COGNOMS:

NOM: DNI/NIE:

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. (5 puntos)

Los supercomputadores modernos están compuestos por miles de nodos de cálculo interconectados entre si. Un nodo de cálculo tiene una configuración similar a un computador personal (ver tabla de la siguiente pagina). Se quiere actualizar un supercomputador cambiando los procesadores actuales (fabricados en tecnología de 14 nm) por la nueva generación (fabricada en tecnología de 10 nm). Los nuevos procesadores son compatibles a nivel de zócalo por lo que no se requiere cambiar ningún componente más. La siguiente tabla muestra algunas características de ambos procesadores:

tecnología	área	dados	die yield	F	C	V	I	cores por dado	IPC(A)
14 nm	400 mm2	140	90%	3,5 GHz	12,5 nF	1,2 V	10 A	20	2
10 nm	600 mm2	90	70%	3,0 GHz	15,0 nF	1,0 V	15 A	40	2,5

La columna “dados” indica el número de dados (dies) completos (una vez descontados los incompletos de los bordes) que caben en cada oblea. Sabemos que el coste de procesar una oblea es el mismo en ambos casos y que el procesador de 14nm se vendió por 5000 €. También sabemos que en ambos casos el fabricante vende un procesador al doble del coste de un dado funcional (en este margen se incluye el empaquetado, testeo, comercialización y beneficio del fabricante).

a) **Calcula** el precio de venta de cada procesador de 10 nm (pista: calcula los dados buenos en cada caso).

Las columnas F, C, V, I indican: frecuencia (F), carga capacitiva equivalente del chip (C), voltaje de alimentación (V) y intensidad (corriente) de fugas (I). Para el consumo de energía solo tendremos en cuenta la potencia de conmutación y la potencia de fugas, la corriente de cortocircuito es despreciable, por lo que no la tendremos en cuenta.

b) **Calcula** la potencia consumida por cada uno de los dos procesadores.

c) **Calcula** la ganancia en consumo (**en %**) del procesador de 10 nm respecto al de 14 nm.

Una aplicación A utiliza un solo core de estos procesadores. El lenguaje máquina y el compilador es el mismo en ambos procesadores por lo que ambos ejecutan el mismo número de instrucciones dinámicas. La columna IPC(A) muestra el IPC (instrucciones por ciclo) que medimos en cada procesador al ejecutar dicha aplicación.

d) **Calcula** el speed-up (**en %**) del procesador de 10 nm respecto al de 14 nm en la aplicación A.

Un Ingeniero de computadores ha paralelizado la aplicación A de forma que en un nodo la parte paralela se ejecuta con un speedup proporcional al número de cores. La parte paralela representa el 96% del tiempo de ejecución de A

e) **Calcula** el speed-up de la versión paralela de A respecto a la versión secuencial en un nodo del procesador de 10 nm (40 cores).

La aplicación A se puede paralelizar para ejecutarse en múltiples nodos. Al igual que antes, el speed-up de la parte paralela es proporcional al numero de cores (40 por nodo en el procesador de 10 nm).

f) **Calcula** el número mínimo de nodos con procesadores de 10 nm (40 cores por nodo) que serán necesarios para obtener al menos un speed-up de 24 en la aplicación A respecto la versión secuencial.

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

Componente	CPU	Placa base	Fuente Alimentación	DIMMs memoria	Disco duro
Cantidad	1	1	1	8	2
MTTF (en horas)	1.000.000	500.000	250.000	1.000.000	200.000

g) **Calcula** el tiempo medio hasta fallo de un nodo (MTTF)

El supercomputador consta de 10.000 nodos idénticos. Cuando un nodo falla el supercomputador puede seguir funcionando, aunque con un rendimiento levemente degradado. El personal de mantenimiento cambia diariamente todos los nodos que fallan por nodos nuevos.

h) **Calcula** cuantos nodos cambian (en media) cada día

COGNOMS:

NOM: DNI/NIE:

Problema 2. (5 puntos)

Dado el siguiente código escrito en C, que compilamos para un sistema linux de 32 bits:

```
typedef struct {
  char c;
  char d[4];
  short e[2];
  float *f;
  int g;
  double h;
 } s1;
```

```
typedef struct {
  s1 v[1000];
  char a;
  } s2;
```

a) **Dibuja** como quedarían almacenadas en memoria las estructuras **s1** y **s2**, indicando claramente los desplazamientos respecto al inicio, el tamaño de todos los campos y el tamaño de los structs.

b) **Escribe** UNA ÚNICA INSTRUCCIÓN que permita mover **x.v[100].c** al registro **%al**, siendo **x** una variable de tipo **s2** cuya dirección está almacenada en el registro **%ecx**. Indica la expresión aritmética utilizada para el cálculo de la dirección.

c) **Escribe** UN CONJUNTO DE 2 INSTRUCCIONES que permita mover **x.v[y.g].d[2]** al registro **%al**, siendo **x** una variable de tipo **s2** cuya dirección está almacenada en el registro **%ecx** e **y** una variable de tipo **s1** cuya dirección está almacenada en el registro **%ebx**. Indica la expresión aritmética utilizada para el cálculo de la dirección.

Dado el siguiente código escrito en C:

```
int examen(int i, int v[10]) {
  int w[2];
  w[0] = 1;
  if (i <= 10)
    w[0] = v[i];
  pepe();
  return w[0];
}
```

d) **Dibuja** el bloque de activación de la rutina examen, indicando claramente los desplazamientos respecto a **%ebp** y el tamaño de todos los campos.

e) **Traduce** a ensamblador del x86 la rutina examen.