Universidad Nacional de Ingeniería
Facultad de Ciencias

Introducción a la Ciencia de la Computación

Sistemas de Numeros

Prof: J. Solano
2011-I

Objetivos

Despues de estudiar este capitulo el estudiante sera capaz de:

- ❑ **Entender el concepto de sistemas de numeros.**
- ❑ **Distinguir entre sistemas de numeros posicionales y no-posicionales.**
- ❑ **Describir el sistema decimal, binario, hexadecimal y octal.**
- ❑ **Convertir un numero en binario, octal o hexadecimal a un numero en el sistema decimal.**
- ❑ **Convertir un numero en el sistema decimal a un numero en binario, octal y hexadecimal.**
- ❑ **Convertir un numero en binario a octal y vice versa.**
- ❑ **Convertir un numero en binario a hexadecimal y vice versa.**
- ❑ **Hallar el numero de digitos necesario en cada sistema para representar un valor particular.**

INTRODUCCION

Un **sistema de numeros** define como un numero puede ser representado usando distintos simbolos. Un numero puede ser representado diferentemente en diferentes sistemas. Por ejemplo, los numeros $(2A)_{16}$ y $(52)_8$ ambos se refieren a la misma cantidad, $(42)_{10}$, pero sus representaciones son diferentes.

Varios sistemas de numeros han sido usados en el pasado y pueden ser categorizados en dos grupos: sistemas **posicionales** y **no-posicionales**. Se discutiran principalmente sistemas posicionales de numeros, pero tambien se daran ejemplos de sistemas no-posicionales.

SISTEMAS POSICIONALES DE NUMEROS

En un **sistema posicional de numeros,** la posicion que un simbolo ocupa en el numero determina el valor que representa. En este sistema, un numero representado como:

$$\boxed{\pm (S_{k-1} \ldots S_2\, S_1\, S_0 .\, S_{-1}\, S_{-2} \ldots S_{-l})_b}$$

tiene el valor de:

$$n = \pm \; S_{k-1} \times b^{k-1} + \ldots + S_1 \times b^1 + S_0 \times b^0 \; + \; S_{-1} \times b^{-1} + S_{-2} \times b^{-2} + \ldots + S_{-l} \times b^{-l}$$

En el que S es el conjunto de simbolos, b es la **base** (o **radix**).

El sistema decimal (base 10)

La palabra **decimal** es derivada de la raiz Latina **decem** (diez). En este sistema la **base b = 10** y usamos diez simbolos

$$\boxed{S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}}$$

Los simbolos en este sistema son frecuentemente referidos como **digitos decimales** o simplemente **digitos**.

ENTEROS

$$N = \pm \; S_{k-1} \times 10^{k-1} + S_{k-2} \times 10^{k-2} + \ldots + S_2 \times 10^2 + S_1 \times 10^1 + S_0 \times 10^0$$

	10^{k-1}	10^{k-2}	$\bullet\bullet\bullet$	10^2	10^1	10^0	Place values
$\pm$	S_{k-1}	S_{k-2}	$\bullet\bullet\bullet$	S_2	S_1	S_0	Number

$$N = \pm\, S_{k-1} \times 10^{k-1} + S_{k-2} \times 10^{k-2} + \bullet\bullet\bullet + S_2 \times 10^2 + S_1 \times 10^1 + S_0 \times 10^0 \quad \text{Values}$$

Ejemplo 1

Este ejemplo muestra los valores posicionales para el entero +224 en el sistema decimal.

			10^2		10^1		10^0	Place values
			2		2		4	Number
N	=	+	2×10^2	+	2×10^1	+	4×10^0	Values

Notar que el digito 2 en posicion 1 tiene el valor 20, pero el mismo digito en posicion 2 tiene el valor 200. Tambien notar que normalmente eliminamos el signo mas, pero esta implicito.

Ejemplo 2

Este ejemplo muestra los valores posicionales para el numero decimal -7508. Usamos 1, 10, 100 y 1000 en lugar de potencias de 10.

			1000		100		10		1		Place values
			7		5		0		8		Number
N	$=$	$-($	7×1000	$+$	5×100	$+$	0×10	$+$	8×1	$)$	**Values**

REALES

	Integral part		Fractional part
$R = \pm$	$S_{k-1} \times 10^{k-1} + \ldots + S_1 \times 10^1 + S_0 \times 10^0$	$+$	$S_{-1} \times 10^{-1} + \ldots + S_{-l} \times 10^{-l}$

Ejemplo 3

Aqui se muestran los valores posicionales para el numero real +24.13

	10^1		10^0		10^{-1}		10^{-2}	Place values
	2		4	•	1		3	Number
$R = +$	2×10	$+$	4×1	$+$	1×0.1	$+$	3×0.01	Values

El sistema binario (base 2)

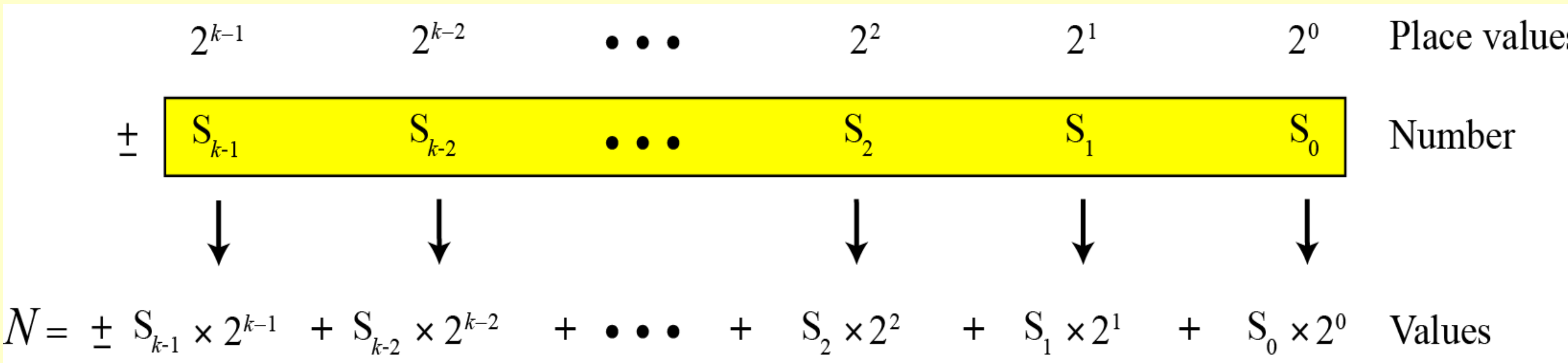
La palabra **binario** es derivada de la raiz Latina **bini** (dos). En este sistema la **base b = 2** y usamos solo dos simbolos

$$\boxed{S = \{0, 1\}}$$

Los simbolos en este sistema son frecuentemente referidos como **digitos binarios** o **bits** (digito binario).

