

SISTEMAS DE NUMERACIÓN

- Sistema egipcio
 - Símbolos para representar cantidades
 - Sistema aditivo

						
1	10	100	1.000	10.000	100.000	1.000.000



= 42



= 2235

- Sistema romano

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

$$\text{III} = 1 + 1 + 1 = 3$$

$$\text{XL} = 50 - 10 = 40$$

$$\text{MCM} = 1000 + (1000 - 100) = 1900$$

Representación posicional de los números

- Cada símbolo cambia su valor según su posición (dígito)
- El número se representa:

$$.....d_7d_6d_5d_4d_3d_2d_1d_0$$

- Y su valor será:

$$N =d_7 \cdot b^7 + d_6 \cdot b^6 + d_5 \cdot b^5 + d_4 \cdot b^4 + d_3 \cdot b^3 + d_2 \cdot b^2 + d_1 \cdot b^1 + d_0$$

- Donde
 - N = valor del número
 - b = valor de la base
 - d_i = valor del dígito i-simo

Bases más utilizadas

- Decimal
 - Base = 10. Dígitos 0 a 9.
 - Ejemplo: $1025_{10} = 1 \cdot 10^3 + 2 \cdot 10 + 5$
- Binario
 - Base 2. Dígitos 0 y 1.
 - Ejemplo: $100111_2 = 1 \cdot 2^5 + 1 \cdot 2^2 + 1 \cdot 2 + 1$
- Hexadecimal
 - Base 16. Dígitos 0 a F
 - Ejemplo: $3F0A_{16} = 3 \cdot 16^3 + 15 \cdot 16^2 + 10$
- Octal
 - Base 8. Dígitos 0 a 7.
 - Ejemplo: $1025_8 = 1 \cdot 8^3 + 2 \cdot 8 + 5 \neq 1025_{10}$

Conversión entre bases

- Conversión a base 10 -> “calcular el valor”

$$100111_2 = 1 \cdot 2^5 + 1 \cdot 2^2 + 1 \cdot 2 + 1 = 39_{10}$$

$$3F0A_{16} = 3 \cdot 16^3 + 15 \cdot 16^2 + 10 = 16138_{10}$$

$$1025_8 = 1 \cdot 8^3 + 2 \cdot 8 + 5 = 533_{10}$$

- Conversión entre bases potencias de 2 -> agrupamientos binarios

- Agrupamientos hexadecimal (4) -> $3FA = \underline{0011} \ \underline{1111} \ \underline{1010} = 1111111010_2$

- Agrupamientos octal (3) -> $1111111010_2 = \underline{001} \ \underline{111} \ \underline{111} \ \underline{010} = 1772_8$

- Conversión de decimal a cualquier base -> dividir por la base

$$153_{10} = 10011001_2$$

$$1869_{10} = 74D_{16}$$

$$\begin{array}{r}
 1869 \overline{) 16} \\
 \underline{13} \\
 116 \overline{) 16} \\
 \underline{4} \\
 7 \overline{) 16} \\
 \underline{0} \\
 \text{MSB } 7
 \end{array}$$

↑
D

$$\begin{array}{r}
 153 \overline{) 2} \\
 \underline{1} \\
 76 \overline{) 2} \\
 \underline{0} \\
 38 \overline{) 2} \\
 \underline{0} \\
 19 \overline{) 2} \\
 \underline{1} \\
 9 \overline{) 2} \\
 \underline{1} \\
 4 \overline{) 2} \\
 \underline{0} \\
 2 \overline{) 2} \\
 \underline{0} \\
 1 \overline{) 2} \\
 \underline{1} \\
 0
 \end{array}$$

MSB 1

Aritmética binaria

- Suma binaria

$$\begin{array}{r} \textcircled{1} \qquad \qquad \textcircled{1} \quad \rightarrow \text{acarreo} \\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1 \\ + 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0 \\ \hline \textcircled{1} \leftarrow \text{sobrepasamiento} \ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \end{array}$$

