Problema 1. (3 puntos) Una CPU de 32 bits tindrà un temps de cicle (TC) de 10 ns. A l'executar un programa P (que executa 5x1 instruccions) en un simulador on tots els accessos a memòria tarden 1 cicle s'ha mesurat un CPI de 1,8 cicle instrucció (que anomenarem CPI _{Ideal}). a) Calculeu el temps d'execució (T _{exec}) del programa P en aquest sistema de memòria ideal (en segons). Per mesurar l'impacte de la cerca d'instruccions en el rendiment, hem modificat el simulador per analitzar un sister amb una cache d'instruccions (que anomenarem 1,1) i una memòria principal SDRAM (els accessos a dades segueix tardant 1 cicle). La mida de les instruccions es de 4 bytes. La mida de bloc de L1 es de 32 bytes i el temps d'acces cas d'encert a L1 (Tn) es de 1 cicle. Pel programa P la taxa de fallades (m₂) de L1 es del 10%. b) Calculeu quants accessos a L1 fa el programa P. La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de fila de 4 cicles, u la tabacida de columna de 4 cicles in un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra a passos en cas de fallada a L1. En el cicle 1 s'accedeix a L1 les detecta que es una fallada de cache. En el cicle 2 s'envia la comanda READ l'adreça de columna. Al cicle 1 (d cicles després de RD) apareixen les dades (4 cicles bytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etternama READ l'adreça de columna. Al cicle 1 (d cicles després de RD) apareixen les dades (4 cicles bytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etternama READ l'adreça de columna. Al cicle 1 (d cicles després de RD) apareixen les dades (4 cicles bytes/cicle) al bus de dades (8 us 5). Enalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció a CPU (DADA) i en paral·lel s'escriu el bloc a L1 (cart.1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRECHARGE de CB)											
Una CPU de 32 bits tindrà un temps de cicle (Tc) de 10 ns. A l'executar un programa P (que executa 5x1 instruccions) en un simulador on tots els accessos a memòria tarden 1 cicle s'ha mesurat un CPI de 1,8 cicle instrucció (que anomenarem CPI _{idea}). a) Calculeu el temps d'execució (T _{exec}) del programa P en aquest sistema de memòria ideal (en segons). Per mesurar l'impacte de la cerca d'instruccions en el rendiment, hem modificat el simulador per analitzar un sister amb una cache d'instruccions (que anomenarem L1) i una memòria principal SDRAM (els accessos a dades segueix tardant 1 cicle). La mida de les instruccions es de 4 bytes. La mida de bloc de L1 es de 32 bytes i el temps d'accès: cas d'encert a L1 (T _h) es de 1 cicle. Pel programa P la taxa de falladaes (m ₁) de L1 es del 10%. b) Calculeu quants accessos a L1 fa el programa P. La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de fila de 4 cicles, u latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle després (cicle s'envia la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle després (cicle s'envia la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle després (cicle s'envia la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle sesprés (cicle s'envia la comanda PRECHARGE per tancar la pàgina (CICle) el bus (cicles ettiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRE LI MISS) la coma de l'emportation de l'accès carla la carla la pagina (PRE LI MISS) la carla la	3er Control Arquitect	tura de Com	outadors						Curs	201	1-2012
Una CPU de 32 bits tindrà un temps de cicle (Tc) de 10 ns. A l'executar un programa P (que executa 5x1 instruccions) en un simulador on tots els accessos a memòria tarden 1 cicle s'ha mesurat un CPI de 1,8 cicle instrucció (que anomenarem CPI _{idea}). a) Calculeu el temps d'execució (T _{exec}) del programa P en aquest sistema de memòria ideal (en segons). Per mesurar l'impacte de la cerca d'instruccions en el rendiment, hem modificat el simulador per analitzar un sister amb una cache d'instruccions (que anomenarem L1) i una memòria principal SDRAM (els accessos a dades segueix tardant 1 cicle). La mida de les instruccions es de 4 bytes. La mida de bloc de L1 es de 32 bytes i el temps d'accès: cas d'encert a L1 (T _h) es de 1 cicle. Pel programa P la taxa de falladaes (m ₁) de L1 es del 10%. b) Calculeu quants accessos a L1 fa el programa P. La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de fila de 4 cicles, u latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle després (cicle s'envia la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle després (cicle s'envia la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle després (cicle s'envia la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicle sesprés (cicle s'envia la comanda PRECHARGE per tancar la pàgina (CICle) el bus (cicles ettiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRE LI MISS) la coma de l'emportation de l'accès carla la carla la pagina (PRE LI MISS) la carla la	Droblema 1 (2 nuntos)										
instruccións) en un simulador on tots els accessos a memòria tarden 1 cicle s'ha mesurat un CPI de 1,8 cicle instrucció (que anomenarem CPI _{Ideal}). a) Calculeu el temps d'execució (T _{exec.}) del programa P en aquest sistema de memòria ideal (en segons). Per mesurar l'impacte de la cerca d'instruccions en el rendiment, hem modificat el simulador per analitzar un sister amb una cache d'instruccions (que anomenarem L1) i una memòria principal SDRAM (els accessos a dades segueix tardant 1 cicle). La mida de les instruccions es de 4 bytes. La mida de bloc de L1 es de 32 bytes i el temps d'accès: cas d'encert a L1 (T _h) es de 1 cicle. Pel programa P la taxa de fallades (m ₁) de L1 es del 10%. b) Calculeu quants accessos a L1 fa el programa P. La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de fila de 4 cicles, u latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra i la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria 14 cicles després (cicle s'envia la comanda READ i l'adreça de columna. Al cicle 10 (4 cicles després de RD) apareixen les dades (4 cicles yetyes/cicle) al bus de dades (Bus D). Les dades es van carregant a ran sm'es tot cicles CB) a mesura que van apareixe pet bus (cicles etiquetats D). Finalment al cicle 14, un cop s'ha transm'es tot cicles CB) a mesura que van apareixe pet bus (cicles etiquetats D). Finalment al cicle 11, accivia la comanda PRECHARGE per tancar la pàgina (PRE CICLE). La MISS la la mesura que coma la comanda PRECHARGE per tancar la pàgina (PRE Buffer La La MISS). La man man man man man man man man man ma				4.0	,				5.4	,	
