2 Representación de números naturales

2.1 Introducción

Existen diferentes formas de representar valores numéricos. De momento, en este capítulo, nos limitamos a representar números del conjunto de los naturales: $\mathsf{N} = \{0, 1, 2, \ldots\}$ (en inglés, *unsigned integers* o *non-negative integers*). Más adelante, en el capítulo 4, veremos cómo representar números del conjunto de los enteros: $\mathsf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ (*signed integers* o simplemente *integers*). En este capítulo, si no se indica lo contrario, cuando hablemos de números nos referimos siempre a números naturales.

Un número es un ente abstracto. Podemos pensar en tres manzanas y representarlas en nuestra imaginación fácilmente. Sabemos desde bastante pequeños qué significa el número tres y aprendemos a representarlo mediante un símbolo: el 3. No obstante, este sistema de representación de un símbolo diferente para cada número no es escalable y se hace inviable. Para números grandes necesitamos un sistema que sea potente, que permita representar cualquier número por grande que sea. Hemos aprendido a representar los números naturales en el sistema decimal. Los computadores cuando representan y procesan números naturales lo hacen en un sistema de representación muy parecido al decimal, pero más eficiente para el computador: el sistema binario. Por ello empezaremos por repasar lo que sabemos del sistema decimal.

2.2 Sistema convencional en base b

El sistema convencional en una base genérica b es una generalización del sistema decimal. El sistema binario es un caso particular del sistema convencional en base b, el caso en que b vale 2, al igual que el decimal es el caso con b igual a 10.

2.2.1 Decimal

En el sistema decimal tenemos diez símbolos diferentes para representar cada uno de los números naturales del cero al nueve: 0, 1, 2, 3,..., 8, y 9. Estos símbolos se denominan **dígitos**, tal vez porque son diez y tenemos diez dedos (en la escuela también se les denomina “cifras”). Pero, ¿qué pasa cuando se trata de un número más grande? No podemos inventarnos símbolos diferentes para cada uno de los infinitos números naturales. En el sistema decimal, un número mayor que 9 se representa mediante un **vector de dígitos**. Así, representamos el cinco mil doscientos cincuenta y seis como: 5256. Decimos vector de dígitos porque la posición que ocupa cada dígito en la secuencia es importante: el decimal es un sistema **posicional**. No "cuenta" lo mismo el cinco que está a la izquierda de la cadena que el que está entre el 2 y el 6, aunque los dos son el mismo símbolo. El **peso** de cada dígito depende de la posición que ocupa en el vector que representa el número y estos pesos son las potencias de 10. Así, el vector de dígitos 5256 es una representación del número que resulta de calcular:

$$5 \times 1000 + 2 \times 100 + 5 \times 10 + 6 \times 1$$

Que expresado usando las potencias de 10 (10^0=1, 10^1=10, 10^2=100, 10^3=1000,...) es:

$$5 \times 10^3 + 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$$

En general, el vector de dígitos $x_{n-1}x_{n-2} \ldots x_2x_1x_0$ con $x_i \in \{0, 1, 2, ..., 8, 9\}$ representa, en el sistema decimal, el número natural con valor:

$$x_{n-1}10^{n-1} + x_{n-2}10^{n-2} + \cdots + x_2 10^2 + x_1 10 + x_0$$

que usando la notación del sumatorio se expresa como:

$$\sum_{i=0}^{n-1} x_i 10^i$$

Esta fórmula expresa el valor del número en función de los dígitos que lo representan. Si realizamos las operaciones de multiplicación de los dígitos por las potencias de 10 y las sumas de los resultados usando el sistema decimal, obtenemos, claro está, el número representado en decimal.

2.2.2 Binario

En un computador los números naturales se representan en el sistema binario. Una de las diferencias con el decimal es que en binario sólo hay dos símbolos o dígitos diferentes en vez de los 10 del decimal: el 0 y el 1. El valor numérico que representan estos dos símbolos es el mismo que representan el 0 y el 1 en decimal. En binario, cada uno de los dígitos se denomina dígito binario, o simplemente **bit** (***bi****nary digi****t***). La otra diferencia entre binario y decimal es que el peso de cada dígito según la posición que ocupa en el vector que representa al número sigue las potencias de 2 en vez de las de 10.

