

Representación de la Información (2)

Alberto Ruiz Cristina

Introducción

- Sabemos que el ordenador utiliza el sistema binario, pero ¿cómo se almacenan los números?
- Normalmente se asignan n bits para representar un número.
- Esos n bits son la longitud de una palabra, es decir, la información que puede procesarse de una vez

Introducción

- El tamaño de palabra es muy importante en un equipo, ya que afecta...
 - A la representación de los números
 - Al tamaño idóneo de los registros
 - A la longitud de las direcciones de memoria
 - A la longitud de las instrucciones
- Hablamos de palabra, media palabra, doble palabra...
- Cuando los caracteres se almacenaban con 6 bits, los tamaños de palabra solían ser múltiplos de 6. Ahora lo son de 8 (16, 32, 64...)

Representación de los números

- Vamos a estudiar algunos convenios de representación de números enteros y racionales en binario
- **Rango de representación:** intervalo comprendido entre el menor y el mayor número representable.
 - Con n bits podremos representar 2^n combinaciones y por tanto números distintos.
 - Ejemplo: con 2 bits puedo representar $2^2=4$ números distintos (00, 01, 10, 11)

Coma fija sin signo

- Coma fija: el lugar donde se sitúa el punto de referencia es fijo. Para números enteros estará a la derecha del todo, y no se representa
- Esta representación coincide con el binario puro
- Ejemplo: $27 = 00011011$
- En todos los ejemplos manejaremos tamaño de palabra 8
- Rango de representación: $0, 2^n - 1$

Coma fija con signo

- Lo llamamos “signo-magnitud”. El primer bit indica si el número que va a continuación es positivo (0) o negativo (1)
- Ejemplo:
 - 27 = 00011011
 - 27 = 10011011
- Perdemos un bit para representar números, así que el rango será: $-(2^{n-1}-1), 2^{n-1}-1$

Complemento a la base menos 1

- Es el complemento a la base menos 1
- Ejemplo en base 10:
 - El Ca10 era:
 - $72 \rightarrow 100 - 72 = 28$
 - $28 \rightarrow 100 - 28 = 72$
 - El Ca9 es:
 - $72 \rightarrow 100 - 1 - 72 = 27$
 - $27 \rightarrow 100 - 1 - 27 = 72$

Complemento a 1

- Ejemplo en base 2:
 - El Ca2 era:
 - $101 \rightarrow 1000 - 101 = 11$
 - $011 \rightarrow 1000 - 011 = 101$
 - El Ca1 es:
 - $101 \rightarrow 1000 - 1 - 101 = 10$
 - $010 \rightarrow 1000 - 1 - 010 = 101$
- En este caso hay doble representación del cero:
 - $1000 = 0000 = 0$

Complemento a 1

- En este caso, si hay acarreo, se suma al resultado:
- $A - B = A + \text{Complemento}(B) + \text{acarreo}$
- Ejemplo en base 10:
 - ¿ $72 - 43$?
 - Complemento a 9 de 43 $\rightarrow 100 - 1 - 43 = 56$
 - $72 - 43 = 72 + (-43) = 72 + 56 = 128 \rightarrow 28 + 1 = 29$
- Ejemplo en base 2:
 - ¿ $111 - 101$?
 - Complemento a 1 de 100 $\rightarrow 1000 - 1 - 101 = 10$
 - $111 - 101 = 111 + 010 = 1001 \rightarrow 001 + 1 = 10$

Cálculo del Complemento a 1

- Método simplificado:
 - Invierto todos los bits del número
 - $1001010 \rightleftarrows 0110101$
- Ca1 frente a Ca2:
 - En Ca1 hay doble representación del 0
 - El Ca1 se calcula más fácilmente
 - Al operar con Ca1 se tiene en cuenta el acarreo
- Rango del Ca1:
 - $-2^{n-1}-1, 2^{n-1}-1$

	Binario	S.Magn.	Ca1	Ca2
000	0	0	0	0
001	1	1	1	1
010	2	2	2	2
011	3	3	3	3
100	4	-0	-3	-4
101	5	-1	-2	-3
110	6	-2	-1	-2
111	7	-3	-0	-1

Números reales

- La coma fija exige reservar un número determinado de dígitos para la parte entera y para la fraccionaria
- El resultado es desaprovechar espacio en ocasiones
- Además, el rango de valores es muy reducido
- A cambio, los circuitos son más simples

Efecto de mover la coma

- ¿Qué le ocurre a un número al mover la coma?
- En base 10:
 - 12.25
 - 122.5 \rightarrow
 - 1.225 \leftarrow
- En base 2:
 - 1100.01 (12.25)
 - 11000.1 (24.5) \rightarrow
 - 110.001 (6.125) \leftarrow

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 19 INF2

Efecto de mover la coma

- ¿Qué le ocurre a un número al mover la coma?
- En base 10:
 - 12.25
 - 122.5 \rightarrow *10
 - 1.225 \leftarrow /10
- En base 2:
 - 1100.01 (12.25)
 - 11000.1 (24.5) \rightarrow *2
 - 110.001 (6.125) \leftarrow /2

