

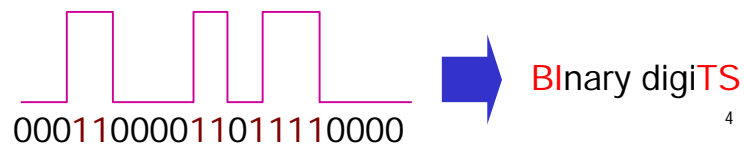
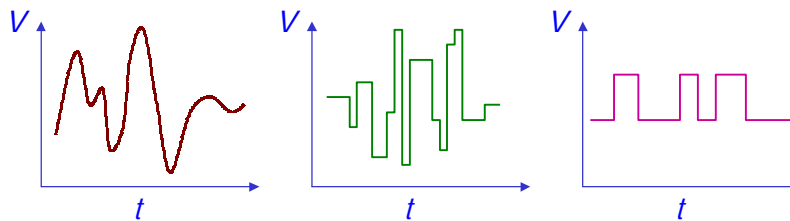
2.1 Codificación binaria

- Bits
- Representación octal
- Representación hexadecimal
- Conversiones

3

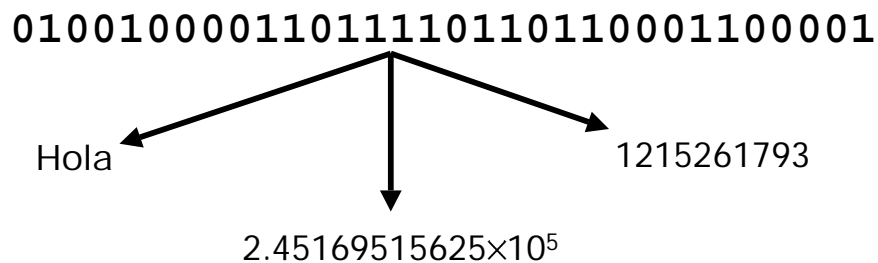
Bits

- Los ordenadores utilizan señales eléctricas digitales
- Sólo dos niveles de voltaje: más sencillo



Codificación de la información

- La *información* se representa con *símbolos* de un *código*
- Cada *símbolo* corresponde a una *combinación de bits*
- El *significado* de los bits depende de la *interpretación*



5

Sistema binario

Bits individuales: poco prácticos

8 bits = 1 byte (*octeto*) → unidad básica de memoria

16, 32, 64 bits = 1 *palabra* de 16, 32, 64 bits

2¹⁰ bytes = 1,024 bytes = 1 KB (*kilobyte*)

2²⁰ bytes = 1,048,576 bytes = 1,024 Kb = 1 MB

2³⁰ bytes = 1,073,741,824 bytes = 1 GB

...

Relación *complicada* entre binario y decimal

6

Números binarios (en base 2)

7 6 5 4 3 2 1 0 ← posiciones

11011011

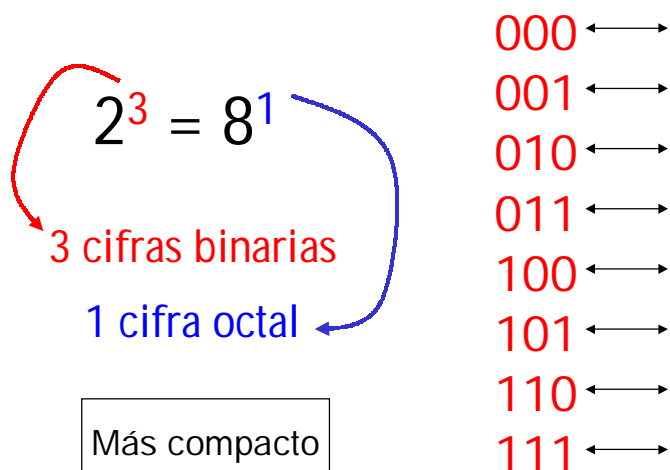
$1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

Número equivalente en base 10:

