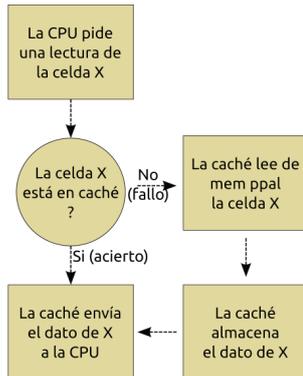


# Guía de ejercicios # 11 - Jerarquía de memoria

Organización de Computadoras 2017

UNQ

## Funciones de correspondencia



Cuando la Unidad de Control pide una determinada celda, la memoria caché debe, en primer lugar, determinar si la dirección corresponde a una celda cacheada. Para responder esta pregunta, debe aplicar una **función de correspondencia**

### 1 Correspondencia asociativa

En una **correspondencia asociativa** el contenido de cada celda leída se **asocia** a una **etiqueta** que identifica su origen, es decir, su dirección de memoria principal. Esto permite que las celdas puedan ser almacenadas en cualquier ranura, por lo que se debe chequear en todas ellas si alguna contiene la dirección buscada como tag.

Por ejemplo, considerará una memoria principal con direcciones de 6 bits (64 celdas) y donde cada celda de almacena 1 byte. En la siguiente memoria caché las celdas cacheadas son las 010000 a 010011:

|    |        |          |
|----|--------|----------|
| 00 | 010000 | 11111111 |
| 01 | 010001 | 11111111 |
| 10 | 010010 | 11111111 |
| 11 | 010011 | 11111111 |

Normalmente, las memorias caché tienen capacidad para almacenar varias celdas en un misma línea<sup>1</sup>, para aprovechar el principio de localidad espacial. En estos casos el tag ya no es simplemente la dirección de una celda, sino el número de bloque (que está relacionado con las direcciones que lo componen). Siguiendo el ejemplo anterior, si utilizamos una

<sup>1</sup>ranura=línea=slot

caché con bloques de 4 celdas, se tienen  $\frac{64}{4} = 16$  bloques, y entonces el número de bloque tiene 4 bits, pues  $2^4 = 16$ . De aquí que el bloque número 0000 tiene las celdas 000000 a 000011. bloque número 0001 tiene las celdas 000100 a 000111. Notar que se subrayó el número de bloque dentro de la dirección de la celda.

|    |      |                                  |
|----|------|----------------------------------|
| 00 | 1100 | 01010101111111110101010111111111 |
| 01 | 1010 | 1111111101010101111111110101010  |
| 00 | 0000 | 01010101111111110101010111111111 |
| 01 | 0110 | 1111111101010101111111110101010  |

Veamos una lectura de ejemplo: ¿cómo determina la caché si la celda 011010 está cacheada? Para eso utiliza los primeros 4 bits que corresponden al número de bloque (0110). ¿Qué parte de la línea envía como respuesta a la CPU? Para eso utiliza los últimos 2 bits (10), que se denominan **bits de palabra**, para extraer una porción de la línea.

|          |          |          |          |
|----------|----------|----------|----------|
| 11111111 | 10101010 | 11111111 | 10101010 |
| 00       | 01       | 10       | 11       |

De esta manera, el dato que envía es el tercer byte (11111111)

### Ejercicios

Todos los ejercicios tienen una memoria caché con **correspondencia asociativa**

- Suponer una memoria principal de 32 celdas de un byte y una memoria caché con 4 líneas y capacidad de almacenar 1 celda en cada línea.
  - ¿Qué tamaño tiene el tag?
  - ¿Qué capacidad total tiene la caché?
- Suponer una memoria principal de 32 celdas de un byte y una memoria caché con 4 líneas y capacidad de almacenar 4 celdas en cada línea.
  - ¿Qué tamaño tiene el tag?
  - ¿Qué capacidad total tiene la caché?

3. Suponer que una caché que tiene el siguiente contenido:

|    |     |                          |
|----|-----|--------------------------|
| 00 | 345 | F434C0F434C0F434C0F434C0 |
| 01 | 346 | F434C0F434C0F434C0F434C0 |
| 10 | 347 | B000AAAABBBB000AAAABBB   |
| 11 | 348 | 2B05000B000AAAABBBCCCO   |

- ¿Está cacheada la celda 3451?
- ¿Qué valor retorna la caché para esa celda?

