COGNOMS:																											
NOM:]	DNI/N	IIF.									
NOIVI.		_]	Divi	•1.									
empezar el que no hay bordes. Use de los recu automática Problema Queremos p	exam a tach e un ú uadros , no so 1. (3,	none nico s, to egui	Escies ni cua odo ir co unte	riba i bo idro lo d rred	un rro en que ctar	solo nes blai hay nent	o cara y que nco p /a fu te es	ácter e cac era era tas ir	por da ca sepa de d	r re arác arac ello ucci	cua cter r los os es ione	dro, que ape s igr es pu	en ede e ellide nora uede	may enm os y do. e cor	rúscu narca nom La mpo	ulas ado abre idei rtar	y lo dent es con ntifica no t	más ro c mpu ació ene	s cla de si uesto n d r no	ram u re os s el a ta.	ento cuad i es d lum	e pos dro s el cas no s	sible sin II so. I	e. Es legai No e ealiz	imp r a t scri a d	oorta cocar ba fu e fo	inte los iera rma
direcciones void refe	de 32	bit	s, m	ape	o di	rect	o, taı	nañc	de	cac	che I	16 K	byte	es, t	ama	ño (de blo	oque	e 64	byt	es. I	∠a ru	tina	a pr	ogr	amaı	
a) Escrib																									LLU	La	
,	unsi				_		-								l b				•								
	unsi	_				_	_		//]		-																
	unsi						a_mc										se dich		-			bl	oqu	е			
En la implei datos globa		ació	n de	la d	cach	ne C o	ору l	oack	+ W	/rite	e All	loca	te , ւ	ın e	stud	lian	te ha	ded	clara	ido	las s	iguie	ente	s est	truc	tura	s de
unsigned		tag	gs[2	256];	//	vec.	tor	COl	rre	spc	ondi	ent	ie a	a la	a m	emor	ria	de	et	iqu	etas	3				
unsigned											_										dez						
unsigned b) Escrib addr	e un	frag	mer	nto	de	códi	igo e	n C	para	а са	alcu	lar I	a va	riab	ole b	ool		mi	ss c	uan					la d	lirec	ción
c) Escrib aparta		_				_		-						-		orre	spon	dier	nte.	Pue	des	usar	las	vari	able	es de	los

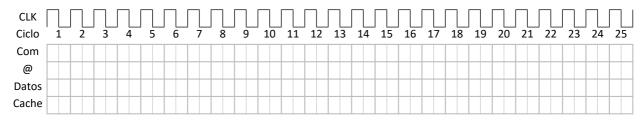
Esta cache está conectada a una memoria principal formada por un único módulo DIMM estándar de 4 Gbytes. Este DIMM tiene 8 chips de memoria **DDR**-SDRAM (**Double Data Rate** Synchronous DRAM) de un byte de ancho cada uno. El DIMM esta configurado para leer/escribir ráfagas de 64 bytes (justo el tamaño de bloque de las caches). La latencia de fila es de 4 ciclos, la latencia de columna de 3 ciclos y la latencia de precarga de 1 ciclo.

La cache no tiene buffers de escritura, por lo que siempre se escribe en memoria principal primero el bloque reemplazado (en caso que sea pertinente) antes de leer el nuevo bloque, tampoco tiene continuación anticipada ni transferencia en desorden. El controlador de memoria envía los comandos necesarios a la DDR-SDRAM de forma que ambos bloques sean transferidos lo más rápidamente posible y se maximice el ancho de banda.

En los siguientes cronogramas indica la ocupación de los distintos recursos de la memoria DDR: bus de datos (Datos), bus de direcciones (@), bus de comandos(Com). Indica también la ocupación de la cache (Cache), especifica cuando se accede a la cache si es acierto (A) o fallo (F), y cuando la CPU lee el dato de cache (LC) o escribe el dato en cache (EC). En todos los cronogramas supondremos que no hay ninguna página de DRAM abierta.

Para el caso fallo y escritura con dirty bit = 0

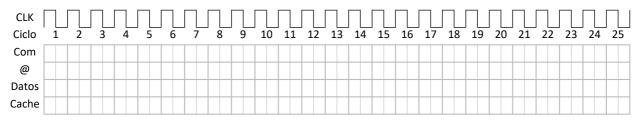
d) Rellena el siguiente cronograma.



