Cognoms:	Nom:
Examen Final d'Arquitectura de Computadors	Curs 2012-2013 Q1

Problema 1. (4 puntos)

Disponemos del diseño de dos procesadores que son totalmente compatibles a nivel de lenguaje màquina. Estos procesadores se denominan: HP de "High Performance" y LP de "Low Power". La siguiente tabla muestra las caracteristicas más relevantes: carga capacitiva equivalente (C), voltaje de alimentación (V), intensidad de fugas (I_f), frecuencia de funcionamiento (F) y CPI de una aplicación A (obtenido con el mismo ejecutable para ambos procesadores).

Procesador	С	V	I _f	F	СРІ
НР	10 nF	1,2 V	10 A	4 GHz	0,5 c/i
LP	2 nF	0,9 V	1 A	1 GHz	1,5 c/i

a)	a) Calcula el speed-up del procesador HP respecto el LP al ejecutar la aplicación A.											

Para calcular la disipación de potencia tendremos en cuenta la potencia debida a conmutación y la debida a fugas (la potencia debida a cortocircuito es despreciable).

b) **Escribe** la fórmula para cada una de las fuentes de disipación de potencia y **calcula** la potencia disipada por dicha causa para ambos procesadores.

	Potencia conmutación (Pc)	Potencia fugas (Pf)	Potencia total (Pt)
Formula			
procesador HP			
procesador LP			

c)	Calcula la ganancia en energía del procesador LP respecto al procesador HP al ejecutar la aplicación A.

El TDP (Thermal Design Point) de un chip es la potencia máxima que el fabricante garantiza puede disipar sin que su funcionamiento se vea comprometido. Nuestros diseñadores de circuitos nos han asegurado que el área no va a ser un problema, pero nos han comentado que disponemos de un TDP màximo de 92W para los procesadores, por lo que en un chip podemos poner o bien 1 solo procesador HP (que denominaremos chip H-1) o bien 36 procesadores LP (que denominaremos chip L-36).

Es posible paralelizar parte de la aplicación A de forma que el trabajo se distribuya equitativamente entre los procesadores disponibles (el overhead debido a sincronización es despreciable). La parte sequencial se seguirá ejecutando en un solo procesador independientemente de los que haya disponibles en el chip. Tanto la parte sequencial como la paralela consiguen el mismo CPI que la aplicación original en los respectivos procesadores (1,5 c/i en un procesador LP y 0,5 c/i en un procesador HP).

A uno de los diseñadores se le ha ocurrido que, dado que el área no es un problema, podríamos poner 1 procesador HP y 36 LP en el mismo chip, de forma que la parte sequencial de la aplicación se ejecute en el procesador HP y la paralela en los 36 procesadores LP. Durante la ejecución de la parte secuencial los 36 procesadores LP estarán apagados, mientras que durante la ejecución de la parte paralela el procesador HP estará apagado, de forma que nunca se superará el TDP máximo 92W. A esta tercera opción la denominaremos chip HL-136.

d) **Rellena** la siguiente tabla con el **speed-up** que se obtendría con cada uno de los chips mencionados, en función del % de la parte paralela, **respecto a 1 procesador LP** (justifica muy brevemente cada respuesta).

Chip	0% (totalmente secuencial)	100% (totalmente paralela)	60% (el 40% restante es secuencial)
H-1			
(1 procesador HP)			
L-36			
(36 procesadores LP)			
HL-136			
(1 procesador HP + 36 procesadores LP)			

En el chip HL-136 ejecutamos una aplicación B (que tarda 2 horas) compuesta por 3 fases:
• La Fase 1 (30 minutos) es totalmente sequencial y se ejecuta en el procesador HP (los 36 procesadores LP están apagados).
• La Fase 2 (30 minutos) es totalmente paralela y se ejecuta en los 36 procesadores LP (el procesador HP está apagado).
• La Fase 3 (1 hora) es una fase de entrada/salida en que la CPU tiene una actividad muy baja, por lo que no importa la velocida de la CPU. Para reducir el consumo, la Fase 3 se ejecuta en uno de los procesadores LP (el procesador HP y 35 de la procesadores LP estan apagados).
e) Calcula la potencia media disipada por el chip HL-136 al ejecutar la aplicación B
La Fase 3 de entrada/salida se corresponde exclusivamente a acesos a disco y realiza un 60% de lecturas y un 40% de escrituras. sistema de disco esta formado por un RAID 5 de 10 discos. Cada uno de estos discos tiene una capacidad de 1 Terabyte, un ancide banda efectivo de 200 Mbytes/s. Tanto las escrituras como las lecturas a disco se corresponden a accesos aleatorios. controlador del RAID ordena las peticiones de acceso para aprovechar al máximo la concurrencia entre discos. En ambos cas consideraremos que disponemos de suficientes peticiones de lectura/escritura de forma que el controlador del RAID siemp puede aprovechar el ancho de banda de todos los discos físicos.
f) Calcula la capacidad útil y el ancho de banda efectivo del RAID 5.
Este RAID 5 lo sustituimos por un RAID 51 formado por 2 grupos de 10 discos.
g) Calcula la capacidad útil y el ancho de banda efectivo del nuevo RAID 51.
Sabemos que el tiempo de la Fase 3 depende exclusivamente de la velocidad a la que se transfieren los datos hacia/desde sistema de disco.
h) Calcula el tiempo de la aplicación B con el RAID 51.

h)	Calcula el tiempo de la aplicación B con el RAID 51.

