Repaso de C

Estructura de Computadores

Autores:

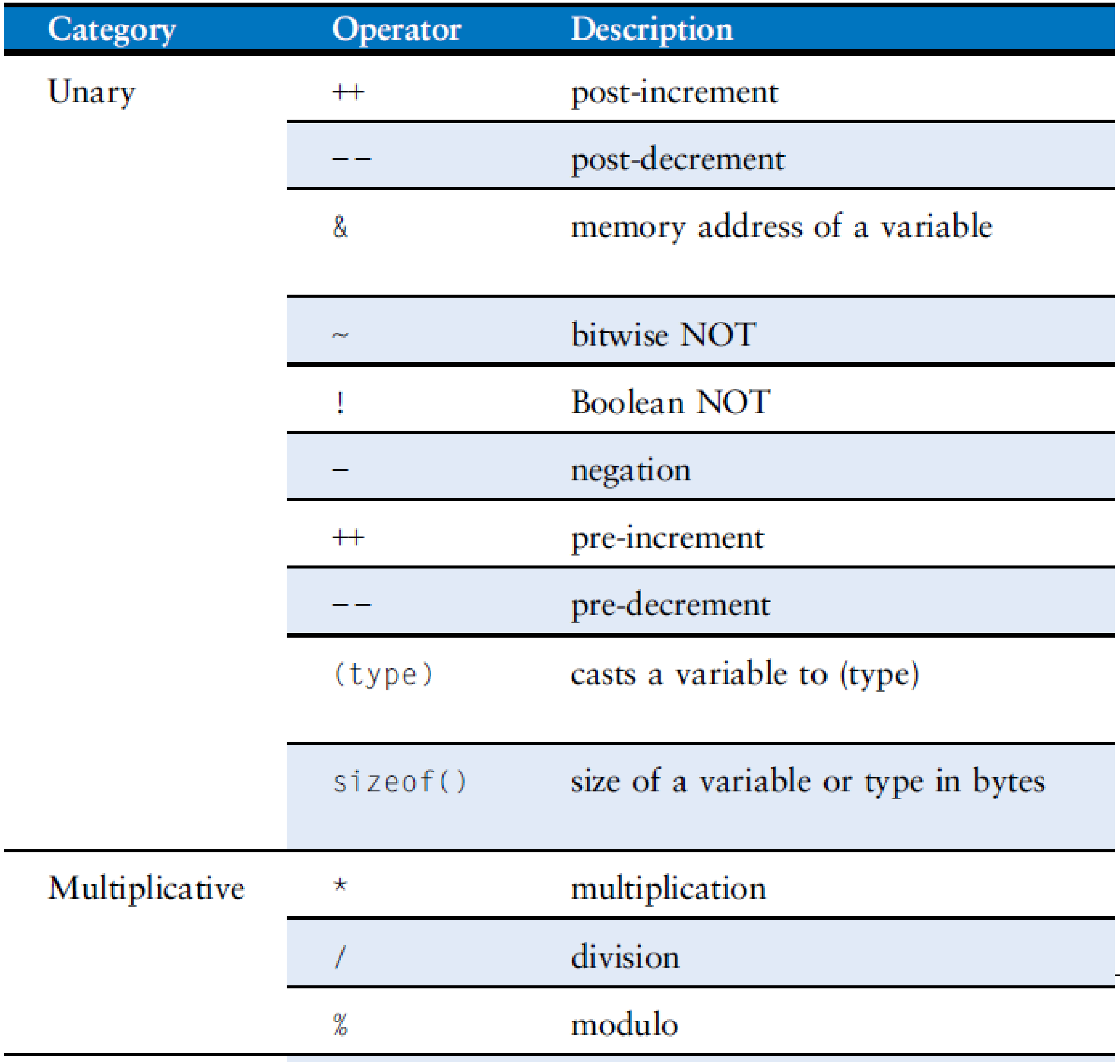
Katzalin Olcoz

Bibliografía:

Apéndice C. Digital Design and Computer Architecture. ARM edition. S. Harris y D.M. Harris.

