PR 02

SISTEMAS DE NUMERACIÓN

1. NÚMEROS DECIMALES

El sistema de numeración que utilizamos en la vida cotidiana es el sistema decimal.

En dicho sistema de numeración poseemos 10 símbolos (dígitos) diferentes para representar cantidades:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

En adelante llamaremos Base de un sistema de numeración: al número de símbolos diferentes que se utiliza en dicho sistema de numeración. Por lo tanto en el sistema decimal, su base será 10.

Evidentemente en el sistema decimal no solo utilizamos cantidades del 0 al 9, sino que expresamos cantidades superiores combinando los dígitos ej. 537.

En el anterior número, cada uno de los dígitos tiene un peso diferente.

El número 537 lo podemos descomponer como sigue:

$$537 = 500 + 30 + 7$$

O lo que es lo mismo :

$$537 = 5 * 100 + 3 * 10 + 7 * 1$$

Que también se puede expresar como:

$$537 = 5 * 10^2 + 3 * 10^1 + 7 * 10^0$$

La anterior fórmula se conoce como expresión polinómica, y se puede asegurar que cualquier número representado en cualquier base, puede representarse de forma polinómica mediante la siguiente expresión:

$$N = a_n\ b^n + a_{n-1}\ b^{n-1} + \ldots\ldots\ A_0\ b^0 + a_{-1}\ b^{-1} + a_{-2}\ b^{-2} \ldots\ldots$$

Donde:

N → Número en base 10

b → Base.

a → Un dígito o número de la cifra a representar (cualquiera)

n → Posición que ocupa "a". La posición empieza desde cero y de derecha a izquierda.

En la anterior expresión,

$$a_n\ b^n + a_{n-1}\ b^{n-1} + \ldots\ldots.\ A_0\ \ b^0$$

Nos da la parte entera, mientras que

$$+\ a_{-1}\ \ b^{-1} + a_{-2}\ \ b^{-2}\ \ldots..$$

No da la parte fraccionaria.

2. NÚMEROS BINARIOS

El sistema de numeración binario solo posee dos dígitos 0 y 1. A cada uno de éstos dígitos se les denomina **bit.**

La base del sistema binario es 2.

Este es el sistema de numeración que se emplea en los circuitos electrónicos digitales y para almacenar información digital.

El número máximo que se pueden expresar con el sistema binario depende del número de bits que utilicemos y de acuerdo con la siguiente fórmula:

$$N = 2^n - 1$$

Así con 1 bit podemos contar hasta el 1. : 0 y 1; ($N = 2^1 - 1$)

Con 2 bits podemos contar desde 0 hasta el 3: 0, 1, 2, 3; ($N = 2^2 - 1$)

Con 3 bits podemos contar desde 0 hasta 7: 0, 1, 2, 3, 4, 5, 6, 7; ($N = 2^3 - 1$)

Con 4 bits podemos contar desde 0 hasta 15: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15. ($N = 2^4 - 1$); y así sucesivamente.

Los pesos o valores posicionales en un número binario que posee solo parte entera son:

$\mathbf{2^{n-1}}$	**.....**	$\mathbf{2^4}$	$\mathbf{2^3}$	$\mathbf{2^2}$	$\mathbf{2^1}$	$\mathbf{2^0}$
		16	8	4	2	1

Los pesos o valores posicionales en un número binario que y en su parte decimal es:

2^{-1}	2^{-2}	2^{-3}	2^{-4}	**....**
0,5	0,25	0,125	0,0625	

Ejercicio calcule los bits que necesitamos para representar el número decimal 1032 en binario.

