

COGNOMS:	<input type="text"/>		
NOM:	<input type="text"/>	DNI/NIE:	<input type="text"/>

IMPORTANTE leer atentamente antes de empezar el examen: Escriba los apellidos, el nombre y el DNI/NIE antes de empezar el examen. Escriba un solo carácter por recuadro, en mayúsculas y lo más claramente posible. Es importante que no haya tachones ni borrones y que cada carácter quede enmarcado dentro de su recuadro sin llegar a tocar los bordes. Use un único cuadro en blanco para separar los apellidos y nombres compuestos si es el caso. No escriba fuera de los recuadros, todo lo que haya fuera de ellos es ignorado. La identificación del alumno se realiza de forma automática, no seguir correctamente estas instrucciones puede comportar no tener nota.

Problema 1. (4 puntos)

Una empresa de automoción está desarrollando un sistema de conducción autónoma. Uno de los problemas a resolver consiste en procesar imágenes provenientes de una cámara mediante una red neuronal. Para realizar el procesamiento se ha seleccionado un modelo concreto de CPU. Procesar una sola imagen mediante la red neuronal requiere 10×10^9 operaciones de punto flotante para lo que se ejecutan un total de 15×10^9 instrucciones. Para garantizar tiempos de respuesta y niveles de seguridad significativamente superiores a los de los mejores conductores humanos se ha estimado que se necesitaría procesar 10 imágenes por segundo.

- a) **Calcula** el rendimiento en Gflops que necesitamos obtener para poder procesar 10 imágenes/s.

<input type="text"/>

Nuestra red neuronal tiene una implementación que usa un solo procesador (core) de la CPU. Se ha procesado una imagen y se ha estimado que para poder procesar 10 imágenes/s sería necesario que la CPU funcionase a una frecuencia de 50 GHz.

- b) **Calcula** el IPC (instrucciones por ciclo) del procesador de la CPU al procesar una imagen mediante la red neuronal.

<input type="text"/>

Lamentablemente, la tecnología actual no puede alcanzar frecuencias tan altas. Según el fabricante, la CPU puede funcionar a una frecuencia máxima de 5GHz. La empresa ha contratado un ingeniero de computadores que ha observado que la CPU dispone de 4 procesadores integrados (cores), pero que la implementación secuencial de la red neuronal solo aprovecha uno. Después de mucho esfuerzo, se ha conseguido paralelizar el 96% de la aplicación para un número arbitrario de cores, el 4% restante sigue siendo secuencial y se ejecuta en un solo core.

- c) **Calcula** el speed-up al procesar una imagen con la versión paralela respecto a la versión secuencial en un sistema con 4 procesadores (cores).

<input type="text"/>

Los programadores sugieren que se use una CPU de gama alta con más cores para poder procesar 10 imágenes/s.

- d) **Calcula** cuantos procesadores (cores) serían necesarios para poder procesar 10 imágenes/s con la CPU funcionando a 5 GHz

<input type="text"/>

El director del proyecto argumenta que una CPU con tantos cores no sería viable para el proyecto por el coste hardware y por el elevado consumo. El ingeniero de computadores propone programar los bucles principales en ensamblador para aprovechar las instrucciones SIMD de 256 bits. Con 256 bits es posible empaquetar 8 datos de simple precisión. Se ha observado que las instrucciones SIMD solo son aplicables a la parte paralela. El 80% del tiempo de la parte paralela son instrucciones de punto flotante que se han podido implementar completamente con SIMD consiguiendo un speed-up de 8x, el 20% restante ha visto reducido su tiempo de ejecución a la mitad debido a la reducción de las iteraciones de los bucles principales.

e) **Calcula** el speed-up de la parte paralela al optimizar el código usando instrucciones SIMD

f) **Calcula** a que frecuencia deberíamos hacer funcionar la CPU en un sistema con 4 cores e instrucciones SIMD de 256 bits una vez paralelizado y optimizado el código, para procesar 10 imágenes/s

