

Examen Test (3.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.
Cada respuesta vale 3/30 si es correcta, 0 si está en blanco o claramente tachada, -1/30 si es errónea.
Anotar las respuestas (a, b, c o d) en la siguiente tabla.

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

Examen de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.
Cada respuesta vale 4/20 si es correcta, 0 si está en blanco o claramente tachada, -1.33.../20 si es errónea.
Anotar las respuestas (a, b, c o d) en la siguiente tabla.

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

Examen de Problemas (3.0p)

1. Bucles while (0.5 puntos).

A.

```
int fun_a(unsigned x) {
    int val = 0;
    while (x) {
        val ^= x;
        x >>= 1;
    }
    return val & 0x1;
}
```

B. Calcula la paridad de x (devuelve 1 si hay nº impar de 1s, 0 si par)

2. Representación y acceso a estructuras (0.5 puntos).

A. struct P1 { int i; char c; int j; char d; };

i	c	j	d	total	alineamiento
0	4	8	12	16	4

B. struct P2 { int i; char c; char d; int j; };

i	c	d	j	total	alineamiento
0	4	5	8	12	4

C. struct P3 { short w[3]; char c[3]; };

w	c	total	alineamiento
0	6	10	2

D. struct P4 { short w[3]; char *c[3]; };

w	c	total	alineamiento
0	8	20	4

E. struct P5 { struct P1 a[2]; struct P2 *p };

a	p	total	alineamiento
0	32	36	4

3. Unidad de control (0.3 puntos).

La UC puede controlar simultáneamente $n+k$ señales de control, aunque la unidad de procesamiento tiene un total de $n \cdot 2^m + k$ señales de control (si todos los 2^m códigos de cada uno de los n campos codificados son válidos)

Ver dibujo adjunto

4. Unidad de control (0.2 puntos).

Ahorro ninguno, se gastan 160 bits más.

$$340 \times 16 < (340 \times 8 + 180 \times 16) = (170 + 180) \times 16 = 350 \times 16$$

$340 \times 16 < 350 \times 16$, se gastan 10×16 bits más

Ver dibujo adjunto

5. Entrada/Salida (0.5 puntos).

$210_{\text{perif}} \times 4_{\text{puertos}} = 840_{\text{puertos}} = 0x348$

Si empiezan desde el final, $0x2000 - 0x0348 = 0x1cb8$

E/S ocuparía $0x1cb8 \dots 0x1fff$

Si pregunta M máxima, $0x0000 \dots 0x1cb7 = 8K - 840 = 7352$ palabras

Ver dibujo adjunto

6. Diseño del sistema de memoria (0.5 puntos).

ROM $0x0000-0x1fff = 2^{13} = 8K$ pal = $8K \times 16$. En módulos de $4K \times 4 \times (2 \times 4) \rightarrow 8$ módulos

SRAM $0xC000-0xffff = 3 \times 2^{12} = 12K$ pal = $12K \times 16$. En m.de $4K \times 8 \times (3 \times 2) \rightarrow 6$ módulos

Todos los módulos conectados a $A_{11} \dots A_0$, decodificación con $A_{15} \dots A_{12}$ (1^{er} dígito hex)

Ver dibujo adjunto

7. Memoria cache (0.5 puntos).

- A. Directa: $20 = 7 + 8 + 5$. etiqueta+marco+palabra. $2^{13} \text{pal} / \text{cach} / 2^5 \text{pal} / \text{bloq} = 2^8 \text{bloq} / \text{cache}$
- B. Asociativa: $20 = 15 + 5$. etiqueta +palabra.
- C. Asociativa Conjuntos 4vías: $20 = 9 + 6 + 5$. etiqueta+conjnt+palabra. $2^8 \text{bloq} / \text{cach} / 2^2 \text{blq} / \text{conj} = 2^6 \text{conj} / \text{cache}$
- D. Sectores de 4palabras: $20 = 13 + 2 + 5$. etiqueta+bloque+palabra. $4 = 2^2 \text{bloq} / \text{sector}$

Ver dibujo adjunto