- Resta binaria (poco usada)

$$\begin{array}{r} \textcircled{1} \qquad \qquad \textcircled{1} \quad \rightarrow \text{préstamo} \\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \\ - 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0 \\ \hline 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1 \end{array}$$

- Multiplicación binaria

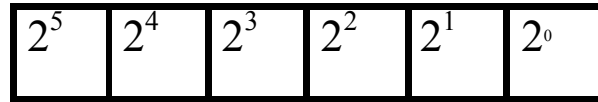
$$\begin{array}{r}
 1100 \\
 X 1011 \\
 \hline
 1100 \\
 1100 \\
 0000 \\
 1100 \\
 \hline
 100000100
 \end{array}$$

- División binaria

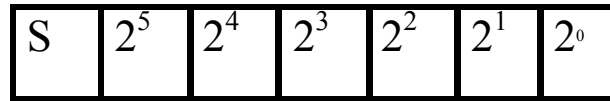
$$\begin{array}{r}
 101111 \overline{) 100} \\
 \underline{- 100} \\
 111 \\
 \underline{- 100} \\
 111 \\
 \underline{- 100} \\
 11
 \end{array}$$

Representación de números enteros

- ◆ Binario puro (sin signo)



- ◆ Módulo y signo



- ◆ Complemento a 1:

- Los números positivos igual que en binario puro (MSB = 0)
- Los números negativos invierten todos sus bits (MSB = 1)

- ◆ Complemento a 2:

- Se forma el complemento a 1 del número
- Se le suma 1
- También va a coincidir que el MSB es 0 para los números positivos y 1 para los negativos

- ◆ Exceso Z

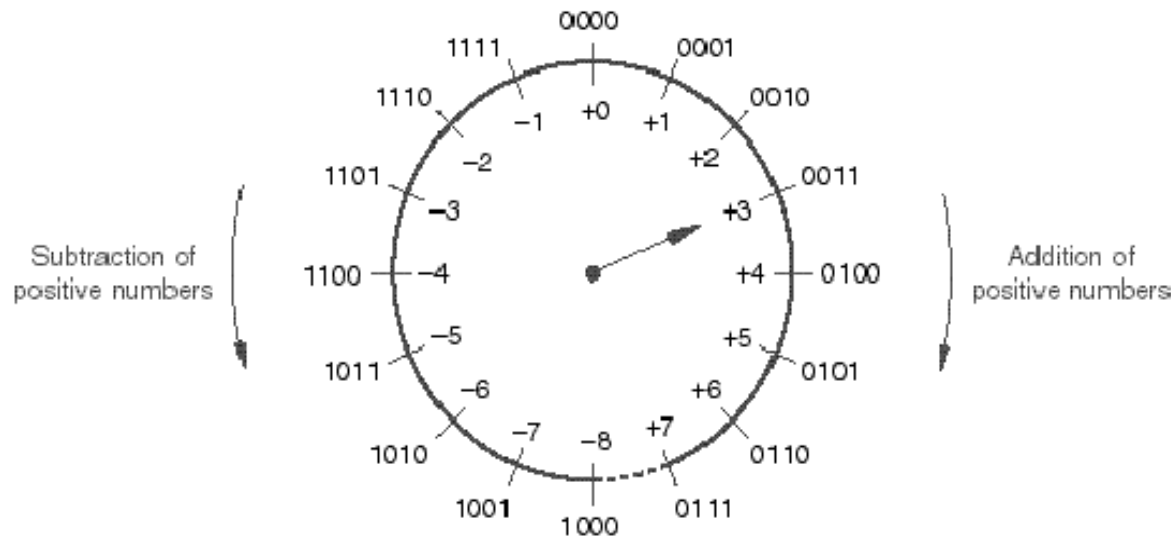
- Se obtiene sumando Z al número en binario puro
- El código exceso $2^{(n-1)}$ es igual que el complemento a 2 pero invirtiendo el MSB