Facultad
de
Informática

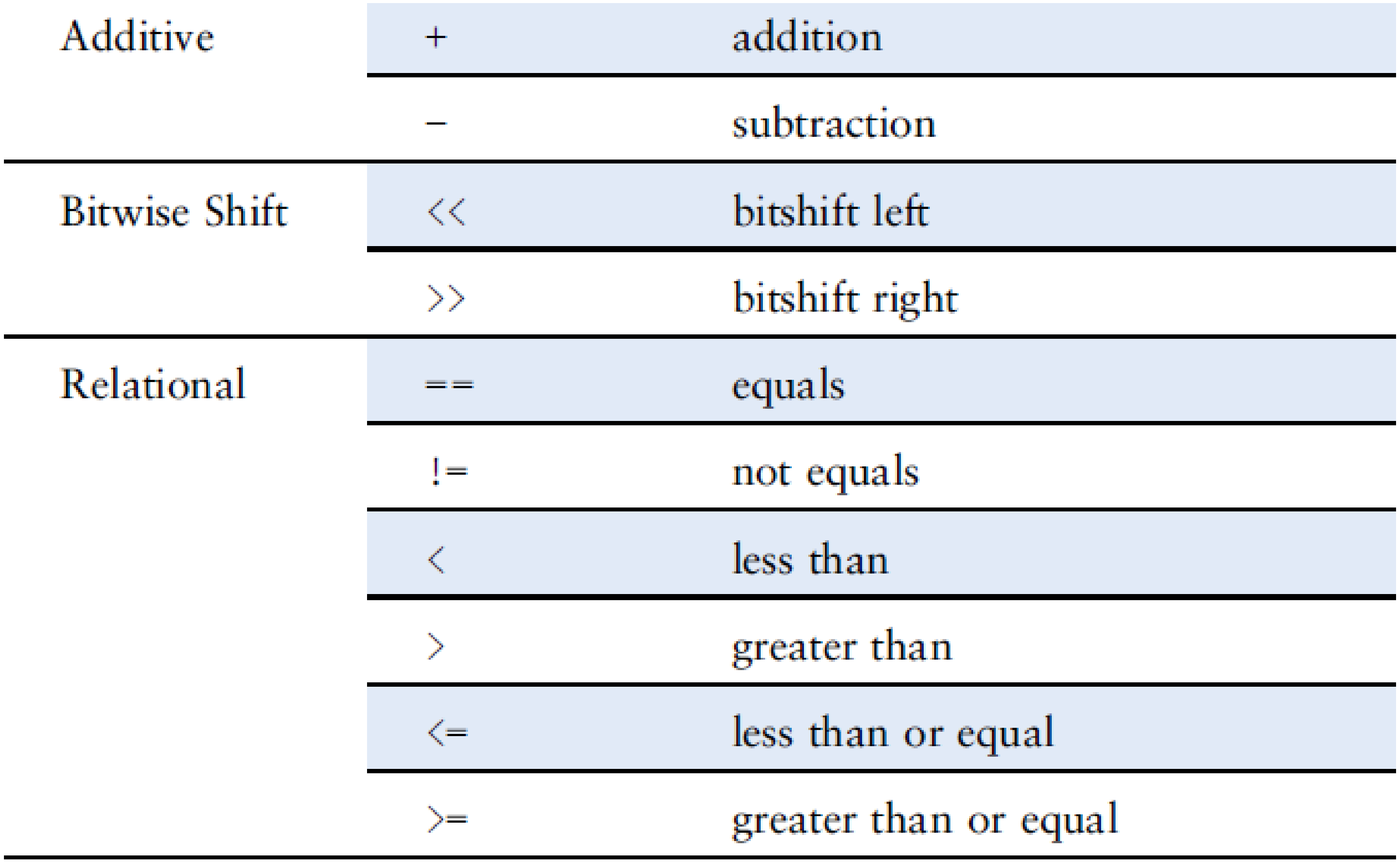
Operadores (de mayor a menor prioridad)



Category	Operator	Description
Unary	++	post-increment
	--	post-decrement
	&	memory address of a variable
	~	bitwise NOT
	!	Boolean NOT
	-	negation
	++	pre-increment
	--	pre-decrement
	(type)	casts a variable to (type)
	sizeof()	size of a variable or type in bytes
Multiplicative	*	multiplication
	/	division
	%	modulo