ENTEROS

$$N = \pm\ S_{k-1} \times 2^{k-1} + S_{k-2} \times 2^{k-2} + \ldots + S_2 \times 2^2 + S_1 \times 2^1 + S_0 \times 2^0$$



Valores posicionales para un entero en sistema binario

Ejemplo 4

Este ejemplo muestra que el numero $(11001)_2$ en binario es el mismo que 25 en decimal. El subscrito 2 muestra que la base es 2

		2^4		2^3		2^2		2^1		2^0	Place values
		1		1		0		0		1	Number
N	=	1×2^4	+	1×2^3	+	0×2^2	+	0×2^1	+	1×2^0	Decimal

El numero decimal equivalente es N = 16 + 8 + 0 + 0 + 1 = 25

REALES

Integral part	•	Fractional part

$$R = \pm \; S_{k-1} \times 2^{k-1} + \ldots + S_1 \times 2^1 + S_0 \times 2^0 \; + \; S_{-1} \times 2^{-1} + \ldots + S_{-l} \times 2^{-l}$$

Ejemplo 5

Aqui se muestra que el numero $(101.11)_2$ en binario es equivalente al numero 5.75 en decimal

		2^2		2^1		2^0		2^{-1}		2^{-2}	Place values
		1		0		1	•	1		1	Number
R	=	1×2^2	+	0×2^1	+	1×2^0	+	1×2^{-1}	+	1×2^{-2}	Values

El sistema hexadecimal (base 16)

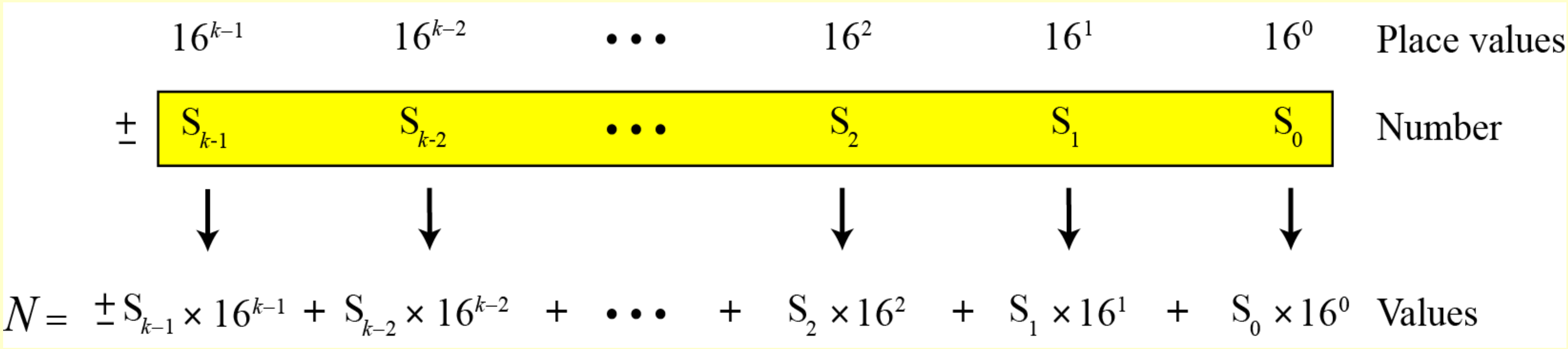
La palabra **hexadecimal** es derivada de la raiz griega **hex** (seis) y la raiz latina **decem** (diez). En este sistema la **base b = 16** y usamos dieciseis simbolos para representar un numero.
El conjunto de simbolos es

S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Notar que los simbolos A, B, C, D, E, F son equivalentes a 10, 11, 12, 13, 14 y 15 respectivamente. En este sistema los simbolos son referidos como **digitos hexadecimales**.

ENTEROS

$$N = \pm S_{k-1} \times 16^{k-1} + S_{k-2} \times 16^{k-2} + \ldots + S_2 \times 16^2 + S_1 \times 16^1 + S_0 \times 16^0$$



Valores posicionales para un entero en sistema hexadecimal

Ejemplo 6

Este ejemplo muestra que el numero $(2AE)_{16}$ en hexadecimal es eqivalente a 686 en decimal.

		16^2		16^1		16^0	Place values
		2		A		E	Number
N	=	2×16^2	+	10×16^1	+	14×16^0	Values

El numero decimal equivalente es $N = 512 + 160 + 14 = 686$

El sistema octal (base 8)

La palabra **octal** es derivada de la raiz latina **octo** (ocho). En este sistema la **base b = 8** y usamos ocho simbolos para representar un numero.

El conjunto de simbolos es

$$\boxed{S = \{0, 1, 2, 3, 4, 5, 6, 7\}}$$

ENTEROS

$$N = \pm S_{k-1} \times 8^{k-1} + S_{k-2} \times 8^{k-2} + \ldots + S_2 \times 8^2 + S_1 \times 8^1 + S_0 \times 8^0$$

	8^{k-1}	8^{k-2}	$\cdots$	8^2	8^1	8^0	Place values
$\pm$	S_{k-1}	S_{k-2}	$\cdots$	S_2	S_1	S_0	Number
	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$	

$$N = \pm S_{k-1} \times 8^{k-1} + S_{k-2} \times 8^{k-2} + \cdots + S_2 \times 8^2 + S_1 \times 8^1 + S_0 \times 8^0 \quad \text{Values}$$

Valores posicionales para un entero en sistema octal

Ejemplo 7

Este ejemplo muestra que el numero $(1256)_8$ en octal es eqivalente a 686 en decimal.

		8^3		8^2		8^1		8^0	Place values
		1		2		5		6	Number
N	=	1×8^3	+	2×8^2	+	5×8^1	+	6×8^0	Values

El numero decimal equivalente es N = 512 + 128 + 40 + 6 = 686

Resumen de los 4 sistemas posicionales

Table 2.1 Summary of the four positional number systems

System	*Base*	*Symbols*	*Examples*
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	2345.56
Binary	2	0, 1	$(1001.11)_2$
Octal	8	0, 1, 2, 3, 4, 5, 6, 7	$(156.23)_8$
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$(A2C.A1)_{16}$

La tabla muestra como el numero decimal del 0 al 15 es representado en los diferentes sistemas.