219

7

Representación en base 8 (octal)



8

Binario a octal

10100101110100110001011

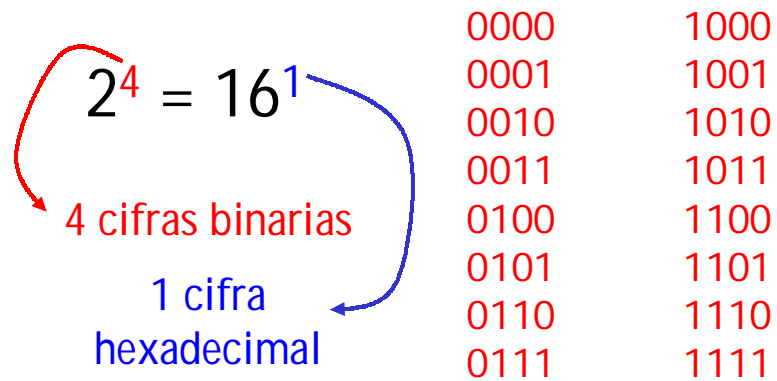
9

Octal a binario

24564613₍₈₎

10

Representación en base 16 (hexadecimal)



11

Binario a hexadecimal

10100101110100110001011

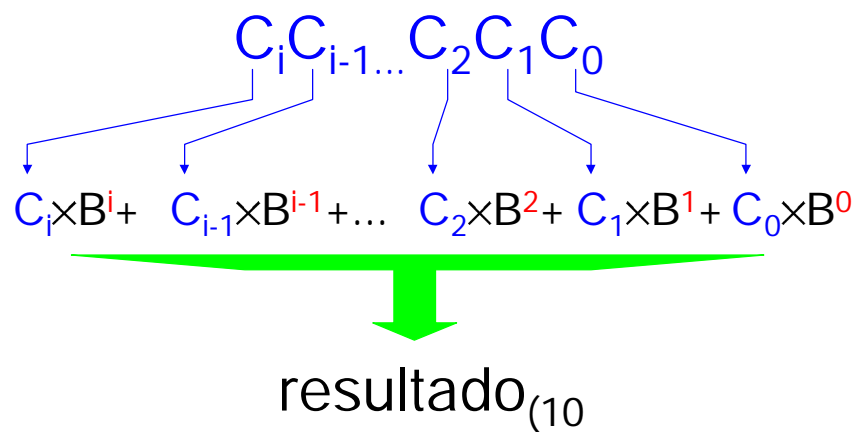
12

Hexadecimal a binario

52E98B₍₁₆₎

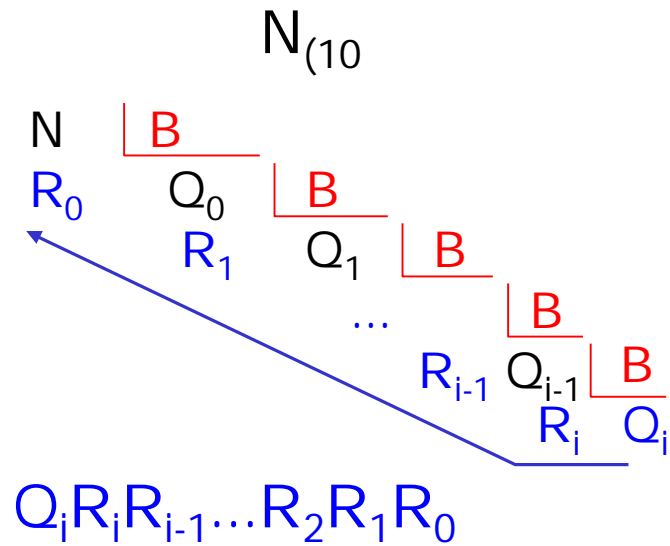
13

De base B a decimal (base 10)



