Organización de Computadoras

Clase 2

Temas de Clase

- Representación de datos
  - Números con signo
- Operaciones aritméticas
- Banderas de condición
- Representación de datos alfanuméricos

Temas de Clase

- Representación de datos
  - Números con signo
    - Binario con signo (BCS) o signo y módulo
    - Complemento a 1 (o a la base reducida)
    - Complemento a 2 (o a la base)
    - Exceso

Repaso de clase 1
Números enteros sin signo

- Con n bits podemos representar 2^n números distintos.
- El rango obtenido va desde 0 hasta 2^n-1

Repaso de clase 1
Números enteros sin signo

Ej: si usamos 3 bits

- tenemos $2^3 = 8$ números distintos
- el rango es $[0,2^3-1]= [0,7]$

Binario	Decimal
000	0
001	1
…	…
111	7

Repaso de clase 1
Números enteros sin signo

Ej: en 8 bits

- ➢ tenemos $2^8 = 256$ números distintos
- ➢ el rango es $[0,2^8-1]= [0,255]$

Binario	Decimal
00000000	0
00000001	1
…	…
01111111	127
10000000	128
…	...
11111111	255

Repaso de clase 1
Números enteros sin signo

RECORDAR:

La cantidad de representaciones distintas depende del número de bits

- Si usamos n bits, entonces:

$$N^{o}s \text{ distintos} = 2^{n}$$

Números enteros **con** signo

- **Para representar números con signo, deberemos asignar algunas representaciones para números positivos y otras para los negativos.**

Si usamos n bits, entonces:

$$\text{N}^{\text{o}}\text{s distintos} = 2^n$$

Números enteros **con** signo

- **Para representar números con signo, deberemos asignar algunas representaciones para números positivos y otras para los negativos.**

Si usamos n bits, entonces:

Idealmente la mitad serán positivos y la otra mitad negativos

Nºs distintos = 2^n

Números enteros **con** signo

- **Para representar números con signo, deberemos asignar algunas representaciones para números positivos y otras para los negativos.**

Vamos a estudiar 4 formas de realizar estas asignaciones, la que proponen los sistemas de representación BCS, Ca1, Ca2 y Exceso

Idealmente la mitad serán positivos y la otra mitad negativos

Nºs distintos = 2^n

Representación en BCS

- Con n bits, 1 bit representa al signo y n-1 bits a la magnitud o módulo.

n-1	n-2	0
SIGNO	MAGNITUD	

- El bit n-1 (extremo izquierdo) representa sólo al signo
- Los bits 0 a n-2 la magnitud

BCS o Binario con signo

- Un 0 en el bit de signo indica que el número es positivo
- Un 1 en el bit de signo indica que el número es negativo
- Los bits 0 $\longrightarrow$ n-2 representan el valor absoluto en binario
- El rango: $-(2^{n-1} - 1) \longrightarrow +(2^{n-1} - 1)$ con 2 ceros

Binario con signo (2)

- Ejemplos

$$+32_{10} = \underbrace{0010000}_{32}0 \qquad -32_{10} = \underbrace{1010000}_{32}0$$

$+7_{10} =$ 00000111 $\qquad - \ 7_{10} =$ 10000111

$+41_{10} =$ 00101001 $\qquad -41_{10} =$ 10101001

BCS - Binario con signo (3)

Ejemplo: BCS con 8 bits

	Binario	Decimal
Números Positivos	00000000	+0
	00000001	+1
	...	...
	01111111	$+127= +(2^{n-1}-1)$
Números Negativos	10000000	-0
	10000001	-1
	...	...
	11111111	$-127= -(2^{n-1}-1)$

Binario con signo (4)

Ejemplo: BCS en 3 bits

	Binario	Decimal
Números Positivos	000	+0
	001	+1
	010	+2
	011	$+3 = +(2^2-1)$
Números Negativos	100	-0
	101	-1
	110	-2
	111	$-3 = -(2^2-1)$

Resumen: BCS

- ✓ El intervalo es simétrico
- ✓ El bit de la izquierda sólo indica el signo
  - Es cero (0) en los positivos
  - Es uno (1) en los negativos
- ✓ Hay dos ceros
- ✓ Números distintos: 2^n

Técnica de Complementos

- El complemento a un número N de un número A (A menor que N) es igual a la cantidad que le falta a A para ser N

Complemento a N de A = N - A

- El complemento a un número N del número (N-A) es igual a A.