De la representación de un número a su valor

Definimos el sistema binario como sigue. El vector de n bits $X = x_{n-1}x_{n-2} \ldots x_2x_1x_0$, con $x_i \in \{0,1\}$, representa en el sistema binario al número natural (*unsigned*) que denotamos X_u, siendo:

$$X_u = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \cdots + x_2 2^2 + x_1 2^1 + x_0 2^0$$

que también se puede expresar usando la notación del sumatorio como:

$$X_u = \sum_{i=0}^{n-1} x_i 2^i$$

Una vez más, si realizamos los cálculos de la expresión anterior usando el sistema de representación decimal obtendremos el valor del número representado en decimal. La propia definición del sistema nos da la forma de pasar un número de su representación en binario a su representación en decimal.

Ejemplo 1

$X = 01011$ representa en binario al valor: $X_u = 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$, que en decimal se representa por 11.

Ejemplo 2

En la columna de la izquierda de la tabla se muestra la representación (vector de bits) de los números naturales del 0 al 15 usando el sistema binario y en la columna de la derecha el número natural que representa cada vector de bits (usando para ello la representación en decimal).

Vector de bits $X = x_3\ x_2\ x_1\ x_0$	**Valor** X_u
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	10
1 0 1 1	11
1 1 0 0	12
1 1 0 1	13
1 1 1 0	14
1 1 1 1	15

Notación

En el computador, un vector de n bits (las tensiones eléctricas binarias en n cables ordenados) puede representar muchas cosas diferentes, según la regla de codificación usada. Así que, dado un vector de bits, para saber qué representa hay que saber con qué sistema de representación hay que interpretarlo.

En general, al vector de bits lo denotamos con una letra mayúscula o un nombre en mayúsculas, por ejemplo X o DATA, y al valor, al número que representa ese vector de bits, lo denotaremos con la misma letra o nombre del vector pero con el subíndice “u” de *unsigned*, que indica que hemos usado el sistema de representación de números naturales, que en el computador es el sistema binario, para interpretar el vector de bits. Por ejemplo, X_u o $DATA_u$.

2.2.3 Generalización: sistema convencional en base b

Como ya hemos visto, tanto el sistema de representación decimal como el binario son sistemas posicionales: la posición del dígito dentro del vector es importante para la representación. Son sistemas con base fija: el peso del dígito i es b^i (la base b es la misma para todas las posiciones i del vector de dígitos). La diferencia entre ellos es que en decimal la base b es 10 y en binario es 2. En ambos sistemas un dígito puede valer $0 \le x_i \le b-1$. A continuación definimos el “sistema convencional en base b” que es una generalización de estos tipos de sistemas.

2.1 En el **sistema convencional en base b**, para $b \in N$ y $b \ge 2$, el vector de n dígitos $X = x_{n-1}x_{n-2} \ \ldots \ x_2x_1x_0$ con $x_i \in \{0,..., b\text{-}1\}$, representa al número con valor:

$$X_u = x_{n-1}b^{n-1} + x_{n-2}b^{n-2} + x_{n-3}b^{n-3} + \cdots + x_2b^2 + x_1b + x_0 , \qquad \textbf{(EQ 1)}$$

que con la notación del sumatorio se expresa como:

$$X_u = \sum_{i=0}^{n-1} x_i b^i \qquad \textbf{(EQ 2)}$$

1

2.2.4 Hexadecimal

De los sistemas convencionales en base b, además del decimal y del binario, también usaremos el hexadecimal (b=16). Por convenio, en hexadecimal, a los dígitos con valor 10, 11,..., 15 se les asigna los símbolos A, B,..., F, respectivamente. El valor del número representado en hexadecimal por 2BD5 es: $2 \times 16^3 + 11 \times 16^2 + 13 \times 16^1 + 5 \times 16^0$, que en decimal queda representado por el vector de dígitos 11221.

Una vez más, obtenemos la representación del número en decimal si representamos b y el valor de los dígitos x_i en decimal y realizamos las operaciones de EQ (2) en decimal.

Notación

Cuando en el computador tenemos un vector X de n bits (en un computador n suele ser grande: 8, 16, 32 o 64) es cómodo usar el hexadecimal para representar por escrito el vector de n bits, ya que en hexadecimal necesitamos solamente $n/4$ dígitos. Por ejemplo, para $n = 16$, si tenemos $X = 1000110101111010$ es más claro escribir X = 8D7A. Cuando la representación sea en hexadecimal, para diferenciarla de la representación en binario, pondremos el prefijo 0x antes del vector de dígitos hexadecimales, X = 0x8D7A. Por último, usaremos el sistema decimal para expresar el valor concreto X_u, que es el sistema usual entre nosotros. Por ejemplo, para X = 0x8D7A escribiremos $X_u = 36218$.

