



Estructura de Computadores

(Grado MI)

Memorias

Enunciados de problemas

marzo 2019

1 Sea un computador con tamaño de palabra de 32 bits, direccionamiento a nivel de byte y una memoria caché asociativa de 32 KB con bloques de 16 B y política de escritura aplazada con actualización (CBWA). En este computador se va a ejecutar un programa, del que se ha extraído el siguiente fragmento de código, que a la derecha se muestra en lenguaje de alto nivel:

<pre> and r20, r0, 0 and r1, r0, 0 buc: ld r9, r4, r20 ld r10, r5, r20 add r10, r9, r10 st r10, r6, r20 ld r9, r7, r20 add r3, r3, r9 add r20, r20, 4 add r1, r1, 1 cmp r15, r1, 2000 bbl lt r15, \$buc </pre>	<pre> for (i=0; i<2000; i++){ c[i] = b[i] + a[i]; r = r + g[i]; } </pre>
---	---

- Indique de forma razonada qué tipos de proximidad de referencias se observan y a qué se deben.
- Calcule la tasa de aciertos de la caché sabiendo que está inicialmente invalidada, las instrucciones ocupan una palabra, cada elemento de los vectores ocupan también una palabra, y que tanto el código como los vectores están alineados a bloque. Indique qué fallos se deben a instrucciones y cuáles a datos.
- Calcule el tiempo medio empleado en los accesos a memoria sabiendo que el tiempo de acceso de la memoria principal es de 50 ns y el de la caché de 2 ns.
- Indique cómo cambiarían la tasa de aciertos y el tiempo medio de acceso calculados en los apartados anteriores si la caché tuviera política de escritura inmediata sin actualización (WTWNA). ¿Con cuál de las dos políticas de escritura se obtendría mejor tiempo de acceso?

2 En un computador con palabras y direcciones de 64 bits, direccionamiento de memoria a nivel de byte, memoria caché de 32 KB con ubicación asociativa e inicialmente vacía y bloques de 16 B, se ejecuta el siguiente fragmento de código almacenado a partir de la dirección 0, en el que todas las instrucciones ocupan una palabra. La memoria caché dispone de política de escritura WTWNA (inmediata sin actualización) y su tiempo de acceso es de 4 ns.

```

LD .R3, #0
LD .R7, #0
LD .R2, #1000
LD .R5, #500
etqt: INC .R3
      LD .R4, [.R2++]
      ADD .R7, .R4
      CMP .R3, .R5
      BNZ /etqt
      MOVE .R7, .R4
      HALT

```

- Especifique la traza de referencias a memoria en la ejecución de las cuatro primeras iteraciones, indicando los fallos y aciertos en los accesos a memoria caché. Seguidamente, obtenga el número total de fallos y el número total de aciertos de memoria caché en la ejecución completa del código, detallando los debidos a instrucciones y a datos.
- Calcule la tasa de aciertos en memoria caché que se obtendría al ejecutar este código.
- Calcule el tiempo de acceso efectivo a memoria, considerando que el tiempo de acceso a memoria principal es de 40 ns.

3 Sea un procesador con tamaño de palabra de 32 bits, direccionamiento a nivel de byte y una caché asociativa de 16KB con bloques de 16B y política de escritura aplazada con actualización (CBWA). En este computador se ejecuta el siguiente fragmento de código, que se muestra a la derecha en lenguaje de alto nivel, almacenado a partir de la dirección 0, donde r1 y r2 contienen las direcciones de comienzo de los vectores b y c respectivamente, siendo sus valores iniciales r1=1024 y r2=1424.

```

        and .r7, #0
buc:    ld .r3, #0[.r1]          for (i=0; i<100; i=i+1)
        mult .r3, .r4           c[i] = a*b[i]
        st .r3, #0[.r2]
        add .r1, #4
        add .r2, #4
        add .r7, #1
        cmp .r7, #100
        bnz $buc

```

a) Calcule el número de bloques que ocupa el código y los vectores sabiendo que tanto las instrucciones como los elementos de los vectores ocupan una palabra e indique razonadamente si la caché tiene capacidad suficiente para albergarlos.

b) Especifique la traza de referencias a memoria correspondiente a la ejecución de las 4 primeras iteraciones del bucle, indicando razonadamente en cuales se produce fallo en caché. Suponga que la caché está inicialmente invalidada. Indique además qué referencias muestran proximidad espacial y cuáles proximidad temporal.

c) Calcule el número de aciertos y fallos que se producen en la caché en la ejecución completa del código, así como la tasa de aciertos de la caché.

d) Calcule el tiempo medio de acceso a memoria (tiempo efectivo) sabiendo que el tiempo de acceso de la caché es 2ns y el de la Mp es 40ns.