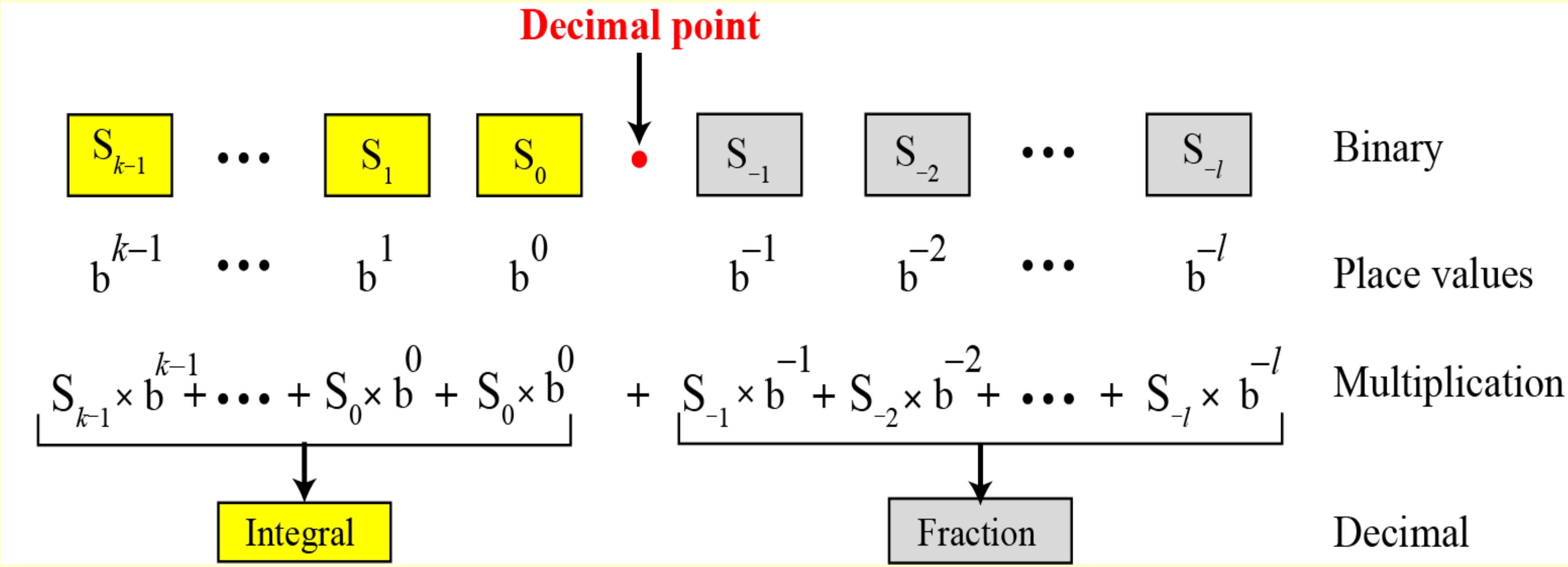
Table 2.2 Comparison of numbers in the four systems

Decimal	*Binary*	*Octal*	*Hexadecimal*
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversion

Necesitamos saber como convertir un numero en un sistema al numero equivalente en otro sistema. Desde que el sistema decimal es mas familiar que los otros sistemas, mostramos primero como convertir de cualquier base a decimal. Luego mostramos como convertir de decimal a cualquier base. Finalmente, mostraremos como convertir facilmente de binario a hexadecimal u octal y viceversa.

Conversion de cualquier base a decimal



Convirtiendo otras bases a decimal

Ejemplo 8

Este ejemplo muestra como convertir el numero binario $(110.11)_2$ a decimal. $(110.11)_2 = 6.75$

Binary	1		1		0	•	1		1
Place values	2^2		2^1		2^0		2^{-1}		2^{-2}
Partial results	4	+	2	+	0	+	0.5	+	0.25

Decimal: 6.75

Ejemplo 9

Este ejemplo muestra como convertir el numero hexadecimal $(1A.23)_{16}$ a decimal.

Hexadecimal	1		A	•	2		3
Place values	16^1		16^0		16^{-1}		16^{-2}
Partial result	16	+	10	+	0.125	+	0.012
Decimal: 26.137							

Notar que el resultado en la notacion decimal no es exacto, porque $3 \times 16^{-2} = 0.01171875$. Hemos redondeado este valor a tres digitos (0.012).

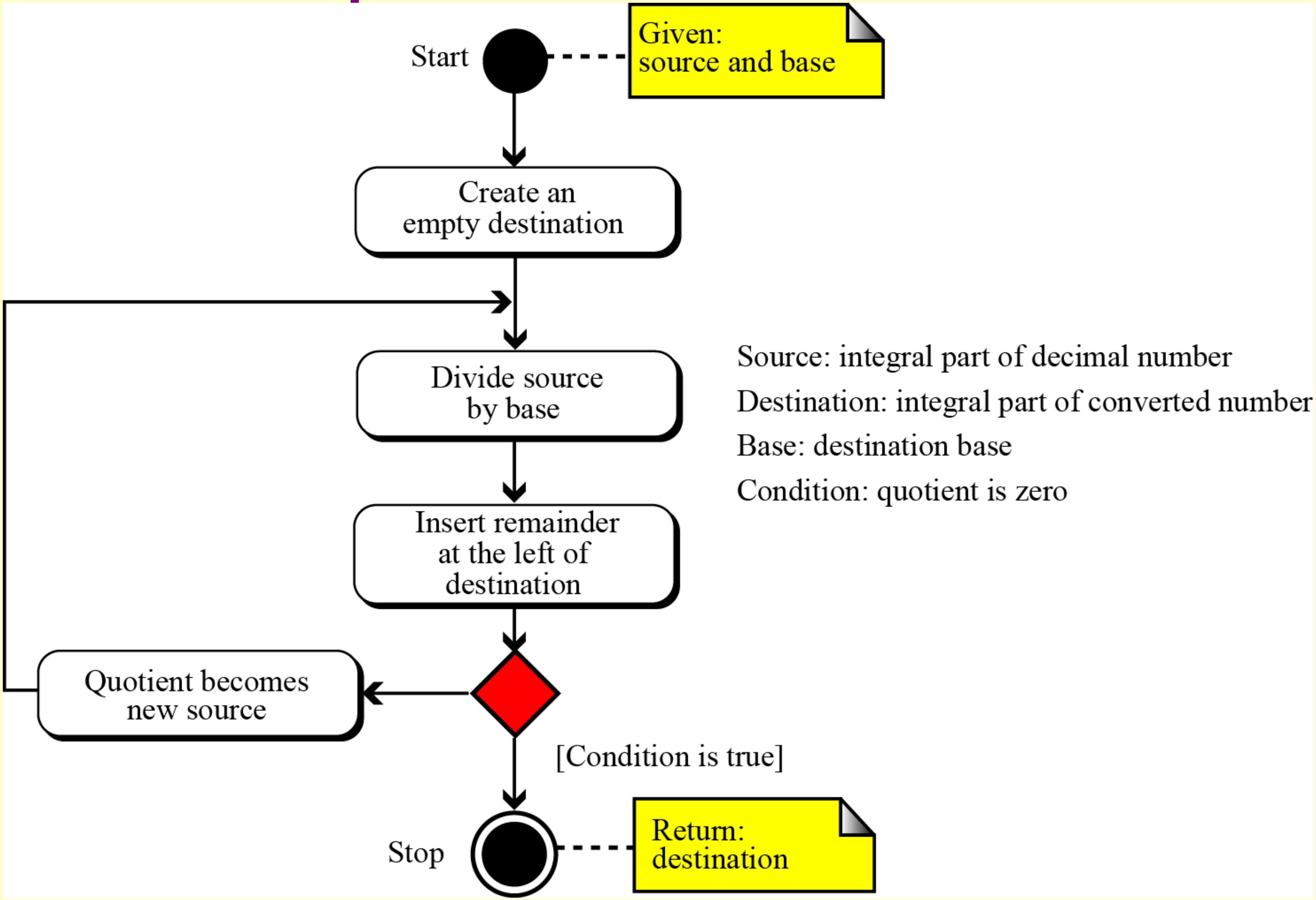
Ejemplo 10

Este ejemplo muestra como convertir $(23.17)_8$ a decimal.

