EC Reavaluació. Sessió 1: Rendiment i consum. Punters

Exercici 1 (Examen Final gener de 2013)

Un processador disposa de 4 tipus d'instruccions diferents: A, B, C i D. La següent taula mostra quin és el número d'instruccions executades per a un programa sota consideració i el CPI de cada tipus d'instrucció. El processador té un rellotge a 2GHz.

Tipus d'instrucció	CPI	#instruccions
A	1	$8 * 10^9$
B	2	$6 * 10^9$
C	1	$4 * 10^9$
D	3	$2 * 10^9$

Indica:

1. Calcula el CPI mitjà del programa sota consideració.
2. Indica quin és el temps d'execució (en segons) del programa sota consideració.
3. Indica quin seria el guany (speed-up) obtingut si s'aconseguís reduir el CPI de les instruccions de tipus B a 1 cicle.

Exercici 2 (Examen Final 2018-2019 Q1)

Considerem un computador amb un processador MIPS funcionant a una freqüència de 0,5Ghz, i que dissipa una poteència de 10 W. Suposem que la cache d'instruccions és ideal (sempre encerta), i que la cache de dades té un temps de servei en cas d'encert th = 1 cicle. El temps necessari per copiar un bloc de memòria principal a cache és tblock = 99 cicles. Els CPI dels diversos tipus d'instruccions (en abseència de fallades) són:

	Salts	Loads	Resta d'instruccions
CPI	3	5	2

A través de simulacions amb un programa de test hem mesurat una taxa de fallades de la cache de dades del 4,4% (és a dir, m = 0,044). Totes les referències a memòria són lectures. El nombre d'instruccions executades és:

	Salts	Loads	Resta d'instruccions
núm. instr.	$1 * 10^9$	$3 * 10^9$	$6 * 10^9$

a) Calcula el CPI_{ideal} del programa (CPI promig amb cache ideal sense fallades)

b) Calcula, en segons,el temps d'execució (incloent-hi fallades de cache)

c) Calcula l'energia total consumida durant l'execució del programa, en Joules

d) Calcula, en cicles, el temps d'accés mitjaè a memòria dels loads per a aquest programa

Exercici 3 (Examen Parcial 2018-2019 Q2)

S'executa un programa de test en un ordinador que té 3 tipus d'instruccions (A,B,C), amb CPI diferents. La següent taula especifica el nombre d'instruccions de cada tipus que executa el programa i el seu CPI.

Tipus d'instrucció	Nombre d'instruccions	CPI
A	$8 * 10^9$	7
B	$6 * 10^9$	5
C	$4 * 10^9$	4

a) Sabem que la freqüència de rellotge és de 1,5GHz i que la potècia dissipada és P=100W. Calcula el temps d'execució en segons i l'energia consumida en Joules durant l'execució del programa de test.

b) Es vol modernitzar l'equipament comprant un nou ordinador amb una CPU amb ISA diferent. Aquesta nova CPU funciona amb el mateix voltatge però te més transistors, així que la seva capacitància agregada (C) és un 50% més gran i el factor d'activitat (α) un 20% superior. A més a més, els seu ISA té 4 tipus d'instruccions (A,B,C,D). Un cop recompilat el programa, el nombre d'instruccions de cada tipus que executa i el seu CPI són les indicades en la següent taula:

Tipus d'instrucció	Nombre d'instruccions	CPI
A	$5 * 10^9$	2
B	$3 * 10^9$	4
C	$2 * 10^9$	5
D	$2 * 10^9$	1

Sabem que amb aquesta nova CPU obtenim un speedup de 2. A quina freqüència (en GHz) va la nova CPU? ¿Quina potència dissipada en Watts consumeix l'execució del programa (podem suposar que la potència estàtica és zero tant en la CPU original com en la nova)?

Exercici 4 (Examen Final gener 2013)

Donada la següent declaració de variables globals, que s'ubica a memòria a partir de l'adreça 0x10010000:

```
.data
v1: .byte 12
v2: .half 0x13F
v3: .dword -4
v4: .asciiz "01234" # codi ASCII '0' = 0x30
v5: .byte 3
v6: .space 5
v7: .float 3.5
```

a) Omple la següent taula amb el contingut de memòria en hexadecimal. Les variables s'emmagatzemen a partir de l'adreça 0x10010000. Les posicions de memòria sense inicialitzar es deixen en blanc.

