

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
d	a	d	a	c	d	c	d	c	a	a	c	a	a	d	a	d	d	b	c	a	d	b	c	c	c	b	d	b	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
a	d	b	a	a	a	d	a	c	a	d	b	b	d	c	c	b	b	c	a

Examen de Problemas (3.0p)

1. Ensamblador (1 puntos).

Se puntúa **0.1p** por cada función (total **0.1 x 10 = 1p**).

Múltiples soluciones válidas, cualquiera de ellas puntúa, sólo se sugieren algunas de ellas, en primer lugar la generada por gcc -Og. Si el resultado es correcto y cumple las restricciones no se pide más.

Algunas instrucciones u operandos mostrados como alternativas **no se han explicado en clase**, **no se esperaba** que nadie respondiera con dichas alternativas, se indican sólo como curiosidad.

	Opción 1	Opción 2	Opción 3	Opción 4
return x + x;	leaq (%rdi,%rdi), %rax	mov %rdi, %rax add %rdi, %rax	mov %rdi, %rax shl %rax	
return x * 5;	leaq (%rdi,%rdi,4), %rax	mov %rdi, %rax shl \$2, %rax add %rdi, %rax		
return x * 8;	leaq 0(, %rdi,8), %rax	mov %rdi, %rax shl \$3, %rax		
return x * 16;	movq %rdi, %rax salq \$4, %rax	mov %rdi, %rax shl \$4, %rax	lea (,%rdi,8), %rax add %rax, %rax	lea (,%rdi,8), %rax shl %rax
return x / 16;	movq %rdi, %rax shrq \$4, %rax	xor %rax, %rax shld \$-4, %rdi, %rax		
return x * 12;	leaq (%rdi,%rdi,2), %rdx leaq 0(,%rdx,4), %rax	lea (...2), %rax shl \$2, %rax	lea (...2), %rax add %rax, %rax add %rax, %rax	lea (...4), %rax add %rdi, %rax add %rax, %rax
return x % 8;	movq %rdi, %rax andl \$7, %eax	mov %rdi, %rax ror \$3, %rax shr \$61, %rax	mov %rdi, %rax ror \$3, %rax shr \$-3, %rax	xor %rax, %rax shrd \$3, %rdi, %rax rol \$3, %rax
return x & 0xffffffff;	movl %edi, %eax	mov %rdi, %rax and \$-1, %eax	mov %rdi, %rax and \$0xffffffff, %eax	
return x & 1;	movq %rdi, %rax andl \$1, %eax	mov %rdi, %rax and \$1, %rax	xor %rax, %rax rcr %rdi rcl %rax	
return x != 0;	testq %rdi, %rdi setne %al movzbl %al, %eax	xor %rax, %rax test %rdi, %rdi setnz %al		

2. Unidad de Control (0.4 puntos).

Se puntúa **0.1p** por cada dibujo y **0.1p** por el valor final correcto. (total **0.1 x 3 + 0.1 x 1 = 0.4p**).

Ejemplo tomado de https://images.slideplayer.com/11/3265187/slides/slide_26.jpg

Ver dibujo al final de la plantilla

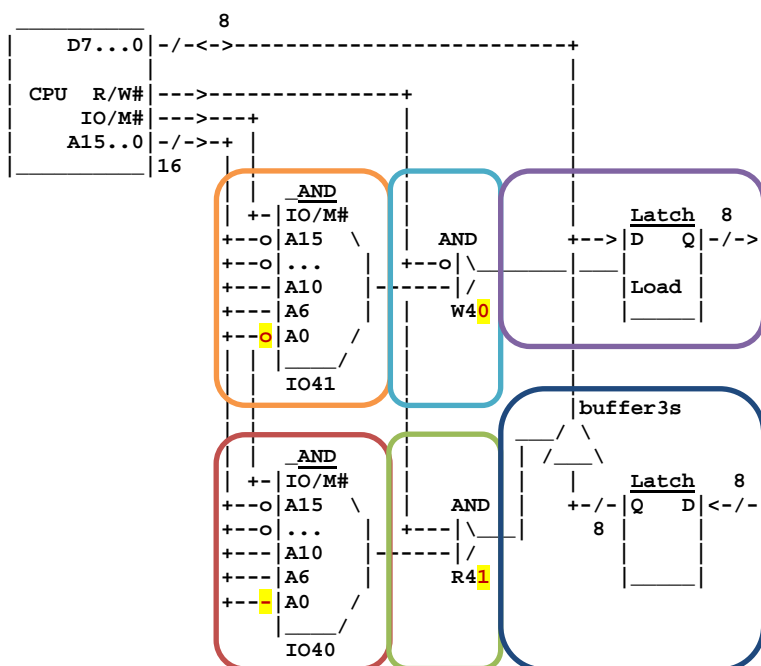
$$2048 * 41 - (2048 * 7 + 100 * 41) = 83968 - 18436 = \mathbf{65532}$$