14

Decimal a base B



15

Hexadecimal a decimal

$169CB_{(16)}$

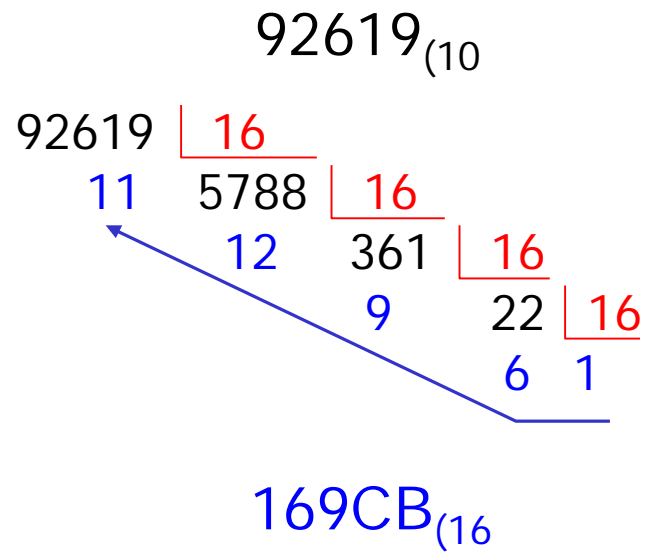
$$1 \cdot 16^4 + 6 \cdot 16^3 + 9 \cdot 16^2 + 12 \cdot 16^1 + 11 \cdot 16^0$$

\downarrow

$92619_{(10)}$

16

Decimal a hexadecimal



17

2.2 Tipos de datos básicos

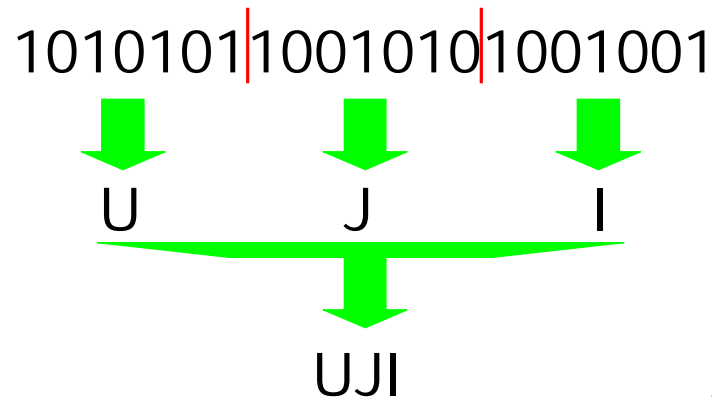
- Carácter
- Número entero
- Número real
- Booleano

18

Codificación de caracteres

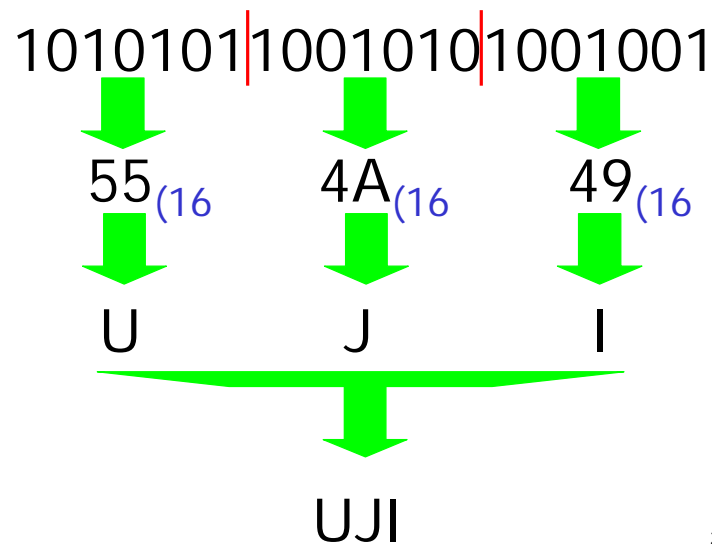
ASCII: American Standard Code for Information Interchange

