

FUNDAMENTOS DE COMPUTADORES

Boletín de ejercicios resueltos del tema 4

1. Memoria Principal

1. Si un sistema tiene 1 GB de memoria principal, ¿cuántos bits tienen las direcciones de memoria física? ¿cuál es el rango de direcciones de memoria válidas? Indica si son válidas las siguientes direcciones

0x200C0008
0x0000FFF8
0x53FA47AC
0x47AC
0x153FA47AC

Solución: Como 1 GB son 2^{30} bytes, las direcciones de memoria física tendrán 30 bits de longitud. Si las expresamos en hexadecimal, necesitaremos $\lceil 30/4 \rceil = 8$ caracteres hexadecimales, de los cuales, el más significativo representa solo $30 \% 4 = 2$ bits. Entonces, el rango de direcciones válidas será de 0x0000 0000 a 0x3FFF FFFF.

0x200C0008 SI
0x0000FFF8 SI
0x53FA47AC NO, porque está fuera de rango (requiere 31 bits)
0x47AC SI (es igual a 0x000047AC)
0x153FA47AC NO, porque está fuera de rango (requiere 33 bits)

2. En un sistema las direcciones físicas tienen 32 bits. ¿Cuánto es el máximo de memoria principal que podemos utilizar? ¿Qué ocurre si tenemos 2 GB de memoria? ¿Y si tenemos 8 GB?

Solución: Con 32 bits podremos direccionar $2^{32} \text{ bytes} = 4 \text{ GB}$ de memoria. Si solo tenemos 2 GB (2^{31} bytes), todas aquellas direcciones que necesiten más de 31 bits no serán legales. Si tenemos 8 GB de memoria (2^{33} bytes), no podremos acceder a la mitad de la memoria, porque necesitaríamos un bit adicional.

3. Dada una memoria entrelazada de orden superior con 2^{28} posiciones de memoria y con 2^{25} posiciones de memoria por modulo, ¿qué bits de la dirección indican el número de modulo?

Solución: Si hay 2^{28} posiciones de memoria, nuestras direcciones tendrán 28 bits. Como cada módulo contiene 2^{25} posiciones de memoria, hay $\frac{2^{28}}{2^{25}} = 2^{28-25} = 2^3$ módulos. Entonces, 3 bits de la dirección servirán para identificar el módulo, y el resto para la posición dentro del módulo. Como la memoria tiene entrelazamiento de orden superior, los primeros 3 bits seleccionan el módulo, y los 25 restantes la posición dentro del módulo.

4. Un programa recorre secuencialmente un array unidimensional, accediendo por tanto a posiciones de memoria consecutivas. Si en nuestro equipo disponemos de varios módulos de memoria, ¿qué tipo de entrelazamiento será más beneficioso?

Solución: Como podemos acceder simultáneamente a datos en diferentes módulos, dado el orden en el que accedemos en el programa es preferible que posiciones de memoria consecutivas se alojen en módulos diferentes. Esto se consigue con entrelazamiento de orden inferior.

5. Sea una memoria principal compuesta por 8 módulos de 4 palabras por modulo. El procesador emite la siguiente secuencia de direcciones físicas (todas son peticiones de lectura): 09, 20, 15, 18, 19, 21, 22 y 23. Calcula el número de palabras por ciclo que se pueden leer: a) Con entrelazamiento de orden superior, b) Con entrelazamiento de orden inferior.

Solución: Podremos leer varias direcciones de memoria en el mismo ciclo siempre y cuando los módulos de memoria en los que se encuentran sean diferentes, independientemente del orden en el que accedamos a ellos.

a) Entrelazamiento de orden superior

La memoria se estructura de la siguiente forma:

M0	M1	M2	M3	M4	M5	M6	M7
00	04	08	12	16	20	24	28
01	05	09	13	17	21	25	29
02	06	10	14	18	22	26	30
03	07	11	15	19	23	27	31

Cada dirección x se encuentra en el módulo $\lfloor x/8 \rfloor$.

- Ciclo 1: podemos leer las direcciones 09, 20, 15 y 18 (módulos 2, 5, 3 y 4).
- Ciclo 2: direcciones 19 y 21 (módulos 4 y 5)
- Ciclo 3: dirección 22 (módulo 5)
- Ciclo 4: dirección 23 (módulo 5)

Tardamos por tanto 4 ciclos en leer las 8 palabras

b) Entrelazamiento de orden inferior

La memoria se estructura de la siguiente forma:

M0	M1	M2	M3	M4	M5	M6	M7
00	01	02	03	04	05	06	07
08	09	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Cada dirección x se encuentra en el módulo $x \% 8$.