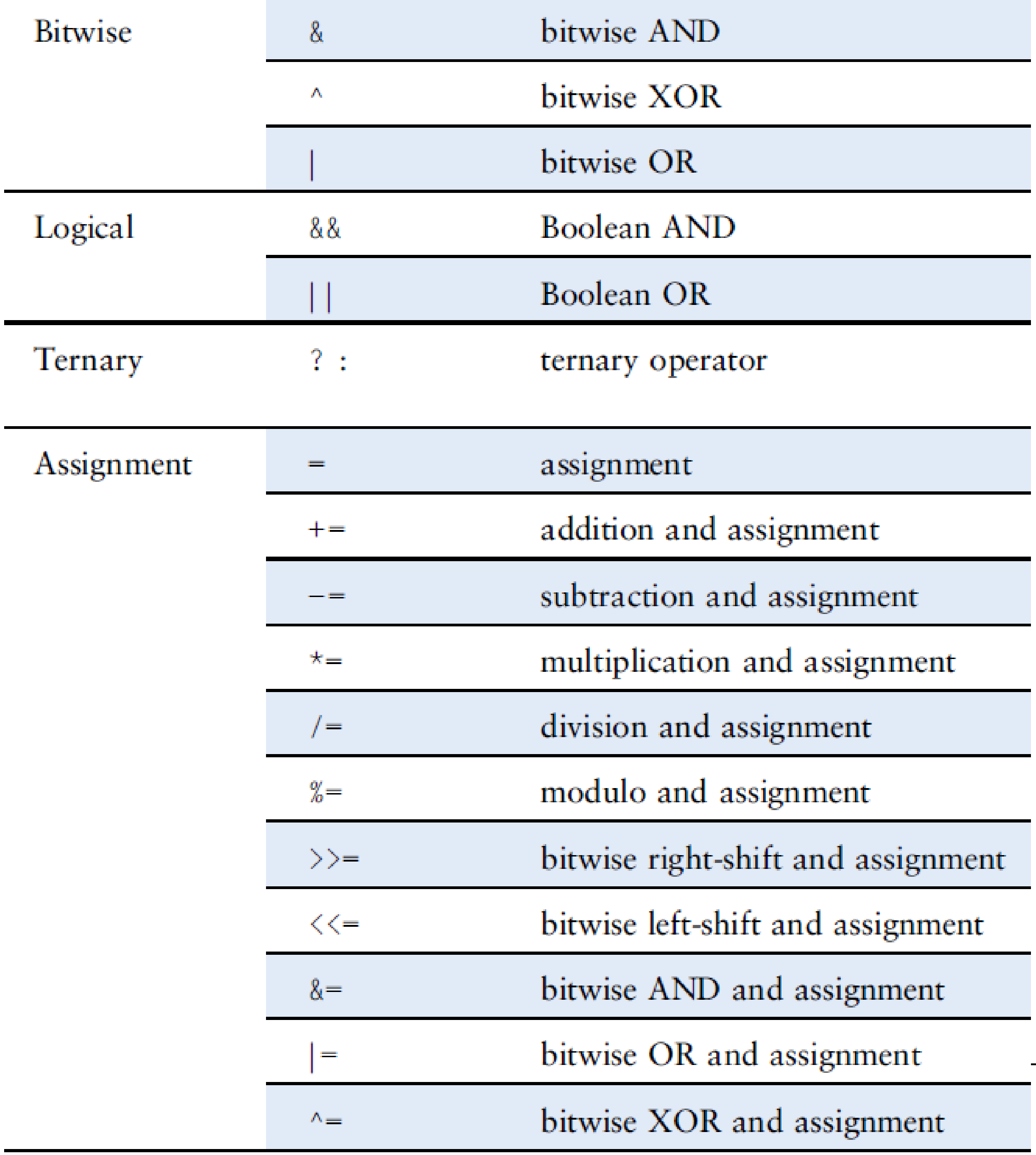
Facultad
de
Informática

Operadores (cont)



Additive	+	addition
	−	subtraction
Bitwise Shift	<<	bitshift left
	>>	bitshift right
Relational	==	equals
	!=	not equals
	<	less than
	>	greater than
	<=	less than or equal
	>=	greater than or equal

Operadores (cont 2)