7 bits = 1 carácter $\rightarrow 2^7=128$ caracteres



19

Codificación de caracteres



20

Tabla de códigos ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Caracteres de control

Signos de puntuación

Letras mayúsculas y minúsculas

Números y signos aritméticos

21

Codigos ASCII *extendidos*

Codificación en 1 byte: 128 caracteres adicionales

Extensiones d'IBM

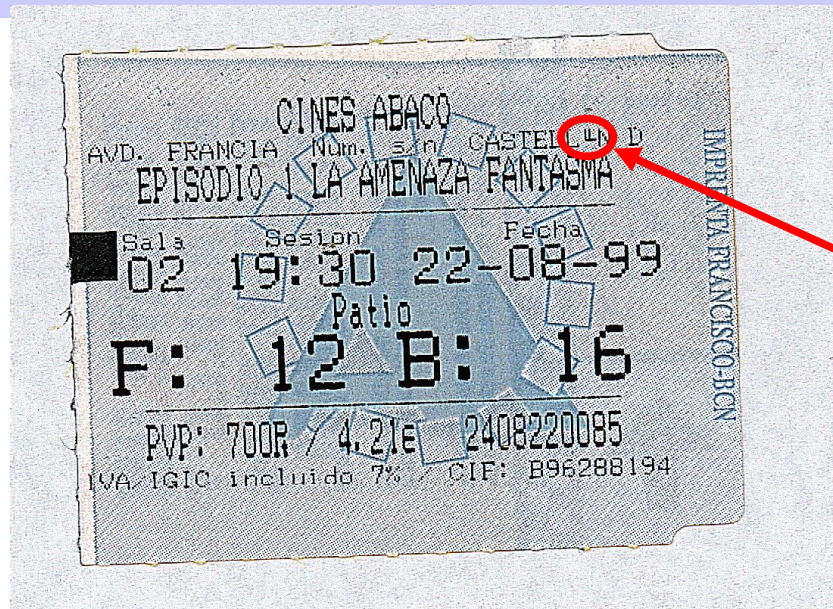
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G
2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H
3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I
4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J
5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K
6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L
7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M
8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N
9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U

Extensiones de Windows

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G
2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H
3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I
4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J
5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K
6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L
7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M
8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N
9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U

22

Errores de traducción



El estándar Unicode

- Consorcio
(IBM, Microsoft, etc.)
- Internacional
- 16 bits
- Amplia el ASCII

ASCII/8859-1 Text

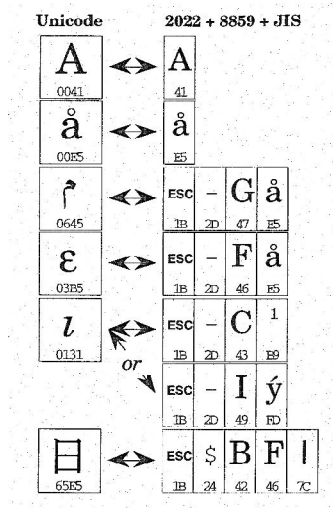
A	0100 0001
S	0101 0011
C	0100 0011
I	0100 1001
I	0100 1001
/	0010 1111
8	0011 1000
8	0011 1000
5	0011 0101
9	0011 1001
-	0010 1101
l	0011 0001
	0010 0000
t	0111 0100
e	0110 0101
x	0111 1000
t	0111 0100

Unicode Text

A	0000 0000 0100 0001
S	0000 0000 0101 0011
C	0000 0000 0100 0011
I	0000 0000 0100 1001
I	0000 0000 0100 1001
	0000 0000 0010 0000
天地	0101 1001 0010 1001
	0101 0111 0011 0000
	0000 0000 0010 0000
س	0000 0110 0011 0011
J	0000 0110 0100 0100
ا	0000 0110 0011 0111
م	0000 0110 0100 0101
	0000 0000 0010 0000
α	0000 0011 1011 0001
κ	0010 0010 0111 0000
γ	0000 0011 1011 0011