- Ciclo 1: podemos leer las direcciones 09, 20, 15, 18, 19, 21 y 22 (módulos 1, 4, 7, 2, 3, 5 y 6).
- Ciclo 2: dirección 23 (módulo 7)

En este caso tardamos solo 2 ciclos en leer las 8 palabras

2. Memoria Caché

6. Un computador dispone de un sistema de memoria central constituido por una memoria principal Mp y una cache Mc. Mp tiene una dimensión de 128K palabras y está estructurada como un conjunto de módulos de 8K palabras con entrelazado de orden inferior. La Mc tiene un tamaño de 2K palabras con líneas de 256 palabras y una correspondencia directa. Se pide:
- Número de módulos de Mp.
 - Interpretación de los bits de la dirección física del sistema de memoria para Mp
 - Interpretación de los bits de la dirección física del sistema de memoria para Mc.

Solución:

- $Mp = 128K \text{ palabras} = 2^{17} \text{ palabras}$; Módulos = $8K \text{ palabras} = 2^{13} \text{ palabras}$; $2^{17}/2^{13} = 2^4 = 16 \text{ módulos}$
- Las direcciones físicas tienen 17 bits. Los 4 bits menos significativos determinan el módulo, y el resto la dirección dentro del módulo.

n = 17 bits

13 bits	4 bits
dirección dentro del módulo	módulo

- Tamaño de línea = $256 = 2^8 \text{ palabras}$; Tamaño de caché = $2K = 2^{11} \text{ palabras}$
Número de líneas = $2^{11}/2^8 = 2^3 = 8 \text{ líneas}$

De los 17 bits de la dirección física, 8 indican el desplazamiento dentro de la línea, 3 es el índice para seleccionar la línea, y el resto es la etiqueta.

n = 17 bits

6 bits	3 bits	8 bits
etiqueta	índice	desplazamiento

7. Una memoria caché asociativa por conjuntos consta de 16 conjuntos con 4 vías por conjunto, con un tamaño de línea de 512 bytes. La memoria principal tiene una capacidad de 4MB. A la dirección de memoria principal, 0x2864C0 (en binario, 10 1000 0110 0100 1100 0000), ¿qué conjunto le corresponde (expresado en decimal)?

Solución:

DATOS

- Memoria caché (Mc) asociativa por conjuntos.
- 16 (2^4) conjuntos
- 4 (2^2) vías/ conjunto
- Memoria principal $C_{Mp} = 4 \text{ M}$ (2^{22}) bytes
- 512 (2^9) bytes/línea

Puesto que la Mp tiene 2^{22} bytes se requieren direcciones de 22 bits de longitud.

En segundo lugar se va a calcular el número de líneas L de la Mc:

$$L = 16 \text{ conjuntos} \times 4 \text{ líneas/conjunto} = 2^6 \text{ líneas en Mc}$$

El formato de una dirección de la memoria caché que utiliza una función de correspondencia asociativa por conjuntos es:

n = 22 bits		
9 bits	4 bits	9 bits
etiqueta	índice	desplazamiento

Luego la dirección que se nos plantea es:

etiqueta	índice	desplazamiento
101000011	0010	011000000

El conjunto de Mc al que hace referencia esta dirección es $0010 = (2)_{10}$

8. Un computador tiene una unidad de memoria de 8MB y una memoria caché de 4KB con un tamaño de línea de 256 bytes. Suponer que se hace una referencia a la dirección de memoria principal 0x006CA0 (en binario, 000 0000 0110 1100 1010 0000).
- Si la memoria caché utiliza correspondencia directa, ¿en qué línea de la memoria caché sería posible encontrar esa dirección de memoria principal? ¿Qué etiqueta habría que buscar en esa línea para saber si esa dirección se encuentra en la memoria caché?
 - Si la memoria caché utiliza correspondencia totalmente asociativa, ¿qué etiqueta habría que buscar
 - Si la memoria caché utiliza asociatividad por conjuntos de 2 vías, ¿en qué conjunto de la memoria caché debemos buscar esa dirección? ¿qué etiqueta habría que buscar para saber si esa dirección se encuentra en la memoria caché?

Solución: DATOS

- Memoria principal $C_{Mp} = 8 \text{ M } (2^{23})$ bytes
- Memoria caché $C_{Mc} = 4 \text{ K } (2^{12})$ bytes
- 256 (2^8) bytes/línea

De los datos del enunciado se deduce que el número de líneas L de Mc es:

$$L = 2^{12} \text{ bytes} / (2^8 \text{ bytes/línea}) = 2^4 \text{ bloques en Mc}$$

a) La memoria caché utiliza una función de correspondencia directa. Por lo tanto el formato de una dirección desde el punto de vista de la memoria caché es :

n = 23 bits		
11 bits	4 bits	8 bits
etiqueta	índice	desplazamiento

Luego dada la dirección en binario del enunciado,

etiqueta	índice	desplazamiento
00000000110	1100	10100000

La línea de la Mc a la que hace referencia es la número 12 (1100).