Complemento a N de (N-A) = N - (N-A) = A

Técnica de Complementos

- El complemento a un nú
  número A (A menor que
  cantidad que le falta a A par ser N

Ejemplo:
El complemento a 9 del 3 es 6

Complemento a N de A = N - A

- El complemento a un número N del número (N-A) es igual a A.

Complemento a N de (N

Ejemplo:
El complemento a 9 del 6 es 3

Técnica de Complementos

- El complemento a un núr número A (A menor que c ar ser N

Ejemplo:
El complemento a 9 del 3 es 6

Podríamos decir que la definición de complemento nos permite "vincular" pares de números.
En nuestro ejemplo de complemento a 9: 3 y 6, 2 y 7, etc.

A = N - A

- E ero N del número (N-A) es igual a A.

Complemento a N de (N

Ejemplo:
El complemento a 9 del 6 es 3

Técnica de Complementos (2)

En un sistema con n dígitos llamamos:

Complemento a la base disminuida si $N = base^n - 1$

(En sistema binario se llama **Complemento a 1** ó **Ca1**)

(si n=8, $N = 2^8 - 1 = 11111111$)

Complemento a la base cuando $N = base^n$

(En sistema binario se llama **Complemento a 2** ó **Ca2**)

(si n=8, $N = 2^8 = 100000000$)

Técnica de Complementos (3)

Ej: El complemento a 1 en 8 bits de 01010100 es:

```
     11111111
-    01010100
  ____________
     10101011     =>Ca1(01010100)=10101011
```

Observar que, el Ca1 en 8 bits de 10101011 es

```
     11111111
-    10101011
  ____________
     01010100     =>Ca1(10101011)=01010100
```

- Cada uno de los resultados es el Ca1 del otro!!
- Donde uno tiene 0s el otro tiene 1s y viceversa.

Técnica de Complementos (3)

Ej: El complemento a 1 en 8 bits de 01010100 es:

```
     11111111
-    01010100
  ____________
     10101011      =>Ca1(01010100)=10101011
```

Observar que, el Ca1 en 8 bits

```
     11111111
-    10101011
  ____________
     01010100      =>Ca1(10101011)=01010100
```

No hace falta que hagamos cuentas para encontrar el Ca1 de un número!!!! Simplemente invertimos los bits, donde hay 0 ponemos 1 y viceversa

- Cada uno de los resultados es el Ca1 del otro!!
- Donde uno tiene 0s el otro tiene 1s y viceversa.

Técnica de Complementos (4)

Ej: El complemento a 2 en 8 bits de 01010100 es:

```
    100000000
-    01010100
_____________
     10101100     =>Ca2(01010100)=10101100
```

Observar que, el Ca2 en 8 bits de 10101100 es

```
    100000000
-    10101100
_____________
     01010100     =>Ca2(10101100)=01010100
```

- ➢ cada uno de los resultados es el Ca2 del otro!!
- ➢ "mirando" desde la derecha coinciden hasta el 1er uno

Técnica de Complementos (4)

Ej: El complemento a 2 en 8 bits de 01010100 es:

```
    100000000
-    01010100
_____________
     10101100     =>Ca2(01010100)=10101100
```

Observar que, el Ca2 en 8 bits

```
    100000000
-    10101100
_____________
     01010100     =>Ca2(10101100)=01010100
```

No hace falta que hagamos cuentas para encontrar el Ca2 de un número!!!! Simplemente copiamos hasta el primer uno desde la derecha y luego invertimos los bits, donde hay 0 ponemos 1 y viceversa

