push %ebp
mov %esp, %ebp
```

```
mov 8(%ebp), %eax # arg. ptr
mov $100, %ecx # cont/idx
```

```
.bucle:
rolw $8, -2(%eax,%ecx,2)
```

```
dec %ecx # 100..1
```

```
jnz .bucle
```

```
pop %ebp
ret
```

**big2little:**

```
push %ebp
mov %esp, %ebp
```

```
mov 8(%ebp), %eax
xor %ecx, %ecx
```

```
.bucle:
rorw $8, (%eax,%ecx,2)
```

```
inc %ecx # 0..99
```

```
cmp $100, %ecx
```

```
jnz .bucle
pop %ebp
ret
```

**ALTERNATIVAS CUERPO BUCLE**

```
mov (%eax,%ecx,2), %dl
mov 1(%eax,%ecx,2), %dh
mov %dh, (%eax,%ecx,2)
mov %dl, 1(%eax,%ecx,2)
```

```
mov (%eax,%ecx,2), %dh
mov 1(%eax,%ecx,2), %dl
mov %dx, (%eax,%ecx,2)
```

```
mov (%eax,%ecx,2), %dx
mov %dh, (%eax,%ecx,2)
mov %dl, 1(%eax,%ecx,2)
```

```
mov (%eax,%ecx,2), %dx
xchg %dh, %dl
mov %dx, (%eax,%ecx,2)
```

### 3. Unidad de Control (0.5 puntos).

Solución (inspirada en <http://slideplayer.com/slide/11064764/>)

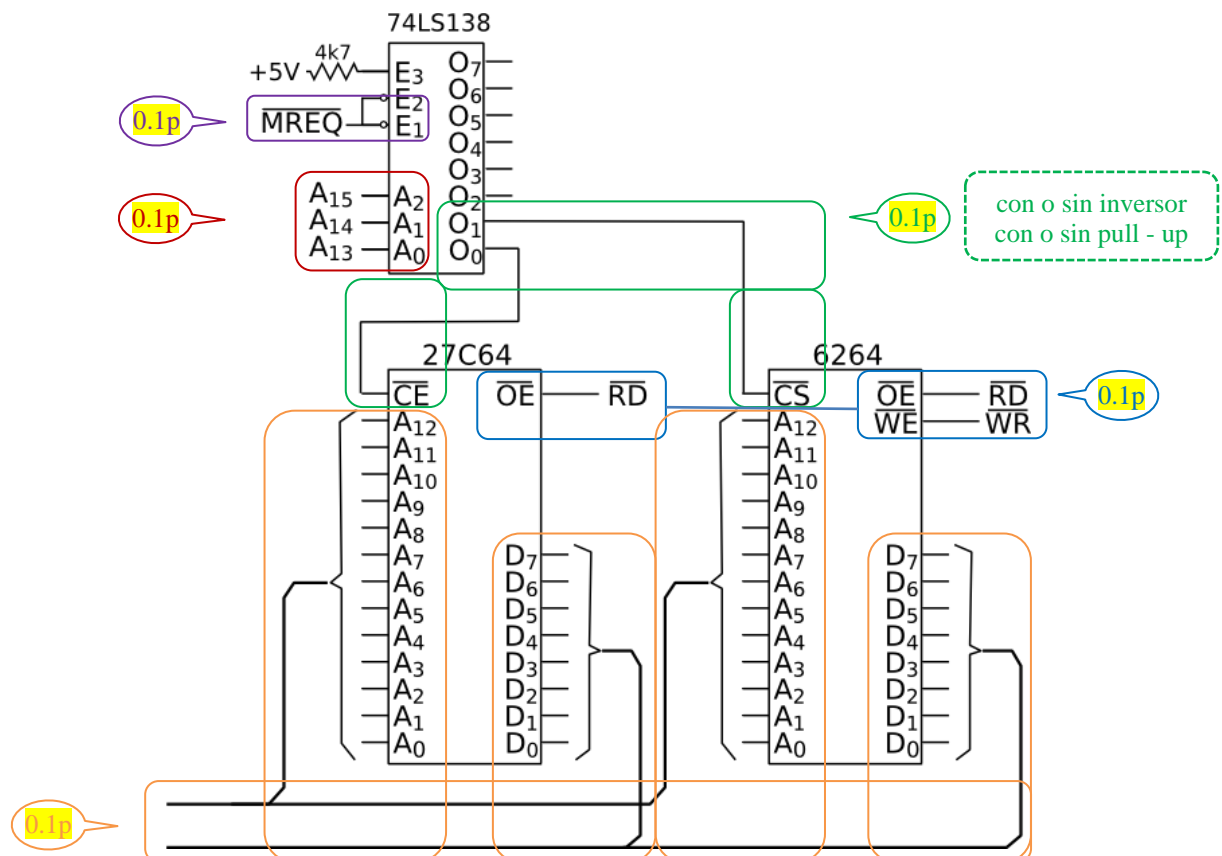
Se puntúa **0.05p** por micro-instrucción (aproximadamente 11 micro-instrucciones, una de regalo, total **0.05 x 10 = 0.50p**)