Objetivos del diseño

- Universal
- Eficiente
- Uniforme
- No-ambiguo



25

Representación de números enteros

- Valor absoluto en binario
- Bit adicional para el signo

$$45_{(10)} = 2D_{(16)} = 00101101_{(2)}$$

$$+45_{(10)} = 000101101_{(2)}$$

$$-45_{(10)} = 100101101_{(2)}$$

26

Signo y magnitud



rango: $[-2^{n-1}+1, +2^{n-1}-1]$

Dos representaciones del cero

27

Complemento a dos

Positivos: igual que signo y magnitud

Negativos:

1. Invertir todos los bits del positivo
2. Sumar 1 al resultado anterior
3. Despreciar el último acarreo, si hay

➡ *Idem para pasar de negativo a positivo*

28

Complemento a dos: ejemplo con 8 bits

$+45 = 00101101$

1. Invertir bits 11010010

2. Sumar 1 11010011

3. Acarreo? no

$11010011 = -45$

29

Complement a dos

rango: $[-2^{n-1}, +2^{n-1}-1]$

Única representación del cero

Fácil para sumar y restar: *más utilizado*

Cuantos bits?

8, 16, 32 ...

30

Errores de rango

Resultado de operación fuera del rango

El comportamiento depende de la implementación

Ejemplo:

21419	01010011110101011
+ 18397	+ 0100011111011101
<hr/>	<hr/>
	1001101110001000

31





Representación *por exceso*

No utiliza bit de signo

$$\text{código} = \text{valor} + \text{exceso}$$

Para n bits  $\text{exceso} = 2^{n-1}$

Ejemplo con 8 bits: $\text{exceso} = 2^7 = 128$

+45		+45+128 = 173	₍₁₀₎		10101101	₍₂₎
-45		-45+128 = 83	₍₁₀₎		01010011	₍₂₎

32

Representación de números reales

Aproximaciones: π , e , etc.

Notación científica:

$$\boxed{2.99792458} * 10^8$$

mantisa

Representación en *coma flotante*:



33

Conversión de números reales

$$\begin{array}{c} 26.4713_{(8)} \\ \swarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \searrow \\ 2 * 8^1 + 6 * 8^0 + 4 * 8^{-1} + 7 * 8^{-2} + 1 * 8^{-3} + 3 * 8^{-4} \\ \downarrow \\ 22.612060546875_{(10)} \end{array}$$

34

Estándards del IEEE

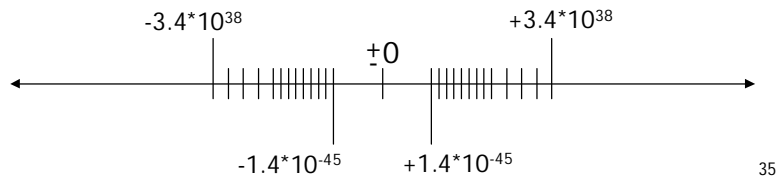
Precisión simple (32 bits):

1 bit de signo

8 bits de exponente: $[-126, +127]$
(*exceso 127*)

23 bits de mantisa *normalizada*,
con el primer bit *implícito*

rango:



mantisa normalizada

Sólo una cifra *a la izquierda* del punto.

$$295.02 \times 10^{-4} = 2.9502 \times 10^{-2}$$

El número 2.9502 está subrayado en rojo.

sumar 2 al exponente

2 lugares a la izquierda

En binario:

$$110.01 \times 2^{-4} = 1.1001 \times 2^{-2}$$

$$0.0011 \times 2^{-5} = 1.1000 \times 2^{-8}$$

Ejemplo

11000011100101100000000000000000

valor =

37

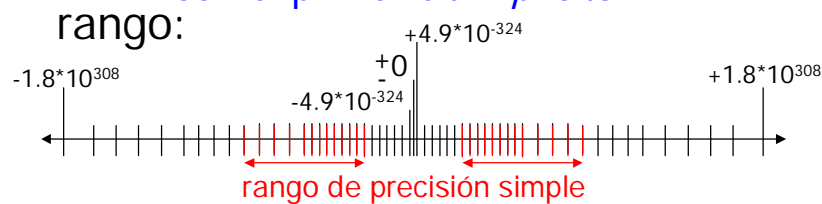
Estándards del IEEE

Precisión doble (64 bits):