@Memòria	Dada	@Memòria	Dada	@Memòria	Dada	@Memòria	Dada
0x10010000		0x10010010		0x10010020		0x10010030	
0x10010001		0x10010011		0x10010021		0x10010031	
0x10010002		0x10010012		0x10010022		0x10010032	
0x10010003		0x10010013		0x10010023		0x10010033	
0x10010004		0x10010014		0x10010024		0x10010034	
0x10010005		0x10010015		0x10010025		0x10010035	
0x10010006		0x10010016		0x10010026		0x10010036	
0x10010007		0x10010017		0x10010027		0x10010037	
0x10010008		0x10010018		0x10010028		0x10010038	
0x10010009		0x10010019		0x10010029		0x10010039	
0x1001000A		0x1001001A		0x1001002A		0x1001003A	
0x1001000B		0x1001001B		0x1001002B		0x1001003B	
0x1001000C		0x1001001C		0x1001002C		0x1001003C	
0x1001000D		0x1001001D		0x1001002D		0x1001003D	
0x1001000E		0x1001001E		0x1001002E		0x1001003E	
0x1001000F		0x1001001F		0x1001002F		0x1001003F	

b) Calcula el valor final del registre $t0 en hexadecimal després d'executar el següent codi.

```
li $t0, -1
lui $t0, 0x32A4
srl $t0, $t0, 8
sw $t0, 0($sp)
lh $t0, 0($sp)
```

c) Calcula el valor final del registre $t0 en hexadecimal després d'executar el següent codi.

```
li $t0, 10247
li $t1, 10
div $t0, $t1
mflo $t0
mfhi $t1
addu $t0, $t0, $t1
```

Exercici 5 (Examen Parcial primavera 2013)

Donada la següent declaració de variables globals d'un programa escrit en llenguatge C:

```
char a='c'; /* el codi ascii de la 'a' és el 0x61 */
int b=1,c=-10;
char d[2][3]={{0xFE,0x01,0xFB},{0x0A,0xF4,0x04};
long long e=0x8811223344556677;
short f[3]={0xFFFB,0x0003,0xFFFC};
unsigned int g=35;
```

a) Tradueix-la al llenguatge assemblador del MIPS.

b) Omple la següent taula amb el contingut de memòria en hexadecimal. Les variables s'emmagatzemen a partir de l'adreça 0x10010000. Les posicions de memòria sense inicialitzar es deixen en blanc.

@Memòria	Dada	@Memòria	Dada	@Memòria	Dada	@Memòria	Dada
0x10010000		0x10010010		0x10010020		0x10010030	
0x10010001		0x10010011		0x10010021		0x10010031	
0x10010002		0x10010012		0x10010022		0x10010032	
0x10010003		0x10010013		0x10010023		0x10010033	
0x10010004		0x10010014		0x10010024		0x10010034	
0x10010005		0x10010015		0x10010025		0x10010035	
0x10010006		0x10010016		0x10010026		0x10010036	
0x10010007		0x10010017		0x10010027		0x10010037	
0x10010008		0x10010018		0x10010028		0x10010038	
0x10010009		0x10010019		0x10010029		0x10010039	
0x1001000A		0x1001001A		0x1001002A		0x1001003A	
0x1001000B		0x1001001B		0x1001002B		0x1001003B	
0x1001000C		0x1001001C		0x1001002C		0x1001003C	
0x1001000D		0x1001001D		0x1001002D		0x1001003D	
0x1001000E		0x1001001E		0x1001002E		0x1001003E	
0x1001000F		0x1001001F		0x1001002F		0x1001003F	

c) Quin és el valor final del registre $t4, en hexadecimal, després d'executar el següent fragment de codi?

```
la $t0,c
lb $t1,8($t0)
lh $t2,28($t0)
or $t3,$t1,$t2
li $t4,0x00FFFFFF
and $t4,$t3,$t4
```

d) Quin és el valor final dels registres $t4 i $t5, en hexadecimal, després d'executar el següent fragment de codi?

```
la $t1,g
lw $t2,0($t1)
li $t3,3
div $t2,$t3
mfhi $t4
mflo $t5
```

Exercici 6 (problema 2.29 de la col.lecció)

Donades les següents declaracions de variables globals, en llenguatge C:

```
char *punterc;
short *punterh;
int *punteri;
long long *punterd;
```

Tradueix a assemblador MIPS les següents sentències:

a) `punterc++;`

b) `punteri++;`

c) `punterh++;`

d) `punterd++;`

e) `*punteri = *punteri + 5;`

f) `*punterh = *punterh + 10;`

Exercici 7 (problema 2.28 de la col.lecció)

Donades les següents declaracions de variables globals, en C:

```
int vec[6] = {2, 4, 0, 6, 8, 0};
int *punter;
```

Tradueix a una única sentència en C el conjunt d'iinstruccions de cada apartat:

a)
```
la $t0, punter
la $t1, vec + 8
sw $t1, 0($t0)
```

b)
```
la $t0, punter
lw $t1, 0($t0)
addiu $t1, $t1, 4
sw $t1, 0($t0)
```

c)
```
la $t1, vec
lw $t3, 0($t1)
lw $t4, 4($t1)
addiu $t3, $t3, $t4
sw $t3, 4($t1)
```

d)
```
la $t1, vec
la $t0, punter
lw $t2, 0($t0)
lw $t3, 0($t2)
addiu $t3, $t3, 1
sw $t3, 8($t1)
```

e)
```
la $t0, punter
lw $t2, 0($t0)
lw $t3, 0($t2)
addiu $t3, $t3, 1
sw $t3, 8($t2)
```

Exercici 8 (Examen Parcial novembre de 2012)

Fes un programa en MIPS en 3 instruccions que intercanviï els 16 bits de menys pes amb els 16 bits de més pes del registre $s0.