Per mesurar l'impacte de la cerca d'instruccions en el rendiment, hem modificat el simulador per analitzar un sister amb una cache d'instruccions (que anomenarem L1) i una memòria principal SDRAM (els accessos a dades segueix tardant 1 cicle). La mida de les instruccions es de 4 bytes. La mida de bloc de L1 es de 32 bytes i el temps d'accés: cas d'encert a L1 (T _h) es de 1 cicle. Pel programa P la taxa de fallades (m ₁) de L1 es del 10%. b) Calculeu quants accessos a L1 fa el programa P. La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de filla de 4 cicles, u latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra e passos en cas de fallada a L1. En el cicle 1 s'accedeix a L1 ies detecta que es una fallada de cache. En el cicle 2 s'envia la comanda ACTIVE i l'adreça de columna. Al cicle 10 (4 cicles després de R0) apareixen les dades (4 cicles s'envia la comanda READ I l'adreça de columna. Al cicle 10 (4 cicles després de R0) apareixen les dades (4 cicles bytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRE CICLE) (L1 L1 L	instruccions) en un simulado	or on tots els a									
amb una cache d'instruccions (que anomenarem L1) i una memòria principal SDRAM (els accessos a dades segueix tardant 1 cicle). La mida de les instruccions es de 4 bytes. La mida de bloc de L1 es de 32 bytes i el temps d'accés cas d'encert a L1 (T _h) es de 1 cicle. Pel programa P la taxa de fallades (m ₁) de L1 es del 10%. b) Calculeu quants accessos a L1 fa el programa P. La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de fila de 4 cicles, u latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra e la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicles després (cicle s'envia la comanda READ i l'adreça de columna. Al cicle 10 (4 cicles després de RD) apareixen les dades (4 cicles c'envia la comanda READ i l'adreça de columna. Al cicle 10 (4 cicles després de RD) apareixen les dades (4 cicles cytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRE CICLE) (Cicle 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 CPU (DADA) (Cicles CB) (CICLE)	a) Calculeu el temps d'exe	cució (T _{exec}) de	l programa	P en aqı	uest sis	stema	de me	mòria	ideal (e	en seg	ons).
amb una cache d'instruccions (que anomenarem L1) i una memòria principal SDRAM (els accessos a dades segueix tardant 1 cicle). La mida de les instruccions es de 4 bytes. La mida de bloc de L1 es de 32 bytes i el temps d'accés cas d'encert a L1 (T _h) es de 1 cicle. Pel programa P la taxa de fallades (m ₁) de L1 es del 10%. b) Calculeu quants accessos a L1 fa el programa P. La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de fila de 4 cicles, u latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra i passos en cas de fallada a L1. En el cicle 1 s'accedeix a L1 i es detecta que es una fallada de cache. En el cicle 2 s'em la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicles després (cicle s'envia la comanda READ i l'adreça de columna. Al cicle 10 (4 cicles després de RD) apareixen les dades (4 cicles ytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRE CLK											
La memòria SDRAM esta formada per un DIMM de 8 bytes d'amplada amb una latència de fila de 4 cicles, u latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra a passos en cas de fallada a L1. En el cicle 1 s'accedeix a L1 i es detecta que es una fallada de cache. En el cicle 2 s'em la comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicles després (cicle s'envia la comanda READ i l'adreça de columna. Al cicle 10 (4 cicles després de RD) apareixen les dades (4 cicles bytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRE CLK Cicle 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 DADA I COPU L1 MISS COM ACT RD RD DADA I CARL1 COM ACT RD RD DADA I CARL1 PRE DADA I CARL1 DADA I CARL	amb una cache d'instruccions tardant 1 cicle). La mida de le	s (que anomena es instruccions e	rem L1) i un es de 4 byte	a memò s. La mio	ria prii da de b	ncipal : loc de	SDRAN L1 es	Л (els a de 32	ccesso bytes i	s a da	des segu
latència de columna de 4 cicles i un temps per la comanda PRECHARGE de 1 cicle. El següent cronograma mostra de passos en cas de fallada a L1. En el cicle 1 s'accedeix a L1 i es detecta que es una fallada de cache. En el cicle 2 s'envila comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicles després (cicle s'envila la comanda READ i l'adreça de columna. Al cicle 10 (4 cicles després de RD) apareixen les dades (4 cicles bytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRECLIC LICLE LICL	b) Calculeu quants accesso	os a L1 fa el pro	grama P.								