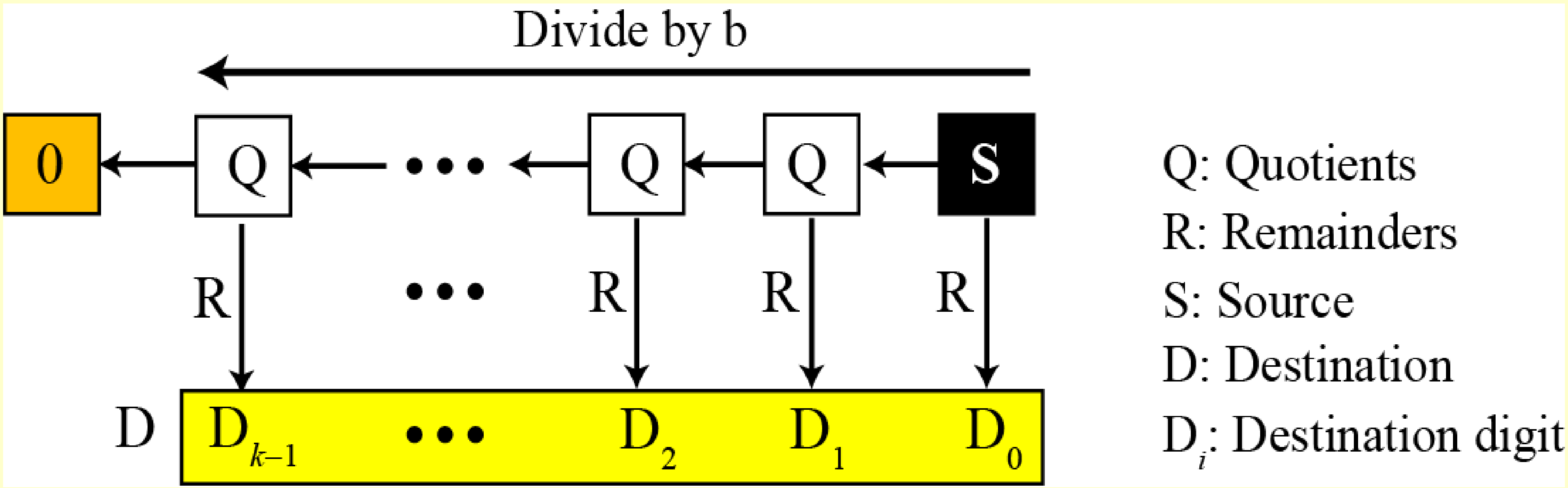
Octal	2		3	•	1		7
Place values	8^1		8^0		8^{-1}		8^{-2}
Partial result	16	+	3	+	0.125	+	0.109
Decimal: 19.234							

Esto significa que $(23.17)_8 \sim 19.234$. Otra vez hemos redondeado a $7 \times 8^{-2} = 0.109375$.

Decimal a cualquier base



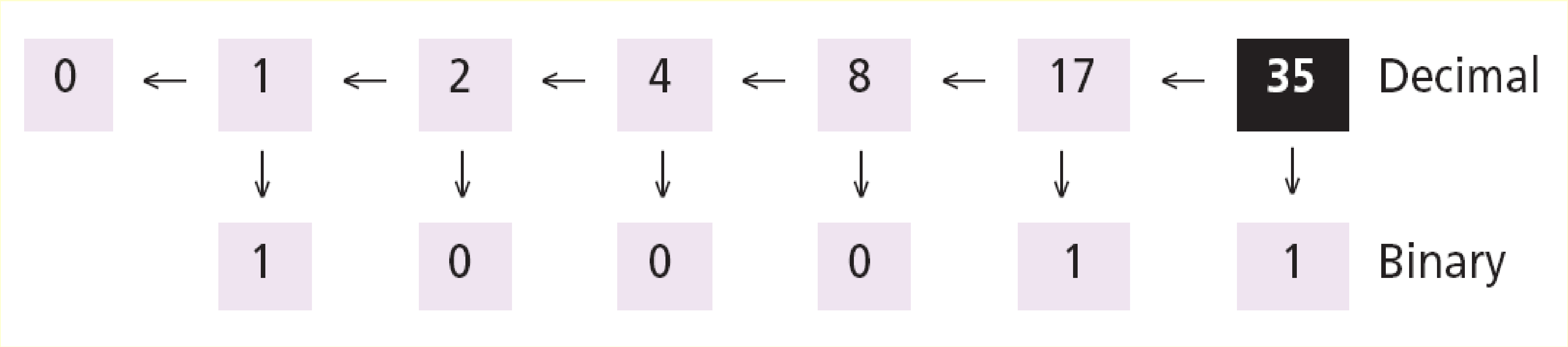
Convirtiendo otras bases a decimal (parte entera)



Convirtiendo la parte integral de un numero decimal a otras bases

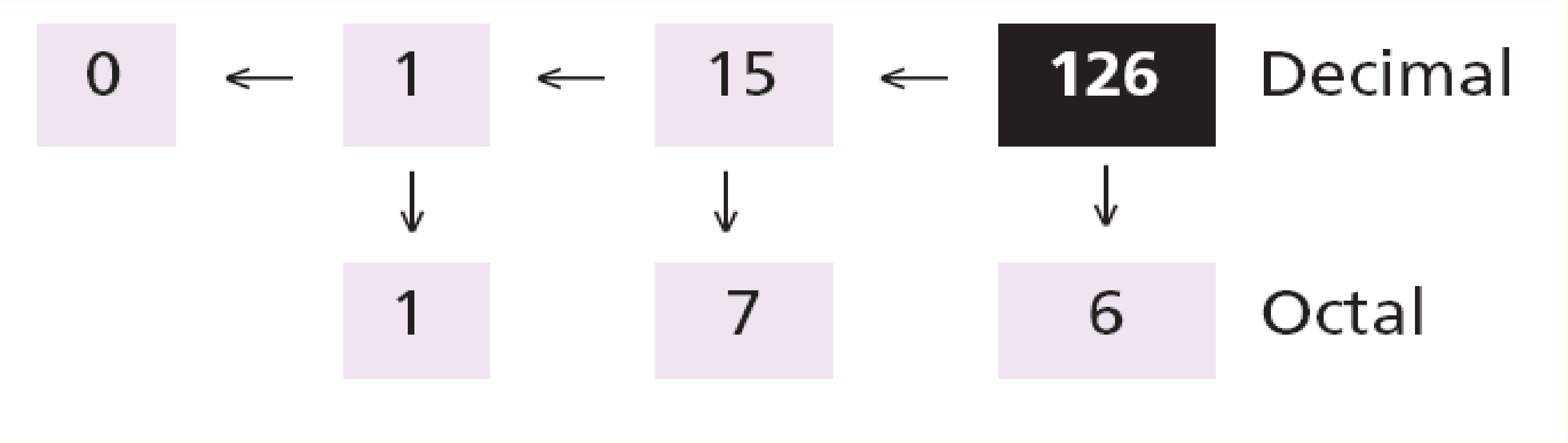
Ejemplo 11

Este ejemplo muestra como convertir 35 en decimal a binario. Enpezamos con el numero en decimal, nos movemos hacia la izquierda mientras que hallamos continuamente los cocientes y restos de la division por 2. El resultado es 35 = $(100011)_2$