Finalmente, después de algunas optimizaciones adicionales, es posible procesar 10 imágenes/s haciendo funcionar la CPU a 4 Ghz. La CPU se alimenta con 1,2 V, tiene una carga capacitiva equivalente de 15 nano Faradios (nF) y una corriente de fugas de 10 Amperios (A).

g) **Calcula** la potencia de fugas, la potencia de conmutación y la potencia total de la CPU

La siguiente tabla muestra los componentes del sistema, la cantidad de componentes usados y el tiempo medio hasta fallo en horas:

Componente	Cantidad	MTTF
CPU	1	1.000.000
Placa Base	1	200.000
DIMM de memoria	2	500.000
Fuente alimentación	1	100.000

h) **Calcula** el tiempo medio hasta fallo del sistema (MTTF)

[illegible]

NOM:											DNI/NIE:								
------	--	--	--	--	--	--	--	--	--	--	----------	--	--	--	--	--	--	--	--

Problema 2. (3 puntos)

Dado el siguiente código escrito en C:

```
typedef struct {
    char *a;
    short *b;
    short int c[2019];
    char d;
} X;

X S;
```

- a) **Dibuja** la estructura S identificando claramente el tamaño de la estructura, el tamaño de cada elemento y el offset de cada elemento respecto al inicio de la estructura.



- b) **Escribe** UNA ÚNICA INSTRUCCIÓN que permita almacenar S.d en la parte baja del registro **%eax** suponiendo que la dirección de la estructura S está almacenada en el registro **%ebx**.
Indica claramente la expresión aritmética que has usado para el cálculo de la dirección de S.d.

- c) **Escribe** una instrucción que permita almacenar `S.c[i]` en la parte baja del registro `%edi` suponiendo que la dirección de la estructura `S` está almacenada en el registro `%ebx` y que la variable `i` está almacenada en el registro `%ecx`. **Indica** claramente la expresión aritmética que has usado para el cálculo de la dirección.

--

COGNOMS:

NOM: DNI/NIE:

Problema 3. (3 puntos)

Dado el siguiente código escrito en C

```
void traspuesta(int A[10][10],int B[10][10],int dim);
int d,A[10][10],B[10][10];
int main(){
    ...
    traspuesta(A,B,d);
}
void traspuesta(int A[10][10],int B[10][10],int dim){
    int i,j;
    for( i = 0; i < dim; i++ )
        for( j = 0; j < dim; j++ )
            B[j][i]=A[i][j];
}
```

- a) **Enmarca** qué instrucciones en ensamblador del IA32 se corresponden con los 13 pasos de la Gestión de Subrutinas. Enmarca las instrucciones como se muestra para el caso **(6)**. Marca un paso sin instrucciones con una caja vacía.

- (1) Paso de parametros
- (2) Llamada subrutina
- (3) Enlace dinamico, puntero bloque de activacion
- (4) Reserva espacio variables locales
- (5) Salvar estado llamador
- (6) Cuerpo subrutina
- (7) Mover resultado a eax
- (8) Restaura estado
- (9) Elimina variables locales
- (10) Deshacer enlace dinamico
- (11) Retorno de subrutina
- (12) Elimina parametros
- (13) Recoger/usar el resultado

```
main:
    pushl d
    pushl $B
    pushl $A
    call traspuesta
    addl $12, %esp
```

...

```
traspuesta:
    pushl %ebp
    movl %esp, %ebp
    subl $8, %esp
    pushl %ebx
    pushl %esi
    pushl %edi
```

```
movl 8(%ebp), %ebx
movl 12(%ebp), %esi
movl 16(%ebp), %edi
# resto de instrucciones
# del cuerpo de la subrutina
```

(6)

```
popl %edi
popl %esi
popl %ebx
movl %ebp, %esp
popl %ebp
ret
```

- b) **Traduce** a ensamblador del IA32 las instrucciones de la rutina traspuesta que faltan en la caja etiquetada como (6) en el apartado a).