3. CONTAR EN BINARIO

La manera de contar en binario es totalmente similar a la forma en que contamos en decimal, lo único que tenemos que recordar es que solo posemos 2 dígitos, seguidamente se da una tabla con los números desde el 0 hasta el 15 decimal expresados en binario (necesitamos 4 bits)

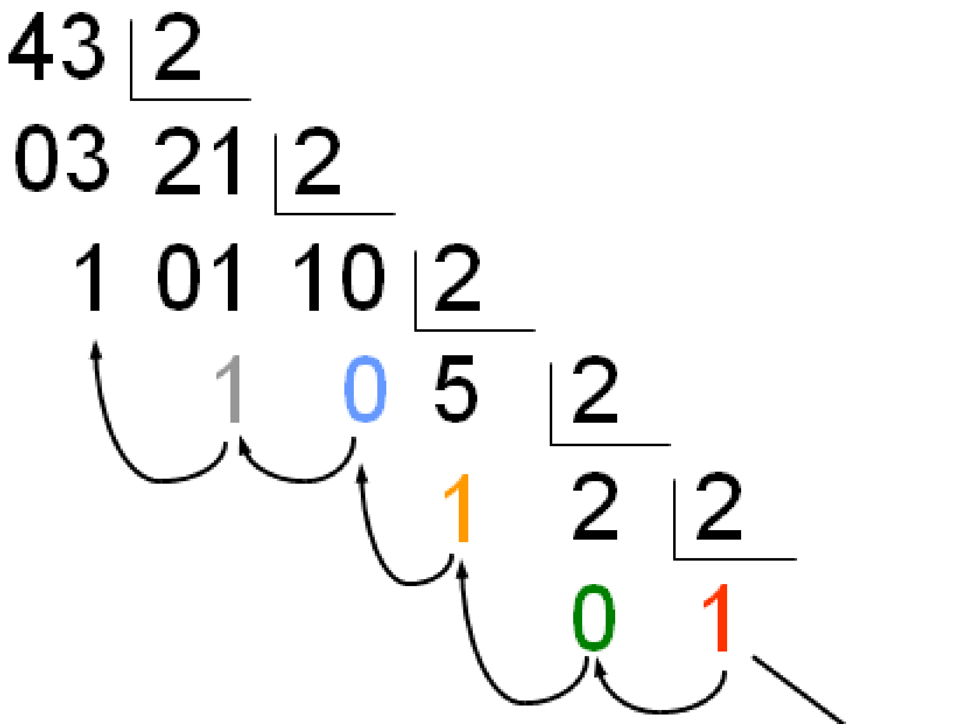
Nº decimal	**Nº binario**	**Nº decimal**	**Nº binario**
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

4. CONVERTIR DECIMAL EN BINARIO

El método más utilizado es el de la divisiones sucesivas y consiste en tomar, el número decimal y dividirlo por 2 (base del sistema binario) hasta que el resto sea 1 ó 0, al cociente resultante se vuelve a dividir por 2 y así hasta que ya no se puedan efectuar más divisiones, posteriormente se compone el número binario (solo puede haber 0´s y 1´s) en orden inverso al que fue obtenido.

Como ejemplo se convierte el número 43 (en base 10) a binario:

Conversión del número decimal 43 en binario



Número binario: 1 0 1 0 1 1

Para convertir una parte fraccionaria expresada en decimal a binario, utilizaremos el método de multiplicaciones sucesivas por 2.

Ejemplo: tenemos el número fraccionario 0,81 y lo queremos convertir en binario,

Tomamos el número decimal y lo multiplicamos por 2:

0,82 x 2 = 1,61

Del resultado obtenido la parte que se ha convertido en entera la tomamos como primer bit de la parte fraccionaria en binario, en el caso mostrado 1.

Nos quedamos con la parte fraccionaria obtenida 0,61 y repetimos el proceso:

0,61 x 2 = 1,22

Del resultado obtenido, volvemos a tomar la parte entera como segundo bit de la parte fraccionaria en binario, en el caso mostrado 1.

Nos volvemos a quedar con la parte fraccionaria : 0,22 y repetimos el proceso

$$0,22 \times 2 = 0,44$$

En este caso como la parte entera obtenida es 0 , tomamos 0 como el tercer bit de la parte fraccionaria en binario.