1 bit de signo

11 bits de exponente: $[-1022, +1023]$
(exceso 1023)

52 bits de mantisa *normalizada*,
con el primer bit *implícito*



38

Valores especiales

S	E	M
---	---	---

Cero ⊗ 00000000 000000000000000000000000

Dos representaciones para el cero

Infinito ⊗ 11111111 000000000000000000000000

Infinitos positivo y negativo

NaN ⊗ 11111111 xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Not a Number

39

Números en Matlab

Todos los números son reales
con precisión doble (64 bits)

$$\text{eps} = 2.2204\text{e-}016 = 2^{-52}$$

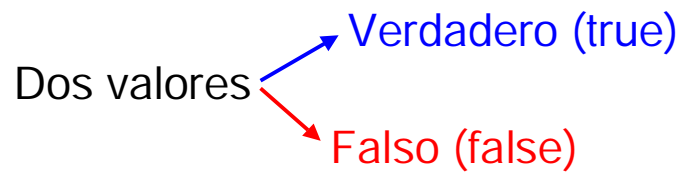
Se pueden definir enteros de 8, 16 y 32 bits

Pero no se pueden hacer operaciones!

Se han de convertir en reales para operar

40

Tipos de datos booleanos



Representación habitual:

Verdadero	1
Falso	0

41

2.3 Operaciones básicas

- Aritméticas
- Relacionales
- Lógicas

42

2.3 Operaciones básicas

- Aritméticas

+ - * /

- Relacionales

< <= == ~= >= >

- Lógicas

& (and) | (or) ~ (not)

43

Operaciones aritméticas

número op número
↓
número

237 + 1411 7.2 * 3.01 18 / 4

44

Operaciones relacionales

número op número
↓
booleano

340 > 2501 1.1 < 1.18 27 ~= 3.2

45

Operaciones lógicas

booleano op booleano
↓
booleano

true & false true | true ~ false

46

2.4 Expresiones

Combinación de operandos y operadores

La expresión se evalúa y da un resultado

$(4 + 8 * 6) / 10 \quad \sim((5 < 8) \& (3 > 4))$

47

Orden de evaluación

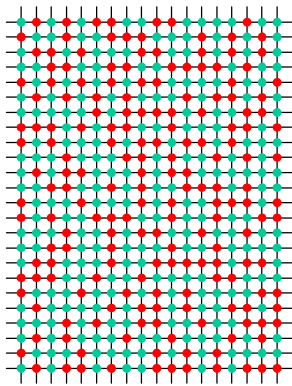
Prioridad entre operadores

↑	+	()	paréntesis
	~ -		negación
	* /		producto, división
	+ -		suma, resta
	< <= > >=		relacionales de orden
	== ~=		igualdad
	&		Y lógico
↓	-		O lógico

48

2.5 Almacenamiento

Memoria: circuito electrónico matricial



```

10101001010101011101010101010
10101101010101010100111000110
01101000001011100010101000001
11001010001010100000011010111
11010101001100101000101010101
01010101010101010101010100101
01000101000000010011110101110
11101011101100010001010110011
00011001010101010100111000110
01101000001011100010101000001
11001010001010100000011010111
11010101001100101000101010101
01010101010101010101010100101
01000101000000010011110101110
    
```

Señales eléctricas

Bits

49

Organización de la memoria

D i r e c t i ó n	2A7C3	...	2A7C2	...
	2A7C4	10010111	2A7C4	1001011100111011
	2A7C5	00111011	2A7C6	1011011000001101
	2A7C6	10110110	2A7C8	0110101111100010
	2A7C7	00001101	2A7CA	0101110101110000
	2A7C8	01101011	2A7CC	1101101001110001
	2A7C9	11100010	2A7CE	0010011000001011
	2A7CA	01011101	2A7D0	...
	2A7CB	01110000		
	2A7CC	11011010		
	2A7CD	01110001		
	2A7CE	00100110		
	2A7CF	00001011		
	2A7D0	...		

palabras de 16 bits
(32 bits, 64 bits, etc.)