- cada uno de los resultados es el Ca2 del otro!!
- "mirando" desde la derecha coinciden hasta el 1er uno

Representación en Ca1

- Los n bits representan al número

n-1 0

Número

- Información del signo

Ca1

- Si el número es positivo, los n bits tienen la representación binaria del número (como siempre)
- Si el número es negativo, los n bits tienen el Ca1 del binario correspondiente al número opuesto positivo.
  - El Ca1 de un número en base 2 se obtiene invirtiendo todos los bits

Ca1

- Los positivos tienen el bit de la izquierda en cero (0)
- Los negativos lo tienen en uno (1)
- El rango va desde

  $$-(2^{n-1}-1) \text{ a } +(2^{n-1}-1)$$

  con dos ceros

Ca1

- Ejemplos

$+$ $-$

$+32_{10} = 00100000$ $-32_{10} = 11011111$

$+7_{10} = 00000111$ $-7_{10} = 11111000$

$+41_{10} = 00101001$ $-41_{10} = 11010110$

Ca1

Ca1 en 8 bits

	Binario	Decimal
Números Positivos	00000000	+0
	00000001	+1
	...	...
	01111111	$+127= +(2^{n-1}-1)$
Números Negativos	10000000	$-127= -(2^{n-1}-1)$
	10000001	-126
	...	...
	11111111	-0

Ca1

Ca1 en 8 bits

	Binario	Decimal
Números Positivos	00000000	+0
	00000001	+1
	...	...
	01111111	+127= $+(2^{n-1}-1)$
Números Negativos	10000000	-127= $-(2^{n-1}-1)$
	10000001	-126
	...	...
	11111111	-0

Observar que ese “1” vale $-(2^7-1)$= -127

Ca1

Ca1 en 3 bits

	Binario	Decimal
Números Positivos	000	+0
	001	+1
	010	+2
	011	+3= $+(2^2-1)$
Números Negativos	100	-3= $-(2^2-1)$
	101	-2
	110	-1
	111	-0

Ca1

Ca1 en 3 bits

	Binario	Decimal
Números Positivos	000	+0
	001	+1
	010	+2
	011	$+3= +(2^2-1)$
Números Negativos	100	$-3= -(2^2-1)$
	101	-2
	110	-1
	111	-0

Observar que ese "1" vale $-(2^2-1)= -3$

Ca1

Dada una cadena de bits ¿qué número decimal representa si lo interpretamos en Ca1?

- Cuando es positivo:

$$01100000 = 1 \times 2^6 + 1 \times 2^5 = 64+32= 96$$

Como siempre

Ca1

- Cuando es negativo, puedo hacer dos cosas:
- Ca1 del número y obtengo el positivo

Ej.

11100000 = - 31

Ca1(11100000) → 00011111 = +31

Ca1

- Otro método: el peso que tiene el primer dígito ahora es $-(2^{n-1}-1)$ y el resto de los dígitos con pesos positivos *como siempre*
-

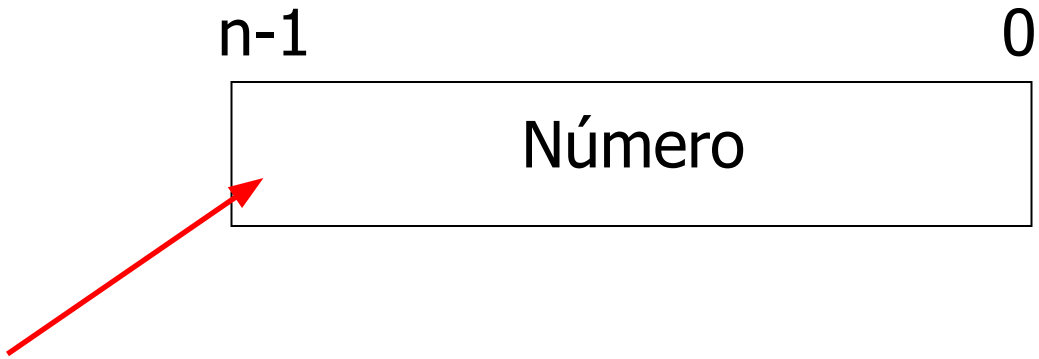
$$11100000 = -1x(2^7-1) + 1x2^6 + 1x2^5 =$$
$$= -127 + 64 + 32 = -31$$