- Ejemplo de representación con 4 bits

<i>Decimal</i>	<i>Two's Complement</i>	<i>Ones' Complement</i>	<i>Signed Magnitude</i>	<i>Excess 2^{n-1}</i>
-8	1000	—	—	0000
-7	1001	1000	1111	0001
-6	1010	1001	1110	0010
-5	1011	1010	1101	0011
-4	1100	1011	1100	0100
-3	1101	1100	1011	0101
-2	1110	1101	1010	0110
-1	1111	1110	1001	0111
0	0000	1111 or 0000	1000 or 0000	1000
1	0001	0001	0001	1001
2	0010	0010	0010	1010
3	0011	0011	0011	1011
4	0100	0100	0100	1100
5	0101	0101	0101	1101
6	0110	0110	0110	1110
7	0111	0111	0111	1111

Suma y resta en complemento a dos

- La suma se hace normalmente
- Para la resta, se complementa a dos el sustraendo y se suma: $A - B = A + (-B)$
- n bits $\Rightarrow$ posibilidad de representar números entre -2^{n-1} y $+2^{n-1}-1$
- Consideraciones sobre el desbordamiento (en la suma):
 - Si A y B son positivos, $A+B$ es positivo. Si hay desbordamiento $\Rightarrow$ error
 - Si A y B son negativos, $A+B$ es negativo. Si hay desbordamiento $\Rightarrow$ error
 - Si A y B son de distinto signo, si hay desbordamiento se ignora.



Errores en la suma

- Suma de dos números positivos

$$125_{10} = 0111\ 1101$$

$$58_{10} = 0011\ 1010$$

$$\begin{array}{r} 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ + \\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \\ \hline \boxed{1}0\ 1\ 1\ 0\ 1\ 1\ 1 \end{array}$$

Signo incorrecto →

Resultado incorrecto → - 73

- Suma de dos números negativos

$$125_{10} = 0111\ 1101 \text{ Complemento a 2} = 1000\ 0010 + 1 = 1000\ 0011$$

$$58_{10} = 0011\ 1010 \text{ Complemento a 2} = 1100\ 0101 + 1 = 1100\ 0110$$

$$\begin{array}{r} 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ + \\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0 \\ \hline \boxed{1}\boxed{0}1\ 0\ 0\ 1\ 0\ 0\ 1 \end{array}$$

Acarreo que se descarta →

Signo incorrecto →

Resultado incorrecto → + 73

Otras operaciones

- Multiplicación y división en C-2 -> se hacen tratando de forma separada la magnitud y el signo
- Suma en complemento a 1
 - Si hay acarreo del MSB hay que sumarlo al resultado.
 - Igual que en C-2, puede haber error si los dos números tienen el mismo signo.
- Multiplicación y división por potencias de 2
 - Multiplicación = desplazamiento a izquierda
 - División = desplazamiento a derecha

Códigos para números decimales

- Representar números decimales en binario
- Cada dígito se representa con 4 (o más) bits, $59_{10} = 0101\ 1001_{\text{BCD}}$
- Códigos más típicos
 - BCD natural -> los 9 dígitos representados en binario
 - BCD exceso 3 -> a partir de BCD natural, sumando 3.
 - BCD aiken -> los 4 dígitos binarios tienen pesos 2-4-2-1 en vez de 8-4-2-1

DECI MAL	BCD nat 8421	Aitken 2421	5421	BCD Exceso 3	BIQUI- NARIO	1 DE 10
0	0000	0000	0000	0011	01 00001	1000000000
1	0001	0001	0001	0100	01 00010	0100000000
2	0010	0010	0010	0101	01 00100	0010000000
3	0011	0011	0011	0110	01 01000	0001000000
4	0100	0100	0100	0111	01 10000	0000100000
5	0101	1011	1000	1000	10 00001	0000010000
6	0110	1100	1001	1001	10 00010	0000001000
7	0111	1101	1010	1010	10 00100	0000000100
8	1000	1110	1011	1011	10 01000	0000000010
9	1001	1111	1100	1100	10 10000	0000000001

