CODIFICACION

Las computadoras operan solamente con unos y ceros, o dicho de otra manera, los circuitos de una computadora, para que sean más sencillos, solamente son capaces de distinguir entre dos estados: circulación de corriente o no circulación, encendido o apagado, lo cual puede asociarse a un uno y a un cero**,** por ejemplo:

1= ENCENDIDO
0 = APAGADO

Es así entonces que podríamos operar con un sistema de numeración binaria, pero esto tiene el inconveniente que es difícil de leer e interpretar, además que para cantidades relativamente pequeñas se necesitan grandes cantidades de unos y ceros. Para probar esto, obsérvese el número del ejemplo $(2142)_{10}$, en que se necesitan doce dígitos binarios para representar una cantidad formada con cuatro dígitos decimales.-

A fin de simplificar las cosas, se ha decidido representar los números mediante un código binario, o sea que cada digito decimal es reemplazado por una cierta combinación de ceros y unos.

Para ello si tomamos un conjunto de cuatro dígitos binarios (BIT = Binary digit) con un conjunto de cuatro bits, se tienen dieciséis combinaciones posibles, de las cuales solamente se utilizan diez.

Lo más simple es la condición de peso, o sea que el dígito binario tome un valor en función de su ubicación, tal como lo hace en el sistema binario natural.

Si los pesos son iguales a este, el código así formado es conocido como BCD (Binary Coded Decimal), que significa decimal codificado en binario.

De esta manera, tendremos:

	Peso			
	2^3	2^2	2^1	2^0
Decimal	2	2	2	2
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

No es éste el único empleado, dado que los pesos 8, 4, 2 y 1 pueden ser variados, obteniendo así algunos códigos particulares, útiles para otros fines.-

Podemos citar entre otros los códigos 2421, 5211, 5421.

Este último, se aplica para lograr diferencias entre las cifras superiores a 5 e inferiores a él.

	Pesos			
Decimal	5	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1

4	0	1	0	0
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0

Obsérvese que los números comprendidos entre el 5 y el 9 llevan un 1 en primer lugar, mientras que los restantes no.

Otras veces, es necesario transmitir datos en forma tal que se detecte fácilmente un error. La forma más simple es que cada cifra posea en su formación una cantidad igual de unos y ceros. Así se ha desarrollado el sistema biquinario:

Decimal	Biquinario
0	1010000
1	1001000
2	1000100
3	1000010
4	1000001
5	0110000
6	0101000
7	0100100
8	0100010
9	0100001

También a menudo se debe hacer ahorro de energía, lo cual significa que un código debe tener la menor cantidad posible de unos, lo cual es logrado por el de pesos 7421, que es:

	Pesos			
Decimal	7	4	2	1
0	0	0	0	0
1	0	0	0	
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	1	0	0	0
8	1	0	0	1
9	1	0	1	0

Otros códigos son aquellos que para formar sus codificaciones siguen reglas no ponderadas, como ser los códigos continuos, como el código GRAY (binario reflejado) o el código Jonson, que forma sus códigos de forma adyacente (se diferencian en un BIT) y además la primera y la última codificación también son adyacentes, lo que lo hace un código cíclico.

Decimal	GRAY (4 bits)	JOHNSON (8 bits)
0	0000	00000000
1	0001	00000001
2	0011	00000011
3	0010	00000111
4	0110	00001111
5	0111	00011111
6	0101	00111111
7	0100	01111111
8	1100	11111111
9	1101	11111110

10	1111	11111100
11	1110	11111000
12	1010	11110000
13	1011	11100000
14	1001	11000000
15	1000	10000000

Los códigos BCD (binary codes decimal) son códigos que trabajan en decimal, pero están codificados en binario, por ello tienen 10 codificaciones diferentes que se corresponden con las del sistema decimal. Los más importantes pueden ser los siguientes:

Decimal	BCD natural 8 4 2 1	BCD exceso 3	Aiken 2 4 2 1
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0010
3	0011	0110	0011
4	0100	0111	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111

Para operar con los códigos BCD se pueden hacer grupos de 4 bits, de forma que cada grupo corresponde a un dígito decimal, de este modo pasar al sistema decimal y viceversa es una tarea sencilla.

Decimal	BCD natural	BCD exceso 3	Aiken
4 5	0100 0101	0111 1000	0100 1011

Cuando se opera en BCD natural, por ejemplo en una suma, hay que tener cuidado ya que el resultado de esa operación puede ser una codificación fuera del código. Para realizar la corrección se suma la cantidad 6 (0110) obteniendo resultados correspondientes al código BCD.

En computadoras prácticamente todas las operaciones se reducen a sumas.

Complementación Decimal Y Binaria

Generalizando, podemos decir que si en lugar de restar un número con signo negativo, sumamos su "complemento al módulo", obtendremos el mismo resultado. El módulo en cada caso es igual al número mayor que puede escribirse con una cantidad de cifras más uno. Por ejemplo para un sistema de 5 cifras decimales, el número mayor será 99999 y el módulo será 100000.