Repetimos el proceso:

$$0,44 \times 2 = 0,88$$

Cuarto bit = 0

$$0,88 \times 2 = 1,76$$

quinto bit = 1

El proceso se repite hasta que obtengamos como resultado del producto 0, o cuando consideremos que la precisión obtenida es suficiente.

4. CONVERTIR BINARIO EN DECIMAL

El valor decimal de cualquier número binario se puede obtener sumando los pesos de todos los bits que son 1.

Como ejemplo obtenemos el valor del número binario 1001110.

Nº binario	1	0	0	1	1	1	0
Pesos	64	32	16	8	4	2	1
Valor	64	0	0	8	4	2	0

El valor resultante será = 64+8+4+2 = 78

Veamos ahora el caso de un número binario que además posee parte entera y parte fraccionaria, número binario: 1010. 101

Nº binario	1	0	1	0	.	1	0	1
Pesos	8	4	2	1	.	0.5	0.25	0.125
Valor	8	0	2	0	.	0.5	0	0.125

El número decimal equivalente = 8+2+0,5+0,125 = 10,625

En el sistema binario encontramos las siguientes agrupaciones básicas de bits.

NIBBLE	Formado por 4 bits
BYTE	Formado por 8 bits
WORD	Formado por 16 bits
DOUBLE WORD	Formado por 32 bits
QUADRUPLE WORD	Formado por 64 bits

6. SISTEMA HEXADECIMAL

El sistema hexadecimal o base dieciséis, utiliza 16 símbolos para su representación.

Los 10 primeros son los dígitos del 0 al 9. Para los restantes se completan las letras de la 'A' a la 'F'.

La A tiene el valor 10, la B el 11, la C el 12 y así Sucesivamente.

Sus valores posicionales son:

16^4	16^3	16^2	16^1	16^0
65536	4096	256	16	1

Conteo en hexadecimal:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 2A, ...

Para **convertir un número hexadecimal en decimal** se utiliza el método de multiplicar cada dígito por su valor posicional y luego se realiza la suma de todos los valores obtenidos.

Ejemplo convertir el número hexadecimal 3C05 en decimal

Nº HEX	**3**	**C (12 en decimal)**	**0**	**5**
Pesos	4096	256	16	1
Valor	3 x 4096 = 12288	12 x 256 = 3072	0	5

El número hexadecimal 3C05 expresado en decimal =

12288 + 3072 + 5 = 15365

Para convertir un número decimal en hexadecimal, lo más sencillo, es convertir primero el número decimal en binario.

Ejemplo convertir el número decimal 94 en hexadecimal:

1.- Convertimos 94 en binario; $94_{10} = 1011110_{2}$

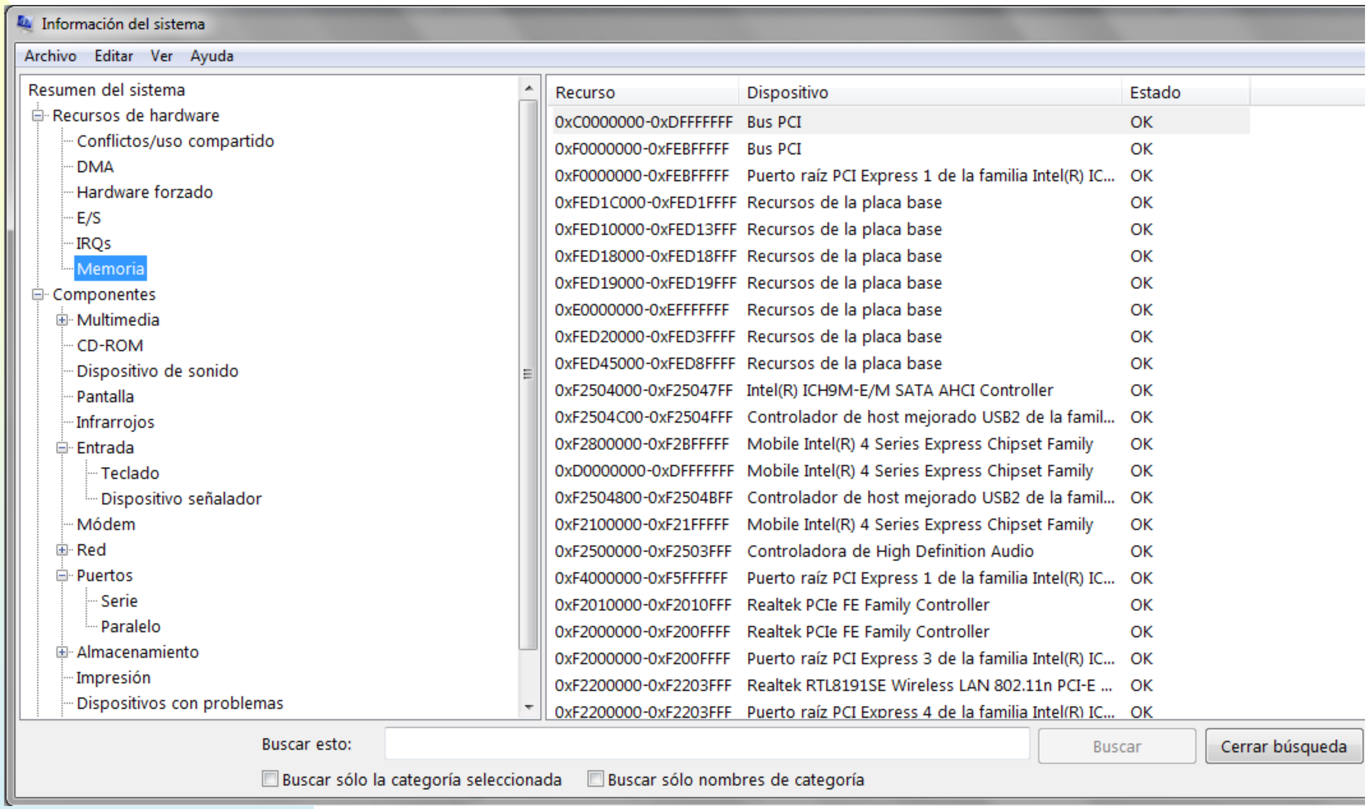
2.- Comenzando desde la derecha del número binario obtenido hacemos grupos de 4 bits: 101 1110

3.- Sustituimos cada grupo de 4 bits por su equivalente en hexadecimal: 101 = 5 ; 1110 = E.

4.- Componemos el número: 5E.

Decimal	Binario	Hex	Decimal	Binario	Hex
0	0000	0	16	1 0000	10
1	0001	1	17	1 0001	11
2	0010	2	18	1 0010	12
3	0011	3	19	1 0011	13
4	0100	4	20	1 0100	14
5	0101	5	21	1 0101	15
6	0110	6	22	0110	16
7	0111	7	23	0111	17
8	1000	8	24	1000	18
9	1001	9	25	1001	19
10	1010	A	26	1010	1ª
11	1011	B	27	1011	1B
12	1100	C	28	1100	1C
13	1101	D	29	1101	1D
14	1110	E	30	1110	1E
15	1111	F	31	1111	1F

El código hexadecimal es muy utilizado en el mundo de la informática y en la electrónica digital, por ejemplo para indicar direcciones de memoria o de dispositivos.



Nótese que para indicar que un número es HEX se le añade el prefijo 0X

7. CÓDIGOS BCD (Decimal Codificado en Binario)

Los códigos BCD son sistemas de numeración que nos permiten representar cada uno de los dígitos decimales mediante un código binario.