4. Suponer una memoria principal con direcciones de 16 bits y una memoria caché con 256 líneas y capacidad de almacenar **4 celdas** en cada línea. Si la CPU pide la lectura de la celda FA32, ¿Que tag se debe buscar en la caché?

5. Cuantos bits de una dirección se destinan a: *tag* y *palabra* en el siguiente esquema:

- Una memoria principal de 256 celdas:
- Una memoria cache con:
  - (a) Bloques de 4 celdas
  - (b) 8 líneas
  - (c) correspondencia asociativa

6. Cuantos bits de una dirección se destinan a: *tag* y *palabra* en el siguiente esquema:

- Una memoria principal de 65536 celdas
- Una memoria cache con:
  - (a) Bloques de 16 celdas
  - (b) 16 líneas
  - (c) correspondencia asociativa

7. El chip 80286 (fabricado entre 1982 y 1993) tenía un bus de datos de 16 bits, pero un bus de direcciones de 24 bits, lo que lo hacía la primera arquitectura de Intel capaz de soportar 16Mb de RAM. Suponer la siguiente memoria cache, adaptada a dicha arquitectura:

- (a) 32 celdas por bloque
- (b) 256 líneas
- (c) correspondencia asociativa

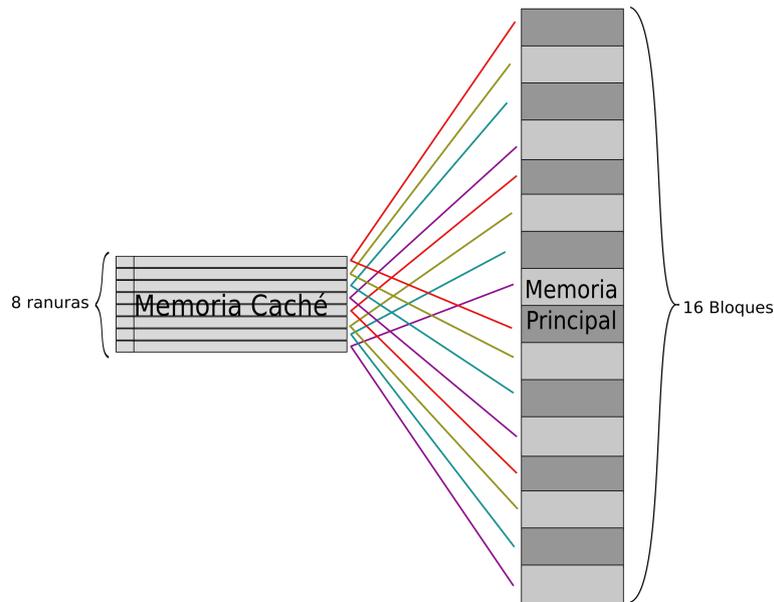
Determinar:

- (a) ¿Cómo se divide una dirección de memoria en *tag* y *palabra*?
- (b) ¿Cómo se decide si la dirección FAFABA está en caché?
- (c) ¿Cuántas celdas contiene dicha memoria cache?

## 2 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. Siguiendo el ejemplo de la memoria descrita en la sección anterior, considerar una caché con 8 líneas y bloques de 4 celdas. El criterio que se usa para corresponder bloques con líneas es circular, como describe la siguiente figura.

Esto permite simplificar la búsqueda de un tag en la caché, pues se debe buscar en una sola línea, la que corresponde al bloque buscado. ¿Cómo determina qué línea es esa? Notar primero que el *tag* es mas pequeño que en el caso del mapeo asociativo: Se tienen  $\frac{16}{8} = 2$  bloques por ranura. Es decir que con un bit (pues  $2^1 = 2$ ) alcanza para identificar cual



de esos 2 bloques está almacenado en la ranura en cuestión. Notar también que la ranura 000 almacena el bloque 0000 o el bloque 1000, y de aquí que el bit mas significativo dentro del número de bloque puede usarse como *tag*.