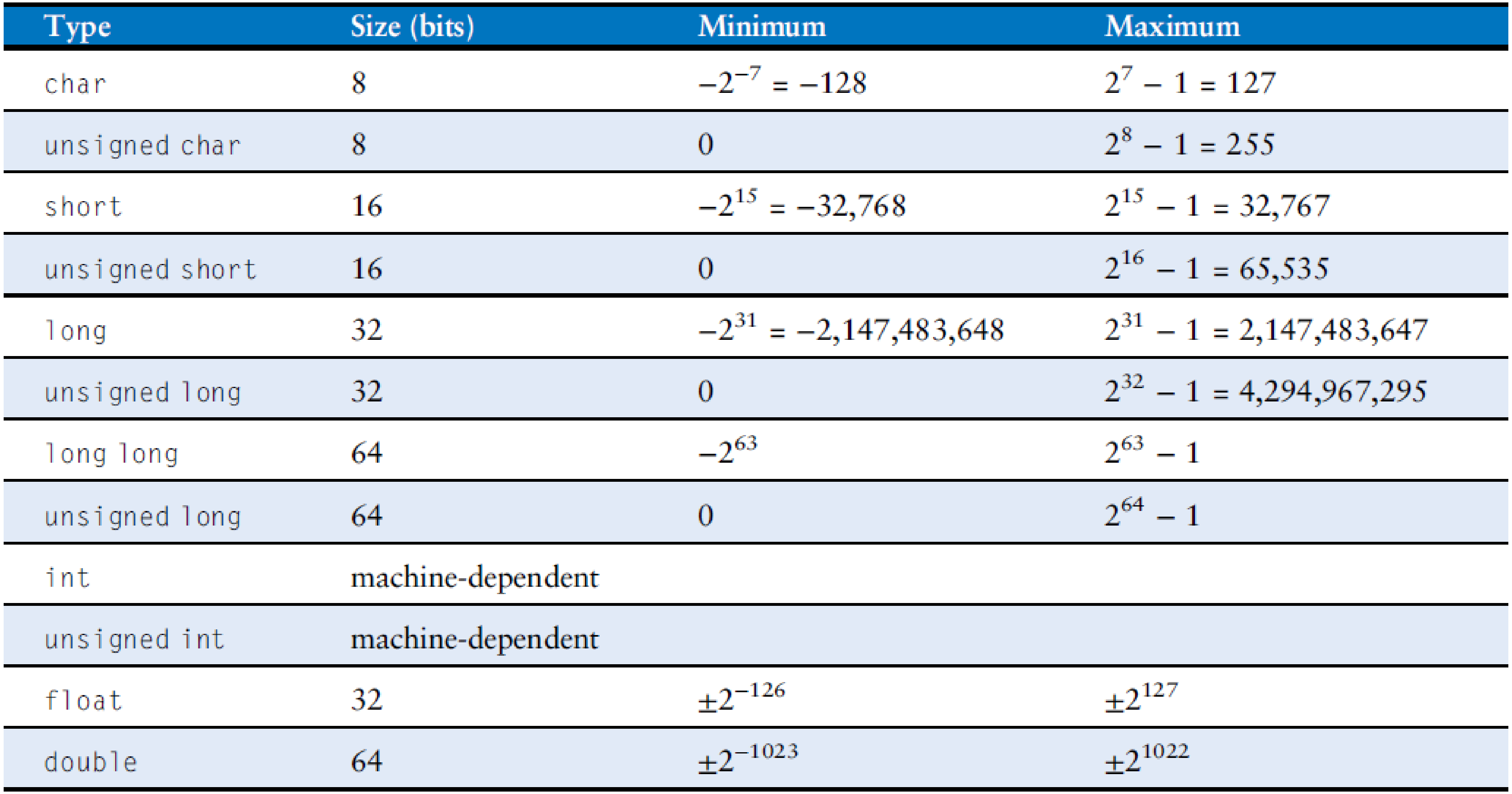


Bitwise	&	bitwise AND
	^	bitwise XOR
	|	bitwise OR
Logical	&&	Boolean AND
	||	Boolean OR
Ternary	? :	ternary operator
Assignment	=	assignment
	+=	addition and assignment
	−=	subtraction and assignment
	*=	multiplication and assignment
	/=	division and assignment
	%=	modulo and assignment
	>>=	bitwise right-shift and assignment
	<<=	bitwise left-shift and assignment
	&=	bitwise AND and assignment
	|=	bitwise OR and assignment
	^=	bitwise XOR and assignment

Variables

- Una variable tiene:
  - Tipo
  - Nombre
  - Valor
  - Ubicación en memoria
- Distinguimos entre:
  - Globales
  - Locales