2.2.5 Rango de la representación en un sistema convencional

El rango de la representación indica qué números se pueden representar usando un vector de n dígitos. De la EQ (1), o de la EQ (2), se deduce que el menor valor para X_u se da cuando todos los sumandos valen cero: todos los dígitos valen 0 (ya que X_u no puede tomar valores negativos, pues los dígitos y la base son positivos). Por lo tanto, el menor valor de X_u es 0.

¿Cuál es el mayor número que se puede representar en base b con n dígitos? De las mismas ecuaciones se deduce que el mayor valor de X_u se obtiene cuando el valor de cada dígito sea el mayor posible: cuando todos los dígitos valgan b-1. ¿Cuál es este número? Se obtiene fácilmente sabiendo que

$\sum_{n=0}^{n-1} b^i = (b^n - 1)/(b - 1)$ (ver nota[1]):

$\sum_{i=0}^{n-1} (b-1)b^i = (b-1)\sum_{i=0}^{n-1} b^i = b^n - 1$. Por lo que el mayor número es el b^n-1.

Por último, sabemos que hay b^n combinaciones diferentes de n dígitos ya que cada dígito puede tomar b valores diferentes. Estos b^n diferentes vectores de n dígitos representan b^n números naturales diferentes en el sistema convencional en base b. Esos números no pueden ser otros que los b^n que hay entre el mínimo (0) y el máximo (b^n-1) representables, ambos incluidos. De todo esto concluimos lo siguiente.

⚐ 2.2

En el sistema convencional en base b, el **rango** de representación con n dígitos es:

$$0 \le X_u \le b^n - 1 \qquad \textbf{(EQ 3)}$$

✍ 2

2.2.6 Extensión de rango en un sistema convencional

Dado el vector de n dígitos que representa a un número natural en el sistema convencional en base b, ¿cuál es la representación del mismo número en el mismo sistema pero con n+1 dígitos? La respuesta se encuentra resolviendo el siguiente problema.

---

1. Sea $S = \sum_{n=0}^{n-1} b^i$. Entonces $S \times b + 1 = b^n + S$, ya que $1 = b^0$, y por lo tanto $S = (b^n - 1)/(b - 1)$.

Dado $X = x_{n-1}x_{n-2} \ \ldots \ x_2x_1x_0$ con $0 \le x_i \le b-1$, hay que encontrar $W = w_nw_{n-1}w_{n-2} \ \ldots \ w_2w_1w_0$ tal que se cumpla:

$$\sum_{i=0}^{n-1} x_ib^i = \sum_{i=0}^{n} w_ib^i \text{ con la restricción } 0 \le w_i \le b-1 .$$

La solución es trivial:

$w_i = x_i$ para i = 0, 1, ..., n-1 y
$w_n = 0$.

Esto ya lo sabíamos para el sistema decimal, pero es generalizable a los sistemas convencionales de cualquier base, como acabamos de demostrar.

2.3

> Dada la representación con n dígitos de un número natural en un sistema convencional en base b (y por tanto en decimal, binario o hexadecimal), para obtener la representación de ese mismo número en ese mismo sistema (con la misma base) pero con n+1 dígitos solamente hay que añadir un dígito con valor 0 a la izquierda de la representación original. Este proceso de **extensión de rango** se puede aplicar repetidas veces hasta obtener la representación con los dígitos deseados.

Ejemplo 3

La representación del número binario 10110 con 8 bits es 00010110. Otro ejemplo, la representación del número hexadecimal 0x2A con cuatro dígitos es 0x002A. 3

2.3 Cambios de base entre sistemas convencionales

2.3.1 De base b a base 10 (de X a Xu)

A partir de la representación de un número en base b, $X = x_{n-1}x_{n-2} \ \ldots \ x_2x_1x_0$ con $x_i \in \{0,..., b-1\}$, debemos encontrar la representación del mismo número en base 10. Como ya hemos dicho, al evaluar la EQ (2) expresando los dígitos y b en decimal y multiplicando y sumando en decimal obtenemos el valor del número representado en decimal, que es lo que buscamos.