a comanda ACTIVE i l'adreça de fila (Bus A) per activar la pàgina corresponent de memòria i 4 cicles després (cicle s'envia la comanda READ i l'adreça de columna. Al cicle 10 (4 cicles després de RD) apareixen les dades (4 cicles bytes/cicle) al bus de dades (Bus D). Les dades es van carregant a un buffer (cicles CB) a mesura que van apareixe pel bus (cicles etiquetats D). Finalment al cicle 14, un cop s'ha transmès tot el bloc al buffer, es passa la instrucció la CPU (DADA) i en paral.lel s'escriu el bloc a L1 (carL1) i s'activa la comanda PRECHARGE per tancar la pàgina (PRECECK CICLE 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 CPU L1 MISS COM ACT RD COM ACT RD COM ACT RD RD PRECENSIA RD PRECENSIA RD RD RD D D D D CCAlculeu el temps mig d'accés a memòria (T _{mam}) (en nanosegons).	latència de columna de 4 cicle	es i un temps pe	er la coman	da PREC	HARGE	de 1 d	cicle. E	l segü	ent cro	nograi	ma most
CLK		de fila (Bus A) p	er activar la	a pàgina	corres	ponen	it de m	emòri	a i 4 cio	cles de	sprés (ci
Cicle 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 CPU	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I	Bus D). Les dad Finalment al cic	es es van ca le 14, un co	rregant p s'ha t	a un <i>b</i> ransmè	uffer (cicles (el bloc	CB) a r al <i>buf</i> j	nesura <i>fer,</i> es	que v passa	an apare la instruc
L1 MISS	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). l la CPU (DADA) i en paral.lel s'	Bus D). Les dad Finalment al cic	es es van ca le 14, un co	rregant p s'ha t	a un <i>b</i> ransmè	uffer (cicles (el bloc	CB) a r al <i>buf</i> j	nesura <i>fer,</i> es	que v passa	an apare la instruc
Buffer Com ACT RD RD PRE Bus A Bus D D D D D D C Calculeu el temps mig d'accés a memòria (T _{mam}) (en nanosegons).	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). l la CPU (DADA) i en paral.lel s'	Bus D). Les dad Finalment al cic escriu el bloc a	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un <i>b</i> ransmè la com	uffer (es tot e anda I	cicles (el bloc PRECH	CB) a r al <i>bufj</i> ARGE	nesura fer, es per tan	que v passa icar la	an apare la instruc pàgina (F
Bus A Bus D D D D D Calculeu el temps mig d'accés a memòria (T _{mam}) (en nanosegons).	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3	Bus D). Les dad Finalment al cic escriu el bloc a	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un <i>b</i> ransmè la com	uffer (es tot e anda I	cicles (el bloc PRECH	CB) a r al <i>bufj</i> ARGE	mesura fer, es per tan 14 DADA	que v passa icar la	an apare la instruc pàgina (F
Bus D D D D D C Calculeu el temps mig d'accés a memòria (T _{mam}) (en nanosegons).	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). la CPU (DADA) i en paral.lel s' CLK	Bus D). Les dad Finalment al cic escriu el bloc a	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un bransme	uffer (cicles (el bloc PRECH	CB) a r al <i>bufj</i> ARGE	mesura fer, es per tan 14 DADA	que v passa icar la	an apare la instruc pàgina (F
c) Calculeu el temps mig d'accés a memòria (T _{mam}) (en nanosegons).	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3 CPU L1 MISS Buffer Com ACT	Bus D). Les dad Finalment al cic escriu el bloc a	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un bransme	uffer (cicles (el bloc PRECH	CB) a r al <i>bufj</i> ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3 CPU L1 MISS Buffer Com ACT Bus A @F	Bus D). Les dad Finalment al cic escriu el bloc a	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un b ransme la com	uffer (es tot e anda I 11 CB	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
d) Calculeu el temps d'execució (T _{exec}) del programa P (en segons).	cytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK	Bus D). Les dad Finalment al cic escriu el bloc a 4 5 6 RD @C	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un bransme	uffer (es tot es anda l	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
d) Calculeu el temps d'execució (T _{exec}) del programa P (en segons).	cytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK	Bus D). Les dad Finalment al cic escriu el bloc a 4 5 6 RD @C	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un bransme	uffer (es tot es anda l	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
d) Calculeu el temps d'execució (T _{exec}) del programa P (en segons).	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3 CPU L1 MISS Buffer Com ACT Bus A @F Bus D	Bus D). Les dad Finalment al cic escriu el bloc a 4 5 6 RD @C	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un bransme	uffer (es tot es anda l	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
d) Calculeu el temps d'execució (T _{exec}) del programa P (en segons).	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3 CPU L1 MISS Buffer Com ACT Bus A @F Bus D	Bus D). Les dad Finalment al cic escriu el bloc a 4 5 6 RD @C	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un bransme	uffer (es tot es anda l	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3 CPU L1 MISS Buffer Com ACT Bus A @F Bus D	Bus D). Les dad Finalment al cic escriu el bloc a 4 5 6 RD @C	es es van ca le 14, un co L1 (carL1) i	arregant op s'ha t s'activa	a un bransme	uffer (es tot es anda l	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3 4 CPU L1 MISS Buffer Com ACT Bus A Bus D C) Calculeu el temps mig d	Bus D). Les dad Finalment al cic escriu el bloc a RD @C accés a memò	es es van ca le 14, un co L1 (carL1) i 7 8	en nano	a un bransme	uffer (es tot es anda l	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F
	bytes/cicle) al bus de dades (pel bus (cicles etiquetats D). I la CPU (DADA) i en paral.lel s' CLK Cicle 1 2 3 4 CPU L1 MISS Buffer Com ACT Bus A Bus D C) Calculeu el temps mig d	Bus D). Les dad Finalment al cic escriu el bloc a RD @C accés a memò	es es van ca le 14, un co L1 (carL1) i 7 8	en nano	a un bransme	uffer (es tot es anda l	cicles (el bloc PRECH.	CB) a r al bufj ARGE	nesura fer, es per tan 14 DADA carL1	que v passa icar la	an apare la instruc pàgina (F