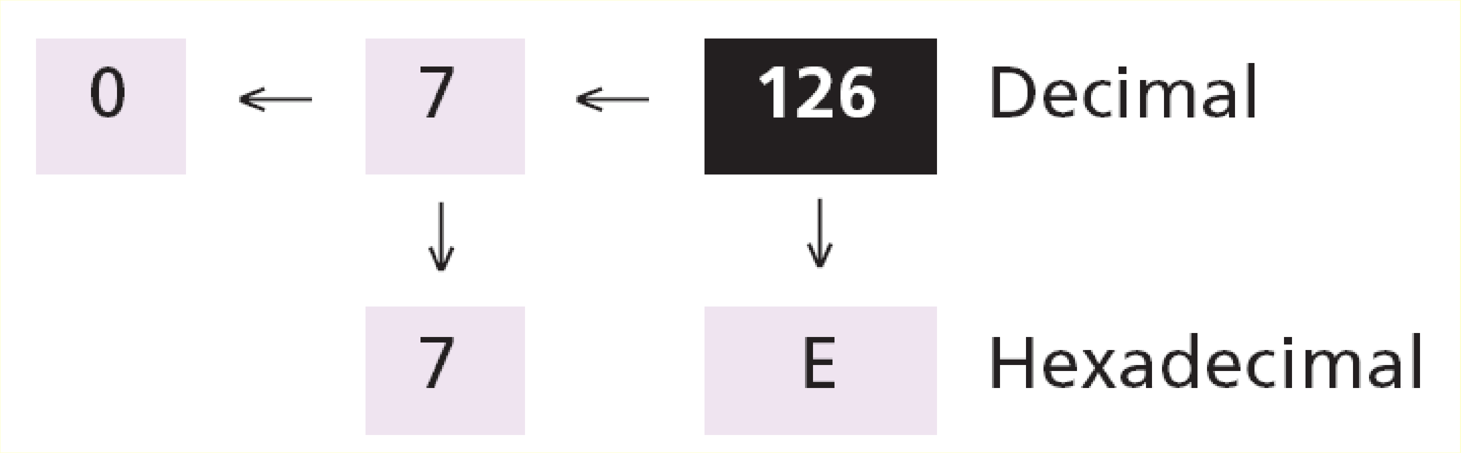
Ejemplo 12

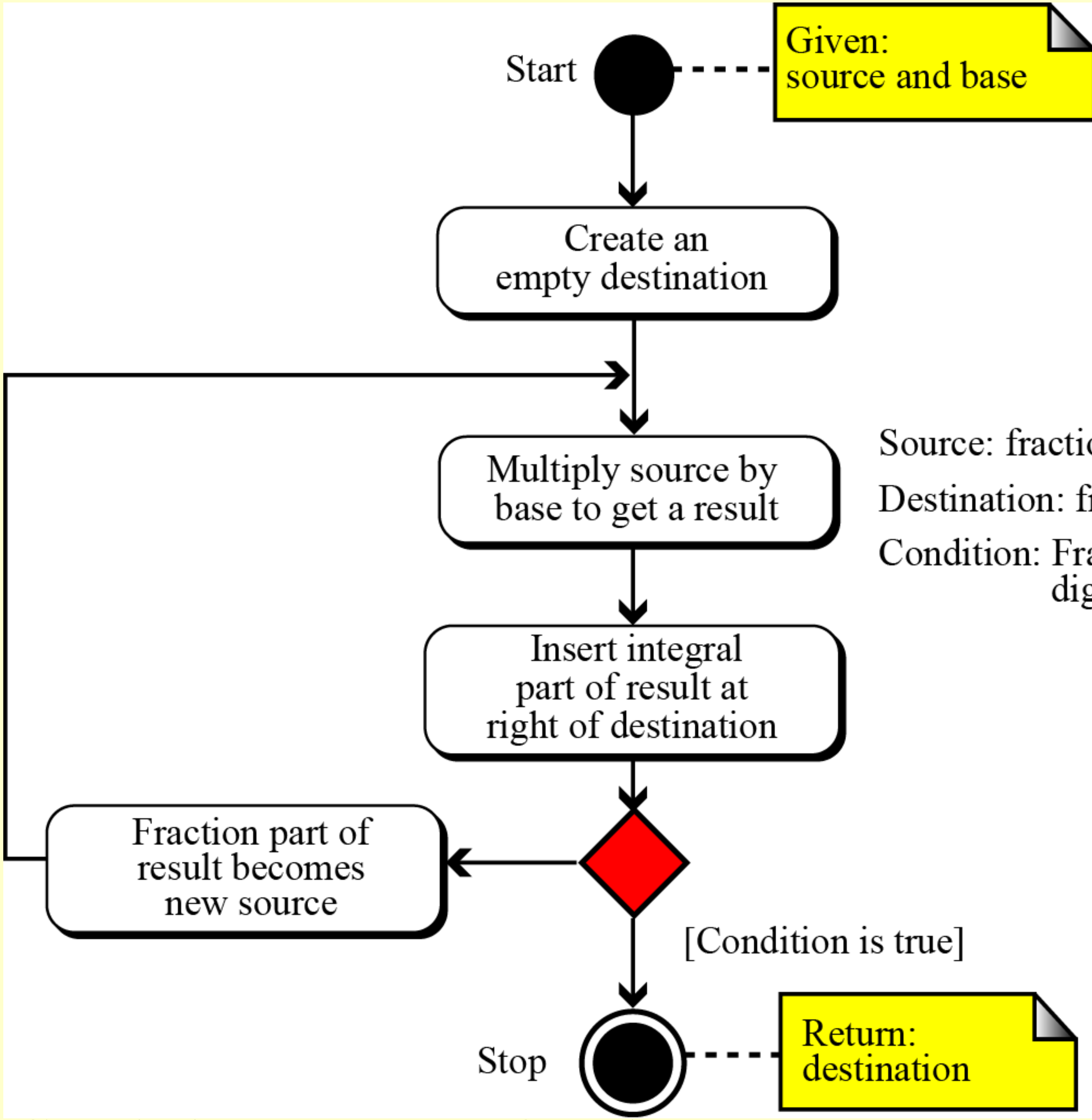
Este ejemplo muestra como convertir 126 en decimal a su equivalente en sistema octal. Nos movemos hacia la izquierda mientras que hallamos continuamente los cocientes y restos de la division por 8. El resultado es 126 = $(176)_8$