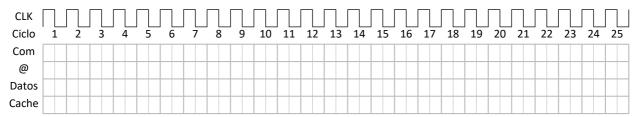
Para el caso fallo y lectura con dirty bit = 1

Rellena los siguientes cronogramas en función de la ubicación de los bloques involucrados.

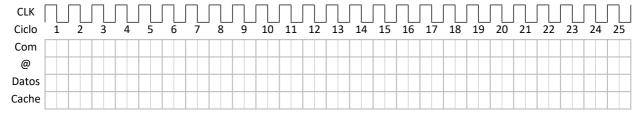
e) Ambos bloques están ubicados en el mismo banco pero en páginas distintas.



f) Ambos bloques están ubicados en la misma página.



g) Ambos bloques están ubicados en bancos distintos.



COGNOMS:																											
NOM:																	DNI	/NIE:									
Problema	2. (3	pui	nto	s)																							
En este prol que a lo lar centraremo en donde to (que usaren por tanto 20	go do s en odos nos a	el pr los a los los	robl acce acc argo	ema esos esos o de	a ası a in: s a n tod	umir struc nem o el	emos ccion oria t probl	s qu es. I tard lem	ie lo Hem lan i a) y	os ac nos s un c her	cceso simu ciclo nos o	os a lado En obte	dat o una el P enida	os s a ve idea o los	e pr rsiór al he s sigi	od n id em uie	duce deal os s ente	n siei de ui imula s date	mpr n pro ido	e ba oces la ej	ijo c ado ecu	ond r qu ción	icion e Ilaı de ı	nes io mare una i	deale emos aplic	es y s Picaci	nos deal, ón A
Llamaremos principal. El				-																							
a) Calcul a	a el C	CPI d	lel p	oroc	esac	dor P	mp a	al ej	ecu	tar l	а ар	lica	ción	A (0	Plp	mŗ	p).										
b) Calcula	el t	iem	ро	de p	ena	lizac	ión d	e lo	s ac	ces	os a	mei	mori	ia (e	n cio	clo	s) d	el pro	ces	ado	r Pn	np re	espe	cto e	el Pic	lea	l
Para mejora denominare tardan 2 cic fallo es de 2	mos los, e	pro es de	oces ecir	ado que	r Pc e en	i). Pa caso	ara p	ode acie	r m	ante tene	ener emo	la f s un	recu ia pe	ienc enali	ia de izaci	e 2 ón	2 GH n de	z, los	aco	eso	s a l	а са	che	de ir	nstru	ıcci	ones
c) Calcul a	a el t	iem	ро	de e	jecu	ıción	de la	а ар	lica	ciór	ı A e	n el	pro	cesa	dor	Pc	ci										

cache de instrucciones consume 5 nano joules, decodificar una instrucción CISC consume 3 nanojoules, y ejecutar micro-operación (uop) consume 4 nanojoules.	una
d) Calcula la energía consumida por el procesador Pci durante la ejecución de la aplicación A	
Algunos procesadores incluyen, ademas de una cache de instrucciones de nivel 1, una cache de uops (que denom de nivel 0) donde se guardan las uops que ya han sido traducidas para que en caso de hit en la uop-cache no necesario descodificar y traducir las instrucciones CISC. La introducción de una uop-cache reduce tanto el numero instrucciones CISC descodificadas cómo el número de accesos a la cache de instrucciones, el numero de accesos a se mantiene.	sea o de
En nuestro procesador (que ahora llamaremos Puop) hemos introducido una pequeña uop-cache. Para la aplicació gracias a la uop-cache, solo el 20% de las instrucciones CISC deben ser leídas de la cache de instruccion descodificadas. Sabemos que el tamaño de bloque de la cache de instrucciones es de 32 bytes, que cada instrucción tiene un tamaño de 4 bytes y cada uop tiene un tamaño de 6 bytes.	es y
e) Calcula el número de bytes leídos de la uop-cache, el numero de bytes leídos de la cache de instrucciones número de bytes que se leen de la memoria principal durante la ejecución de A.	y e
Una de las ventajas de la cache de micro-uops es que se produce un ahorro de energía ya que no es neces descodificar todas las instrucciones dinámicas. Un acceso a la uop-cache para leer una uop consume 2 nanojoules. f) Calcula la energía consumida por el procesador Puop durante la ejecución de la aplicación A.	