Para calcular el complemento podemos restar el dígito menos significativo de 10 y los demás de nueve.

```
 1000          9 9 10          9 9 9
-              -               -
  354           3 5  4          3 5 4

  646           6 4  6          6 4 5
                                +
                                      1
                                ---------
                                6 4 6
```

La última operación nos muestra como en lugar de restar la última cifra de 10 le restamos 9 y luego si sumamos uno obtenemos el mismo resultado. En las dos primeras calculamos el

complemento a 10 y en la última encontramos el complemento a 9, y al sumarle uno hallamos el complemento a 10.
De la misma forma al operar con números binarios podemos hallar el complemento a dos y el complemento a 1. en el complemento a dos se resta el dígito menos significativo de 2 y los demás de 1; mientras que en el complemento a 1, se restan todos de 1.

```
 1 1 1 10                          1 1 1 1
-          complemento a 2      -          complemento a 1
 1 0 1  1                          1 0 1 1
 0 1 0  1                          0 1 0 0
```

El complemento a 1 se calcula rápidamente cambiando ceros por unos y unos por ceros. Por ello son útiles los códigos autocomplementarios (BCD exceso tres)

Números en coma fija y en coma flotante

Nuestro sistema de operar es conocido como punto fijo o coma fija, pues se encolumnan los números por el lugar donde va la coma decimal, antes de operarlos.

Por ejemplo, el número 324,58 en binario quedará:

```
0011  0010  0100   0101  1000
                |
        lugar de la coma decimal
```

Si suponemos que solamente se pueden representar números de hasta 6 cifras enteras y 4 decimales, no se podrá procesar ninguna cantidad superior a 999999,9999 ni inferior a 0,0001, lo que resulta un rango estrecho para cálculos científicos.
Esto se puede solucionar con la notación en punto coma o punto flotante, donde un número se expresa como:

$$N = M * B^{X}$$

Siendo N la cantidad a representar, M la mantisa o cantidad significativa, B la base (generalmente 10) y X el exponente que tiene en cuenta el número de ceros que se debe colocar antes o después de la parte significativa del número. En esta notación se emplean números normalizados, es decir que poseen una sola cifra significativa a la izquierda de la coma decimal. Por ejemplo:

32320482 será: $3,2320482 * 10^{7}$

El número 0,000542876 en binario y coma flotante será:

```
S1 0000  0100  0101  0100  0010  1000  0111  0110  S2
 ↓ Exponente          Mantisa                    ↓
Signo del exponente                        signo de la mantisa
```

Vemos como se define el signo mediante un BIT de signo. Si el valor del BIT de signo es 0, el signo es positivo y si es 1, el signo es negativo.

Códigos Alfanuméricos

No solo es necesario representar cantidades en una máquina, sino también caracteres alfabéticos, signos de puntuación y códigos de control de periféricos, para avanzar el papel en una impresora, avanzar una línea, y otros por el estilo.

De acuerdo a las necesidades, se podrán emplear seis o más dígitos binarios para ello, llegando hasta 8 o 9 en el caso de usar control de paridad (Parity check).

En la mayoría de los casos, la información es tratada en paquetes de bits, cuya cantidad varía normalmente de cuatro (nibble) a ocho (bytes).

Sabemos que mediante unos cuatro bits (nibble) es posible representar las diez cifras decimales, con un resto de seis combinaciones, que servirán para representar los números en el sistema hexadecimal, de aquí la importancia de este sistema que aprovecha en su totalidad el espacio disponible.

Así es que las fábricas han desarrollado para sus máquinas diversos códigos, entre los cuales los más empleados son el EBCDIC (Extended Binary Coded Decimal Interchange Code) que significa decimal codificado en binario para intercambio de información y el código ASCII (Amerícan Standard Code for Information Interchange) o sea Código Americano Standard para el Intercambio de información.

El código ASCII es representable en la siguiente forma:

CARACTER	ASCII (Hexadecimal)
A	4 1
B	4 2
C	4 3
•	•
•	•
•	•
I	4 9
J	4 A
K	4 B
O	4 F
P	50
R	52
Z	5 A
0	3 0
1	3 1
9	3 9

Este código es muy empleado, pues la administración Federal de los EEUU lo ha declarado Standard.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0 0 0 0	1 0 0 0	0 1 0 0	1 1 0 0	0 0 1 0	1 0 1 0	0 1 1 0	1 1 1 0	0 0 0 1	1 0 0 1	0 1 0 1	1 1 0 1	0 0 1 1	1 0 1 1	0 1 1 1	1 1 1 1
0000	UNL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0001	D LE	DC 1	DC 2	DC 3	DC 4	NAK	S YN	E T	CAN	EM	S UB	E S	F S	G S	R S	U S
0010	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	/	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	→	←
1000	DE L	_														