Analicemos una lectura de ejemplo:

- (i) ¿cómo determina la caché si la celda 011010 está cacheada? El número de bloque son los primeros 4 bits (0110) y a ese bloque le corresponde la línea 110. Entonces allí debe verificarse que contenga el tag 0.
- (ii) ¿Qué parte de la línea envía como respuesta a la CPU? Como en el caso asociativo, utiliza los **bits de palabra** (10) para extraer una porción de la línea.

## Ejercicios

2.1 ¿Cómo se divide una dirección de memoria de 16 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 líneas

2.2 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*.

2.3 Considerando el escenario del ejercicio 2, dar los tags y las líneas de la cache en las que se almacenan las direcciones:

| Dirección | Nro. Bloque | Nro. línea |
|-----------|-------------|------------|
| 111000    |             |            |
| 011001    |             |            |
| 111111    |             |            |
| 101000    |             |            |
| 101001    |             |            |

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

2.5 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 | Nro. Bloque | Nro. línea | ¿F/A? |
|-----------|-------------|------------|-------|
| 111000    |             |            |       |
| 011001    |             |            |       |
| 011111    |             |            |       |
| 011101    |             |            |       |
| 111111    |             |            |       |
| 111000    |             |            |       |
| 101000    |             |            |       |
| 101001    |             |            |       |

2.6 ¿Cómo se divide una dirección de memoria de 16 bits en *tag*, *línea* y *palabra* si la memoria caché con tiene 4 celdas por bloque y 256 líneas? ¿Cómo se decide si la dirección FA32 está en caché?

2.7 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 | Nro. Bloque | Nro. línea | tag | ¿F/A? |
|-----------|-------------|------------|-----|-------|
|           |             |            |     |       |
|           |             |            |     |       |
|           |             |            |     |       |
|           |             |            |     |       |

2.8 Cuantos bits de una dirección se destinan a: *tag*, *línea* 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

2.9 Considerar la arquitectura del ejercicio 2.8:

- (a) Dar las líneas en las que se almacenan las direcciones: 111B, C334, D01D, AAAA.
- (b) Listar todas las direcciones que se almacenarán en el mismo bloque que la dirección 1A1A.

(c) Suponer que la cache está vacía para completar la siguiente tabla con una secuencia de lecturas:

| Celda | Nro. Bloque | Nro. línea | tag | ¿F/A? |
|-------|-------------|------------|-----|-------|
| 111B  |             |            |     |       |
| 1100  |             |            |     |       |
| C334  |             |            |     |       |
| D01D  |             |            |     |       |
| AAAA  |             |            |     |       |
| 1118  |             |            |     |       |
| D01A  |             |            |     |       |

2.10 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
```

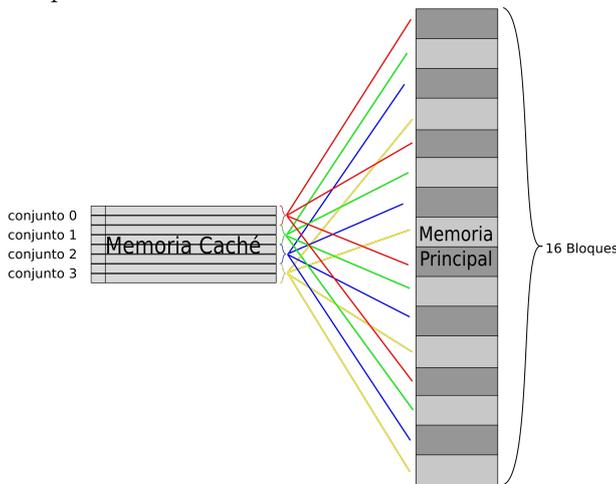
y la siguiente rutina ensamblada a partir de la celda B012

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

### 3 Correspondencia asociativa por conjuntos

Esta correspondencia combina aspectos de las dos anteriores. Las líneas se agrupan en conjuntos para corresponder de

manera directa cada bloque de memoria principal con un conjunto dentro de la caché, y dentro de cada conjunto los bloques se almacenan con un criterio asociativo. En la siguiente figura se ejemplifica esta nueva forma de correspondencia.



Cuando la caché recibe una dirección, debe determinar a que conjunto corresponde el bloque de la celda buscada y luego buscar asociativamente dentro del conjunto el tag que corresponde.

Suponer una maquina como la de la figura, con 16 bloques de 4 celdas en la memoria principal, y una memoria caché con 4 conjuntos de 2 línea cada uno.

Por ejemplo, el conjunto 00 puede almacenar los bloques 0000,0100,1000 y 1100. Si se pide la lectura de la celda 110011, que corresponde al bloque 1100, entonces se deben analizar las líneas del conjunto 00, buscando el tag 11.

## Ejercicios

3.1 Cuantos bits de una dirección se destinan a: *tag*, *conjunto* y *palabra* en el siguiente esquema:

- Una memoria principal de 32 celdas de un byte
- Una memoria cache con:
  - (a) Bloques de 4 celdas
  - (b) 4 líneas
  - (c) correspondencia asociativa por conjuntos, conjuntos de 2 líneas

## 4 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      | línea  | 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

- 4.1 Calcular la tasa de fallos se produjo en las ejecuciones del ejercicio 2.10. ¿Cómo pueden disminuirse los fallos?
- 4.2 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?
- 4.3 Se dispone de una arquitectura como la del ejercicio 2.8. Se sabe además que cada línea tiene 8 bytes (que además es el tamaño de la palabra de esta máquina). En la memoria de esta computadora está cargado a partir de la posición 0 un arreglo de números de 8 bytes, con un largo total de 256 números. El siguiente programa suma algunos de los números del arreglo, tarea que repite 1000 veces. El PC comienza en la etiqueta *main*.