A aquest sistema afegim un segon nivell de cache (L2) de forma que, si es falla a L1 s'accedeix a L2 i nomes en cas de fallar al segon nivell s'ha d'accedir a memòria principal. La taxa local de fallades (m_2) de L2 es del 30%. La mida de bloc de L2 es també de 32 bytes.

e) C a	alculeu			•		s que	fallen	a L1 i €	encerte	n a L2	i el pe	ercenta	itge d'a	access	os que	e fallen	a L1 i a
																	cta que
cicle 3	(CMP)	es com	nparen	les eti	iquete	s i es c	ompro	va que	e es hit	a L2, e	n el ci	cles 4 i	5 es ll	egeix l	a mem	nòria de	es, en el e dades
de la L (DADA	-	i RD2)). Final	lment	al cicle	e 6 s'e	scriu e	l bloc	a L1 (d	arL1)	i, en p	aral.le	l, es p	assa la	ı instru	a òissu	la CPU
CLK Cicle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
CPU	2.4166					DADA											
L1 L2	MISS	TAG	CMP	RD1	RD2	carL1											
						1											que es
compr	ova qu n sol ni	e es <i>mi</i> ivell de	iss a L2 e cache	. Dels e. Un d	cicles 4 cop ter	4 al 15 nim el	es llego bloc al	eix el b I <i>buffe</i>	loc de	la SDR	AM ta	l com j	a s'ha e	explica	at per l	a confi	etes i es guració (2 cicles
CLK		ΠL	ΠL	ΠL	ΠL	ΠL	ΠL	ΠL	ΠL		ΠL			ΠL	ΠL	ΠL	ПП
Cicle CPU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 DADA	17
L1	MISS															carL1	
L2 Buffer		TAG	CMP									СВ	СВ	СВ	СВ	WR1	WR2
Com				ACT				RD				CD		- 65		PRE	
@				@F				@C									
Datos												D	D	D	D		
f) Ca	alculeu	ı el ten	nps mi	g d'ac	cés a n	nemòr	ia (T _{ma}	_{am}) (en	nanos	egons).						
g) Ca	alculeu	ı el ten	nps d'e	execuc	ió (T _{ex}	_{ec}) del	progra	ama P	(en se	gons).							
h) Ca	alculeu	ı el gua	any (sp	eed-u	<i>p</i>) del s	sistem	a amb	L1 i L2	respe	cte el s	sistem	a que i	només	te L1.			