Resumen Ca1

- El intervalo es simétrico
- Los n bits representan al número
- Los positivos tienen 0 en bit izquierdo.
- Los negativos tienen 1 en bit izquierdo.
- Hay dos ceros
- Números distintos 2^n

Representación en Ca2

- Los n bits representan al número



- Información del signo

Representación en Ca2

- Si el número es positivo, los n bits tienen la representación binaria del número (como siempre)
- Si el número es negativo, los n bits tienen el Ca2 del binario correspondiente al opuesto positivo.

Ca2

- El Ca2 de un número se obtiene, “mirando” desde la derecha, escribiendo el número igual hasta el primer “1” inclusive y luego invirtiendo los demás dígitos
- Otra forma es invirtiendo todos los bits (Ca1) y luego sumándole 1.
- Otra forma: *aplicando la definición de Complemento a la base* Ca2(N) = b^n - N

Ca2

- Los positivos tienen el bit de la izquierda en cero (0)
- Los negativos lo tienen en uno (1)
- Hay un solo cero
- El rango es asimétrico y va desde

$$-(2^{n-1}) \text{ a } +(2^{n-1}-1)$$

Ca2

- Ejemplos

  $+32_{10} = 00100000$ ← "mirando" desde la derecha

  $-32_{10} = 11100000$

- ✓ Los dígitos en rojo se copiaron igual
- ✓ Los dígitos en azul se invirtieron

Ca2 (otra forma)

$+32_{10}=00100000$

1111

11011111 invierto todos los bits

\+ 1 le sumo 1

$-32_{10}=11100000$ en Ca2

Ca2 (otra forma)

- $Ca2 = b^n - N^o = 2^8 - 32 = 256-32=224$
- Hagamos la cuenta en base 2

$$
\begin{array}{rcccccccc}
 & 0 & 1 & 1 & & & & & \\
 & \cancel{1} & \cancel{1}\,0 & \cancel{1}\,0 & 1\,0 & 0 & 0 & 0 & 0 \\
- & & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
-32= & & 1 & 1 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

en Ca2

Ca2

Ca2 en 8 bits

	Binario	Decimal
Números Positivos	00000000	+0
	00000001	+1
	…	…
	01111111	$+127 = +(2^{n-1}-1)$
Números Negativos	10000000	$-128 = -(2^{n-1})$
	10000001	-127
	…	...
	11111111	-1

Ca2

Ca2 en 8 bits

	Binario	Decimal
Números Positivos	00000000	+0
	00000001	+1
	...	...
	01111111	$+127 = +(2^{n-1}-1)$
Números Negativos	10000000	$-128 = -(2^{n-1})$
	10000001	-127
	...	...
	11111111	-1

Observar que ese “1” vale $-(2^7) = -128$

Ca2

Ca2 en 3 bits

	Binario	Decimal
Números Positivos	000	+0
	001	+1
	010	+2
	011	$+3= +(2^2-1)$
Números Negativos	100	$-4= -(2^2)$
	101	-3
	110	-2
	111	-1

Ca2

Ca2 en 3 bits

	Binario	Decimal
Números Positivos	000	+0
	001	+1
	010	+2
	011	$+3= +(2^2-1)$
Números Negativos	100	$-4= -(2^2)$
	101	-3
	110	-2
	111	-1

Observar que ese “1” vale $-(2^2)= -4$

Ca2

Dada una cadena de bits ¿qué número decimal representa si lo interpretamos en Ca2?

- Cuando es positivo:

$01100000 = 1 \times 2^6 + 1 \times 2^5 = 64 + 32 = 96$

Como siempre

Ca2

- Cuando es negativo, puedo hacer dos cosas:
- Ca2 el número y obtengo el positivo

Ej.

