Arquitectura de Computadores

I2

16 noviembre 2021

Inicio: 8.30 am

Entrega: hasta las 9.00 pm

0) Responde esta pregunta de manera manuscrita.

a) Nombre completo y número de alumno: _____

b) Me comprometo a no preguntar ni responder dudas de esta prueba, ya sea directa o indirectamente, a nadie que no sea parte del equipo docente del curso. **Firma**: _____

Responde sólo TRES de las siguientes cuatro preguntas

1) **a**) i) Si el *hit ratio* de una cierta pieza de código es 0.2, el tiempo de acceso a la cache es 20 nano-segundos y el tiempo de acceso a la memoria es 1 microsegundo; ¿cuál es el tiempo efectivo de acceso a memoria de esta pieza de código?

ii) Un científico escribe el siguiente código para sumar todos los elementos de una matriz muy grande:

```
float a[32768, 32768], sum
…
for i = 0, …, 32767:
       for j = 0, …, 32767:
              sum = sum + a[j,i]
```

El científico se queja de que el código corre muy lento. ¿Qué cambio simple se puede hacer al código para aumentar la velocidad de ejecución, y por qué?

b) Considera un computador en que las direcciones de memoria son de 32 de bits y tiene una cache que puede almacenar 4096 (4K) entradas.

i) Si cada entrada de la cache almacena una dirección de memoria y un byte de datos, ¿qué tan grande es la cache? Justifica tu respuesta.

ii) Si en cambio la cache usa *direct mapping*, y cada entrada almacena un *tag* y una línea de datos de 4 bytes, ¿qué tan grande es la cache? Justifica tu respuesta.

c) En clases vimos que las caches direct mapping y fully associative en general no son las más convenientes. En la práctica se usan más a menudo caches set associative. Explica detalladamente las diferencias entre una cache **2-way** set associative y una cache **8-way** set associative:

i) ¿Cómo se busca una dirección de memoria en cada caso?

ii) Si las caches son básicamente del mismo tamaño, ¿bajo qué condiciones de ejecución del programa puede ser preferible una y la otra? Justifica tu respuesta.

Arquitectura de Computadores

I2

16 noviembre 2021

2) Se tiene un computador con memoria RAM de 1 MB. Sin embargo, los programas corren con una memoria virtual de 4 MB. Se sabe que el computador utiliza páginas de 256 KB y sus direcciones de memoria virtual son de 12 bits.

Por otra parte, ustedes saben que el computador emplea una tabla de páginas en donde cada entrada almacena 1 bit de presencia, 1 *dirty bit*, *m* bits suficientes para indicar el marco (*page frame*) y todos los bits de metadata necesarios para el criterio de reemplazo, el cual será LRU. Pueden suponer que la tabla de páginas se almacena "mágicamente" en otra memoria aparte, con suficiente espacio para que todo quepa correctamente.

Finalmente, las especificaciones del computador indican que este cuenta con una unidad TLB pequeña, capaz de almacenar sólo 2 traducciones y con criterio de reemplazo FIFO.

En base a esta información y a la siguiente secuencia de 20 accesos a memoria (direcciones virtuales), responde y justifica tus respuestas:

a) Cantidad de marcos (*page frames*)

b) Cantidad de páginas

c) Cantidad de *page faults*

d) *Hit rate* del TLB

Accesos a memoria:

1. 0xA26	8. 0x4D7	15. 0xDB4
2. 0x09A	9. 0xB57	16. 0x566
3. 0x0C4	10. 0x5B1	17. 0xC9B
4. 0x714	11. 0x688	18. 0xDF5
5. 0xAC8	12. 0x6D6	19. 0xEBB
6. 0x5A5	13. 0x993	20. 0x53A
7. 0x8FE	14. 0x168	

3) Considera que la instrucción **`SUB A,B`** se demora 20ns en ejecutarse, **`ADD (var),A`** se demora 23ns en ejecutarse, **`MOV B,(var)`** se demora 25ns en ejecutarse, **`JEQ label`** se demora 21ns en ejecutarse, y **`CMP A,0`** se demora 18ns en ejecutarse. Además, la etapa Instruction Fetch se demora 10ns.

```
DATA:
var1  3
var2  3
res   0
i     0

CODE:
        MOV A, (var1)
        MOV (i), A
        start:
                MOV A, (res)
                ADD A, (var2)
                MOV (res), A
                MOV A, (i)
                SUB A, 1
                MOV (i), A
                CMP A, 0
                JNE start
```

a) Calcula cuánto se demora (en nanosegundos) en ejecutar el programa si el computador no tiene paralelismo ni pipelining integrado.

b) Calcula cuánto se demora (en nanosegundos) en ejecutar el programa si el computador tiene paralelismo con pipelining de 5 etapas (IF-ID-EX-MEM-WB). (El manejo de *stalling* es por software, es decir con la instrucción NOP, y no hay predicción de saltos).

c) Calcula cuánto mejor (en porcentaje) es utilizar pipelining que no utilizarlo en el programa anterior.

Arquitectura de Computadores

I2

16 noviembre 2021

4) Se tienen dos CPUs con paralelismo UMA. Cada una cuenta con una caché propia y las cachés funcionan en conjunto bajo el protocolo de coherencia de cachés MESI.

Se requiere que la CPU 0 (dueña de la caché P0) sume X con Arr[0] y lo guarde en Arr[1], y luego le reste Arr[1] a X y lo guarde en X.

Al mismo tiempo, se requiere que la CPU 1 (dueña de la caché P1) reste Arr[0] a Arr[1] y lo guarde en X, para luego sumar X y Arr[1] y guardar el resultado en Arr[0].

a) [**1.5 pts.**] Escribe el código en assembly que ejecuta cada CPU.

b) [**4.5 pts.**] Suponiendo que los valores iniciales de las variables en la memoria compartida son los siguientes:

Dirección	variable	valor
0x64206F68	X	5
0x64206F6C	Arr	1
0x64206F70		2

ejecuta el código de cada CPU, llenando la siguiente tabla MESI en el proceso

	P0												P1											
	X				Arr[0]				Arr[1]				X				Arr[0]				Arr[1]			
Inst	M	E	S	I	M	E	S	I	M	E	S	I	M	E	S	I	M	E	S	I	M	E	S	I

e indica los valores finales de X, Arr[0] y Arr [1] en cada caché.