```

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     //se avanza el contador
           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?

4.4 Analizar el programa del Ejercicio 4.3, pero con una cache asociativa (con las mismas características: 256B de datos, con un tamaño de línea de 8B). ¿Cuál es ahora la tasa de fallos para los ítems a) y b)? ¿Cuál es entonces la ventaja de una cache de mapeo directo?

4.5 Considerar una maquina con una memoria cache de correspondencia directa de 1024 lineas, un tamaño de bloque de 4 bytes y una memoria principal de 64 KBytes con celdas de un byte.

- (a) Dada una dirección de memoria calcular la cantidad de bits que se destinan a: *tag*, *línea* y *palabra*.
- (b) Suponer que la cache esta vacía, y que se realizan lecturas de datos cuyas direcciones están en el siguiente orden:

```
0xEA00
0xEA01
0x9A03
0xEA02
0xEA04
0xEA05
0xEA07
0xEA04
```

Determinar para cada lectura si esta produjo un fallo o un acierto.

- (c) Sabiendo que el tiempo de acceso a la memoria ram es de  $0,5\mu s$ , calcular el tiempo máximo de acceso a la cache para que el tiempo total de los accesos sea inferior a los  $2,4\mu s$ .

4.6 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? |
|-----------|-----|-------|---------|-----|-----------|
|           |     |       |         |     |           |

## 5 Ejercicios integradores

- La arquitectura **Q** se utiliza para implementar programas de procesamiento de imágenes de camaras de vigilancia...
- Suponga la siguiente secuencia de accesos a memoria, considerando una cache de mapeo directo de 16 celdas por bloque y 16 lineas:  
ABOC, ABOD, 9901, 9902, ABOE, ABOF, ABOF, ABOE
  - Indique, para cada acceso, si ocurre un fallo o un acierto, y que principio de localidad (temporal o espacial) se manifiesta.
  - ¿Cómo afecta a la tasa de aciertos si se la cambia por una cache asociativa?
- HTTP es el protocolo que se utiliza para navegar en Internet. Existe algo llamado HTTP cache. Averiguar qué es. ¿Qué parecido tiene con el cache del procesador?
- ¿Que significa “cachear” una celda de memoria? ¿Es posible que un programa funcione mejor sin memoria cache que con cache?
- La memoria caché es una memoria asociativa ¿Por qué? (Notar que no hablamos de mapeo asociativo).

## Bibliografía

- Apunte sobre jerarquía de memorias:* [http://orga.blog.unq.edu.ar/wp-content/uploads/sites/5/2015/04/orga\\_apunteMemorias1.pdf](http://orga.blog.unq.edu.ar/wp-content/uploads/sites/5/2015/04/orga_apunteMemorias1.pdf)
- Organización y Arquitectura de Computadoras*, William Stallings, Capítulo 4