Representación de la Información

(Parte 1)

David Alejandro González Márquez

Clase disponible en: `https://github.com/fokerman/computingSystemsCourse`

Hay 10 tipos de personas:

Las que conocen el código binario y las que no.

Hay 10b tipos de personas:

Las que conocen el código binario y las que no.

Introducción

Representación de la información hace referencia a la **forma** en que traducimos símbolos a entidades.

Por ejemplo:

- Una mano como ✌ es el símbolo usado para representar la cantidad de dos.
- Pero también puede significar *Paz*.
- Incluso, si rotamos la mano como 🤘 su significado puede cambiar a ser el símbolo de *tijeras*.

Es decir,

Estamos dando un **significado** a un **símbolo**.

Símbolos

Un número es una expresión de una cantidad, que representa una magnitud.

Por ejemplo,

- “9” es el símbolo usado para representar **nueve** unidades en el sistema decimal.
- “6” es el símbolo usado para representar **seis** unidades en el sistema decimal.
- …

El sistema decimal es un **sistema de numeración posicional** en base 10,
donde el valor de cada símbolo depende de su posición.

Ahora vamos conocer otros sistemas posicionales utilizando diferentes bases.

Sistemas de numeración

Existen distintos sistemas de numeración **posicionales**, cada uno depende de su **base**, es decir, la cantidad de símbolos distintos que tiene el sistema.

- Sistema decimal (base 10)

  0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- Sistema binario (base 2)

  0, 1

- Sistema octal (base 8)

  0, 1, 2, 3, 4, 5, 6, 7

- Sistema hexadecimal (base 16)

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

Sistemas de numeración

Magnitud	Decimal (10)	Binario (2)	Octal (8)	Hexadecimal (16)
0	0	0	0	0
1	1	1	1	1
2	2	10	2	2
3	3	11	3	3
4	4	100	4	4
5	5	101	5	5
6	6	110	6	6
7	7	111	7	7
8	8	1000	10	8
9	9	1001	11	9
10	10	1010	12	A
11	11	1011	13	B
12	12	1100	14	C
13	13	1101	15	D
14	14	1110	16	E
15	15	1111	17	F
16	16	10000	20	10
17	17	10001	21	11
18	18	10010	22	12
19	19	10011	23	13
20	20	10100	24	14
21	21	10101	25	15

Notación

Indicación de bases

Sistema decimal (base 10)

Ejemplo: | 248_{10} | 248 |

Sistema binario (base 2)

Ejemplo: | 101001_{2} | 101001b |

Sistema octal (base 8)

Ejemplo: | 755_{8} | 755o |

Sistema hexadecimal (base 16)

Ejemplo: | $F8AA_{16}$ | F8AAh | 0xF8AA |

Cambio de base

Para realizar operaciones entre distintas magnitudes, estas deben estar representadas en la misma base. Dependiendo entre qué bases se busque convertir, se pueden usar distintos métodos.

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \mapsto$$

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \mapsto$$

$$\begin{array}{r|l} 367 & 2 \\ \cline{2-2} 16 & 183 \\ 07 & \\ 1 & \end{array}$$

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \mapsto$$

$$\begin{array}{r|l} 367 & 2 \\ \cline{2-2} 16 & 183 \\ \cline{1-1} 07 & \\ 1 & \end{array} \qquad \begin{array}{r|l} 183 & 2 \\ \cline{2-2} 03 & 91 \\ \cline{1-1} 1 & \end{array}$$

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \mapsto$$

$$\begin{array}{rrrrr} 367 & 2 & & & \\ \cline{2-2} 16 & 183 & 2 & & \\ \cline{3-3} 07 & 03 & 91 & 2 & \\ \cline{4-4} 1 & 1 & 11 & 45 & \\ & & 1 & & \end{array}$$

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \mapsto$$

$$\begin{array}{rrrrrr|l}
367 & & & & & & 2 \\
16 & 183 & & & & & 2 \\
07 & 03 & 91 & & & & 2 \\
1 & 1 & 11 & 45 & & & 2 \\
 & & 1 & 05 & 22 & & \\
 & & & 1 & & &
\end{array}$$