Ejemplo 4 De binario a decimal

Vamos a obtener la representación en decimal del número binario 10110111. Para ello evaluamos la EQ (2) en decimal. Representar en decimal un dígito binario es trivial, la representación del 0 y del 1 en binario y en decimal es la misma. Expandiendo la fórmula y sustituyendo por los valores del ejemplo queda:

$$X_u = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 =$$

$$=1 \times 128 + 0 \times 64 + 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 =$$

$$=128 + 32 + 16 + 4 + 2 + 1 = 183$$

Para pasar de binario a decimal hay que sumar las potencias de dos, 2^i, para las posiciones i del vector en las que el dígito vale 1. Para hacerlo con rapidez es conveniente saber de memoria las potencias de 2. La siguiente tabla muestra las primeras de ellas. ✍ 4

Tabla 2.1 Primeras potencias de dos

i	12	11	10	9	8	7	6	5	4	3	2	1	0
2^i	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Ejemplo 5 De hexadecimal a decimal

A30E representa en hexadecimal al número natural que en decimal se representa por 41742. Esto es, si X=0xA30E, los dígitos son x_3=A, x_2=3, x_1=0 y x_0=E. Ahora hay que representar estos dígitos en decimal y realizar las operaciones de la EQ (2) en decimal, que es como las sabemos hacer:

$$X_u = 10 \times 16^3 + 3 \times 16^2 + 0 \times 16^1 + 14 \times 16^0 =$$

$$=10 \times 4096 + 3 \times 256 + 14 = 40960 + 768 + 14 = 41742$$

✍ 5

2.3.2 De base 10 a base b (de Xu a X)

Dado un número representado en base 10 tenemos que encontrar su representación en otra base b. Este problema es equivalente a encontrar el valor de las incógnitas x_{n-1}, x_{n-2},..., x_2, x_1 y x_0, tales que cumplan la siguiente ecuación (X_u y b son conocidos, no son ninguna incógnita):

$$X_u = x_{n-1}b^{n-1} + x_{n-2}\, b^{n-2} + ... + x_2\, b^2 + x_1\, b + x_0 \qquad \textbf{(EQ 4)}$$

En general, una sola ecuación con n incógnitas, para $n>1$, tiene infinitas soluciones. Una de estas soluciones es, por ejemplo, $x_0 = X_u$ y $x_{n-1} = x_{n-2} = ... = x_2 = x_1 = 0$. Pero para valores de X_u mayores o iguales que b esta solución no nos satisface, ya que en estos casos X_u no puede ser un dígito. Así que, para que las soluciones de la EQ (4) sean dígitos en el sistema en base b, las soluciones tienen que cumplir una restricción que no habíamos enunciado todavía:

$$0 \le x_i < b \qquad \text{para} \qquad x_i = 0, \cdots, n-1 \qquad \textbf{(EQ 5)}$$

Para encontrar los valores de x_i que cumplen con EQ (4) y con la restricción EQ (5), sacamos el factor común b de todos los sumandos de EQ (4) excepto de x_0 (ya que b no multiplica a x_0).

$$X_u = (x_{n-1}b^{n-2} + x_{n-2}b^{n-3} + x_{n-3}b^{n-4} + \cdots + x_2 b^1 + x_1)b + x_0 \qquad \textbf{(EQ 6)}$$

Esto es, dividiendo X_u entre b (esta división la sabemos hacer en decimal, ya que tenemos expresado en decimal X_u) obtenemos que el cociente, que llamamos C_1, es:

$$C_1 = x_{n-1}b^{n-2} + x_{n-2}b^{n-3} + x_{n-3}b^{n-4} + \cdots + x_2b^1 + x_1 \qquad \textbf{(EQ 7)}$$

El resto de la división es el dígito x_0 de la representación del número X_u en base b, que cumple con la restricción, es menor que b, ya que es el resto de una división por b. Ya tenemos un dígito.

¿Cómo encontrar los dígitos que faltan? Si nos fijamos en la EQ (7) vemos que tiene la misma forma que la EQ (4) pero con un término menos. Si repetimos lo hecho anteriormente, sacando el factor común b de todos los sumandos de la EQ (7) excepto de x_1, tenemos,

$$C_1 = (x_{n-1}b^{n-3} + x_{n-2}b^{n-4} + \cdots + x_3b^1 + x_2)b + x_1$$

Si ahora dividimos C_1 (que tenemos expresado en decimal) entre b, el resto de esta nueva división es el dígito x_1 que estamos buscando. El cociente de esta última división, que denotamos por C_2, es:

$$C_2 = x_{n-1}b^{n-3} + x_{n-2}b^{n-4} + \cdots + x_3b^1 + x_2$$

Esta expresión es como la EQ (7) pero con un sumando menos. Así que podemos repetir el proceso y seguir dividiendo cada cociente resultante por la base hasta que el cociente resulte ser cero. Los restos de estas divisiones son los dígitos que estamos buscando.