	Cognoms:
Ľ	er Control Arquitectura de Computadors Curs 2011-2012 Q1
Pro	plema 2. (3 puntos)
Mby	onemos de 20 discos físicos de 1 Tbyte de capacidad por disco, que ofrecen un ancho de banda efectivo de 250 tes/s por disco. Con estos discos deseamos montar un sistema de almacenamiento basado en RAID 50 nizado en 4 grupos de 5 discos.
	(Raid 0)
	Grupo (Raid 5)
a)	Calcular la cantidad de información útil (datos) que puede almacenar el sistema RAID50
cons del l cons	analizar el ancho de banda en escritura consideraremos por separado el caso en que realizamos accesos a tiras ecutivas y el caso en que realizamos accesos a tiras aleatorias. En el caso de accesos aleatorios, el controlador RAID ordena las peticiones de acceso para aprovechar al máximo la concurrencia entre discos. En ambos casos ideraremos que disponemos de suficientes peticiones de escritura de forma que el controlador del RAID siempre le aprovechar el ancho de banda de todos los discos físicos.
	l caso de escrituras secuenciales es posible esperar a tener una cantidad de datos suficiente para calcular la lad correspondiente y escribir simultáneamente en todos los discos.
b)	Calcular el ancho de banda efectivo (datos), si hacemos escrituras secuenciales, para el sistema RAID50.
calc	l caso de escrituras aleatorias es necesario leer los datos antiguos y la tira de paridad correspondiente para Ilar la nueva paridad y además escribir tanto datos como paridad por cada escritura que se desea realizar rivese que para escribir 1 tira de datos es necesario realizar 4 operaciones de disco.
c)	Calcular el ancho de banda efectivo (datos), si hacemos escrituras aleatorias, para el sistema RAID50.
	disco tiene un tiempo medio entre fallos (MTTF _{disco}) de 100.000 horas y el tiempo de recuperar la información aso de tener que reemplazar un disco (MTTR _{disco}) es de 10 horas.
d)	Calcular el tiempo medio entre fallos de un grupo de 5 discos (MTTF _{GRUPO}) suponiendo que los discos son e único componente que puede fallar.