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \mapsto$$

$$\begin{array}{rrrrrrr|l}
367 & 183 & 91 & 45 & 22 & 11 & 5 & 2 \\
16 & 03 & 11 & 05 & 02 & 01 & 1 & \\
07 & 01 & 01 & 01 & 00 & 1 & & \\
1 & 1 & 1 & 1 & 0 & & &
\end{array}$$

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \longmapsto$$

$$\begin{array}{rrrrrrrrr}
367 & \big\lfloor\, 2 & & & & & & & \\
16 & 183 & \big\lfloor\, 2 & & & & & & \\
07 & 03 & 91 & \big\lfloor\, 2 & & & & & \\
1 & 1 & 11 & 45 & \big\lfloor\, 2 & & & & \\
 & & 1 & 05 & 22 & \big\lfloor\, 2 & & & \\
 & & & 1 & 02 & 11 & \big\lfloor\, 2 & & \\
 & & & & 0 & 1 & 5 & &
\end{array}$$

Convertir desde decimal a binario

Método de las divisiones sucesivas

$$367_{10} \mapsto$$

$$
\begin{array}{rrrrrrrrr}
367 & 2 & & & & & & & \\
16 & 183 & 2 & & & & & & \\
07 & 03 & 91 & 2 & & & & & \\
1 & 1 & 11 & 45 & 2 & & & & \\
 & & 1 & 05 & 22 & 2 & & & \\
 & & & 1 & 02 & 11 & 2 & & \\
 & & & & 0 & 1 & 5 & 2 & \\
 & & & & & & 1 & 2 &
\end{array}
$$

Convertir desde decimal a binario

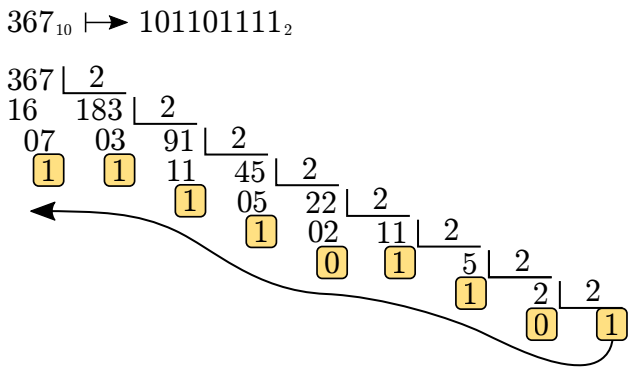
Método de las divisiones sucesivas

$$367_{10} \mapsto$$

$$
\begin{array}{rrrrrrrrrr}
367 & \big\lfloor\, 2 & & & & & & & & \\
16 & 183 & \big\lfloor\, 2 & & & & & & & \\
07 & 03 & 91 & \big\lfloor\, 2 & & & & & & \\
1 & 1 & 11 & 45 & \big\lfloor\, 2 & & & & & \\
 & & 1 & 05 & 22 & \big\lfloor\, 2 & & & & \\
 & & & 1 & 02 & 11 & \big\lfloor\, 2 & & & \\
 & & & & 0 & 1 & 5 & \big\lfloor\, 2 & & \\
 & & & & & & 1 & 2 & \big\lfloor\, 2 & \\
 & & & & & & & 0 & 1 &
\end{array}
$$

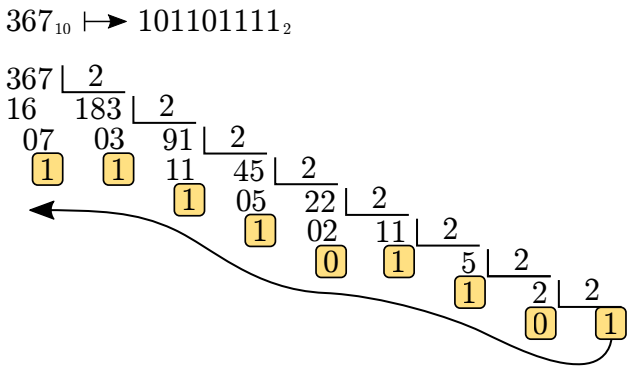
Convertir desde decimal a binario

Método de las divisiones sucesivas



Convertir desde decimal a binario

Método de las divisiones sucesivas



Si bien utilizamos este método para convertir a binario, podemos usarlo para cualquier base.