2.4 **Procedimiento para pasar de decimal a base b**. Se divide (división entera en decimal) el número por la base b. El resto de esta división es el valor del dígito de menor peso de la representación del número en base b. Se repite el proceso de división por b tomando como dividendo el cociente de la división anterior hasta que el cociente de la última división sea cero. Los restos de esta secuencia de divisiones son los valores de dígitos que buscamos, ordenados de menor a mayor peso.[a]

a. Si se sigue dividiendo el cociente con valor 0 entre b se obtiene cociente cero y resto cero. Esto no es problema ya que se pueden poner todos los dígitos que se quieran con valor 0 en las posiciones de más peso del vector de dígitos y la representación del número sigue siendo la misma

Ejemplo 6 De decimal a binario

Vamos a obtener la representación en binario del número natural 426: dado X_u = 426, encontrar el vector de bits X que lo represente en binario. La figura 2.1 muestra las sucesivas divisiones entre 2, para obtener los bits. El resultado es X = 110101010. 6

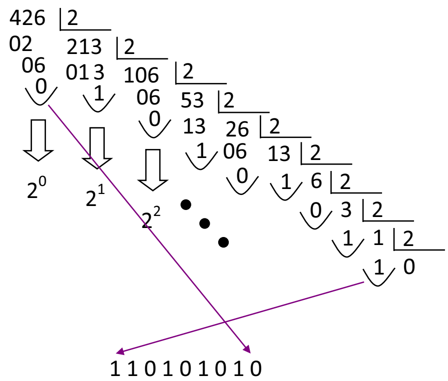

Fig. 2.1 Divisiones sucesivas para encontrar los bits que representan al 426 en binario.

Ejemplo 7 De decimal a hexadecimal

Vamos a representar en hexadecimal el número que acabamos de representar en binario, el 426. La figura 2.2 muestra las sucesivas divisiones por la base16 para obtener los restos, que son los dígitos hexadecimales. El resto con valor decimal 10 se codifica con el símbolo A para formar el dígito hexadecimal. El resultado es X = 0x1AA. ✍ 7

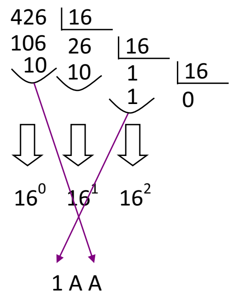

Fig. 2.2 Divisiones sucesivas para encontrar los dígitos de la representación al 426 en hexadecimal.

2.3.3 De hexadecimal a binario y viceversa

Dado que 16 es igual a 2^4 resulta muy sencillo pasar de la representación de un número natural en binario a su representación en hexadecimal y viceversa.

De binario a hexadecimal.

Dada la representación de un número natural en binario, $X = x_{n-1}x_{n-2} \ \dots \ x_2x_1x_0$ con $x_i \in \{0,1\}$, debemos encontrar la representación en hexadecimal de ese mismo número. Esto es, debemos encontrar el vector de dígitos hexadecimales $H = h_{k-1}h_{k-2} \ \dots \ h_2h_1h_0$ con $h_i \in \{0,...,15\}$ tal que $X_u = H_u$, que en función de los dígitos se expresa así:

$$\sum_{i=0}^{n-1} x_i 2^i = \sum_{i=0}^{k-1} h_i 16^i \qquad \textbf{(EQ 8)}$$

Supongamos inicialmente $n = 4k$ y, para que las expresiones resulten más sencillas, vamos a escribirlas para $n = 16$ y $k = 4$. El sumatorio de la izquierda de la igualdad una vez desarrollado, queda así:

$x_{15}2^{15} + x_{14}2^{14} + x_{13}\,2^{13} + x_{12}\,2^{12} +... + x_7 2^7 + x_6\,2^6 + x_5\,2^5 + x_4 2^4 + x_3 2^3 + x_2\,2^2 + x_1\,2 + x_0$

