

Guía de ejercicios # 11 - Jerarquía de memoria

Organización de Computadoras C3

UNQ

Objetivos

Que el estudiante pueda:

- Conocer y comprender la jerarquía de memorias
- Comprender la necesidad de una memoria Cache
- Analizar como funcionan las correspondencias de memoria cache
- Entender cual es el impacto del desempeño de una cache

1 Correspondencia Directa

En una caché con correspondencia directa las celdas se almacenan en una línea predeterminada, es decir que cada bloque, si debe ser cacheado, está destinado a una línea específica.

1. ¿Cómo se divide una dirección de memoria de 5 bits en *tag*, *línea* y *palabra* si se tiene una memoria principal de 32 celdas de un byte y una memoria caché con bloques de 4 celdas y 4 líneas
2. Considerar una computadora con una memoria de 64 celdas de un byte y una memoria cache con 4 líneas y bloques de 8 celdas por línea. Dada una dirección de memoria calcular la cantidad de bits que se destinan a: *tag*, *línea* y *palabra*.
3. Considerando el escenario del ejercicio 2, dar los tags y las líneas de la cache en las que se almacenan las direcciones:

Dirección	Tag	Nro. línea
111000		
011001		
111111		
101000		
101001		

4. Considerando el escenario del ejercicio 2, listar todas las direcciones en la misma línea que la dirección 111000.

5. Suponer que la cache descrita en el ejercicio 2 está vacía, y que se realizan lecturas de direcciones en el siguiente orden. Determinar para cada lectura si ésta produjo un fallo o un acierto.

Dirección	Tag	Nro. linea	¿F/A?
111000			
011001			
011111			
011101			
111111			
111000			
101000			
101001			

6. Considerar una máquina **Q** con una memoria cache de mapeo directo de sólo 2 líneas, con capacidad cada una para almacenar un bloque de 4 celdas. A partir de la posición 0000 está ensamblado el siguiente programa. El PC comienza en la etiqueta **main** y la caché está inicialmente vacía.

```

main: MOV R7, 0x0005 ; iniciar el contador
      MOV R6, 0x0001 ; iniciar el acumulador
arriba: MUL R6, 0x0002
      SUB R7, 0x0001;
      JNE arriba

```

Completar la siguiente tabla

Dirección	Tag	Nro. linea	Palabra	¿F/A?

7. Cuantos bits de una dirección se destinan a: *tag*, *linea* y *palabra* en el siguiente esquema:
- Una memoria principal de 64 Kbytes, con celdas de un byte
 - Una memoria cache con:
 - (a) Bloques de 8 celdas
 - (b) 256 Bytes de capacidad para datos (sin contar tag)
 - (c) correspondencia directa
8. Para los siguientes programas, asumir que se ejecutan en una máquina de arquitectura **Q**, con una memoria cache de bloques de 4 celdas y 4 líneas. Para cada uno calcular la cantidad de accesos a memoria caché y a memoria principal se llevan a cabo durante su ejecución.

```

{a}      MOV R0, 0x0000
          MOV R1, 0x0000
          MOV R2, 0x0000
          MOV R3, 0x0000
          MOV R4, 0x0000
          MOV R5, 0x0000
          MOV R6, 0x0000

```

```

MOV R7, 0x0000
arriba: ADD R0, 0x0001
        ADD R1, 0x0001
        ADD R2, 0x0001
        ADD R3, 0x0001
        ADD R4, 0x0001
        ADD R5, 0x0001
        ADD R6, 0x0001
        ADD R7, 0x0001
        CMP R7, 0x000F
        JNE arriba

```

{b} El siguiente programa está ensamblado a partir de la celda A012:

```

MOV R0, 0xAAAA
arriba: MOV R1, 0x0004
        CALL shiftN
        CMP R0, 0x0000
        JNE arriba

```

y la siguiente rutina ensamblada a partir de la celda B012

```

shiftN: CMP R1, 0x0000
        JE salir
        DIV R0, 0x0002
        SUB R1, 0x0001
        JMP shiftN
salir:  RET