Convertir de binario a decimal

Usando las potencias de la base representada.

$$101101111_2 \mapsto$$

Convertir de binario a decimal

Usando las potencias de la base representada.

$101101111_2 \mapsto$

Convertir de binario a decimal

Usando las potencias de la base representada.

$$101101111_2 \mapsto$$

$$2^8 \cdot \boxed{1} + 2^7 \cdot \boxed{0} + 2^6 \cdot \boxed{1} + 2^5 \cdot \boxed{1} + 2^4 \cdot \boxed{0} + 2^3 \cdot \boxed{1} + 2^2 \cdot \boxed{1} + 2^1 \cdot \boxed{1} + 2^0 \cdot \boxed{1}$$

Convertir de binario a decimal

Usando las potencias de la base representada.

$$101101111_2 \mapsto$$

$$2^8 \cdot \boxed{1} + 2^7 \cdot \boxed{0} + 2^6 \cdot \boxed{1} + 2^5 \cdot \boxed{1} + 2^4 \cdot \boxed{0} + 2^3 \cdot \boxed{1} + 2^2 \cdot \boxed{1} + 2^1 \cdot \boxed{1} + 2^0 \cdot \boxed{1}$$

$$256 + 0 + 64 + 32 + 0 + 8 + 4 + 2 + 1$$

Convertir de binario a decimal

Usando las potencias de la base representada.

$$101101111_2 \mapsto 367_{10}$$

$$2^8 \cdot \boxed{1} + 2^7 \cdot \boxed{0} + 2^6 \cdot \boxed{1} + 2^5 \cdot \boxed{1} + 2^4 \cdot \boxed{0} + 2^3 \cdot \boxed{1} + 2^2 \cdot \boxed{1} + 2^1 \cdot \boxed{1} + 2^0 \cdot \boxed{1}$$

$$256 + 0 + 64 + 32 + 0 + 8 + 4 + 2 + 1 = 367_{10}$$

Convertir de binario a decimal

Usando las potencias de la base representada.

$$101101111_2 \mapsto 367_{10}$$

$$2^8 \cdot \boxed{1} + 2^7 \cdot \boxed{0} + 2^6 \cdot \boxed{1} + 2^5 \cdot \boxed{1} + 2^4 \cdot \boxed{0} + 2^3 \cdot \boxed{1} + 2^2 \cdot \boxed{1} + 2^1 \cdot \boxed{1} + 2^0 \cdot \boxed{1}$$

$$256 + 0 + 64 + 32 + 0 + 8 + 4 + 2 + 1 = 367_{10}$$

Si bien utilizamos este método para convertir desde binario a decimal, podemos usarlo para convertir desde cualquier otra base.

Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

$$1101101111_2 \mapsto$$

Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

$$1101101111_2 \mapsto$$

0 0 1 1 0 1 1 0 1 1 1 1

Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

$1101101111_2 \mapsto$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

$$1101101111_2 \mapsto 7$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

$$1101101111_2 \mapsto 57$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

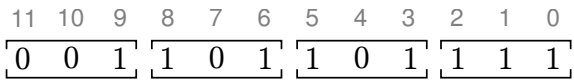
$$1101101111_2 \mapsto 557$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

$$1101101111_2 \mapsto 1557$$



Convertir binario a octal

Tomar de a 3 dígitos binarios y convertirlos a base octal

$$1101101111_2 \mapsto 1557_8$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

$$1101101111_2 \mapsto$$

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

$1101101111_2 \mapsto$

0 0 1 1 0 1 1 0 1 1 1 1

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

$1101101111_2 \mapsto$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

$$1101101111_2 \mapsto \mathrm{F}$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

$$1101101111_2 \mapsto 6F$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

$$1101101111_2 \mapsto 36\mathrm{F}$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

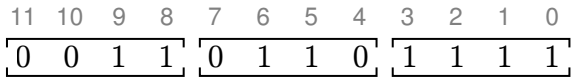
$$1101101111_2 \mapsto 36F_{16}$$

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	1	1	0	1	1	1	1

Convertir binario a hexadecimal

Tomar de a 4 dígitos binarios y convertirlos a base hexadecimal.

$$1101101111_2 \mapsto 36F_{16}$$



Desde binario a cualquier base que sea potencia de dos se puede convertir directamente tomando secuencias de bits y convirtiéndolas una a una.