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 20 INF2

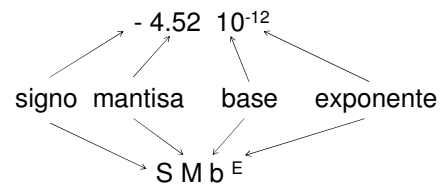
Coma flotante

- Se corresponde con lo que se llama “notación científica”
- Se usa para expresar números muy pequeños o muy grandes
- Ejemplo:
- $4\,520\,000\,000\,000 = 4.52 \cdot 10^{12}$
- $0,000\,000\,000\,004\,52 = 4.52 \cdot 10^{-12}$

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 21 INF2

Coma flotante

- Componentes de la notación en coma flotante:



ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 22 INF2

Coma flotante

- Normalmente, un número se almacena como 2 números en coma fija:
 - Uno representa el signo del número y la mantisa
 - El otro representa el exponente
- La base no se representa, se da por sabida
- La **precisión** de los cálculos viene dada por el tamaño de la mantisa
- El **rango** de representación viene dado por el tamaño del exponente

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 23 INF2

Normalización

- ¿Cómo se escribe el número 6.25 en binario usando notación en coma flotante?
 - $0110.01 \cdot 2^{000}$
 - $011.001 \cdot 2^{001}$
 - $01.1001 \cdot 2^{010}$
 - $0.11001 \cdot 2^{011}$
- Para evitar representaciones múltiples, se adopta el convenio de situar la coma en un lugar fijo.

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 24 INF2

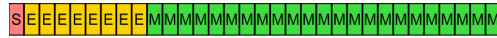
Normalización

- ¿Dónde ponemos la coma? A la izquierda del primer dígito significativo
- $0110.01 \cdot 2^{000} \rightarrow 0.11001 \cdot 2^{011}$
- Ventajas de coma flotante normalizada:
 - Dos números: 4132.524 y 0.03423
 - En coma fija, si reservamos 4 bits para la parte entera y 2 para la fraccionaria:
 - 4132.52 y 0000.03
 - En coma flotante normalizada:
 - $0.413252 \cdot 10^4$ y $0.342300 \cdot 10^{-1}$

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 25 INF2

Estándar IEEE 754

- IEEE: Institute of Electrical and Electronics Engineers
 - 1 bit de signo de la mantisa
 - 8 bits del exponente
 - 23 bits de mantisa



- Total: 32 bits.
- Hay versiones distintas del estándar de 64 bits, aumentando el tamaño de la mantisa

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 26 INF2

IEEE 754

- La mantisa está expresada en signo-magnitud y normalizada: se asume que la coma está a la derecha del primer bit significativo (que valga 1)
- Es decir, el primer bit de la mantisa normalizada siempre será 1
- Por tanto no lo almacenamos, y que el circuito encargado de operar lo restablezca sobre la marcha

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 27 INF2

IEEE 754

- Ejemplo:
 - Número: $0110.01 \cdot 2^{000}$
 - Normalizamos: $1.1001 \cdot 2^{010}$
 - ¿Para qué almacenar el primer 1 si siempre va a estar ahí? $.1001 \cdot 2^{010}$
- Es decir, que todos los números expresados en IEEE 754 tienen la forma 1.xxxx, pero el 1 no se representa

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 28 INF2

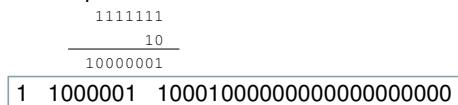
IEEE 754

- Exponente: se almacena en exceso a 127
 - Con 8 bits mi rango es 0, 255
- No voy a almacenar el exponente real E, sino E+127
- Para calcular el auténtico E, restaré 127 a lo almacenado
 - Por tanto el rango real es -127, 128
- Ventaja: el signo va implícito. Almacenar el exponente en positivo facilita algunas operaciones

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 29 INF2

IEEE 754

- Ejemplo: $-6.125_{(10)} = 110.001$
- Normalizamos:
 - $1.10001 \cdot 2^{010}$
 - El bit de signo de la mantisa será 1
 - El exponente será $2+127 =$



ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 30 INF2

IEEE 754

- $-118.625_{(10)} = 1110110.101$
- Movemos la coma: $1.110110101 \cdot 2^6$
- $127 + 6 = 133 = 1000101$
(¡quizá te sea más fácil sumar en binario!)

1 1000101 1101101010000000000000

- En general, la coma flotante complica los circuitos de operación, pero a cambio se ahorra en capacidad de almacenamiento

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 31 INF2

IEEE 754

- ¿Y al revés?

1 1000001 1000100000000000000000

Signo: negativo

Mantisa: 1.10001

Exponente: $1000001 - 01111111 = 10$

1.10001 2^{10}

Es decir, $110.001 = -6.125$

¡No olvidéis el signo!