Calcular el tiempo medio entre fallos del sistema RAID50 (MTTF _{RAID50}) suponiendo que los discos son el único componente que puede fallar.
nuestra implementación, cada grupo de discos tiene su propia fuente de alimentación con un tiempo medio entre os (MTTF _{fuente}) de 1.000.000 horas
Calcular el tiempo medio entre fallos de un grupo de 5 discos teniendo en cuenta también los fallos de las fuentes (MTTF _{GF}).
Calcular el tiempo medio entre fallos del sistema RAID50 (MTTF _{RAID50}) teniendo en cuenta tanto los posibles fallos de disco cómo los de las fuentes
¿Cuál crees que, en lo que a fiabilidad se refiere, es el componente crítico del sistema? Propon alguna solución para mejorar la fiabilidad del sistema (sin variar la capacidad de almacenamiento de datos del sistema).

Cognoms:	. Nom:
3er Control Arquitectura de Computadors	Curs 2011-2012 Q1

Problema 3. (2 puntos)

Debemos decidir el diseño de un procesador de señal (DSP) que ejecutará únicamente el siguiente kernel (observese que es un bucle infinito). Todas las variables son de coma flotante de simple precisión. Las variables cambian de valor en cada iteración ya que estan mapeadas a diversos sensores que las actualizan constantemente. Por ello deben cargarse de y guardarse en memoria cada iteración.

```
while (true) {
    A=(A-B*C)/(D+E)
}
```

Las dos alternativas de las que disponemos son un procesador Acumulador y otro tipo Memoria/Memoria. La descripción del ISA de ambos procesadores es la siguiente:

	Acumulador		Memoria / Memoria	
Tipo ins.	Ejemplo Operación		Ejemplo	Operación
Aritmética	Add A	ACC=ACC+A	Add A, B, C	C=A+B
Memoria	Load A	ACC=A		
Salto	Br Etiq	PC=PC+despl	Br Etiq	PC=PC+despl