Otros códigos

- Códigos Gray
 - El código gray se forma a partir de binario con $G_i = B_i \oplus B_{i+1}$
 - B_i = bit i-simo del código en binario
 - G_i = bit i-simo del código en gray.
- Códigos Johnson
 - Codifica $2n$ combinaciones con n bits
- Ambos códigos son cíclicos, esto quiere decir que:
 - Las diferencias entre una combinación y la siguiente es de un solo bit
 - La diferencia entre la primera y la última combinación es de un solo bit

DECIMAL	BINARIO NATURAL	GRAY 3 BITS	GRAY 4 BITS	JOHNSON 4 BITS	JOHNSON 8 BITS	1 de 10
0	0000	000	0000	0000	00000000	0000000001
1	0001	001	0001	0001	00000001	0000000010
2	0010	011	0011	0011	00000011	0000000100
3	0011	010	0010	0111	00000111	0000001000
4	0100	110	0110	1111	00001111	0000010000
5	0101	111	0111	1110	00011111	0000100000
6	0110	101	0101	1100	00111111	0001000000
7	0111	<u>100</u>	0100	1000	01111111	0010000000
8	1000		1100		11111111	0100000000
9	1001		1101		11111110	1000000000
10	1010		1111		11111100	
11	1011		1110		11111000	
12	1100		1010		11110000	
13	1101		1011		11100000	
14	1110		1001		11000000	
15	1111		<u>1000</u>		10000000	

REPRESENTACIÓN DE NÚMEROS DECIMALES EN COMA FIJA

- Formato:

2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
-------	-------	-------	-------	-------	-------	----------	----------	----------	----------	----------	----------

Se puede utilizar:

- Representación en binario puro (sin signo)
- Representación en módulo y signo
- Representación en complemento a 1
- Representación en complemento a 2

• Problemas:

La suma de dos números puede provocar sobrepasamiento (overflow), al poder llegar hasta $n+1$ bits de longitud

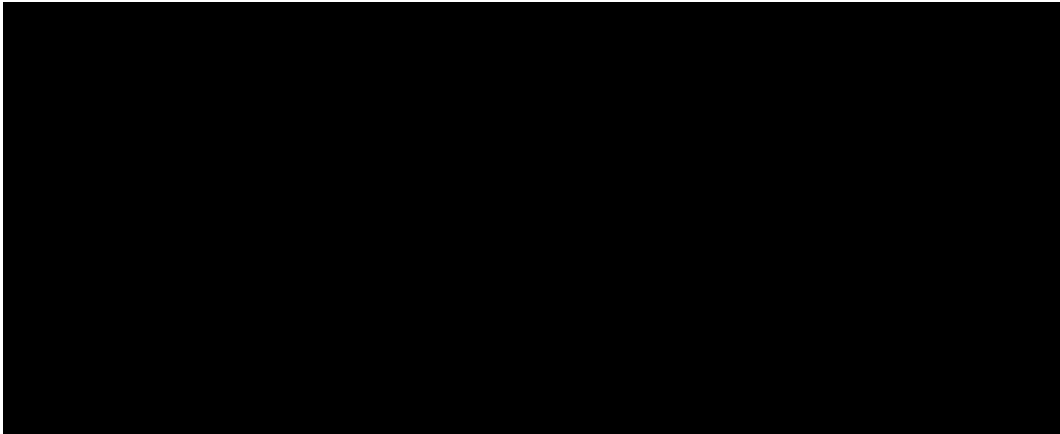
La multiplicación de dos números puede llegar hasta $2n$ bits de longitud

Si se quieren hacer cálculos con la máxima precisión es necesario ir cambiando la coma de posición en cada operación

No permite trabajar a la vez con números muy grandes y muy pequeños

REPRESENTACIÓN DE DECIMALES EN COMA FLOTANTE

Formato: $n^o = mantisa \cdot base^{exponente}$



Mantisa fraccionaria normalizada, sin ceros a la derecha Se representa sin signo y sin el primer bit que es siempre igual a 1.