La etiqueta que habría que buscar es 0x006 (000 0000 0110).

b) La memoria caché utiliza una función de correspondencia totalmente asociativa. Por lo tanto el formato de una dirección desde el punto de vista de la memoria caché es :

n = 23 bits	
15 bits	8 bits
etiqueta	desplazamiento

Luego dada la dirección en binario del enunciado,

etiqueta	desplazamiento
000000001101100	11000000

La etiqueta que habría que buscar es 0x006C (000 0000 0110 1100).

c) La memoria caché tiene asociatividad por conjuntos de 2 vías. Por tanto, las 16 líneas de la memoria caché se estructuran en $16/2=8$ conjuntos (2^3). Entonces, el formato de una dirección desde el punto de vista de la memoria caché es :

n = 23 bits		
12 bits	3 bits	8 bits
etiqueta	índice	desplazamiento

Luego dada la dirección en binario del enunciado,

etiqueta	índice	desplazamiento
000000001101	100	11000000

El conjunto de la Mc a la que hace referencia es el número 4 (100).

La etiqueta que habría que buscar es 0x00D (0000 00000 1101).

9. Un sistema tiene una memoria física de 1MB direccionables a nivel de byte. Está dotado de una única caché de datos de correspondencia directa de 32 bytes con líneas de 4 Bytes. Cada acierto en esa cache se resuelve en 1 ciclo mientras que la penalización de un fallo es de 10 ciclos. Determina el contenido de las etiquetas de todas las líneas de la caché de datos, inicialmente vacía, después de acceder a las siguientes direcciones: $(FF200)_{hex}$, $(FF000)_{hex}$, $(FF204)_{hex}$, $(FF004)_{hex}$, $(FF202)_{hex}$, $(FF008)_{hex}$.

Solución:

La caché tiene 32 bytes y está organizada en conjuntos de una línea (correspondencia directa), donde cada línea tiene $4 = 2^2$ bytes. Por lo tanto la caché tiene $32/4 = 8$ conjuntos (o líneas).

Por otra parte la memoria física es de $1 \text{ MB} = 2^{20}$ bytes, por lo que las direcciones físicas tiene 20 bits.

20 bits		
15 bits	3 bits	2 bits
Etiqueta	Indice	Δ

- Accedemos a la dirección $(FF200)_{hex}$ que se desglosa como:

Etiqueta	Indice	Δ
111 1111 1001 0000	000	00
$7F90_{hex}$	0_{10}	0_{10}

Ello supone la carga del dato con etiqueta $(7F90)_{hex}$ en el conjunto 0. El resto permanecen vacíos.

- Accedemos a la dirección $(FF000)_{hex}$:

Etiqueta	Indice	Δ
111 1111 1000 0000	000	00
$7F80_{hex}$	0_{10}	0_{10}

Se carga el dato con etiqueta $(7F80)_{hex}$ en el conjunto 0 que reemplaza al que acabábamos de cargar.

- Accedemos a la dirección $(FF204)_{hex}$:

Etiqueta	Indice	Δ
111 1111 1001 0000	001	00
$7F90_{hex}$	1_{10}	0_{10}

Se carga el dato con etiqueta $(7F90)_{hex}$ en el conjunto 1.

- Accedemos a la dirección $(FF004)_{hex}$:

Etiqueta	Indice	Δ
111 1111 1000 0000	001	00
$7F80_{hex}$	1_{10}	0_{10}

Se carga el dato con etiqueta $(7F80)_{hex}$ en el conjunto 1 que reemplaza al que acabábamos de cargar.

- Accedemos a la dirección $(FF202)_{hex}$:

Etiqueta	Indice	Δ
111 1111 1001 0000	000	10
$7F90_{hex}$	0_{10}	2_{10}

Se carga el dato con etiqueta $(7F90)_{hex}$ en el conjunto 0 que reemplaza a la etiqueta $(7F80)_{hex}$.

- Accedemos a la dirección $(FF008)_{hex}$:

Etiqueta	Indice	Δ
111 1111 1000 0000	010	00
$7F80_{hex}$	2_{10}	0_{10}

Se carga el dato con etiqueta $(7F80)_{hex}$ en el conjunto 2.

El directorio caché queda de la siguiente forma:

Conjunto 0	7F90
Conjunto 1	7F80
Conjunto 2	7F80
Conjunto 3	
...	
Conjunto 7	

3. Memoria Virtual

10. Si un sistema tiene 2 GBytes de memoria virtual y 256 MBytes de memoria física,
- a) ¿Qué longitud tienen las direcciones virtuales y físicas?
 - b) ¿Cuántas páginas virtuales y físicas hay, si el tamaño de página es de 1 KByte?