4 Un computador con palabras y direcciones de 64 bits y direccionamiento a nivel de byte dispone de una memoria caché, inicialmente vacía, de 24KBytes con ubicación asociativa, política de escritura WTWNA (inmediata sin actualización), tiempo de acceso de 2 ns y bloques de 32Bytes. En este computador se ejecuta el siguiente fragmento de código que se encuentra almacenado a partir de la dirección 0 y en el que todas las instrucciones ocupan una palabra.

```

AND .R1, #0
AND .R2, #0
LD .R3, #512
LD .R4, #4
LD .R5, [.R3++]
OR .R2, .R5
CMP .R1, .R4
BZ $2
ADD .R1, #1
BR $-6
HALT

```

a) Especifique la traza de referencias a memoria en la ejecución del código. Indique los fallos y aciertos en memoria caché, tanto en los accesos a instrucciones como a datos. ¿Cuántos accesos a memoria principal se realizarán durante la ejecución del código?

b) Considerando que la memoria principal tiene un tiempo de acceso de 30 ns, calcule el tiempo efectivo de los accesos a memoria.

5 En un computador con palabras y direcciones de 32 bits y direccionamiento a nivel de byte dotado de una memoria caché de 8 KB con bloques de 16 bytes, ubicación asociativa y política de escritura CBWA (aplazada, con actualización), se ejecuta el siguiente fragmento de programa:

```

bucle: ld  r5, r10, 0          ; for (i=0; i<100; i++) {
      muls r5, r5, r1          ;   a = r1 * inout[i];
      ld  r6, r11, 0          ;   b = r2 * in2[i];
      muls r6, r6, r2          ;   inout[i] = (a + b) / r3;
      add r7, r5, r6          ; }
      divs r7, r7, r3
      st  r7, r10, 0
      addu r10, r10, 4

```

```

addu r11, r11, 4
subu r20, r20, 1
cmp r8, r20, 0
bb1 lt, r8, buc

```

a) Sabiendo que el código y ambos vectores (que constan de 100 elementos) están alineados a bloque, que cada instrucción y cada elemento de los vectores ocupa una palabra y que la memoria caché está inicialmente vacía, calcule el número de accesos a memoria, distinguiendo entre código y datos, y entre accesos de lectura y de escritura. Calcule también la tasa de aciertos considerando: únicamente lecturas, únicamente escrituras y todos los accesos a memoria.

b) Considere que los tiempos de acceso de memoria caché y memoria principal son respectivamente 3ns y 60ns. Calcule el tiempo de acceso efectivo a memoria obtenido al ejecutar el fragmento de programa anterior.

c) Indique cómo cambiarían los resultados del apartado anterior si en lugar de realizarse las escrituras sobre uno de los dos vectores de entrada, se escribiera sobre un tercer vector también alineado a bloque.

6 Sea un computador con tamaño de palabra de 64 bits, direccionamiento a nivel de byte y una memoria caché asociativa de 32KB con bloques de 32B y política de escritura aplazada con actualización (CBWA). En este computador se va a ejecutar un programa, del que se ha extraído el siguiente fragmento de código en ensamblador del 88110, que se muestra a la derecha en lenguaje de alto nivel:

<pre> and r20, r0, 0 and r1, r0, 0 buc: ld r9, r4, r20 ld r10, r5, r20 and r10, r9, r10 st r10, r6, r20 ld r9, r5, r20 add r3, r3, r9 add r20, r20, 8 add r1, r1, 1 cmp r15, r1, 1024 bb1 lt r15, \$buc </pre>	<pre> for (i=0; i<1024; i++){ c[i] = b[i] & a[i]; r = r + a[i]; } </pre>
---	---

a) Indique de forma razonada qué tipos de proximidad de referencias se observan, y a qué se deben.

b) Calcule cuántos bloques ocupan, tanto el código como los vectores, en memoria principal y justifique si la caché tiene capacidad suficiente para albergar a todos ellos. Para ello tenga en cuenta que las instrucciones y cada uno de los elementos de los vectores ocupan una palabra y que, tanto el código como los vectores están alineados a bloque.

c) Calcule la tasa de aciertos de la caché sabiendo que inicialmente está invalidada. Indique qué fallos son de lectura y cuáles de escritura.

d) Calcule el tiempo medio de acceso a memoria sabiendo que el tiempo de acceso de la memoria principal es de 50 ns y el de la caché de 2 ns.