11100000 = - 32

Ca2(11100000) → 00100000 = +32

Ca2

- Otro método: el peso que tiene el primer dígito ahora es $-(2^{n-1})$ y el resto de los dígitos con pesos positivos *como siempre*

$$11100000 = -1x(2^7) + 1x2^6 + 1x2^5$$
$$= -128 + 64 + 32 = -32$$

Resumen Ca2

- El intervalo es asimétrico, hay un - más
- Los n bits representan al número
- Los positivos tienen 0 en bit izquierdo.
- Los negativos tienen 1 en bit izquierdo.
- Hay un solo cero
- Números distintos 2^n

Técnica del Exceso

Representación en exceso

La representación de un número A es la que corresponde a la SUMA del mismo y un valor constante E (o exceso).

Exceso E de A = A + E

Dado un valor, que representa un número en Exceso, para encontrar el número representado se debe RESTAR el valor del exceso.

A = (Exceso E de A) - E

Técnica del Exceso

Representación en exceso

Ej: supongamos 3 bits, exceso $2^2 = 4$.

el binario 001 representa a : 1-4= -3

el binario 101 representa a : 5-4= 1

el binario 111 representa a : 7-4= 3

el binario 000 representa a : 0-4= -4

- Observar que el bit de la izq no indica el signo como antes! (Queda al revés que en los sitemas anteriores)
- En general en n bits, si usamos exceso 2^{n-1}, el rango es $[-2^{n-1}, 2^{n-1}-1]$

Técnica del Exceso

Exceso 2^2 en 3 bits (exceso 4)

	Binario	Decimal
Números **Negativos**	000	$-4 = -(2^2)$
	001	-3
	010	-2
	011	-1
Números **Positivos**	100	+0
	101	+1
	110	+2
	111	$+3 = +(2^2-1)$

Técnica del Exceso

Exceso 2^{n-1} en 8 bits (exceso 128)

	Binario	Decimal
Números **Negativos**	00000000	$-128=-(2^{n-1})$
	00000001	-127
	...	...
	01111111	-1
Números **Positivos**	10000000	+0
	10000001	+1
	...	...
	11111111	$+127== +(2^{n-1}-1)$

Resumen Exceso

- El intervalo es asimétrico, hay un - más
- Los n bits representan al número
- Los positivos tienen 1 en bit izquierdo.
- Los negativos tienen 0 en bit izquierdo.
- Hay un solo cero
- Números distintos 2^n

Resumen - Representación de enteros. Ejemplo para 3bits.

Binario	BCS	Ca1	Ca2	Exc 4	BSS
000	+0	+0	+0	-4	0
001	+1	+1	+1	-3	1
010	+2	+2	+2	-2	2
011	+3	+3	+3	-1	3
100	-0	-3	-4	0	4
101	-1	-2	-3	+1	5
110	-2	-1	-2	+2	6
111	-3	-0	-1	+3	7

Resumen - Representación de enteros.
Ejemplo para 3bits.

Binario	BCS	Ca1	Ca2	Exc 4	BSS
000	+0	+0			
001	+1	+1			
010	+2	+2	2	-2	2
011	+3	+3	+3	-1	3
100	-0	-3	-4	0	4
101	-1	-2	-	+1	5
110	-2	-1			
111	-3	-0			

Pregunta:
¿A que número decimal representa el binario 100?

Respuesta:
Depende cual sea el sistema de representación que estemos usando.
Si es BCS, al -0
Si es Ca1, al -3
Si es Ca2, al -4
Si es Exc, al 0
Si es BSS, al 4

No

Nuevas Banderas aritméticas

- N (negativo): igual al bit más significativo del resultado.
  - Es 1 si el resultado es negativo
- V (overflow): en 1 indica una condición de fuera de rango (desborde) en Ca2.
  - El resultado no se puede expresar con el número de bits utilizado.