Existen diferentes códigos BCD entre los más usuales se encuentran:

BCD 8421 es un código que mediante 4 bits representa digito a dígito un número decimal. Este es un código ponderado, siendo los números 8, 4, 2, y 1 los pesos de los diferentes bits del código.

Ya que con 4 bits se pueden obtener 16 combinaciones diferentes, y solo utilizamos 10, existen 6 combinaciones no válidas.

El conteo en BCD 8421 es muy sencillo veamos una tabla:

Nº Dec	Binario	BCD 8421
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010

Las combinaciones no válidas en BCD 8421 son:

1010, 1011, 1100, 1101, 1110, y 1111.

La transformación de BCD 8421 en decimal y viceversa es inmediata.

Ej. Representar en BCD8421 el número decimal 478:

$$478_{10} = 0100\ \ 0111\ 1000_{\text{ en BCD8421}}$$

Ej.: dado el número 1001 1000 0010 0110 en BCD8421, indicar su equivalente en decimal:

$$1001\ 1000\ 0010\ \ 0110_{\text{en BCD8421}} = 9826_{10}$$

BCD AIKEN 2421 y

BCD AIKEN 5412

Son códigos BCD ponderados.

Para codificar un número decimal en código Aiken, tenemos en cuenta que:

Se asigna un “0” al MSB de los números del 0 a 4 y un “1” al MSB de los números del 5 a 9. El resto de los bits toman el valor para que la suma sea el número decimal deseado.

Dec.	AIKEN 2421	AIKEN 5421
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	1011	1000
6	1100	1001
7	1101	1010
8	1110	1011
9	1111	1100

El código Gray es un código sin peso que se utiliza en algunas aplicaciones de uso industrial, su principal característica es que de una valor al siguiente solo se modifica uno de los bits.

También se le conoce como código reflejado, su construcción es muy sencilla:

Código Gray de 1 bit	Nº decimal
0	0
1	1

Código Gray de 2 bits	Nº decimal
0 0	**0**
0 1	**1**
1 1	**2**
1 0	**3**

El código Gray es un código sin peso que se utiliza en algunas aplicaciones de uso industrial, su principal característica es que de una valor al siguiente solo se modifica uno de los bits.

También se le conoce como código reflejado, su construcción es muy sencilla:

Codigo Gray de 1 bit	Nº decimal
0	0
1	1

Código Gray de 2 bits	Nº decimal
0 0	**0**
0 1	**1**
1 1	**2**
1 0	**3**

Código Gray de 3 bits	**Nº decimal**
0 00	0
0 01	1
0 11	2
0 10	3
1 10	4
1 11	5
1 01	6
1 00	7

Código Gray de 4 bits	**Nº decimal**
0 000	0
0 001	1
0 011	2
0 010	3
0 110	4
0 111	5
0 101	6
0 100	7
1 100	8
1 101	9
1 111	10
1 110	11
1 110	12
1 011	13
1 001	14
1 000	15

El código ASCII (American Standard Code for Information Interchange) es un código alfanumerico que se utiliza en la mayoría de los computadores.

Así cuando en un teclado se pulsa una tecla, es el código ASCII el que se envía al procesador.

Caracteres de control ASCII

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de linea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

Caracteres ASCII imprimibles

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

elCodigoASCII.com.ar

ASCII extendido

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	å	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ò	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	█	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	Ì	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	

Códigos con paridad.

Consiste en añadir un bit a las palabras a transmitir, para lograr que el número de unos sea par (paridad par) o impar (paridad impar).

Así un código BCD que se desea transmitir añadiendo el bit de paridad quedaría como sigue:

BCD8421	**BCD8421 + PARIDAD PAR**	**BCD8421 + PARIDAD IMPAR**
0000	0000 0	0000 1
0001	0001 1	0001 0
...	...	...
0111	0111 1	0111 0
1000	1000 1	1000 0
1001	1001 0	1001 1