7 Sea un computador que dispone de una memoria cache cuyo tiempo de acceso es de 10 ns y una memoria principal en la que se emplean 60 ns para leer o escribir una palabra y 120 ns para leer o escribir un bloque. Obtenga justificadamente el tiempo de acceso que se empleará en cada uno de los siguientes casos de acierto o fallo en cache:

a) Con política de escritura inmediata sin actualización (WTWNA):

a.1) Acierto de lectura.

a.2) Fallo de lectura.

a.3) Acierto de escritura.

a.4) Fallo de escritura.

b) Con política de escritura aplazada con actualización (CBWA):

b.1) Acierto de lectura.

- b.2)** Fallo de lectura y bloque a reemplazar modificado.
- b.3)** Acierto de escritura.
- b.4)** Fallo de escritura y bloque a reemplazar no modificado.

8 Considere un microprocesador con palabras de 16 bits, direcciones de 20 bits y direccionamiento a nivel de byte, cuyo sistema de memoria tiene las siguientes características:

- Memoria principal con tiempo de acceso de 100 ns.
- Memoria caché única, de 16KB, bloques de 8B, ubicación asociativa, política de escritura inmediata sin actualización (WTWNA) y tiempo de acceso 10 ns.

En este sistema se ejecuta un programa del que se ha extraído el siguiente fragmento (cada instrucción y cada dato ocupa una palabra –16 bits–):

```

PROG:  addu  r3, r0, 100    ; r3 <-- 100
BUC:   ld    r1, r2, r0     ; r1 <-- MEM(r2+0)
      add   r2, r2, 2       ; r2 <-- r2+2
      subu  r30, r30, 2     ; (r30 es el puntero de pila)
      st    r1, r30, r0     ; MEM(r30) <-- r1
      subu  r3, r3, 1       ; r3 <-- r3-1
      cmp   r4, r3, r0
      bbl   ne, r4, BUC     ; si r3 != 0 ==> salto a BUC

```

- a)** Justifique cuál es el mínimo y el máximo tiempo de acceso en este sistema y en qué casos se produce cada uno de ellos (lectura/escritura, acceso a instrucciones/datos, acierto/fallo).
- b)** Sabiendo que al comienzo de la ejecución la memoria caché está invalidada, calcule su tasa de aciertos al ejecutar el fragmento de programa dado.
- c)** Calcule el tiempo medio consumido en realizar los accesos a memoria durante la ejecución del fragmento de programa dado.
- d)** Suponiendo que se sustituya la tercera instrucción: `add r2, r2, 2`, por `add r2, r2, 8`, es decir, que se recorra de cuatro en cuatro elementos el vector apuntado por `r2`, calcule la nueva tasa de aciertos al ejecutar el fragmento de programa dado.

9 En un computador con palabras y direcciones de 32 bits, direccionamiento a nivel de byte, una memoria caché asociativa de 64Kbytes inicialmente vacía, bloques de 32bytes y política de escritura inmediata sin actualización (WTWNA), se va a ejecutar el siguiente fragmento de un programa en ensamblador del estándar IEEE en el que tanto las instrucciones como los datos ocupan una palabra. Suponga que este código está almacenado a partir de la dirección 0:

```

LD      .R2, #0
LD      .R7, #4000
LD      .R4, #5
LD      .R3, [.R7]
LD      .R5, #4[.R7]
ADD     .R3, .R5
MOVE    .R3, .R1
ADD     .R2, #1
ADD     .R7, #8
SUB     .R4, #1
BNZ     $-32
WAIT

```

- a)** Represente su traza de ejecución y determine el número de fallos y el número de aciertos que se producirán en los accesos a caché, tanto para instrucciones como para datos. Especifique las direcciones de memoria en las que se producen los fallos.
- b)** Calcule la tasa de aciertos y el tiempo de acceso efectivo a memoria sabiendo que el tiempo de acceso de la memoria principal es de 60 ns y el de la memoria caché de 3 ns.

10 Considerando un computador de 64 bits, con una memoria caché asociativa e inicialmente vacía y una memoria principal cuyos respectivos tiempos de acceso son de 1ns y 15ns, con direccionamiento a byte y bloques de caché de 16 bytes, responda a las siguientes preguntas teniendo en cuenta que la política de escritura es CBWA (aplazada con actualización). Justifique sus respuestas.

Si la probabilidad de que un bloque esté modificado es del 25 %

- ¿Cuál es el tiempo medio de escritura en caso de haber fallo de caché? ¿Y si hubiera acierto?
- ¿Cómo variarían los tiempos, calculados anteriormente, si la política de escritura fuese WTWNA (inmediata sin actualización)?
- ¿Qué influencia tendría, sobre los tiempos de escritura en ambas políticas, que se aumentara el tamaño de los bloques a 32 bytes?

11 Considere un procesador con palabras y direcciones de 32 bits y direccionamiento a nivel de byte, cuyo sistema de memoria tiene las siguientes características:

- Memoria principal con tiempo de acceso de 100 ns.
- Memoria caché única, de 64 KB, bloques de 32 bytes, ubicación asociativa, política de escritura inmediata sin actualización (WTWNA) y tiempo de acceso 10 ns.

