

<b>Nombre:</b>	
<b>DNI:</b>	<b>Grupo:</b>

### Examen de Problemas (3,0 p)

1. **Ensamblador.** (0.5 puntos). Dado el siguiente fragmento de código escrito en C, escriba el código de la función equivalente utilizando el ensamblador de IA32:

```
int max (int a, int b)
{
    if (a > b)
        return a;
    else
        return b;
}
```

Para el desarrollo de este problema ha de seguirse la convención de paso de parámetros `_cdecl`.

2. **Ensamblador** (0.5 puntos). Considerar la siguiente declaración de estructura:

```
struct lio {
    char escalon;
    long ensambl;
    struct toc {
        char web[3];
    } apps;
    int *programar;
    struct lio *mates;
};
```

Debajo se muestran tres funciones C y tres bloques de código Linux x86-64:

```
char *rookie(struct lio *ptr) {
    return &(ptr->apps.web[2]);
}
```

A	mov 0x20(%rdi), %rax mov 0x8(%rax), %rax retq
---	---

```
int hacker(struct lio *ptr) {
    return *(ptr->programar);
}
```

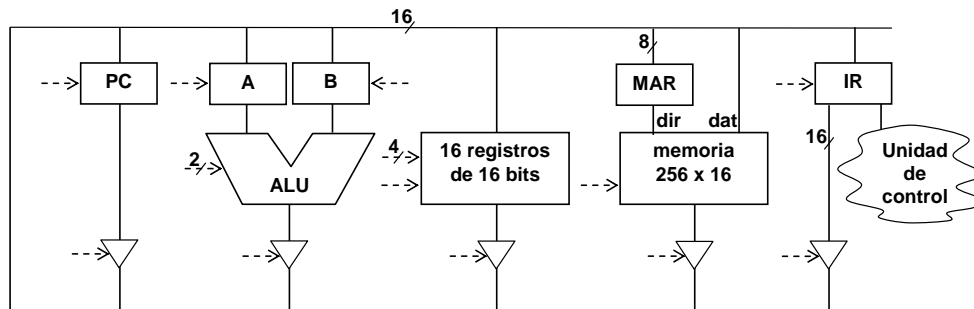
B	lea 0x12(%rdi), %rax retq
---	------------------------------

```
long guru(struct lio *ptr) {
    return ptr->mates->ensambl;
}
```

C	mov 0x18(%rdi), %rax mov (%rax), %eax retq
---	--

Anotar en una tabla de 3 filas y 2 columnas, en la columna izquierda el nombre (A,B,C) de cada bloque de código x86-64, y en la columna derecha el nombre de la función C correspondiente (implementada por dicho código ensamblador).

3. **Unidad de control** (0.5 puntos). La figura muestra el camino de datos de un procesador de 16 bits en el que cada instrucción y cada dato ocupa una palabra. La ALU puede realizar las operaciones  $A+B$ ,  $A-B$ ,  $A+1$ ,  $A$ . Etiquete las señales de control de la figura y escriba (en lenguaje de transferencia de registros o de alto nivel) la fase de captación de instrucción y la sección del microprograma para implementar una instrucción  $\text{ADD } r, [x]$ , que suma el contenido del registro  $r$  al contenido de la dirección de memoria  $x$ , escribiendo el resultado en esa misma dirección de memoria  $x$ .



4. **Entrada/Salida** (0.5 puntos). Un sistema basado en un microprocesador de 8 bits (8 bits de datos y 16 bits de direcciones) con E/S independiente debe conectarse a 8 conmutadores de 1 bit (entradas) y a 8 LED (salidas). Dibuje un esquema detallado del sistema y escriba un programa en pseudocódigo que muestre constantemente, en un bucle, el estado de los conmutadores en los LED (si un conmutador está a 1, el LED correspondiente estará encendido). El bucle finalizará cuando todos los conmutadores estén a 0, dejando por tanto todos los LED apagados. Aclaración: los LED no pueden conectarse directamente a los conmutadores sino que el estado debe pasar por el microprocesador.
5. **Jerarquía de memoria** (0.25 puntos). Suponga una jerarquía de memoria con dos niveles: caché L2 (sin L1) y memoria principal. La caché tiene un tiempo de acceso  $t_{L2}$  de 1,9 ns y una tasa de aciertos  $A_{L2}$  del 98% y la memoria principal tiene un tiempo de acceso  $t_M$  de 55 ns. ¿Cuánto mejora la velocidad promedio de acceso a memoria añadiendo una caché L1 con un tiempo de acceso  $t_{L1}$  de 0,6 ns y una tasa de aciertos  $A_{L1}$  del 95%?
6. **Diseño del sistema de memoria** (0.5 puntos). El ZX Spectrum 48K era un computador con un bus de datos de 8 bits y un bus de direcciones de 16 bits, con la siguiente organización de memoria: el primer bloque de 16 KB consistía en un módulo ROM de  $16K \times 8$ . El segundo bloque, de 16 KB y situado a continuación del primero, estaba constituido por chips de memoria RAM dinámica de  $16K \times 1$ . El tercer y último bloque, de 32 KB, estaba formado por circuitos de memoria RAM dinámica de  $32K \times 1$ . Dibuje el mapa de memoria, con los rangos de direcciones en hexadecimal de cada bloque de memoria, y un esquema de la memoria con todos los chips en el que se detalle la conexión con los buses de datos (D7-D0), direcciones (A15-A0) y control (RD#, WR#, RAS#, CAS#).
7. **Memoria cache** (0.25 puntos). Un procesador Core 2 Duo puede direccionar 64 GB de memoria física y cada núcleo dispone de una caché L1 de 32 KB de instrucciones y otra caché L1 de 32 KB de datos, asociativa por conjuntos con 8 vías y con líneas de 64 bytes. Muestre el esquema de una dirección de memoria física y explique cómo sabe el procesador si una determinada dirección correspondiente a un byte de datos está en la caché L1 de datos o no.