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 32 INF2

IEEE 754

- Quiero representar el 1 $\rightarrow 1.0 \cdot 2^0$
- El exponente será $0 + 127 = 127$

0 01111111 0000000000000000000000

- Quiero representar el 0
- La mantisa siempre será 1.0, no se puede representar. Por convenio, si todos los números están a 0 \rightarrow Es el 0

0 00000000 0000000000000000000000

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 33 INF2

BCD

- Binary Coded Decimal, decimales codificados en binario
- Se usa para displays, calculadoras y aparatos en los que es más sencillo trabajar en decimal
- Necesitaré 4 bits para poder representar del 0 al 9, y sobrarán algunos valores
- No es eficiente ni para almacenar ni para calcular, pero en algunos casos interesa

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 34 INF2

BCD

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Expresar 754 en BCD:

0111 0101 0100

¿Qué número es 01111001?

7 9

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 35 INF2

Código ASCII

- **American Standard Code for Information Interchange**
- *Código Norteamericano Estándar para el Intercambio de Información*
- Creado en 1963
- Usa 7 bits (128 valores)
- El código ASCII extendido (8 bits) permite dos "mapas" de caracteres
 - Primer bit 0 \rightarrow Los otros 7 son el ASCII estándar
 - Primer bit 1 \rightarrow Los otros 7 son el ASCII extendido

ARC – IES Virgen de la Paloma – SIMR DAI Tarde 0809 – Página 36 INF2

Código ASCII

1	!	33	!	65	A	97	a	129	ï	161	í	193	á	225	á
2	"	34	"	66	B	98	b	130	ï	162	î	194	â	226	â
3	#	35	#	67	C	99	c	131	ï	163	ï	195	ã	227	ã
4	\$	36	\$	68	D	100	d	132	ï	164	ï	196	ä	228	ä
5	%	37	%	69	E	101	e	133	ï	165	ï	197	å	229	å
6	&	38	&	70	F	102	f	134	ï	166	ï	198	æ	230	æ
7	'	39	'	71	G	103	g	135	ï	167	ï	199	ç	231	ç
8	(40	(72	H	104	h	136	ï	168	ï	200	è	232	è
9)	41)	73	I	105	i	137	ï	169	ï	201	é	233	é
10	*	42	*	74	J	106	j	138	ï	170	ï	202	ê	234	ê
11	+	43	+	75	K	107	k	139	ï	171	ï	203	ë	235	ë
12	,	44	,	76	L	108	l	140	ï	172	ï	204	ì	236	ì
13	-	45	-	77	M	109	m	141	ï	173	ï	205	í	237	í
14	.	46	.	78	N	110	n	142	ï	174	ï	206	î	238	î
15	/	47	/	79	O	111	o	143	ï	175	ï	207	ï	239	ï
16	:	48	:	80	P	112	p	144	ï	176	ï	208	ð	240	ð
17	;	49	;	81	Q	113	q	145	ï	177	ï	209	ñ	241	ñ
18	<	50	<	82	R	114	r	146	ï	178	ï	210	o	242	o
19	=	51	=	83	S	115	s	147	ï	179	ï	211	ö	243	ö
20	!	52	!	84	T	116	t	148	ï	180	ï	212	ó	244	ó
21	@	53	@	85	U	117	u	149	ï	181	ï	213	ô	245	ô
22	#	54	#	86	V	118	v	150	ï	182	ï	214	õ	246	õ
23	\$	55	\$	87	W	119	w	151	ï	183	ï	215	ö	247	ö
24	%	56	%	88	X	120	x	152	ï	184	ï	216	ø	248	ø
25	&	57	&	89	Y	121	y	153	ï	185	ï	217	ù	249	ù
26	'	58	'	90	Z	122	z	154	ï	186	ï	218	ú	250	ú
27	(59	(91	[123	[155	ï	187	ï	219	û	251	û
28)	60	<	92	\	124	\	156	ï	188	ï	220	ü	252	ü
29	*	61	=	93]	125]	157	ï	189	ï	221	ý	253	ý
30	+	62	>	94	^	126	^	158	ï	190	ï	222	ÿ	254	ÿ
31	,	63	?	95	_	127	_	159	ï	191	ï	223	ÿ	255	ÿ
32	;	64	@	96	~	128	~	160	ï	192	ï	224	ÿ	256	ÿ

ARC - IES Virgen de la Paloma - SIMR DAI Tarde 0809 - Página 37 INF2

En resumen...

- Lo que hay que saber:
 - Entender y expresar números enteros en binario puro, signo-magnitud, complemento a 2 y complemento a 1
 - Entender y expresar números reales en coma flotante (formato IEEE 754)
 - Entender y expresar números en BCD
 - Conocer los rangos de representación de todos estos sistemas de representación

ARC - IES Virgen de la Paloma - SIMR DAI Tarde 0809 - Página 38 INF2