Convertir hexadecimal a binario

Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

$$\mathrm{F5D}_{16} \longmapsto$$

Convertir hexadecimal a binario

Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

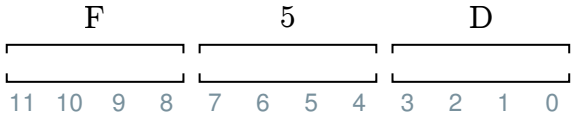
$$F5D_{16} \longmapsto$$

F 5 D

Convertir hexadecimal a binario

Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

$$F5D_{16} \mapsto$$



Convertir hexadecimal a binario

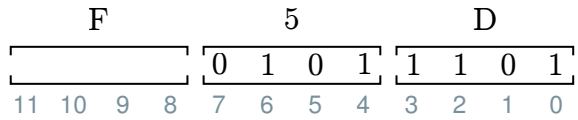
Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

$$F5D_{16} \mapsto$$

Convertir hexadecimal a binario

Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

$$F5D_{16} \mapsto$$



Convertir hexadecimal a binario

Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

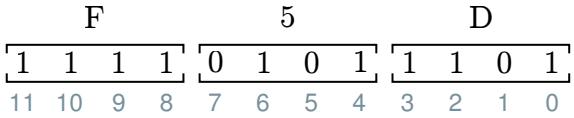
$$F5D_{16} \mapsto$$

F				5				D			
1	1	1	1	0	1	0	1	1	1	0	1
11	10	9	8	7	6	5	4	3	2	1	0

Convertir hexadecimal a binario

Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

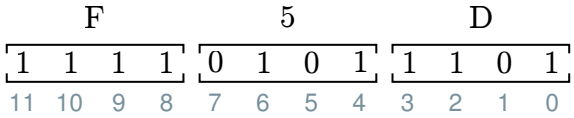
$$F5D_{16} \longmapsto 111101011101_2$$



Convertir hexadecimal a binario

Tomar independientemente cada dígito hexadecimal y convertirlo a binario.

$$F5D_{16} \longmapsto 111101011101_2$$



Este método se puede utilizar también de Octal a binario.

Resumen

La siguiente tabla presenta todos los métodos vistos para convertir entre cualquiera de las bases estudiadas.

		Desde			
		Decimal (10)	Binario (2)	Octal (8)	Hexadecimal (16)
Hasta	Decimal (10)		potencias	potencias	potencias
	Binario (2)	división		por dígito	por dígito
	Octal (8)	división	tomar de a 3		desde binario
	Hexadecimal (16)	división	tomar de a 4	desde binario	

Información binaria

La computadora puede almacenar y operar con **información binaria o digital**.

Intepretamos ceros (0) y unos (1) como ausencia o presencia de una diferencia de tensión (bits).

Tenemos que entender cómo **interpretar** una lista de bits e identificar a qué **número** o entidad corresponde.

Las representaciones están limitadas a operar con una **cantidad fija de bits**.

Para n dígitos en base b existen b^n posibles combinaciones.

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario (2)	Decimal (10)	Animales	Algunos números
0000	0	Perro	1
0001	1	Gato	2
0010	2	Tortuga	3
0011	3	Jirafa	4
0100	4	Búho	5
0101	5	Tigre	10
0110	6	Rata	11
0111	7	León	12
1000	8	Elefante	13
1001	9	Mono	14
1010	10	Cebra	15
1011	11	Gorila	-1
1100	12	Oso	-2
1101	13	Vaca	-3
1110	14	Panda	-4
1111	15	Conejo	-5

Orden de los bits

- El número 15732 en binario es `11110101110100`.
- Necesitamos exactamente 14 bits para poder representarlo.
- Contamos los bits de derecha a izquierda en su representación.

1	1	1	1	0	1	0	1	1	1	0	1	0	0
13	12	11	10	9	8	7	6	5	4	3	2	1	0

- Llamamos "**bit más significativo**", al bit más grande en el orden.

1	1	1	1	0	1	0	1	1	1	0	1	0	0
13	12	11	10	9	8	7	6	5	4	3	2	1	0

- Llamamos "**bit menos significativo**", al bit más chico en el orden.