Contenido

50

Variables

Guardar un valor en memoria

C82E2	...
C82E4	0010100011000100
C82E6	...

Leer un valor de memoria

Contenido de: C82E4?

IDEA: asignar nombre a la dirección de memoria

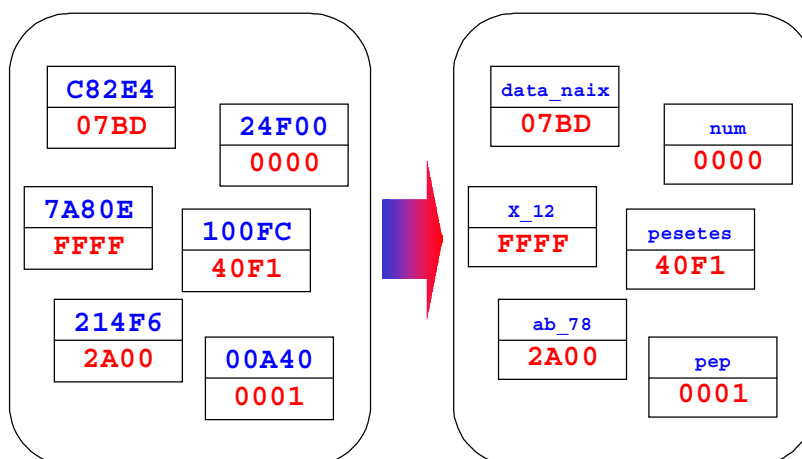
C82E4 \equiv fecha_nacimiento

Guardar un valor: fecha_nacimiento ←

Leer un valor: fecha_nacimiento? →

51

Variables: abstracción de la memoria



52

Asignación

Asignar: **guardar** un valor en una variable.

variable = valor;

Ejemplo: **precio = 1995;**

En general: *variable = expresión;*

Ejemplo: **longitud = 2 * 3.14 * 7;**

53

Lectura

Leer: **obtener** el valor de una variable

Se sustituye el nombre por el valor

Normalmente dentro de una expresión

Ejemplo:

precio_f	-
precio	1995
iva	0.16

precio_f = precio * (1+iva);

54

Otro ejemplo

y	-	a	0.1
x	1.7	b	-0.6
		c	2.3

y = a * x * x + b * x + c;

55

Un programa completo

```
r = 7.5;  
l = 2 * pi * r;  
s = pi * r^2;
```

56

Programa equivalente en C++

```
#include<iostream.h>
void main()
{
    float  pi, r, l, s;

    pi = 3.1416;
    r = 7.5;
    l = 2 * pi * r;
    s = pi * r * r;
    cout << "Longitud: " << l << endl;
    cout << "Area:      " << s << endl;
}
```

57

Bibliografía

- L. Joyanes. *Fundamentos de Programación: Algoritmos y Estructuras de Datos*. McGraw-Hill, 1992. Capítulo 1.
- L. A. Ureña y otros. *Fundamentos de Informática*. Ra-Ma, 1997. Capítulo 2.
- J. García de Jalón et al. *Aprenda Matlab 5.3 como si estuviera en primero*. E.T.S.Ing. Indust., Univ. Polit. Madrid.
- E. Alcalde, M. García. *Informática Básica*. 2ª edición. McGraw-Hill, 1994. Capítulo 3.
- P. Bishop. *Conceptos de Informática*. Anaya Multimedia, 1991. Capítulo 3.
- A. Prieto y otros. *Introducción a la Informática*. 2ª edición. McGraw-Hill, 1995. Capítulo 2.1.

58