

Preparador Informática

www.preparadorinformatica.com

TEMA 9. INFORMÁTICA / S.A.I.

**LÓGICA DE CIRCUITOS. CIRCUITOS
COMBINACIONALES Y SECUENCIALES**

TEMA 9 INF/SAI: LÓGICA DE CIRCUITOS. CIRCUITOS COMBINACIONALES Y SECUENCIALES

1. INTRODUCCIÓN

2. LOGICA DE CIRCUITOS

2.1. OPERACIONES LÓGICAS

2.2. ALGEBRA DE BOOLE

2.3. TEOREMAS DE DEMORGAN

2.4. FUNCIONES LÓGICAS

3. CIRCUITOS COMBINACIONALES

3.1. CODIFICADOR

3.2. DECODIFICADOR

3.3. MULTIPLEXOR

3.4. DEMULTIPLEXOR

3.5. OTROS CIRCUITOS

4. CIRCUITOS SECUENCIALES

4.1. BIESTABLES

4.2. CONTADORES

4.3. REGISTROS

5. CIRCUITOS INTEGRADOS COMERCIALES

6. RECURSOS Y HERRAMIENTAS EDUCATIVAS DE INTERÉS

7. CONCLUSIÓN

8. BIBLIOGRAFÍA



1. INTRODUCCION

El estudio de los circuitos electrónicos se puede clasificar en dos amplias categorías: circuitos analógicos (utiliza magnitudes con valores continuos) y circuitos digitales (utiliza magnitudes con valores discretos).

En las aplicaciones de electrónica, la representación digital presenta ciertas ventajas sobre la representación analógica. La principal ventaja es que los datos digitales pueden ser procesados y transmitidos de forma más fiable y eficiente que los datos analógicos.

En este tema nos vamos a centrar en los sistemas digitales, los cuales se basan en la electrónica digital que utiliza circuitos en los que sólo existen dos estados posibles (0 y 1).

La importancia y justificación de este tema radica en que en la actualidad estamos rodeados de infinidad de dispositivos como lavadoras, televisores, ordenadores, etc. los cuales están formados por circuitos digitales que les permiten realizar todas sus funciones.

Estos circuitos digitales podemos dividirlos en dos grandes grupos: los circuitos combinacionales y los circuitos secuenciales, los cuales serán estudiados en detalle a lo largo del presente tema.

Preparador Informática

2. LOGICA DE CIRCUITOS

El término lógico se aplica a los circuitos digitales que se utilizan para implementar funciones lógicas.

Existen varios tipos de circuitos lógicos, que son los elementos básicos que constituyen los bloques sobre los que se construyen los sistemas digitales más complejos, como por ejemplo un ordenador.

A continuación, se abordan los conceptos previos necesarios para la descripción y diseño de estos circuitos:

- Operaciones lógicas
- Álgebra de Boole

- Teoremas de DeMorgan
- Funciones lógicas

2.1. OPERACIONES LÓGICAS

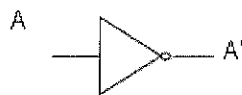
Las puertas lógicas son el bloque fundamental de construcción de todos los circuitos lógicos digitales, de forma que las funciones lógicas se implementan interconectando puertas.

Para representar las puertas lógicas se utilizan las siguientes notaciones:

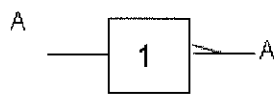
- Símbolos distintivos: corresponden a los utilizados comúnmente por la industria digital.
- Símbolos rectangulares: en 1984 se desarrolló una nueva norma para la representación de símbolos lógicos llamada **IEEE/ANSI 91-1984**. Esta norma es empleada entre otros organismos por el Dpto. de Defensa de los E.E.U.U.

A continuación, se representan las principales puertas lógicas empleando la notación de símbolos distintivos (a) y la de símbolos rectangulares (b). Además, se muestra la tabla de verdad asociada a cada puerta lógica (c):

Puerta NOT: Produce un nivel lógico de salida opuesto al suministrado como entrada.



a)

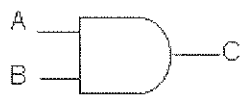


b)

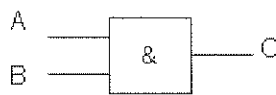
a	s
0	1
1	0

c)

Puerta AND: Produce un nivel alto a la salida sólo cuando todas las entradas están a un nivel alto.



a)

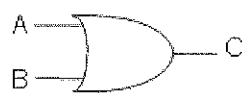


b)

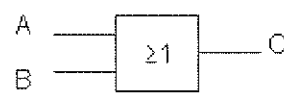
a	b	s
0	0	0
0	1	0
1	0	0
1	1	1

c)

Puerta OR: Produce un nivel alto a la salida cuando cualquiera de sus entradas está a un nivel alto.



a)

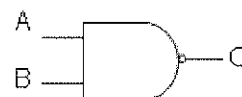


b)

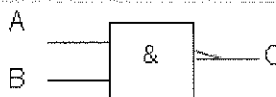
a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

c)

Puerta NAND: Realiza la función inversa a la Puerta AND.



a)



b)

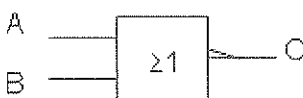
a	b	s
0	0	1
0	1	1
1	0	1
1	1	0

c)

Puerta NOR: Realiza la función inversa a la puerta OR.



a)

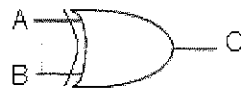


b)

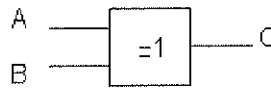
a	b	s
0	0	1
0	1	0
1	0	0
1	1	0

c)

Puerta XOR (OR-exclusiva): Produce un nivel alto a la salida si y solo si una de las dos entradas está a un nivel alto.



a)



b)

a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

c)

Puerta XNOR (NOR-exclusiva): produce un nivel alto a la salida si y solo las dos entradas están al mismo nivel.



a)



b)

a	b	s
0	0	1
0	1	0
1	0	1
1	1	1

c)

Las puertas NAND y NOR se denominan puertas universales, ya que a partir de ellas es posible implementar cualquiera de las demás puertas lógicas.