3. Entrada/Salida (0.6 puntos).

Se puntúa **0.1p** por cada zona del dibujo. (total **0.1 x 3 + 0.1 x 3 = 0.6p**).

Como el Problema 4 de Septiembre 2015, pero intercambiando direcciones puertos. Modificaciones **resaltadas**.
Solución copiada de aquella plantilla, intercambiando el bit menos significativo

```
Hex 0x0  4  4  1
Bin 0000 0100 0100 0001
      A10 A6  A0
```



4. Mapa de memoria (0.6 puntos).

Se puntúa **0.1p** por el valor en a) y **0.05p** por cada dirección en b) (total **0.1 + 0.05 x 10 = 0.60p**).

Ejemplo tomado de <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0201d/I21752.html>

4 M + 4 M + 248 M = 256 M

4096 M / 256 M = $2^{32}/2^{28}=2^4=16$; n = 16 - 1 = **15**

0x0000 0000	Zona de 4MB
0x003F FFFF	
0x0040 0000	Zona de 4MB
0x007F FFFF	
0x0080 0000	Zona de 248MB
0x0FFF FFFF	
0x1000 0000	Zona 1 de 256MB
0x1FFF FFFF	
...	
0xF000 0000	Zona n=15 de 256MB
0xFFFF FFFF	

5. Memoria cache (0.4 puntos).

Se puntúa **0.1p** por cada número, y **0.1p** por los 3 nombres de los campo (total **0.1 x 3 + 0.1 = 0.4p**).

MP: 4 GB = 2^{32} B

Cache: 4 KB = 2^{12} B, 32 B/línea = 2^5 B/línea, 4 vías = 2^2 líneas/conjunto

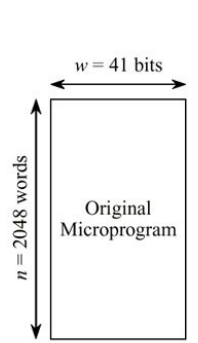
Cache: 2^{12} B / 2^5 B/línea = 2^7 líneas, 2^7 líneas / 2^2 líneas/conjunto = 2^5 conjuntos

Dirección física de memoria principal desde el punto de vista de cache:

etiqueta (22)	conjunto (5)	byte (5)
------------------	-----------------	-------------

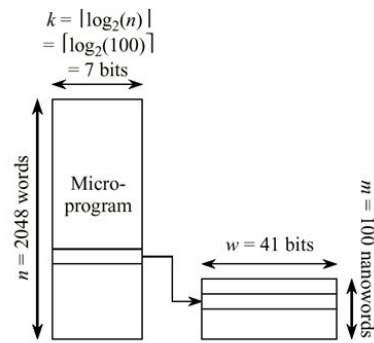
Microprogramming vs. Nanoprogramming

- (a) Microprogramming vs. (b) nanoprogramming.



$$\text{Total Area} = n \times w = 2048 \times 41 = 83,968 \text{ bits}$$

(a)



$$\text{Microprogram Area} = n \times k = 2048 \times 7 = 14,336 \text{ bits}$$

$$\text{Nanoprogram Area} = m \times w = 100 \times 41 = 4100 \text{ bits}$$

$$\text{Total Area} = 14,336 + 4100 = 18,436 \text{ bits}$$

(b)