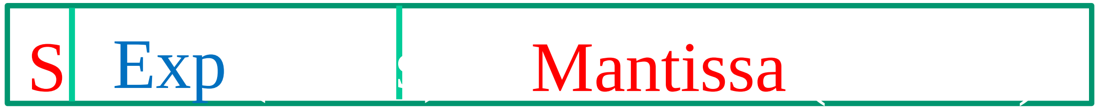
Tipos de datos primitivos



Type	Size (bits)	Minimum	Maximum
`char`	8	$-2^{-7} = -128$	$2^7 - 1 = 127$
`unsigned char`	8	0	$2^8 - 1 = 255$
`short`	16	$-2^{15} = -32{,}768$	$2^{15} - 1 = 32{,}767$
`unsigned short`	16	0	$2^{16} - 1 = 65{,}535$
`long`	32	$-2^{31} = -2{,}147{,}483{,}648$	$2^{31} - 1 = 2{,}147{,}483{,}647$
`unsigned long`	32	0	$2^{32} - 1 = 4{,}294{,}967{,}295$
`long long`	64	-2^{63}	$2^{63} - 1$
`unsigned long`	64	0	$2^{64} - 1$
`int`	machine-dependent		
`unsigned int`	machine-dependent		
`float`	32	$\pm 2^{-126}$	$\pm 2^{127}$
`double`	64	$\pm 2^{-1023}$	$\pm 2^{1022}$

Representación de números en punto flotante

IEEE-754: Define la representación de números de 32 bits (*float*) y 64 bits (*double*). Para 32 bits:



$\equiv (-1)^{S} \times (1.\text{Mantissa}) \times (2^{\text{Exp}-127})$

– Ejemplos:

- 0 10000001 1000 $\overset{\text{all zeroes}}{\cdots\cdots\cdots}$ 000 $= (-1)^0 \times (1.1000\cdots\cdot000) \times (2^{129-127}) = +1.1 \times 2^2 = (110)_2 = (6)_{10}$
- 0 10000101 0000 ········ 000 $= (-1)^0 \times (1.0000\cdots\cdot000) \times (2^{133-127}) = +1.0 \times 2^6 = (1000000)_2 = (64)_{10}$
- 1 10000000 0100 ········ 000 $= (-1)^1 \times (1.0100\cdots\cdot000) \times (2^{128-127}) = -1.010 \times 2^1 = -(10.10)_2 = -(2.5)_{10}$

Facultad
de
Informática

Sección 5.3.2. Digital Design and Computer Architecture. ARM edition. S. Harris y D.M. Harris.

Otros tipos de datos

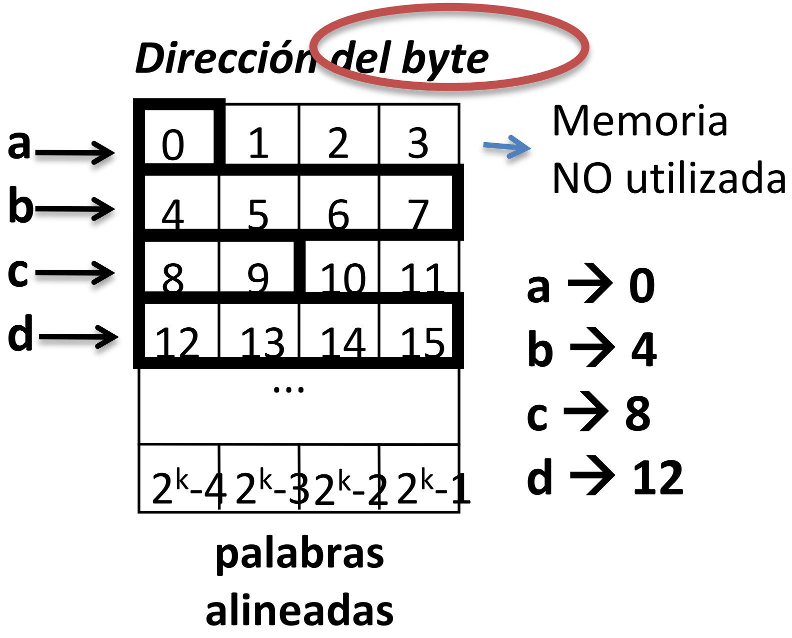
- Punteros
- Arrays
- Cadenas de caracteres
- Estructuras
- Creación de tipos propios usando typedef.

Acceso a estructuras básicas de datos

Tipo C	Número de bits
char	**8**
short	**16**
int	**32**
Los mismo tamaños se aplican para la versión unsigned	

- Ejemplo. Declaramos cuatro variables:
  - char a; int b;

  short int c; int d;



a → 0

b → 4

c → 8

d → 12