1	1	1	1	0	1	0	1	1	1	0	1	0	0
13	12	11	10	9	8	7	6	5	4	3	2	1	0

Números enteros

- **Sin signo**
- **Signo y Magnitud**
- **Complemento a 2**
- **Notación exceso-e**

Números enteros

Notación Sin Signo (n bits)

- Usa n bits para el valor.
- Rango representable 0 a $2^n - 1$.
- No hay números negativos.

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Sin Signo
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Números enteros

Notación Sin Signo (n bits)

- Usa n bits para el valor.
- Rango representable 0 a $2^n - 1$.
- No hay números negativos.

Convertir $X_{10} \Leftrightarrow Y_{SinSigno}$

Decimal a Sin Signo: $Y = base2(X)$

Sin Signo a Decimal: $X = base10(Y)$

Ejemplo:

Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Sin Signo
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Números enteros

Notación Signo y Magnitud (n bits)

- Usa 1 bit para signo y $n - 1$ bits para el valor.
- Rango representable $-(2^{n-1} - 1)$ a $2^{n-1} - 1$.
- No se puede utilizar para hacer operaciones directamente, debe ser interpretado.
- Permite obtener la magnitud sin hacer ninguna operación.

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Con Signo
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-0
1001	-1
1010	-2
1011	-3
1100	-4
1101	-5
1110	-6
1111	-7

Números enteros

Notación Signo y Magnitud (n bits)

- Usa 1 bit para signo y $n - 1$ bits para el valor.
- Rango representable $-(2^{n-1} - 1)$ a $2^{n-1} - 1$.
- No se puede utilizar para hacer operaciones directamente, debe ser interpretado.
- Permite obtener la magnitud sin hacer ninguna operación.
- Bit de signo: 0 = positivo, 1 = negativo.
- La magnitud está determinada por el resto de los bits.

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Con Signo
1 111	-7
1 110	-6
1 101	-5
1 100	-4
1 011	-3
1 010	-2
1 001	-1
1 000	-0
0 000	0
0 001	1
0 010	2
0 011	3
0 100	4
0 101	5
0 110	6
0 111	7

Números enteros

Notación Signo y Magnitud (n bits)

- Usa 1 bit para signo y $n - 1$ bits para el valor.
- Rango representable $-(2^{n-1} - 1)$ a $2^{n-1} - 1$.
- No se puede utilizar para hacer operaciones directamente, debe ser interpretado.
- Permite obtener la magnitud sin hacer ninguna operación.
- Bit de signo: `0` = positivo, `1` = negativo.
- La magnitud está determinada por el resto de los bits.

Convertir $X_{10} \Leftrightarrow Y_{SignoMagnitud}$

Decimal a Signo-Magnitud:

Si $X \geqslant 0$ entonces `Y = concatenar(0, base2(X))`
Si $X \leqslant 0$ entonces `Y = concatenar(1, base2(abs(X)))`

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Con Signo
1 111	-7
1 110	-6
1 101	-5
1 100	-4
1 011	-3
1 010	-2
1 001	-1
1 000	-0
0 000	0
0 001	1
0 010	2
0 011	3
0 100	4
0 101	5
0 110	6
0 111	7

Números enteros

Notación Signo y Magnitud (n bits)

- Usa 1 bit para signo y $n - 1$ bits para el valor.
- Rango representable $-(2^{n-1} - 1)$ a $2^{n-1} - 1$.
- No se puede utilizar para hacer operaciones directamente, debe ser interpretado.
- Permite obtener la magnitud sin hacer ninguna operación.
- Bit de signo: `0` = positivo, `1` = negativo.
- La magnitud está determinada por el resto de los bits.

Convertir $X_{10} \Leftrightarrow Y_{SignoMagnitud}$

Decimal a Signo-Magnitud:

Si $X \geqslant 0$ entonces `Y = concatenar(0, base2(X))`
Si $X \leqslant 0$ entonces `Y = concatenar(1, base2(abs(X)))`

Signo-Magnitud a Decimal:

Si $Y_{n-1} == 0$ entonces `X = base10(`$Y_{n-2} \ldots Y_0$`)`
Si $Y_{n-1} == 1$ entonces `X = -base10(`$Y_{n-2} \ldots Y_0$`)`

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Con Signo
1 111	-7
1 110	-6
1 101	-5
1 100	-4
1 011	-3
1 010	-2
1 001	-1
1 000	-0
0 000	0
0 001	1
0 010	2
0 011	3
0 100	4
0 101	5
0 110	6
0 111	7

Números enteros

Notación Complemento a 2 (n bits)

- Usa n bits para el valor y signo.
- Rango representable $-(2^{n-1})$ a $2^{n-1}-1$.
- Permite hacer operaciones directamente, sin interpretar el número.
- No es posible obtener la magnitud de los números negativos fácilmente.