Solución:

- a) Para direccionar 2 GBytes (2^{31} Bytes) necesitamos 31 bits, y para direccionar 256 MBytes (2^{28} Bytes) necesitamos 28 bits. Por tanto las direcciones virtuales tienen una longitud 31 bits, y las direcciones físicas 28 bits.
 - b) Tanto el espacio virtual como físico se divide en bloques de 1 KByte (2^{10} Bytes). Entonces, habrá $\frac{2GB}{1KB} = \frac{2^{31}}{2^{10}} = 2^{21} = 2,097,152$ páginas virtuales, y $\frac{256MB}{1KB} = \frac{2^{28}}{2^{10}} = 2^{18} = 262,144$ páginas físicas.
11. En un computador con memoria virtual paginada, las direcciones virtuales tienen una longitud de 40 bits. El tamaño de página es de 4 KB y cada entrada en la tabla de páginas ocupa 1 Byte.
- a) ¿Cuál es el tamaño del espacio virtual?
 - b) Si el sistema cuenta con un esquema de traducción directa en un nivel, ¿cuál es el tamaño de cada tabla de páginas?
 - c) Si el sistema utiliza traducción en dos niveles, ¿cuántas tablas de nivel 2 tendremos si cada una de ellas ocupa exactamente una página?
 - d) En el supuesto anterior, ¿cuántas páginas ocupa la tabla de primer nivel?

Solución:

- a) Con 40 bits podemos direccionar 2^{40} Bytes (1 TByte)
 - b) Hay en total $\frac{2^{40}}{2^{12}} = 2^{28}$ páginas virtuales, y por tanto la tabla de páginas tendrá 2^{28} entradas. Como cada una de ellas ocupa 1 Byte, la tabla de páginas ocupará 2^{28} Bytes (256 MBytes).
 - c) En una página de 2^{12} Bytes caben 2^{12} entradas de 1 Byte cada una. Por tanto necesitaremos $\frac{2^{28}}{2^{12}} = 2^{16} = 65536$ tablas de nivel 2.
 - d) Como hay 2^{16} tablas de nivel 2, la tabla de nivel 1 tiene 2^{16} entradas. Ocupa 2^{16} entradas \times 1 Byte/entrada = 2^{16} Bytes, o $\frac{2^{16}}{2^{12}} = 2^4 = 16$ páginas.
12. Un computador tiene una memoria virtual paginada con un esquema de traducción directa en un nivel, donde el espacio virtual es de 4 GBytes (2^{32} B), el espacio físico es de 128 MBytes (2^{27} B) y el tamaño de página es de 4 KBytes (2^{12} B). Cada entrada en la tabla de páginas contiene un bit de residencia y el número de página física.
- a) ¿Cuántas entradas tiene la tabla de páginas?
 - b) ¿Cuánto ocupa cada entrada en la tabla de páginas?
 - c) ¿Cuánto ocupa la tabla de páginas?

- d) Describe el proceso de traducción de la dirección virtual 0xA9347562 a su correspondiente dirección física si en la entrada de la tabla de páginas encontramos el contenido 0xF340.

Solución:

- a) Como el espacio virtual tiene 4 GB y se divide en páginas de 4 KB, habrá en total $\frac{4GB}{4KB} = \frac{2^{32}}{2^{12}} = 2^{20}$ páginas virtuales (PV). La tabla de páginas tiene por tanto 2^{20} entradas.
- b) Como el espacio físico tiene 128 MB y se divide en páginas de 4 KB, habrá en total $\frac{128MB}{4KB} = \frac{2^{27}}{2^{12}} = 2^{15}$ páginas físicas (PF). El número de página física tiene por tanto una longitud de 15 bits. En cada entrada de la tabla de páginas encontramos 1 bit de residencia + 15 bits de PF = 16 bits (2 Bytes).
- c) La tabla de páginas (TP) tiene 2^{20} entradas de 2 Bytes cada una. Ocupa por tanto $2^{20} \times 2 = 2^{21}$ Bytes (2 MB).
- d) El tamaño del espacio virtual determina la longitud de las direcciones virtuales ($2^{32} \text{ Bytes} \rightarrow$ direcciones de 32 bits. Esos 32 bits se dividen en número de página virtual (20 bits para referenciar las 2^{20} PVs) y desplazamiento (12 bits para referenciar el Byte dentro de los 2^{12} Bytes de cada página):

Pág.Virtual	Desplaz.
A9347	562
20 bits	12 bits

La entrada número 0xA9347 de la TP contiene 0xF340, que se divide en 1 bit de residencia y el número de página física:

0xF340	
BR	PF
1	111 0011 0100 0000
1 bit	15 bits (0x7340)

La dirección resultante es la página física concatenada con el desplazamiento: 0x7340562. Tiene 27 bits, como corresponde por el tamaño de la memoria principal (2^{27} Bytes)

Pág.Física	Desplaz.
7340	562
15 bits	12 bits