Ejemplo 13

Este ejemplo muestra como convertir 126 en decimal a su equivalente en sistema hexadecimal. Nos movemos hacia la izquierda mientras que hallamos continuamente los cocientes y restos de la division por 16. El resultado es 126 = $(7E)_{16}$



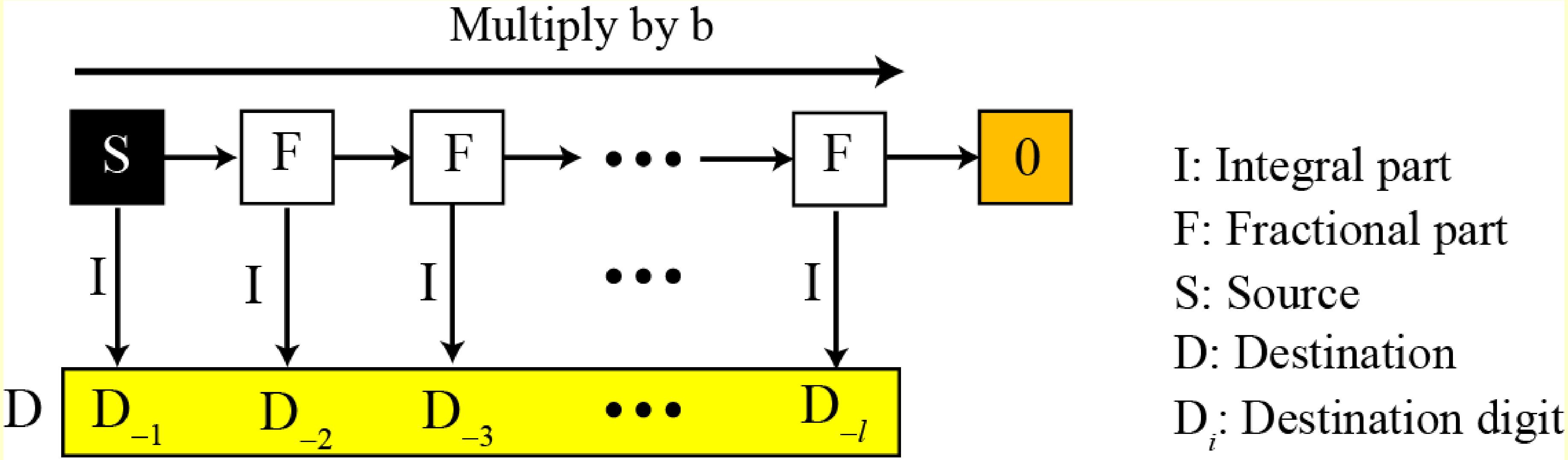


Source: fraction part of decimal number

Destination: fraction part of converted number

Condition: Fraction part is zero or destination digits are enough

Convirtiendo la parte fraccional de un numero decimal a otras bases

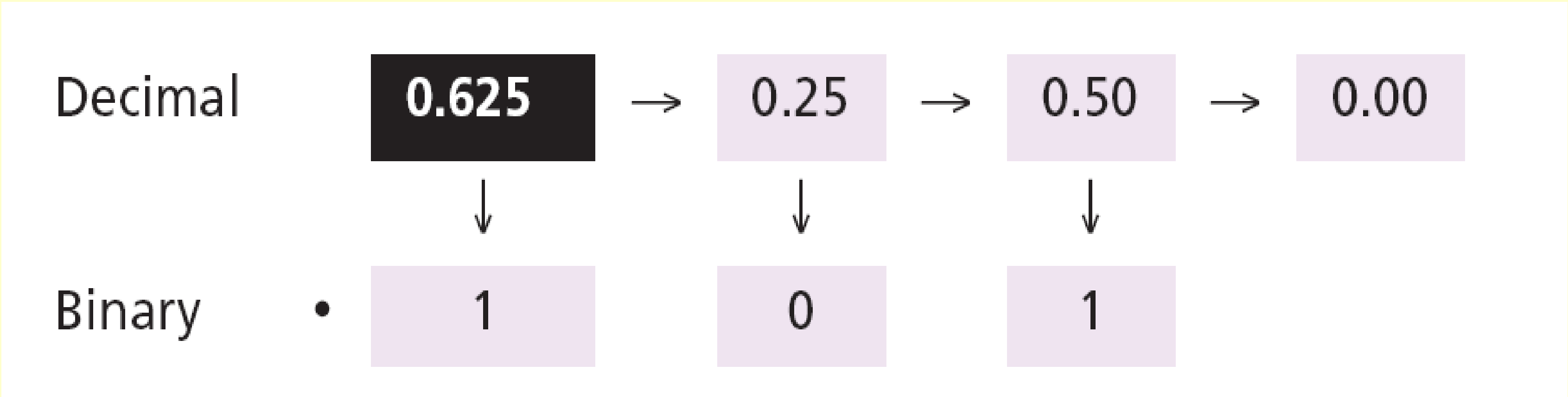


Note:
The fraction may never become zero.
Stop when enough digits have been created.

Convirtiendo la parte fraccional de un numero decimal a otras bases

Ejemplo 14

Convertir el numero decimal 0.625 a binario.



Desde que el numero $0.625 = (0.101)_2$ no tiene parte integral, el ejemplo muestra como la parte fraccional es calculada.

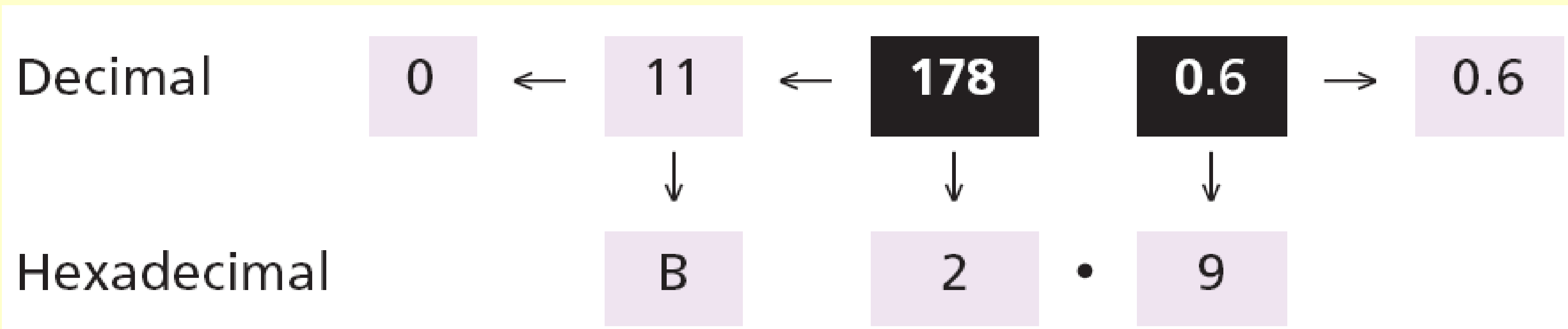
Ejemplo 15

Convertir el numero decimal 0.634 a octal usando un maximo de cuatro digitos. El resultado es $0.634 = (0.5044)_8$. Notar que multiplicamos por 8 (base octal).