Agrupando sumandos de cuatro en cuatro se puede sacar el factor común de una potencia de 2^4, 16, distinta para cada grupo de 4 términos del sumatorio. El resultado es:

$$(x_{15}2^3 + x_{14}\,2^2 + x_{13}\,2 + x_{12})16^3 +... + (x_7 2^3 + x_6\,2^2 + x_5\,2 + x_4)16 + (x_3 2^3 + x_2\,2^2 + x_1\,2 + x_0)$$

Tenemos que encontrar los valores de los dígitos hexadecimales, h_i, tales que la expresión anterior sea igual que la expresión de la derecha de la igualdad de la EQ (8), que una vez desarrollado el sumatorio queda así: $h_3 16^3 + h_2 16^2 + h_1 16 + h_0$. Viendo así las dos expresiones, se deduce que la igualdad entre ellas se cumple si:

$h_0 = x_3 2^3 + x_2\,2^2 + x_1 2 + x_0$; $h_1 = x_7 2^3 + x_6\,2^2 + x_5 2 + x_4$,..., y $h_3 = x_{15}2^3 + x_{14}2^2 + x_{13}2 + x_{12}$

Los dígitos hexadecimales encontrados son válidos ya que su valor máximo, cuando los 4 bits valen 1, es 15. Esto es útil tanto para pasar de binario a hexadecimal como para pasar de hexadecimal a binario.

2.4 **Procedimiento para pasar de binario a hexadecimal**. Se agrupan de 4 en 4 los bits de la representación binaria, empezando por los bits de menor peso. Si al final el número de bits no es múltiplo de 4, se completa la representación binaria con ceros a la izquierda hasta que lo sea (extensión de rango).

Cada grupo de 4 bits representa en binario uno de los dígitos de la representación hexadecimal del número. Los 4 bits de menor peso representan al dígito hexadecimal de menor peso, los 4 siguientes hacia la izquierda representan al dígito de peso 16, y así hasta llegar a los 4 bits de más peso, que representan al dígito hexadecimal de más peso.

Ahora hay que pasar individualmente de la representación binaria de cada dígito hexadecimal a su símbolo en hexadecimal. Esto es, hay que pasar números de 4 bits de binario a decimal y luego cada número decimal mayor que 9 (de 10 a 15) representarlo con la letra que le corresponda como dígito hexadecimal (A, B,...,F). Para los números menores o iguales que 9, el símbolo del dígito decimal coincide con el hexadecimal (0, 1, 2,..., 9).

Ejemplo 8   De binario a hexadecimal

Dado el número representado en binario mediante el siguiente vector con n=16 bits, X=1001001110111010, vamos a obtener su representación en hexadecimal sin efectuar divisiones (encontramos H tal que $H_u = X_u$). Agrupando de 4 en 4 los bits de X obtenemos el valor de los k = n/ 4 = 4 dígitos hexadecimales que representan al número en base 16. La figura 2.3 muestra el proceso. El resultado es H = 93BA. ✍ 8

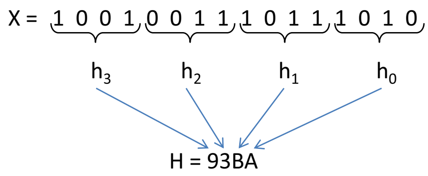

Fig. 2.3 Ejemplo de paso directo de binario a hexadecimal

De hexadecimal a binario.

2.4 **Procedimiento para pasar de hexadecimal a binario** Es el mismo que de binario a hexadecimal pero en sentido contrario. Se pasa cada dígito hexadecimal a binario y se representa cada uno con 4 bits. Aunque un dígito hexadecimal concreto (por ejemplo el 3) se pueda representar con menos de 4 bits (con 2 bits: 11) es necesario hacerlo con 4 bits (0011), de lo contrario el resultado final será incorrecto. Por último, se concatenan los grupos de 4 bits en el mismo orden que los dígitos hexadecimales que representan. Si la representación resultante tiene ceros como bits de más peso, estos se pueden quitar y el valor del número no se modifica.