El procesador Pci es un procesador CISC en donde, en media, cada instrucción CISC se descompone durante la descodificación en 1,5 micro-operaciones (uops). Sabemos que un acceso a MP consume 40 nanojoules, un acceso a la

COGNOMS:																					
NOM:												DN	/NIE:								
Problema :	3. (3,5	puntos)																			
denominare	Se ha ejecutado un programa P en un sistema con un solo procesador y un solo disco D (un PC de sobremesa que denominaremos PC1) y se ha visto que su tiempo de ejecución es de T horas. Para poder estimar el rendimiento del programa P hemos medido (en el PC1) que el programa se ejecuta en tres fases bien diferenciadas:																				
Fase 1: Códi	go SEC	UENCIAL	que	no p	uede p	arale	eliza	irse,	ocup	a el 5	% de	l tiem	oo de	la e	jecu	ción d	de P e	n el	PC1.		
Fase 2: Cóo perfectame	_				-				-		a eje	ecuciór	n de	P er	n el	PC1.	Se p	oued	e pai	raleli	zar
Fase 3: Códi en el PC1.	Fase 3: Código de E/S que pasa todo su tiempo accediendo en el disco D, ocupa el 15% del tiempo de la ejecución de P en el PC1.																				
Con el objeto de reducir el tiempo de ejecución del programa, se baraja la opción de substituir el procesador del PC1 por un sistema multiprocesador de 32 procesadores idénticos al del PC1, manteniendo igual el resto del sistema. A este sistema multiprocesador le llamaremos PC2.																					
a) Calcul a	a el máx	kimo spe	ed-u	p que	e podr	ía co	nse	guirs	se al e	ejecut	ar el	progra	ıma P	con	el P	C2.					
la única qu	Sabemos que el programa P tiene 10^5 instrucciones estáticas y ejecuta 300×10^{12} instrucciones dinámicas. En la Fase 2, la única que realiza cálculos en punto flotante, se ejecutan 260×10^{12} instrucciones, de las cuales 150×10^{12} son instrucciones de coma flotante que realizan un total de 100×10^{12} operaciones de coma flotante en doble precisión.																				
b) Calcul a	a los MI	PS y MF	LOPS	al ej	ecutar	el pr	ogr	ama	P en	el PC	1 si l	a ejecu	ıción	tard	a 10	0x10 ⁵	3 segu	ındo	S.		
c) Calcul a	a los MI	PS y MF	LOPS	al ej	ecutar	el pr	rogr	ama	P en	el PC	2 en	vez de	en P	C1.							\neg

Otra opción que se ha barajado para mejorar el rendimiento del sistema PC1 es añadir un RAID de discos en lugar del disco duro D. El RAID nos permite paralelizar la Fase 3, ya que en esta fase hay suficientes accesos como para saturar el ancho de banda de todos los discos. El disco D tiene una capacidad de 1 Terabyte y un ancho de banda de 400 MBytes/s.

El RAID de discos del que disponemos tiene 6 discos como el D y puede configurarse como RAID 6 o RAID 51.

d)	Describe las principales características de cada uno de estos sistemas RAID, dibujando un esquema de cómo se distribuyen los datos y especificando el tipo de entrelazado, Terabytes de información redundante, número mínimo de discos que han de fallar para que el sistema deje de ser operativo, ancho de banda efectivo de las lecturas en acceso secuencial y ancho de banda efectivo de las escrituras en acceso secuencial.
dis	cidimos configurar el sistema de discos como RAID 6. Al sistema multiprocesador con 32 procesadores y RAID 6 de 6 cos le llamaremos PC3
e)	Calcula el speed-up máximo, sobre el PC2 , que podemos esperar al ejecutar el programa P en el PC3 , asumiendo que todos los accesos a disco son lecturas secuenciales .
	pemos que la Fase 3 tarda 15000 segundos con un solo disco D (PC1 y PC2) y que el 60% del tiempo se realizan lecturas uenciales y el 40% del tiempo se realizan escrituras aleatorias .
f)	Calcula el tiempo de la Fase 3 en el PC3 y el ancho de banda efectivo del RAID 6 durante la Fase 3
Ĺ	
	l l