



Nom:

Cognoms :

Memòria cau (3.5 punts)

1. El nostre computador està format per la CPU, una memòria cau i la memòria principal. Assumiu que, en la nostra arquitectura, el miss penalty és de 70 ns. Per tal de millorar el rendiment del nostre computador, provem les següents memòries cau:

	Mida L1	Miss rate L1	L1 hit time
P1	2 KiB	8.0 %	0.66 ns
P2	4 KiB	4.0 %	0.90 ns

- a) Si l' "L1 hit time" determina el temps de cicle per a les memòries P1 i P2, quines són les seves respectives freqüències de rellotge? (0.25 punts)
- b) Quin és el temps mig d'accés en ambdós sistemes ? (0.5 punts)
- c) Quina memòria cau posaríeu per millorar el rendiment ? Argumenteu breument. (0.25 punts)

a) $P1 \Rightarrow f1 = 1 / 0.66 \text{ ns} \text{ o } 1.515 \text{ GHz}$
 $P2 \Rightarrow f2 = 1 / 0.9 \text{ ns} \text{ o } 1.111 \text{ GHz}$

b) $T = h1T_{ac} + (1-h1)MissP \Rightarrow$
 $T1 = 0.92 * 0.66 \text{ ns} + 0.08 * 70 \text{ ns} = 6.25 \text{ ns}$
 $T2 = 0.96 * 0.90 \text{ ns} + 0.04 * 70 \text{ ns} = 3.66 \text{ ns}$

- c) Clarament triaria la P2. Té més capacitat i a més, tot i tenir un temps d'accés més lent, el temps mig és més baix que la P1



Nom:

Cognoms :

2. Considerem un computador format per una CPU, una memòria cau i una memòria principal. La mida de les paraules és de 64 bits i la memòria principal és adreçable byte a byte. La caché s'organitza mitjançant mapejat directe. El bus d'adreces és de 64 bits i s'organitza de la següent manera:

TAG	Index	Offset
63-10	9-5	4-0

- a) Quina és la mida dels blocs de la caché ? (0.5 punts)
- b) Quantes línies té la caché ? (0.25 punts)
- c) Quin és el ratio entre els bits dedicats a emmagatzemar dades i el nombre total d'informació que ha d'emmagatzemar la caché si considerem que, a més, tenim 1 bit d'status ? (0.5 punts)

(el problema continua en la següent pàgina)

- a) Sé que offset inclou W i B. Sé que les paraules són de 64 bits (8 B) i que cada direccionalment a MP em retorna 1 B, per tant, per aconseguir la paraula sencera necessitem 8 direccions $\Rightarrow B = \log_2 8 = 3 \Rightarrow W = 2 \Rightarrow$ els blocs contenen 2^2 paraules
- b) Tinc caché mapejat directe i 5 bits d'index (i), això vol dir que tinc 2^5 línies (32 línies)
- c) Cada línia de caché guarda 4 paraules de 64 bits, per tant, cada línia guarda 256 bits. Sabem que la caché té 32 línies, per tant podem emmagatzemar $32 * 256$ bits (= 8192 bits) de dades ÚTILS.

Tot i així la caché també necessita guardar el TAG i un bit d'status per cada línia. Això vol dir 54 bits de TAG + 1 bit d'status per línia. En total, $32 * 55$ bits (= 1760 bits) de dades NO útils.

El nombre total d'informació a emmagatzemar és $8192 + 1760$ bits = 9952 bits, per tant, el ratio és igual a $8192 / 9952$ bits (si heu fet el càlcul a l'inversa també ho consideraré correcte)



Nom:

Cognoms :

- d) Amb la caché inicialment buida, les següents adreces són introduïdes (per ordre d'esquerra a dreta)

Address												
Hex	00	04	10	84	E8	A0	400	1E	8C	C1C	B4	884
Dec	0	4	16	132	232	160	1024	30	140	3100	180	2180

Ompliu la següent taula: (1 punt)