En este sistema se ejecuta un programa del que se ha extraído el siguiente fragmento que trabaja con vectores de 256 palabras (cada instrucción y cada dato ocupa una palabra –32 bits–):

```
BUC:  ld r10, r11, 0          ; r10 <-- MEM[r11+0] ; for (i=0; i<256; i=i+1){
      add r15, r15, r10      ; r15 <-- r15 + r1   ;   r = r + a[i];
      ld r2, r12, 0         ; r2 <-- MEM[r12+0] ;   a[i] = b[i];
      st r2, r11, 0         ; MEM[r11+0]<-- r2    ; }
      add r11, r11, 4        ; r11 <-- r11 + 4
      add r12, r12, 4        ; r12 <-- r12 + 4
      add r20, r20, 1        ; r20 <-- r20 + 1
      cmp r7, r20, r3        ; r7 <-- ¿r20 <=> r3?
      bnz r7, $BUC          ; Si r7 != 0 >> salto a BUC ; $ fin
```

- Indique y justifique los tiempos que se emplean en realizar cada acceso de lectura con acierto, lectura con fallo, escritura con acierto y escritura con fallo de caché.
- Calcule el número total de accesos a memoria en el fragmento de programa anterior. Sabiendo que al comienzo de la ejecución la memoria caché está invalidada, calcule cuántos accesos de lectura y cuántos de escritura producen fallo de caché.

12 En un computador con palabras y direcciones de 64 bits y direccionamiento a nivel de byte se dispone de una memoria caché de 64 KB con bloques de 32 B, inicialmente vacía, cuya ubicación es asociativa y su política de escritura WTWNA (inmediata sin actualización). El siguiente fragmento de código está almacenado a partir de la dirección 0 y todas sus instrucciones son de una palabra. Los desplazamientos de las instrucciones de bifurcación se indican en palabras y todos los valores se expresan en decimal.

```
AND .R5, #0
LD .R6, #500
LD .R9, #1024
LD .R8, [.R9++]
AND .R8, #2047
MOVE .R8, R10
CMP .R5, .R6
BZ $2
ADD .R5, #1
BR $-7
WAIT
```

- Obtenga la traza de referencias a memoria en la ejecución de las cinco primeras iteraciones y determine justificadamente el número de fallos de memoria caché, tanto en los accesos a instrucciones como a datos, para la ejecución completa del código.

b) Teniendo en cuenta que los tiempos de acceso de la memoria caché y de la memoria principal son, respectivamente, 3 ns y 40 ns, calcule la tasa de acierto en memoria cache y el tiempo efectivo de los accesos a memoria.

13 Un computador dispone de una memoria caché cuyo tiempo de acceso es de 2 ns y de una memoria principal en la que se emplean 60 ns para leer o escribir una palabra (32 bits). El tamaño del bloque es de cuatro palabras. Determine justificadamente el tiempo de escritura cuando hay un fallo de memoria caché suponiendo que:

a) La política de escritura es CBWA (aplazada con actualización) y

a.1) El bloque a reemplazar está modificado.

a.2) El bloque a reemplazar no está modificado

b) La política de escritura es WTWNA (inmediata sin actualización).

14 Considere un controlador industrial basado en un procesador con palabras y direcciones de 32 bits y direccionamiento a nivel de byte. El sistema posee una memoria caché asociativa de 32 KB inicialmente vacía, con bloques de 16 bytes y política de escritura inmediata sin actualización (WTWNA). En este sistema se ejecuta el siguiente fragmento de un programa almacenado a partir de la dirección 0 en el que tanto las instrucciones como los datos ocupan una palabra:

```
          add    r10, r0, 800
          add    r12, r0, 0
          add    r3, r0, 20
buc:      ld     r20, r10, 0
          ld     r21, r10, 4
          add    r12, r12, r20
          sub    r12, r12, r21
          st     r12, r10, 0
          add    r10, r10, 8
          sub    r3, r3, 1
          cmp    r4, r3, 0
          bbl    gt, r4, buc
```

a) Represente su traza de ejecución para las tres primeras iteraciones del bucle. Considerando la ejecución del fragmento de código completo, calcule el número total de accesos y el número de fallos que se producirán en los accesos a caché, distinguiendo entre los producidos en las instrucciones y en los datos. Especifique en qué direcciones de memoria se producen los fallos.

b) Calcule la tasa de aciertos y el tiempo de acceso efectivo a memoria sabiendo que el tiempo de acceso de la memoria principal es de 80 ns y el de la memoria caché de 4 ns.