Cognoms:	Nom:
Examen Final d'Arquitectura de Computadors	Curs 2012-2013 Q1

Problema 2. (2 puntos)

Dado el siguiente código en C:

a) Traduce la rutina Final a ensamblador del x86.

Dado el siguiente código escrito en C:

```
typedef struct {
  char c1;
  char c2;
  char *pc;
  short M[3][3]
  int i;
} X;

int Rutina(char c, short s, int V[100], X uno, X *dos, int i){
  X vx[10];
  int j;
  ...
}
```

b) **Dibuja** como quedaría almacenada en memoria la estructura **X**, indicando los deplazamientos respecto al inicio y el tamaño de todos los campos. **Dibuja** el bloque de activación de la función **Rutina**, indicando los desplazamientos relativos al registro EBP necesarios para acceder a los parámetros y a las variables locales y el tamaño de todos los parámetros y variables locales.

	gnoms:													• • •			• • •	. Nom:
Proble	ema 3. (2 p	unto	os)														
Dado e	el siguier	ite c	ódig	o es	crito	en e	ensar	mbla	dor	del x	86:							
	<pre>jge end a) movl (%ebx, %esi, 4), %eax b) addl %eax, 4*1024(%ebx, %esi, 4) c) addl 12*1024(%ebx, %esi, 4), %eax addl \$128, %esi jmp for</pre>																	
	tivo con		-			-			_		-	_						
-	ara cada le las 16									s (a),	, (b)	y (c)), inc	dica	a qu	é pá	gina	de la memoria virtual se accede en cada una
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
a																		
b c												<u> </u>						
	ndica cua olanco)	o 0	de la	os ac	cceso	os a 1	mem	noria	7	cutac	dos e	en la	s 16	T	13		racio	nes son fallo de TLB (F) (deja los aciertos en
					<u> </u>				<u> </u>	<u> </u>		<u> </u>	<u> </u>					
c) C	Calcula la	can	tidad	de l	acie	rtos	de T	LB er	n TO	DO e	l bu	cle						
d) C	Calcula la	can	tidac	d de	fallo	os de	TLB	en T	ODC) el b	oucle							

Problema 4. (2 puntos)

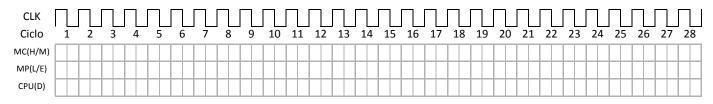
Disponemos de un procesador conectado a un sistema de memoria con una memoria cache de nivel 1 de datos con política de escritura copy back + write allocate.

a) **Define** de forma clara y concisa el comportamiento de la política copy back + write allocate en caso de lectura de un dato con fallo en L1.



Sabiendo que el tiempo de leer o escribir un bloque en Memoria Principal (latencia + transmisión de datos) es de 9 ciclos.

b) Rellena el siguiente cronograma, indicando la ocupación de los distintos recursos del sistema para una operación de lectura de un byte que provoca un fallo en el acceso a la cache, sabiendo que el bloque a reemplazar ha sido modificado en la memoria cache por el acceso inmediatamente anterior (usa H/M para hit/miss de MC, L/E para marcar los ciclos en que se lee/escibe en MP y D para indicar en que ciclo recibe el dato la CPU).



El sistema tiene una memoria principal formada por DIMMs de memoria DDR con 8 chips de 1 byte por DIMM, una latencia de fila de 2 ciclos, una latencia de columna de 2 ciclos y una latencia de precarga de 1 ciclo.

A la memoria DDR se realizan accesos de lectura/escritura en los que se lee/escribe un bloque de 64 bytes. Para indicar la ocupación de los distintos recursos utilizaremos la siguiente nomenclatura:

- AC: ciclo en que se envía el comando ACTIVE
- RD: ciclo en que se envía el comando READ
- PR: ciclo en que se envía el comando PRECHARGE
- @F: ciclo en que se envía la dirección de fila
- @C: ciclo en que se envía la dirección de columna
- D: transmisión de un paquete de datos

Una forma de mejorar el rendimiento de los acceso a MP consiste en aprovechar la localidad espacial de las filas de memoria (que los fabricantes llaman páginas). Para este apartado supondremos que el sistema tiene un controlador de memoria avanzado que no cierra la página (PRECHARGE) después de cada acceso. En caso de que un acceso se realice sobre la página abierta, no es necesario abrirla (ACTIVE). Sin embargo si el acceso se realiza sobre una página distinta, tenemos que cerrar la página anterior (PRECHARGE) y abrir (ACTIVE) la página que se desea acceder.

- c) **Rellena** el siguiente cronograma, indicando la ocupación de los distintos recursos del sistema con el controlador de memoria avanzado para la siguiente secuencia de accesos (inicialmente no hay ningúna página abierta), de forma que la secuencia se realize en el número mínimo de ciclos:
- Lectura de un bloque en la página x
- Lectura de un bloque en la página x
- Lectura de un bloque en la página y