Byte Address	Binary Address	TAG	Index	Offset	Hit/Miss
0x00	0000 0000 0000	0x0	0x00	0x00	M
0x04	0000 0000 0100	0x0	0x00	0x04	H
0x10	0000 0001 0000	0x0	0x00	0x10	H
0x84	0000 1000 0100	0x0	0x04	0x04	M
0xe8	0000 1110 1000	0x0	0x07	0x08	M
0xa0	0000 1010 0000	0x0	0x05	0x00	M
0x400	0100 0000 0000	0x1	0x00	0x00	M
0x1e	0000 0001 1110	0x0	0x00	0x1e	M
0x8c	0000 1000 1100	0x0	0x04	0x0c	H
0xc1c	1100 0001 1100	0x3	0x00	0x1c	M
0xb4	0000 1011 0100	0x0	0x05	0x14	H
0x884	1000 1000 0100	0x2	0x04	0x04	M

- e) Segons aquestes dades, quin és el hit rate ? (0.25 punts)

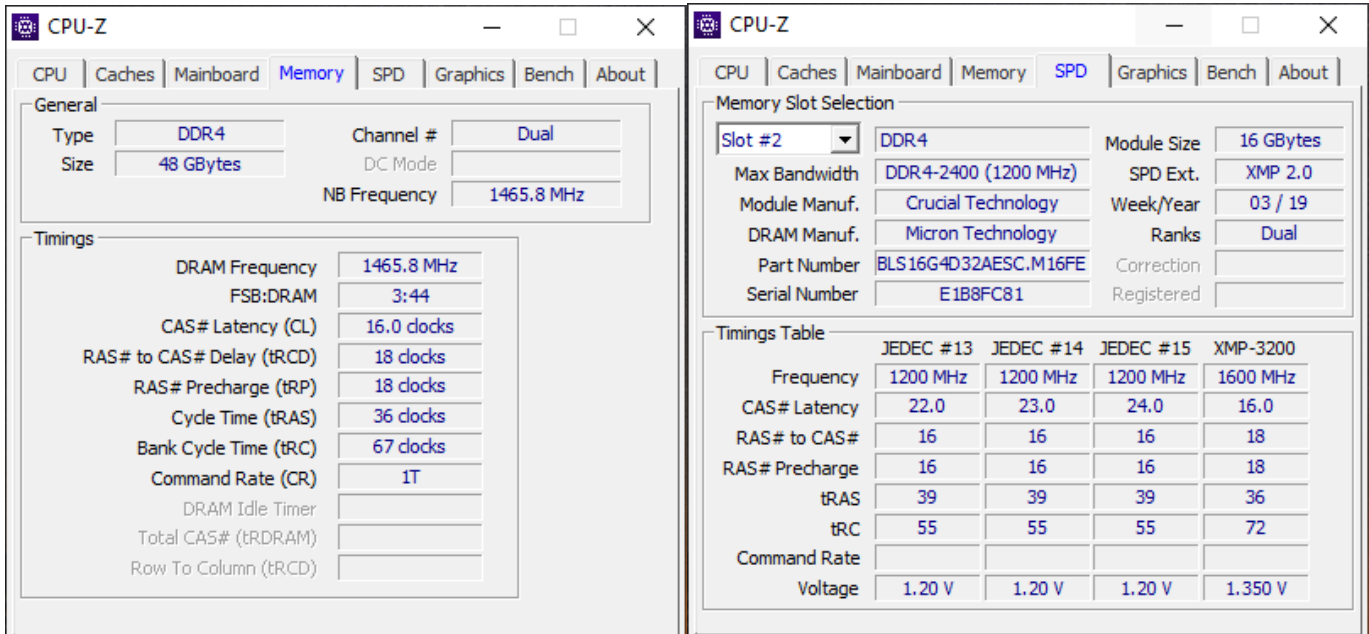
Tenim 4 hits en 12 lectures => $h = 33.33333$

Nom:

Cognoms :

Memòria principal (2.5 punts)

3. Tenim un ordinador amb les següents característiques de Memòria Principal:



General Memory Information:

- Type: DDR4
- Size: 48 GBytes
- Channel #: Dual
- NB Frequency: 1465.8 MHz

Timings:

- DRAM Frequency: 1465.8 MHz
- FSB:DRAM: 3:44
- CAS# Latency (CL): 16.0 clocks
- RAS# to CAS# Delay (tRCD): 18 clocks
- RAS# Precharge (tRP): 18 clocks
- Cycle Time (tRAS): 36 clocks
- Bank Cycle Time (tRC): 67 clocks
- Command Rate (CR): 1T

SPD Memory Slot Selection:

- Slot #2: DDR4
- Module Size: 16 GBytes
- Max Bandwidth: DDR4-2400 (1200 MHz)
- Module Manuf.: Crucial Technology
- DRAM Manuf.: Micron Technology
- Part Number: BLS16G4D32AESC.M16FE
- Serial Number: E1B8FC81
- SPD Ext.: XMP 2.0
- Week/Year: 03 / 19
- Ranks: Dual

Timings Table:

	JEDEC #13	JEDEC #14	JEDEC #15	XMP-3200
Frequency	1200 MHz	1200 MHz	1200 MHz	1600 MHz
CAS# Latency	22.0	23.0	24.0	16.0
RAS# to CAS#	16	16	16	18
RAS# Precharge	16	16	16	18
tRAS	39	39	39	36
tRC	55	55	55	72
Command Rate				
Voltage	1.20 V	1.20 V	1.20 V	1.350 V

a) Ompliu la taula: (0.5 punts)

Amb quina altra nomenclatura podríem trobar les dades a l'etiqueta de la memòria?	PC4-19200
Quantitat total de memòria principal?	48 GB
Número de mòduls de memòria?	3
Freqüència real del BUS?	1200 MHz
Ample banda teòric de cada mòdul de M.P.?	19200 MB/s
Ample de banda teòric del sistema M.P.?	2*19200MB/s (dual)
Aquest processador fa servir un sistema de comunicació "Single Channel", "Dual Channel" o "Triple Channel"?	Dual
Quants cicles de bus es necessiten per fer la transferència de 8 Bytes?	Trcd + CL + 1 clock = 35 clocks
Quants cicles de bus es necessiten per fer la transferència de 16 Bytes contigus?	35 clocks (8 B per flanc)
Quants cicles de bus es necessiten per fer la transferència de 64 Bytes contigus?	42 clocks

(el problema continua en la següent pàgina)



Nom:

Cognoms :

Contesteu les següents qüestions:

- b) Què és el temps de precàrrega ? A què és degut ? (1 punt)
- c) Podem eliminar el temps de precàrrega ? Raoneu la resposta tant si la resposta és afirmativa o negativa. (1 punt)

- b) Quan llegim una cel·la DRAM ens emportem la càrrega. Com volem que la memòria recordi els valors guardats independentment si hem llegit o no, hem de tornar a posar el valor que hi havia en aquella cel·la. Això es fa mitjançant els amplificadors de càrrega.
- c) No es pot eliminar, és inherent a la DRAM. El que podem fer és minimitzar el seu efecte mitjançant l'entrellaçat de memòria, o dit d'una altra forma, treballant amb bancs.



Nom:

Cognoms :

Administració de memòria (3.5 punts)

4. Una matriu de 4096×4096 , amb elements de 8 bits, es normalitzarà de la manera següent: Per a cada columna, es troba l'element més gran i tots els elements de la columna es divideixen pel valor d'aquest element. Assumeix que cada pàgina de la memòria virtual consta de 4KiB i que 1MiB bytes de la memòria principal s'assignen per emmagatzemar dades de matriu durant aquesta computació. Assumeix que es triga 10 ms a carregar una pàgina del disc a la memòria principal quan es produeix un error de pàgina.