Decimal		0.634	→	0.072	→	0.576	→	0.608	→	0.864
		↓		↓		↓		↓		
Octal	•	5		0		4		4		

Ejemplo 16

Convertir el numero decimal 178.6 a hexadecimal usando solo un digito a la derecha del punto decimal. El resultado es 178.6 = $(B2.9)_{16}$. Notar que dividimos o multiplicamos por 16 (base hexadecimal).



Ejemplo 17

Un metodo alternativo para convertir un pequenho entero decimal (usualmente menos de 256) a binario es quebrar el numero como la suma de numeros que son equivalentes a los valores posicionales binarios mostrados:

Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal equivalent	128	64	32	16	8	4	2	1

Decimal 165 =	128	+	0	+	32	+	0	+	0	+	4	+	0	+	1
Binary	1		0		1		0		0		1		0		1

Ejemplo 18

Un metodo similar puede ser usado para convertir una fraccion decimal a binario cuando el denominador es una potencia de dos:

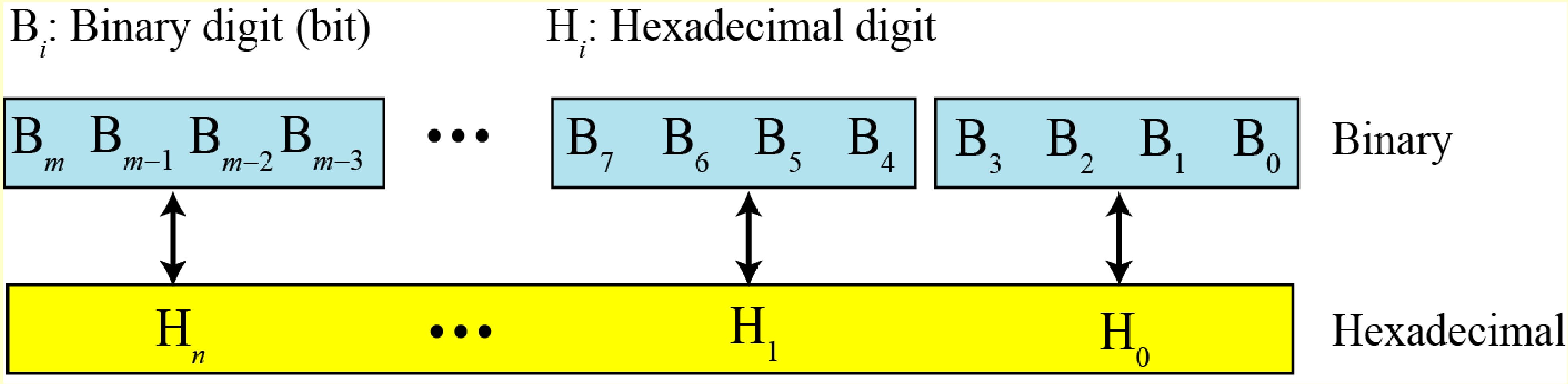
Place values	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Decimal equivalent	1/2	1/4	1/8	1/16	1/32	1/64	1/128

Decimal = 27/64	16/64	+	8/64	+	2/64	+	1/64
	1/4	+	1/8	+	1/32	+	1/64

Decimal 27/64 =	0	+	1/4	+	1/8	+	0	+	1/32	+	1/64
Binary	0		1		1		0		1		1

The answer is then $(0.011011)_2$

Conversion binario-hexadecimal



Conversion binario a hexadecimal y hexadecimal a binario

Ejemplo 19

Mostrar el equivalente hexadecimal del numero binario $(10011100010)_2$

Solucion

Primero arreglamos el numero binario en patrones de 4 bits:

100 1110 0010

Notar que el patron al extremo izquierdo puede tener de uno a cuatro bits. Usamos entonces el equivalente de cada patron mostrado en la tabla para cambiar el numero a hexadecimal: $(4E2)_{16}$.

La tabla muestra el equivalente hexadecimal de cada patron binario: $(100\ 1110\ 0010)_2 = (4E2)_{16}$.

Table 2.2 Comparison of numbers in the four systems

Decimal	*Binary*	*Octal*	*Hexadecimal*
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Ejemplo 20

Mostrar el binario equivalente a $(24C)_{16}$

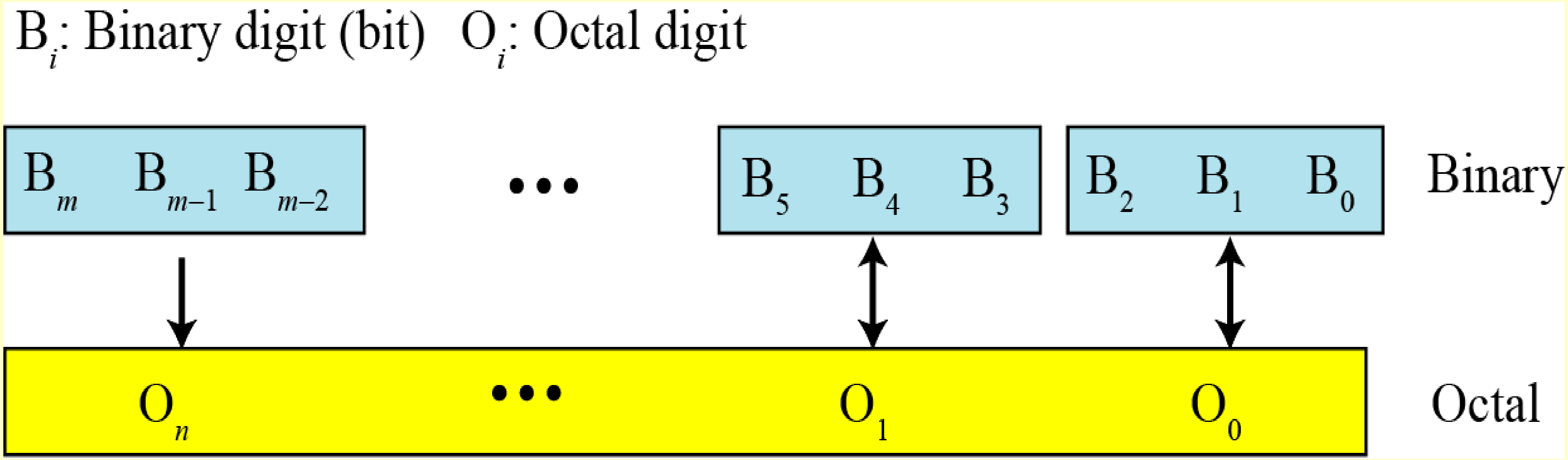
Solucion

Cada digito hexadecimal es convertido a patrones de 4 bits:

2 → 0010, 4 → 0100, y C → 1100

El resultado es $(001001001100)_2$.

