

TIPS DE INFORMACION

1. ¿Por qué a nuestra principal herramienta en sistemas “el Computador” se le conoció como ORDENADOR, luego por COMPUTADOR y finalmente como COMPUTADOR DIGITAL?

En LA HISTORIA los pioneros en el diseño de equipos informáticos, apoyándose en la aritmética binaria y mediante la utilización de tarjetas perforadas codificadas pudieron gestionar los censos grandes de población, a esta máquina la llamaron ORDENADOR, por ser la operación de ordenar las fichas. Luego el nombre de COMPUTADOR debido a que tomaba las decisiones a través de resultado de unos cálculos y realizando comparaciones con otros preestablecidos. Finalmente se conoce como COMPUTADOR DIGITAL debido a que esta información está representada de forma digital en unos y ceros.

2. ¿Porque el termino de PC para referirse a los computadores?

Desde la fabricación de ordenadores con válvulas, pasando por los fabricados con transistores y posteriormente con circuitos integrados de pequeña y media escala de integración, es importante resaltar que el estudio de arquitecturas a partir de la llegada del primer microprocesador de 4 bits que se introdujo en los años 70 y de ahí que IBM lanzo su ordenador personal “PC”.



Se dio inicio al termino PC IBM COMPATIBLE para que las empresas de la competencia simularan que era muy parecido al primer computador de IBM.

<u>Arquitectura</u>	<u>x86</u> , <u>ISA</u> de 8 bits
<u>Procesador</u>	<u>Intel 8088</u> @ 4,77 MHz
<u>Memoria</u>	16KB ~ 640KB
<u>Sistema de audio</u>	Generación de tonos por altavoz
<u>Sistema gráfico</u>	<u>MDA</u> y <u>CGA</u>
<u>Sistema operativo</u>	<u>PC-DOS</u> 1.0+ / <u>IBM Cassette BASIC</u>

3. ¿Sistemas de Codificación BCD, ASCII, UNICODE, FIEL DATA, EBCDIC, GRAY, etc?

BCD (Decimal Codificado Binario), Es la primera codificación para 4 bits (nibble ó cuartero ó 1/2 Byte). Existen varias versiones:

- BCD natural (8 4 2 1) Maneja las potencias de 2 normalmente.
- BCD Aiken (2 4 2 1) Maneja una simetría cambiando los pesos de las potencias.
Para transformar un número en C-9 debe reemplazarse cada dígito por lo que le falta para llegar a 9. Por ejemplo 3 es 0011 y -3 es 1100 que es lo mismo que el 6 en Aiken ósea 1100, lo cual ratifica que es complemento a 9, al número 3 cuanto le falta para ser 9 pues 6.
- BCD Exceso 3 (8+3 4+7 2+3 1+3) Maneja un exceso en 3 unidades de cada valor a representar.
- BCD 5421 (5 4 2 1) Maneja las potencia de manera particular.

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

FIEL DATA Códigos utilizados en el computador para guardar y transmitir información, solo manejaba 6 bits:

- Los números del 0 a 9
- Letras del alfabeto tanto MAYUSCULAS como minúsculas.
- Algunos caracteres especiales (28 en total) como lo son . , ; + * \ < > & \$ () @ # ^ ó / = [] % SP ¿ ? ! - ' -

ASCII (Código Estándar Americano para el Intercambio de Información), Es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales.

Fue creado en 1963 por el Comité Estadounidense de Estándares (ASA, conocido desde 1969 como el Instituto Estadounidense de Estándares Nacionales, o ANSI) como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía.

El código ASCII utiliza 7 bits para representar los caracteres, aunque inicialmente empleaba un bit adicional (bit de paridad) que se usaba para detectar errores en la transmisión.

ASCII fue publicado como estándar por primera vez en 1967 y fue actualizado por última vez en 1986.

Define códigos para 33 caracteres no imprimibles, la mayoría son caracteres de control obsoletos.

Otros 95 caracteres imprimibles que les siguen en la numeración (empezando por el carácter espacio).

Casi todos los sistemas informáticos actuales utilizan el código ASCII o una extensión compatible para representar textos y para el control de dispositivos que manejan texto como el teclado. No deben confundirse los códigos ALT+número de teclado con los códigos ASCII.

EBCDIC (Código para el Intercambio Decimal Codificado Binario Extendido)

A menudo se llama incorrectamente ASCII a otros códigos de caracteres de 8 bits, como el estándar ISO-8859-1 que es una extensión que utiliza 8 bits para proporcionar caracteres adicionales usados en idiomas distintos al inglés, como el español. Este código de 8 bits se le conoce como EBCDIC con 256 caracteres sin bit de paridad. Muy utilizado en arquitectura IBM.

CÓDIGO GRAY

El código binario reflejado o código Gray, nombrado así en honor del investigador Frank Gray, es un sistema de numeración binario en el que dos valores sucesivos difieren solamente en uno de sus dígitos.

El investigador de Laboratorios Bell A. Frank Gray inventó el término código binario reflejado cuando lo patentó en 1947, remarcando que éste "no tenía nombre reconocido aún".¹ Él creó el nombre basándose en el hecho de que el código "puede ser construido a partir del código binario convencional por una suerte de 'proceso reflejante'".

El código fue llamado posteriormente "Gray" por otros investigadores. Dos patentes en 1953 dieron como nombre alternativo "código de Gray" para el "código binario reflejado";^{2 3} uno de ellas también se refiere al código como "minumum error code" (código de error mínimo) y como "cyclic permutation code" (código de permutación cíclica).³

Conversiones

Secuencia	Binario	Gray	Secuencia	Binario	Gray
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Base 2 a Gray

Para convertir un número binario (en [Base 2](#)) a código Gray, simplemente hemos de aplicarle la puerta lógica [XOR](#) al mismo número, con 1 desplazamiento a la derecha

Ejemplo: 1010 ([Base 2](#)) a gray

```

1010
101⊕
----
```

1111

Otros ejemplos:

```

111000
 11100⊕
-----
100100
110101010001
 11010101000⊕
-----
10111111001

```

Gray a Base 2

Tenemos un vector a conteniendo los dígitos en gray y otro vector b destinado a contener los dígitos en Base 2
 a_0 es el dígito que se encuentra en el extremo izquierdo de la representación en código gray

b_0 es el dígito de mayor peso y que se encuentra en el extremo izquierdo en la representación en Base 2

tenemos que: $b_n = b_{n-1} \oplus a_n$ con la excepción de que $b_0 = a_0$, la cual se puede resumir como: *el dígito de más a la izquierda en Base 2 es igual al dígito de más a la izquierda en código gray*

El primer bit empezando por la izquierda del dígito del código gray se respetará para la conversión a base 2, el resultado es obtener el mismo bit para el dígito binario que el que tiene en gray, para conseguir el segundo bit del binario sumaremos el primer bit del dígito del sistema binario por el segundo del sistema gray, sin tener en cuenta los acarreo y respetando la tabla de suma para binarios: 0+0=0 ; 0+1=1 ; 1+0=1 ; 1+1=10

Ejemplo: Con el número 1001 Gray

El primero de base dos es igual al primero en gray que en este caso es (1)

El segundo de base dos es igual a la suma del primero de base 2 con el segundo de gray en este caso es (1)+(0)= (1)

El tercero de base dos es igual a la suma del segundo de base2 con el tercero de gray en este caso es (1)+(0)= (1)

El cuarto de base dos es igual a la suma del tercero de base dos con el cuarto de gray es este caso es (1)+(1)=10 tomamos el cero del 10 descartando el acarreo por lo que tenemos (0)

Esto da como resultado 1110