- a) Supposeu que la matriu es processa una columna a la vegada. Quantes falles de pàgina es produiria i quant temps triga a completar el procés de normalització si els elements de la matriu s'emmagatzemen en ordre de columna en la memòria virtual (primer tots els elements de la columna 0, a continuació els de la columna 1, etc)? (0.75 punts)
- b) Repetiu la part (a) assumint que els elements s'emmagatzemen en ordre de fila (primer tots els elements de la fila 0, a continuació els de la fila 1, etc)? (0.75 punts)
- c) Proposar una manera alternativa de processament de la matriu per reduir el nombre de falles de la pàgina quan la matriu s'emmagatzema a la memòria com en l'apartat (b) (1 punt)

a) Tenim matriu de 4096 files i 4096 columnes, si cada element que hi guardem és de 8 bits, vol dir que una fila completa ocupa 4 KiB (o que una columna completa ocupa 4 KiB) que és just la mida de la pàgina :D

Així doncs, si ordenem les dades per columnes, tinc tantes pàgines com columnes, on cada pàgina és una columna de la matriu. O dit d'una altra manera, porto la columna 0 (faig una falta), busco el màxim, divideixo la columna pel màxim i vaig a buscar la següent columna => tinc tantes faltes com columnes (4096 faltes)

$T = 4096 \text{ faltes} * 10 \text{ ms/falta} = 40.96 \text{ s}$ per dur les pàgines de memòria secundària a memòria principal.

b) Aquí ordenem les dades per files, això vol dir que per buscar el màxim de la columna 0 he de portar totes les files. Per tant, ja tenim 4096 faltes per buscar el màxim de la columna 0. A la mem. Principal només hi caben 256 pàgines, així que per normalitzar la columna, torno a fer 4096 faltes ja que he de portar de nou totes les files (pàgines). Un cop fetes aquestes 8192 faltes, tenim la primera columna normalitzada, així que ho torno a fer per a totes les columnes (4096)

$T = (4096)^2 * 10 \text{ ms/falta} = 335.544,32 \text{ s} \Rightarrow 93.2 \text{ hores} !!!!$

c) Clarament puc millorar el raonament si faig les operacions en blocs de 256 pàgines.



Nom:

Cognoms :

5. En el nostre sistema, les pàgines tenen una mida de 128 paraules. Volem inicialitzar una matriu de 128 files x 128 columnes on cada element té una mida d'una paraula.

Nota: La nostra matriu s'emmagatzema en memòria en "row major order": data[0][0], data[0][1],..., data[0][127], data[1][0], data[1][1],..., data[127][127].

- a) Quantes faltes de pàgina tindrem amb el següent codi en C ? (0.5 punts)

```
for(int j=0; j<128; j++)  
    for (int i=0; i<128; i++)  
        data[i][j] = 0;
```

El sistema ha ordenat la matriu per files, això vol dir que cada fila és una pàgina (tinc 128 paraules en una fila que és just la mida de la pàgina)

El bucle el que fa és fixar la columna i varia 128 cops la fila, això vol dir que tinc 128 faltes per columna; com tenim 128 columnes el nombre de faltes és igual a 128 x 128 (16384)

- b) Quantes faltes de pàgina tindrem amb el següent codi en C ? (0.5 punts)

```
for(int i=0; i<128; i++)  
    for (int j=0; j<128; j++)  
        data[i][j] = 0;
```

El bucle el que fa és fixar la fila i varia 128 cops la columna, això vol dir que tinc 1 falta per fila; com tenim 128 files el nombre de faltes és igual a 128



Nom:

Cognoms :

BUSos (0.5 punts)

6. Calculeu l'ample de banda (MB/s), que necessita un Bus, com a mínim, per poder connectar un sistema de vídeo que suporti una resolució 2048x1536 amb una profunditat de color de 32bits amb una freqüència de refrescament de 100Hz (és a dir 100 imatges per segon). (0.5 punts)

$$(2048 \cdot 1536)[\text{px/Img}] \cdot 4[\text{B/px}] \cdot 100[\text{Img/s}] = 1.258.291.200 \text{ B/s} = 1.228.800 \text{ kB/s} = 1.200 \text{ MB/s} \\ \sim 1.17 \text{ GB/s}$$