Conversion binario-octal



Conversion binario a octal y octal a binario

Ejemplo 21

Mostrar el octal equivalente al numero binario $(101110010)_2$.

Solucion

Cada grupo de tres bits es traducido en un digito octal. El equivalente de cada grupo de 3-bits es mostrado en la tabla anterior

101 110 010

El resultado es $(562)_8$.

Ejemplo 22

Mostrar el binario equivalente al numero $(24)_8$.

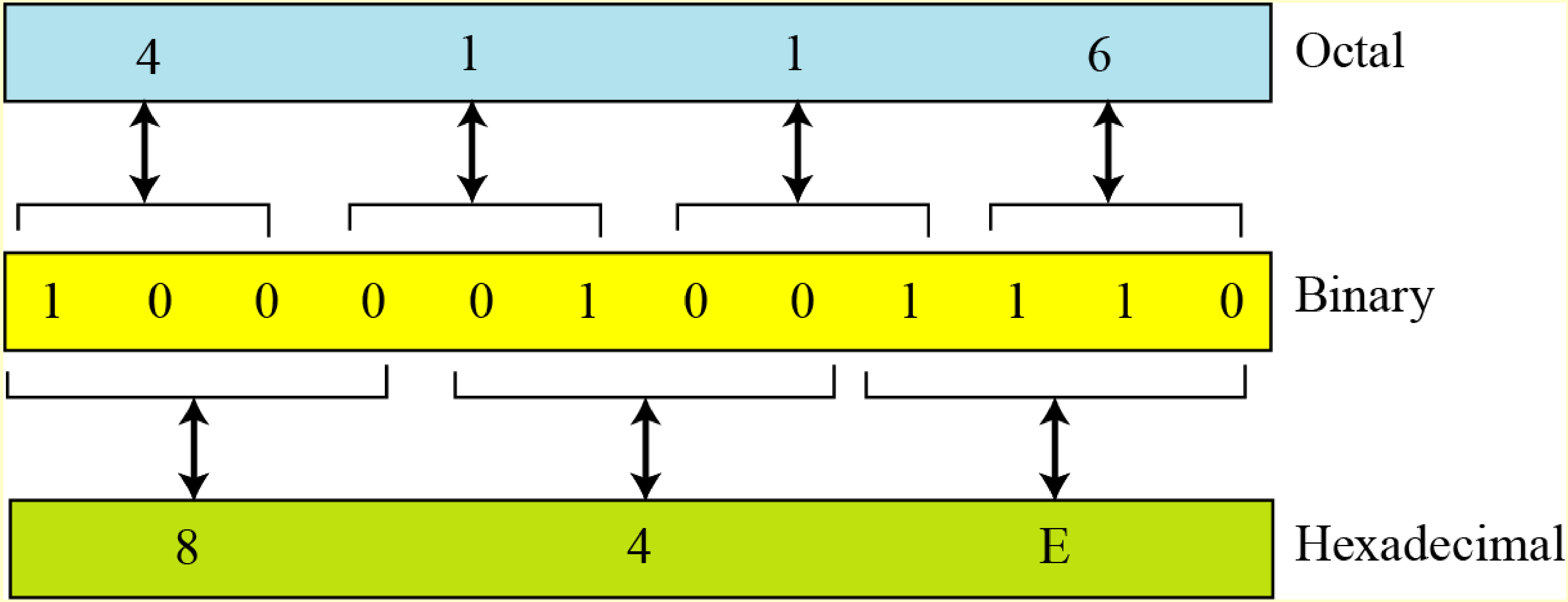
Solucion

Se escribe cada digito octal como su patron de bits equivalente, obteniendose

$$2 \rightarrow 010 \quad y \quad 4 \rightarrow 100$$

El resultado es $(010100)_2$.

Conversion octal-hexadecimal



Conversion octal a hexadecimal y hexadecimal a octal

Numero de digitos

Ejemplo 23

Hallar el numero minimo de digitos binarios requeridos para almacenar enteros decimales con un maximo de seis digitos.

Solucion

$k = 6$, $b_1 = 10$, y $b_2 = 2$. Entonces

$$\mathbf{x = \lceil k \times (\log b_1 / \log b_2) \rceil = \lceil 6 \times (1 / 0.30103) \rceil = 20.}$$

El mayor numero decimal de seis digitos es 999,999 y el mayor numero binario de 20-bits es 1,048,575. Notar que el mayor numero que puede ser representado por un numero de 19-bits es 524287, que es menor que 999,999. Definitivamente necesitamos veinte bits.

SISTEMAS NO-POSICIONALES DE NUMEROS

Aunque **sistemas no-posicionales de numeros** no son usados en computadores, damos una revision corta por comparacion con sistemas de numeros posicionales. Un sistema no-posicional de numeros aun usa un numero limitado de simbolos en el que cada simbolo tiene un valor. Sin embargo, la posicion que un simbolo ocupa en el numero normalmente no tiene relacion con su valor – el valor de cada simbolo es fijo. Para hallar el valor de un numero, anhadimos el valor de todos los simbolos presentes en la representacion.

En este sistema un numero es representado como:

$$\boxed{S_{k-1} \ldots S_2\, S_1\, S_0 \bullet S_{-1}\, S_{-2} \ldots S_{-l}}$$

y tiene los valores de:

	Integral part		*Fractional part*
$n = \pm$	$S_{k-1} + \ldots + S_1 + S_0$	$+$	$S_{-1} + S_{-2} + \ldots + S_{-l}$

Existen algunas excepciones a esta regla de adicion, como muestra el siguiente ejemplo.

Ejemplo 24

Numerales romanos son un buen ejemplo de sistemas no-posicionales de numeros. Este sistema de numeros tiene un conjunto de simbolos S = {I, V, X, L, C, D, M}.

Table 2.3 Values of symbols in the Roman number system

Symbol	*I*	*V*	*X*	*L*	*C*	*D*	*M*
Value	1	5	10	50	100	500	1000

Para hallar el valor de un numero, necesitamos anhadir el valor de los simbolos, sujetos a reglas especificas

Ejemplo 24 (continuacion)

Tabla con algunos numeros romanos y sus valores en el sistema decimal.

III	→	1 + 1 + 1	=	3
IV	→	5 − 1	=	4
VIII	→	5 + 1 + 1 + 1	=	8
XVIII	→	10 + 5 + 1 + 1 + 1	=	18
XIX	→	10 + (10 − 1)	=	19
LXXII	→	50 + 10 + 10 + 1 + 1	=	72
CI	→	100 + 1	=	101
MMVII	→	1000 + 1000 + 5 + 1 + 1	=	2007
MDC	→	1000 + 500 + 100	=	1600