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Complemento a 2
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	−8
1001	−7
1010	−6
1011	−5
1100	−4
1101	−3
1110	−2
1111	−1

Números enteros

Notación Complemento a 2 (n bits)

- Usa n bits para el valor y signo.
- Rango representable $-(2^{n-1})$ a $2^{n-1}-1$.
- Permite hacer operaciones directamente, sin interpretar el número.
- No es posible obtener la magnitud de los números negativos fácilmente.

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Complemento a 2
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Números enteros

Notación Complemento a 2 (n bits)

- Usa n bits para el valor y signo.
- Rango representable $-(2^{n-1})$ a $2^{n-1}-1$.
- Permite hacer operaciones directamente, sin interpretar el número.
- No es posible obtener la magnitud de los números negativos fácilmente.

Convertir $X_{10} \Leftrightarrow Y_{Complemento2}$

Decimal a Complemento2:

Si $X \geqslant 0$ entonces `Y = base2(X)`
Si $X < 0$ entonces `Y = BitwiseNot(base2(abs(X)))+1`

Def. BitwiseNot
Invertir bit a bit

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Complemento a 2
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Números enteros

Notación Complemento a 2 (n bits)

- Usa n bits para el valor y signo.
- Rango representable $-(2^{n-1})$ a $2^{n-1}-1$.
- Permite hacer operaciones directamente, sin interpretar el número.
- No es posible obtener la magnitud de los números negativos fácilmente.

Convertir $X_{10} \Leftrightarrow Y_{Complemento2}$

Decimal a Complemento2:

Si $X \geqslant 0$ entonces `Y = base2(X)`
Si $X < 0$ entonces `Y = BitwiseNot(base2(abs(X)))+1`

Def. BitwiseNot
Invertir bit a bit

Complemento2 a Decimal:

Si $Y_{n-1} == 0$ entonces `X = base10(Y)`
Si $Y_{n-1} == 1$ entonces $X = -$`(base10(BitwiseNot(Y)+1))`

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	Complemento a 2
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Números enteros

Notación exceso-e (n bits)

- Usa n bits, no considera bit para el signo.
- Rango representable $-e$ a $2^n - e - 1$.
- Permite representar un rango arbitrario.
- No se puede identificar el signo o magnitud sin hacer operaciones.

Ejemplo:
Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	exceso-5
0000	-5
0001	-4
0010	-3
0011	-2
0100	-1
0101	0
0110	1
0111	2
1000	3
1001	4
1010	5
1011	6
1100	7
1101	8
1110	9
1111	10

Números enteros

Notación exceso-e (n bits)

- Usa *n* bits, no considera bit para el signo.
- Rango representable $-e$ a $2^n - e - 1$.
- Permite representar un rango arbitrario.
- No se puede identificar el signo o magnitud sin hacer operaciones.

Convertir $X_{10} \Leftrightarrow Y_{exceso-e}$

Decimal a exceso-e:

```
Y = base2(X+e)
```

Ejemplo:

Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	exceso-5
0000	-5
0001	-4
0010	-3
0011	-2
0100	-1
0101	0
0110	1
0111	2
1000	3
1001	4
1010	5
1011	6
1100	7
1101	8
1110	9
1111	10

Números enteros

Notación exceso-e (n bits)

- Usa n bits, no considera bit para el signo.
- Rango representable $-e$ a $2^n - e - 1$.
- Permite representar un rango arbitrario.
- No se puede identificar el signo o magnitud sin hacer operaciones.