La base utilizada es 2

El exponente se representa en exceso 127 para simple precisión y exceso 1023 para doble precisión

El cero se representa por E=0, M=0

Cuando el resultado no tiene sentido (error) se representa por E=255, M=0

Para representar números pequeños (ceranos a 0) se utiliza un formato no normalizado. Se indica con E=0, M distinto de 0. En ese caso el exponente será -126 y la mantisa será 0,M en vez de 1,M

Para simple precisión: $N = (-1)^S \cdot 2^{E-127} \cdot (1.M)$

Para doble precisión: $N = (-1)^S \cdot 2^{E-1023} \cdot (1.M)$

Códigos alfanuméricos

Tabla de códigos ASCII

CÓDIGO CARACTER			CÓDIGO CARACTER			CÓDIGO CARACTER			CÓDIGO CARACTER					
0	0h	NULL	29	1Dh	GS	58	3Ah	:	87	57h	W	116	74h	t
1	1h	SOH	30	1Eh	RS	59	3Bh	;	88	58h	X	117	75h	u
2	2h	STX	31	1Fh	US	60	3Ch	<	89	59h	Y	118	76h	v
3	3h	ETX	32	20h	SP	61	3Dh	=	90	5Ah	Z	119	77h	w
4	4h	EOT	33	21h	!	62	3Eh	>	91	5Bh	[	120	78h	x
5	5h	ENQ	34	22h	"	63	3Fh	?	92	5Ch	\	121	79h	y
6	6h	ACK	35	23h	#	64	40h	@	93	5Dh	]	122	7Ah	z
7	7h	BEL	36	24h	\$	65	41h	A	94	5Eh	^	123	7Bh	{
8	8h	BS	37	25h	%	66	42h	B	95	5Fh	_	124	7Ch	
9	9h	HT	38	26h	&	67	43h	C	96	60h	`	125	7Dh	}
10	Ah	LF	39	27h	'	68	44h	D	97	61h	a	126	7Eh	~
11	Bh	VT	40	28h	(	69	45h	E	98	62h	b	127	7Fh	DEL
12	Ch	FF	41	29h	)	70	46h	F	99	63h	c			
13	Dh	CR	42	2Ah	*	71	47h	G	100	64h	d			
14	Eh	S0	43	2Bh	+	72	48h	H	101	65h	e			
15	Fh	S1	44	2Ch	,	73	49h	I	102	66h	f			
16	10h	DLE	45	2Dh	-	74	4Ah	J	103	67h	g			
17	11h	DC1	46	2Eh	.	75	4Bh	K	104	68h	h			
18	12h	DC2	47	2Fh	/	76	4Ch	L	105	69h	i			
19	13h	DC3	48	30h	0	77	4Dh	M	106	6Ah	j			
20	14h	DC4	49	31h	1	78	4Eh	N	107	6Bh	k			
21	15h	NAK	50	32h	2	79	4Fh	O	108	6Ch	l			
22	16h	SYN	51	33h	3	80	50h	P	109	6Dh	m			
23	17h	ETB	52	34h	4	81	51h	Q	110	6Eh	n			
24	18h	CAN	53	35h	5	82	52h	R	111	6Fh	o			
25	19h	EM	54	36h	6	83	53h	S	112	70h	p			
26	1Ah	SUB	55	37h	7	84	54h	T	113	71h	q			
27	1Bh	ESC	56	38h	8	85	55h	U	114	72h	r			
28	1Ch	FS	57	39h	9	86	56h	V	115	73h	s			