			•	i i	· ·
a)	si lo necesitái los datos son	s. Observad que NO distintos en cada ite	D podéis optimizar el co	ódigo haciendo loads as instrucciones dinám	a. Podéis usar los registros R1a R8 o stores fuera del bucle dado que nicas se ejecutan en cada iteración
	dei bucie y cu	iditios accesos a me	eniona de datos se real	izan tambien para cat	da iteración dei bucie.

b) Traduce el código anterior al ensamblador del procesador Acumulador. Podéis usar las variables temporales temp1 a temp8 si lo necesitáis. Recordad que NO podéis optimizar el código haciendo loads o stores fuera del

bucle. Escribid cuantas instrucciones dinámicas se ejecutan en cada iteración del bucle y cuantos accesos a memoria de datos se realizan también para cada iteración del bucle.
Cualquiera que sea el procesador elegido se va a conectar al mismo subsistema de memoria. En el caso de la memoria de datos, para el kernel anterior, esta es capaz de dar un ancho de banda sostenido de 3 GB/s. En cambio la memoria de instrucciones es capaz de ofrecer, también para el kernel anterior, un ancho de banda sostenido de 4 GB/s. Cada instrucción del procesador Acumulador ocupa 4 bytes y cada instrucción del procesador Memoria/Memoria ocupa 16 bytes. Sabemos que la memoria es el cuello de botella del sistema y por tanto el rendimiento del procesado estará únicamente limitado por el ancho de banda con memoria.
c) Explica de forma razonada y justifica cuantitativamente cuál es el procesador capaz de ejecutar el código más rápidamente y calcula cuál es su ganancia con respecto al más lento para el código dado (Pista: calculad cuantas iteraciones por segundo puede hacer cada procesador).
, ,
Una vez elegido el tipo de ISA del procesador debemos calcular su rendimiento. A partir de aquí y para cumplir los requisitos de tiempo real del programa suponemos que el procesador ejecuta el código a 5E7 iteraciones del bucle por segundo. Sabemos que para conseguirlo el procesador debería funcionar a 2 GHz y con un voltaje de 2 V. En estas condiciones su carga capacitiva equivalente sería de 5nF y su corriente de fugas de 2A.
d) Calcula el rendimiento del procesador elegido en MFLOPS/W.

Cognoms:	Nom:
3er Control Arquitectura de Computadors	Curs 2011-2012 Q1

Problema 4. (2 puntos)

Dado el siguiente código escrito en C:

```
typedef struct {
   int a;
   char b;
   int c;
   int d;
} s1;
```

a) **Dibujad** cómo quedaría almacenada en memoria la estructuras s1, indicando claramente los desplazamientos respecto al inicio y el tamaño de todos los campos.

Traducid a ensamblador del x86 la siguiente sentencia: "X.a = X.a + X.d + (int) X.b", suponed que el registro ebx tiene la dirección de inicio de X.

Disponemos del siguiente código escrito en C:

V[i].a (ESCR)

```
s1 V[1024*256];
for (i=0; i<1024*256; i++)
  V[i].a = V[i].a + V[i].d + (int)V[i].b;</pre>
```

Este código se ejecuta en un procesador con una memoria cache de datos 2-asociativa de 8KB, con bloques de tamaño 32B y política de escritura write through+write no allocate.

 Suponiendo que la dirección de inicio del vector es 0 y que sólo consideramos los accesos al vector V. Calculad: número total de accesos a memoria, número de aciertos/fallos a la cache de datos, la tasa de fallos, el total de bytes leidos de Memoria Principal y el total de bytes escritos en Memoria Principal.
 Para hacer estos cálculos, os ayudará rellener la siguiente tabla:

iteración	iteración 0 1		1		2			3			4				
acceso	@	Con.j MC	Fallo MC	@	Conj. MC	Fallo MC									
V[i].b (LECT)															
V[i].d (LECT)															
V[i].a (LECT)															

#Accesos a Memoria:	Tasa de fallos:					
#Fallos:	Total bytes leidos de MP:					
#Aciertos:	Total bytes escritos en MP:					