Convertir $X_{10} \Leftrightarrow Y_{exceso-e}$

Decimal a exceso-e:

```
Y = base2(X+e)
```

exceso-e a Decimal:

```
X = base10(Y)-e
```

Ejemplo:

Con 4 bits $\rightarrow 2^4 = 16$ combinaciones

Binario	exceso-5
0000	-5
0001	-4
0010	-3
0011	-2
0100	-1
0101	0
0110	1
0111	2
1000	3
1001	4
1010	5
1011	6
1100	7
1101	8
1110	9
1111	10

Resumen de números enteros

- **Sin signo**
  Representa solamente números sin signo. Se utiliza cuando los datos sobre los que se van a operar siempre son positivos. Se debe tener cuidado con las restas.

Resumen de números enteros

- **Sin signo**
  Representa solamente números sin signo. Se utiliza cuando los datos sobre los que se van a operar siempre son positivos. Se debe tener cuidado con las restas.
- **Signo y Magnitud**
  Representa independientemente la magnitud y la codificación del signo. Es complejo para realizar operaciones, porque debe tener en cuenta el signo para operar. Es muy simple para visualizar porque no hay que hacer operaciones para conocer la magnitud del número.

Resumen de números enteros

- **Sin signo**
  Representa solamente números sin signo. Se utiliza cuando los datos sobre los que se van a operar siempre son positivos. Se debe tener cuidado con las restas.
- **Signo y Magnitud**
  Representa independientemente la magnitud y la codificación del signo. Es complejo para realizar operaciones, porque debe tener en cuenta el signo para operar. Es muy simple para visualizar porque no hay que hacer operaciones para conocer la magnitud del número.
- **Complemento a 2**
  Permite realizar operaciones de sumas y restas sin tener en cuenta si el número es positivo o negativo. Además, es posible identificar el signo del número mirando un solo bit.

Resumen de números enteros

- **Sin signo**
  Representa solamente números sin signo. Se utiliza cuando los datos sobre los que se van a operar siempre son positivos. Se debe tener cuidado con las restas.
- **Signo y Magnitud**
  Representa independientemente la magnitud y la codificación del signo. Es complejo para realizar operaciones, porque debe tener en cuenta el signo para operar. Es muy simple para visualizar porque no hay que hacer operaciones para conocer la magnitud del número.
- **Complemento a 2**
  Permite realizar operaciones de sumas y restas sin tener en cuenta si el número es positivo o negativo. Además, es posible identificar el signo del número mirando un solo bit.
- **Notación exceso-e**
  Permite representar un rango variable de números, asimétrico para positivos y negativos. La dificultad para su operación es moderada, ya que se debe tener en cuenta el exceso a la hora de operar.

Tamaño de un número

En la computadora todos los valores almacenados tienen un tamaño.
Decimos que un número está **representado** con/en una determinada **cantidad de bits**.

Tamaño de un número

En la computadora todos los valores almacenados tienen un tamaño.
Decimos que un número está **representado** con/en una determinada **cantidad de bits**.

Ejemplo:

El número 21d en binario es 10101b.
Si representamos este número como:

Notación sin signo de 10 bits →
Notación sin signo de 5 bits →
Notación con signo de 8 bits →
Notación con signo de 5 bits →

Tamaño de un número

En la computadora todos los valores almacenados tienen un tamaño.
Decimos que un número está **representado** con/en una determinada **cantidad de bits**.

Ejemplo:

El número 21d en binario es 10101b.
Si representamos este número como:

Notación sin signo de 10 bits	→	0000010101b
Notación sin signo de 5 bits	→	10101b
Notación con signo de 8 bits	→	00010101b
Notación con signo de 5 bits	→	No es posible.

Importante
Cuando hablamos de **tamaño de un número**, nos referimos a la **cantidad de bits** de su representación binaria.

Bibliografía

- Tanenbaum, “Organización de Computadoras. Un Enfoque Estructurado”, 4ta Edición, 2000.
  - **Apéndice - Números Binarios** - Páginas 631-640.

- Null, “Essentials of Computer Organization and Architecture”, 5th Edition, 2018.
  - **Chapter 2 - Data Representation in Computer Systems:**
    - 2.3 - Converting Between Bases.
    - 2.4 - Signed Integer Representation.

Ejercicios

Con lo visto, ya pueden resolver hasta el ejercicio 5 inclusive de la Práctica 1.

¡Gracias!

Recuerden leer los comentarios adjuntos
en cada clase por aclaraciones.