```
fetch:    MA:=PC; A:=PC
          PC:=A+4
          wait
          Inst:=Mem[MA]
          goto f(IR)
          ...
BEQ:      A:=Reg[rs1]
          B:=Reg[rs2]
          if (A-B != 0) goto fetch
          A:=PC
          B:=Immediate
          PC:=A+B; goto fetch
```

### 4. Configuración de memoria (0.5 puntos).

Solución (ver <https://www.eevblog.com/forum/beginners/z80-single-board-memory-bank-switching/>)

Aproximadamente **0.1p** por zona (**MREQ#**, **A<sub>15-13</sub>**, **CE#/CS#**, **OE#/WE#**, **A<sub>12-0</sub>/D<sub>7-0</sub>**)



### 5. Memoria cache (0.5 puntos).

64 B/línea =  $2^6$  B/línea  $\Rightarrow$  6 bits campo offset (desplazamiento, byte, ...)

L1i 16 x 64KB 4 vías  $\Rightarrow$  4 líneas/conjunto L1i

L1d 16 x 32KB 8 vías  $\Rightarrow$  8 líneas/conjunto L1d

MP 1TB =  $2^{40}$  B  $\Rightarrow$  40 bits dirección física

a) L1 instrucciones **(0.25p)**

$$L1i: 64 \text{ KB} / 64 \text{ B/línea} = 2^{16} \text{ B} / 2^6 \text{ B/línea} = 2^{10} \text{ líneas} (=1024)$$

$$1024 \text{ líneas} / 4 \text{ vías} = 2^{10} \text{ líneas} / 2^2 \text{ líneas/conjunto} = 2^8 \text{ conjuntos} \Rightarrow 8 \text{ bits campo conjunto}$$

$$\text{resto bits: etiqueta} = 40 - 8 - 6 = 26 \text{ bits campo etiqueta}$$

Dirección física de memoria principal desde el punto de vista de L1i: **(0.10p = 0.05p + 0.025p + 0.025p)**

<b>etiqueta (26)</b>	<b>conjunto (8)</b>	<b>byte (6)</b>
----------------------	---------------------	-----------------

Tamaño total en bits ocupado por todas las etiquetas en directorios L1i: **(0.05p)**

$$16 \text{ caches} \cdot 1024 \text{ líneas/cache} \cdot 26 \text{ bits/etiqueta} = 2^4 \times 2^{10} \times 26 \text{ bits} = 2^{14} \times 26 \text{ bits} = 425 \text{ 984 bits}$$

$$\text{alternativamente, una sola cache L1i} = 1024 \text{ líneas} \cdot 26 \text{ bits/etiqueta} = 26 \text{ Kbits}$$

Tamaño total en bits ocupado por todos los datos/instrucciones en L1i: **(0.05p)**

$$16 \text{ caches} \cdot 64 \text{ KB/cache} \cdot 8 \text{ bits/B} = 2^4 \times 2^{16} \times 2^3 \text{ bits} = 2^{23} \text{ bits} = 8 \text{ Mbits} = 8 \text{ 388 608 bits}$$

$$\text{alternativamente, una sola cache L1i} = 2^{16} \times 2^3 \text{ bits} = 2^{19} \text{ bits} = 512 \text{ Kbits}$$

$$\text{Porcentaje Etiquetas / (Datos/Instrucciones)} = 425 \text{ 984} / 8 \text{ 388 608} = 5.08\% \quad \text{(0.05p)}$$

$$\text{alternativamente, una sola cache L1i: } 26 \text{ Kb} / 512 \text{ Kb} = 5.08\%$$

b) L1 datos **(0.25p)**

$$L1i: 32 \text{ KB} / 64 \text{ B/línea} = 2^{15} \text{ B} / 2^6 \text{ B/línea} = 2^9 \text{ líneas} (=512)$$

$$512 \text{ líneas} / 8 \text{ vías} = 2^9 \text{ líneas} / 2^3 \text{ líneas/conjunto} = 2^6 \text{ conjuntos} \Rightarrow 6 \text{ bits campo conjunto}$$

$$\text{resto bits: etiqueta} = 40 - 6 - 6 = 28 \text{ bits campo etiqueta}$$

Dirección física de memoria principal desde el punto de vista de L1d: **(0.10p = 0.05p + 0.025p + 0.025p)**

<b>etiqueta (28)</b>	<b>conjunto (6)</b>	<b>byte (6)</b>
----------------------	---------------------	-----------------

Tamaño total en bits ocupado por todas las etiquetas en directorios L1d: **(0.05p)**

$$16 \text{ caches} \cdot 512 \text{ líneas/cache} \cdot 28 \text{ bits/etiqueta} = 2^4 \times 2^9 \times 28 \text{ bits} = 2^{13} \times 28 \text{ bits} = 229 \text{ 376 bits}$$

$$\text{alternativamente, una sola cache L1d} = 512 \text{ líneas} \cdot 28 \text{ bits/etiqueta} = 14 \text{ Kbits}$$

Tamaño total en bits ocupado por todos los datos/instrucciones en L1d: **(0.05p)**

$$16 \text{ caches} \cdot 32 \text{ KB/cache} \cdot 8 \text{ bits/B} = 2^4 \times 2^{15} \times 2^3 \text{ bits} = 2^{22} \text{ bits} = 4 \text{ Mbits} = 4 \text{ 194 304 bits}$$

$$\text{alternativamente, una sola cache L1d} = 2^{15} \times 2^3 \text{ bits} = 2^{18} \text{ bits} = 256 \text{ Kbits}$$

$$\text{Porcentaje Etiquetas / (Datos/Instrucciones)} = 229 \text{ 376} / 4 \text{ 194 304} = 5.47\% \quad \text{(0.05p)}$$

$$\text{alternativamente, una sola cache L1i: } 14 \text{ Kb} / 256 \text{ Kb} = 5.47\%$$