Ejemplo 9 De hexadecimal a binario

La figura 2.4 muestra el paso de la representación hexadecimal, de 3E2B, a la binaria. Si no es necesario representar el número en binario con 16 bits, se pueden quitar los dos de más peso que valen 0. El resultado es: 11111000101011. Aquí podemos ver que si el dígito hexadecimal con valor 2 se hubiera representado con dos bits, 10, el resultado, 111110101011, sería incorrecto. ✍ 9

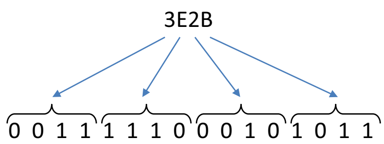

Fig. 2.4 Ejemplo de paso directo de hexadecimal a binario.

Utilidad del hexadecimal.

La facilidad de paso entre las representaciones binario y hexadecimal, en ambos sentidos, sin tener que realizar operaciones aritméticas (la codificación binaria de los 16 dígitos hexadecimales es trivial) nos es útil para:

- Representar un vector de bits de forma compacta. Nuestro computador trabaja con datos e instrucciones codificadas en 16 bits. Es más cómodo y más fácil que no cometamos errores si decimos, por ejemplo, que el resultado de la suma es 0x3A6D que si decimos que es 0011101001101101.
- Para pasar un número de muchos dígitos de decimal a binario, si lo pasamos primero a hexadecimal dividimos por cuatro el número de divisiones a hacer. Para pasar de binario a decimal también se puede pasar primero a hexadecimal. ✍ 10

✍ Ejercicios

1. **Sistema convencional** (⚐ 2.1, nivel B1, sección 2.2.3)

   Dado el vector de dígitos $Q = q_7q_6 \ldots q_2q_1q_0$ con $q_i \in \{0,..., k\text{-}1\}$ escribid la expresión del valor del número natural Q_u que representa ese vector en el sistema convencional en base k.

2. **Rango en el sistema convencional** (⚐ 2.2, nivel B1, sección 2.2.5)

   Expresar el rango de los números naturales que se pueden representar con n dígitos en el sistema de numeración convencional en base b, para los siguientes casos:

   a) n = 6, b = 10; b) n = 10, b = 2; c) n = 3, b = 16; d) n = 4, b = 8.

3. **Extensión de rango en el sistema convencional** (⚐ 2.3, nivel B1, sección 2.2.6)

   Para cada una de las siguientes representaciones de números naturales en el sistema convencional en base b, obtened su representación en la misma base pero con 3 dígitos más: a) 346 (b = 10); b) 01010 (b = 2); c) 0F1A7 (b = 16); d) 346 (b = 16).

4. **De binario a decimal** (⚐ 2.4, nivel B2, sección 2.3.1).

   Expresad en decimal el valor de los siguientes números que se dan representados en binario, X = 1011, Y = 11011100, Z = 0011000111110101.

5. **De hexadecimal a decimal** (⚐ 2.4, nivel B2, sección 2.3.1)

   Para cada una de las siguientes representaciones de números naturales en el sistema hexadecimal, $H = h_{k-1}h_{k-2} \ldots h_2h_1h_0$ con $h_i \in \{0,...,15\}$, indicad el valor del número natural representado, H_u (usando para ello el sistema decimal, como es usual): a) 3A0F, b) 07CB, c) 346, d) 101.

6. **De decimal a binario**(⚐ 2.4, nivel B2, sección 2.3.2)

   Obtener la representación en binario de los siguientes números naturales que representamos en decimal como: a) 135, b) 1025, c) 99, d) 078.

7. **De decimal a hexadecimal** (⚐ 2.4, nivel B2, sección 2.3.2)

   Obtener la representación en hexadecimal de los siguientes números naturales que representamos en decimal como: a) 3245, b) 221, c) 999, d) 1026

8. **De binario a hexadecimal**(⚐ 2.4, nivel B2, sección 2.3.3)

   Obtener la representación en hexadecimal de los siguientes números naturales que representamos en binario como: a) 100110101, b) 11111011, c) 11110, d) 1011001011101

9. **De hexadecimal a binario** (⚐ 2.4, nivel B2, sección 2.3.2)

   Obtener la representación en binario de los siguientes números naturales que representamos en hexadecimal como: a) DE04, b) 110, c) 00F, d) 8217AB

10. **De decimal a binario y a hexadecimal** (⚐ 2.4, nivel B2, sección 2.3.2)

Obtened el vector X de 8 bits que representa en binario cada uno de los siguientes números naturales (expresad X también en hexadecimal). Indicad los casos en que el número no pueda representarse en binario con 8 bits: X_u= 35, Y_u= 79, Z_u= 145, W_u= 284.