Suma en Ca2

- Para sumar dos números en Ca2 se suman los n bits directamente.
  - Si sumamos dos números + y el resultado es – ó si sumamos dos – y el resultado es + hay overflow, en otro caso no lo hay.
- Si los Nºs son de distinto signo nunca puede haber overflow.

Resta en Ca2

- Para restar dos números en Ca2, se restan los n bits directamente. También se puede Ca2 el sustraendo y transformar la resta en suma.
- Si a un Nº + le restamos un Nº – y el resultado es – ó si a un Nº – le restamos un + y el resultado es +

  hay overflow en la resta.
- Si son del mismo signo nunca hay overflow

Bits de condición para la suma

Operación	NZVC	Ca2	Sin signo
0100	0000	+4	+4
+ 0010		+ +2	+ +2
0110		+6	+6

- Los dos resultados son correctos.

Bits de condición para la suma

Operación	NZVC	Ca2	Sin signo
0101	1010	+5	+5
+ 0111		+ +7	+ +7
1100		V -4	+12

- Ca2 incorrecto, sin signo correcto.

Bits de condición para la suma

Operación	NZVC	Ca2	Sin signo
1101 + 0011 = 1 0000	0101	-3 + +3 = 0	13 + 3 = C 0

- Ca2 correcto, sin signo incorrecto.

Bits de condición para la suma

Operación	NZVC	Ca2	Sin signo
1001	0011	-7	9
+ 1100		+ -4	+ 12
1 ← 0101		V +5	C 5

- Los dos resultados son incorrectos.

Bits de condición para la resta

Operación	NZVC	Ca2	Sin signo
1 → 0101	1001	+5	5
− 0111		− +7	− 7
1110		-2	B 14

- ✓ Ca2 correcto, sin signo incorrecto.

Bits de condición para la resta

Operación	NZVC	Ca2	Sin signo
1001	0010	-7	9
− 0100		− +4	− 4
0101		V +5	5

- Ca2 incorrecto, sin signo correcto.

Suma en BCS

$$\begin{array}{r} 1\,001 \\ +\ 1\,001 \\ \hline 1\,010 \end{array} \qquad \begin{array}{r} -1 \\ +\ -1 \\ \hline -2 \end{array}$$

- Para pensar.

Representación alfanumérica

- Letras (mayúsculas y minúsculas)
- Dígitos decimales (0, ..., 9)
- Signos de puntuación
- Caracteres especiales
- “Caracteres” u órdenes de control

Ejemplo

A cada símbolo un código en binario

Ejemplo: x, y, α, β, #, @, [,]

- Ocho símbolos ¿Cuántos bits? ¿Por qué?

000	x	@	...
001	y	[	
010	α	α	
011	β	#	
100	#	β	
101	@	y	
110	[	x	
111	]	]	

Algunos códigos

- FIELDATA
  - 26 letras mayúsculas + 10 dígitos + 28 caracteres especiales
  - Total 64 combinaciones $\Rightarrow$ Código de 6 bits
- ASCII

American Standard Code for Information Interchange

  - FIELDATA + minúsculas + ctrl
  - Total 128 combinaciones $\Rightarrow$ Código de 7 bits

Algunos códigos (2)

- ASCII extendido
  - ASCII + multinacional + semigráficos + matemática
  - Código de 8 bits
- EBCDIC - Extended BCD Interchange Code
  - similar al ASCII pero de IBM
  - Código de 8 bits

Tabla ASCII

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Una extensión al ASCII

128	Ç	144	É	160	á	176	░	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	│	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	Θ	249	·
137	ë	153	Ö	169	_	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	█	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	_
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	φ	253	^{2}
141	ì	158	_	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

mayor información …

- Capítulo 8: Aritmética del computador (8.1., 8.2., 8.3.)
  - Stallings, 5º Ed.
- Sistemas enteros y Punto fijo
  - Apunte 1 de Cátedra
- Capítulo 3: Lógica digital y representación numérica
  - Apuntes COC - Ingreso