```

2 Desempeño (performance) de la caché

Suponer una máquina con arquitectura **Q6** con una memoria caché de correspondencia directa de 64 líneas y 4 celdas por bloque. Se tiene el siguiente programa ensamblado a partir de la celda B110 y se sabe que $R1 = AC00$, $R3 = A702$, $SP = FFEE$ y que la celda A702 tiene el valor 1.

```

B110      CMP [R3],
B111      0x0000
B112      JE fin
B113      ADD R0, [R1]
B114      ADD R2, [R3]
B115  fin: RET

```

Se necesita analizar la performance de la caché en cuanto a la cantidad de fallos que se producen durante la ejecución del programa.

dir(hexa)	dir (binario)	tag	linea	F/A
B110	1011000100010000	10110001	000100	F
B111	1011000100010001	10110001	000100	A
B112	1011000100010010	10110001	000100	A
B113	1011000100010011	10110001	000100	A
AC00	1010110000000000	10101100	000000	F
B114	1011000100010100	10110001	000101	F
A702	1010011100000010	10100111	000000	F
B115	1011000100010101	10110001	000101	A

Se produjeron 8 accesos, de los cuales 4 fueron fallos, es decir que se tiene una **tasa de fallos** de $\frac{4}{8} = 0,5$.

Si el tiempo de acceso a la memoria es de $0.5ms$ y el de la cache es $0.05ms$, se sabe que la ejecución de ese programa llevó en total

$$4 * (0.5ms + 0.05ms) + 4 * 0.05ms$$

Ejercicios

9. Calcular la tasa de fallos que se produjo en las ejecuciones del ejercicio 8. ¿Cómo pueden disminuirse los fallos?
10. Se tiene un sistema con una memoria principal con un tiempo de acceso de 3s, y una memoria caché cuyo tiempo de acceso es de 0,3s y cuya tasa de aciertos es del 90%. ¿Cuánto tiempo se tarda en leer 2000 celdas?

```
main:      MOV R7, 0      //con R7 se cuentan las 1000
                                pasadas
                                MOV R3, 0x80    //en R3 se guarda el 'salto'
comienzo:  MOV R0, 0      //con R0 se recorre el arreglo
                                MOV R1, 0      //en R1 se acumula la suma
loop:      ADD R1, [R0]    //se suma en R1 la posición actual
                                ADD R0, R3      //se avanza R0
                                CMP R0, 0x800   //fin del arreglo (256*8 = 0x800)
                                JL loop         //si no se termino el arreglo, sigo
                                ADD R7, 1
                                CMP R7, 1000    //fin de las pasadas
                                JNE comienzo    //si no hice 1000 pasadas,sigo
                                RET
```

- (a) Dar la tasa de fallos que se produce en la caché (inicialmente vacía), considerando **únicamente los accesos al arreglo** (y no a las instrucciones).
 - (b) Dar la tasa de fallos bajo las mismas condiciones del ítem anterior, pero modificando el valor que se carga en R3 a 0x88 (el salto es ahora de 17 posiciones, dado que $17*8 = 136 = 0x88$).
 - (c) ¿Qué diferencia se observa entre los casos anteriores? ¿A qué se debe?
11. Dado el siguiente programa:

```
ciclo: CMP [R0], 0
      JE fin
      SUB [R0], 1
      JMP ciclo
fin:  ADD [R0], 1
```

- (a) Ensamblar el programa
- (b) Sabiendo que $R0 = C000$, que en la celda C000 hay almacenado un 2, que el programa se encuentra ensamblado a partir de la celda 7001 y el sistema tiene una cache de mapeo directo de 8 celdas por bloque y una capacidad de 2KB, inicialmente vacía. Completar para la ejecución del programa la siguiente tabla:

Dirección	Tag	Línea	Palabra	F/A	Desaloja?

References

- [1] Williams Stallings, *Computer Organization and Architecture*, octava edición, Editorial Prentice Hall, 2010. **Capítulo 4**