2.2. ALGEBRA DE BOOLE

La base matemática en la que se fundamenta la electrónica digital es el álgebra de Boole. Para realizar de forma sistemática tanto el análisis como síntesis de los sistemas digitales se utiliza como eficaz herramienta el álgebra de Boole, que se puede definir como el conjunto de postulados siguientes:

A. Leyes

- Leyes conmutativas

$$AB=BA$$

$$A+B=B+A$$

- Leyes asociativas

$$A(BC)=(AB)C$$

$$A+(B+C)=(A+B)+C$$

- Ley distributiva

$$A(B+C)=AB+AC$$

B. Reglas

1. $A+0=A$	5. $A+A=A$	9. $A=A''$
2. $A+1=1$	6. $A+A'=1$	10. $A+AB=A$
3. $A*0=0$	7. $A*A=A$	11. $A+A'B= A+B$
4. $A*1=A$	8. $A*A'=0$	12. $(A+B)(A+C)= A +BC$

2.3. TEOREMAS DE DEMORGAN

El matemático DeMorgan propuso dos teoremas basados en las teorías de Boole que constituyen una parte muy importante del álgebra de Boole.

1. El complemento de un producto de variables es igual a la suma de los complementos de las variables

$$(A*B)' = A' + B'$$

2. El complemento de una suma de variables es igual al producto de los complementos de las variables

$$(A+B)' = A' * B'$$

2.4. FUNCIONES LÓGICAS

Las funciones lógicas describen cada una de las salidas de un sistema digital para todas las posibles combinaciones de entradas.

Las funciones lógicas se pueden expresar de varias formas:

- A. Expresiones booleanas, de la forma:

$$X = A'B'C' + A'B'C + A'BC + ABC'$$

- B. Circuitos lógicos: (representado mediante puertas lógicas).

C. Tablas de verdad: es una tabla de las posibles combinaciones de valores de las variables de entrada y sus correspondientes valores de salida.

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

D. Mapas de Karnaugh: es una matriz de celdas que proporciona un método sistemático de simplificación de expresiones booleanas.

A B					
		00	01	11	10
0		1	1	1	0
1		1	0	0	0

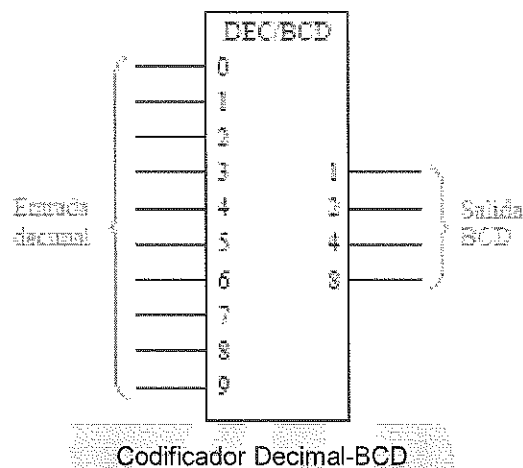
3. CIRCUITOS COMBINACIONALES

Un circuito lógico combinacional es un circuito cuyas salidas están determinadas exclusivamente por el valor de sus entradas en ese mismo instante. Por esta razón se dice que los circuitos combinacionales no cuentan con memoria. A continuación, se presentan algunos de los principales circuitos combinacionales:

- Codificador
- Decodificador
- Multiplexor
- Demultiplexor
- Etc.

3.1. CODIFICADOR

Un codificador es un circuito lógico combinacional que permite que se introduzca en una de sus entradas un nivel activo que representa un dígito, como puede ser un dígito decimal u octal, y lo convierte en una salida codificada, como por ejemplo BCD o binario



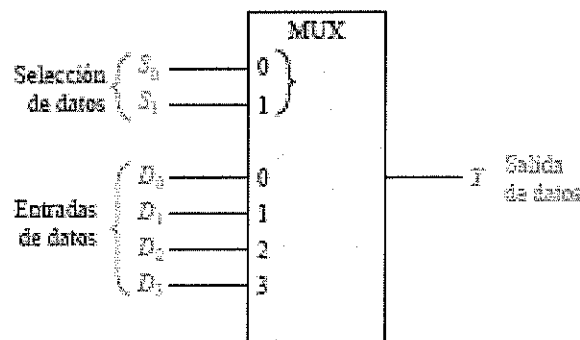
3.2. DECODIFICADOR

Un decodificador es un circuito lógico combinacional que detecta la presencia de una determinada combinación de bits (código) en sus entradas y señala la presencia de este código mediante un cierto nivel de salida. En su forma general, un decodificador consta n entradas para gestionar n bits y en una de las 2^n salidas indica la presencia de una combinación de n bits.

Por ejemplo, en un portero digital cuando vamos a llamar a un piso en concreto, tenemos una leyenda al lado de la placa de pulsadores, en la que nos indican el número/s que tenemos que pulsar (entradas decimales). Pulsamos por ejemplo el 16 para llamar al 2ºB. Esta señal llega a todos los telefonillos, pero solo el del 2ºB la interpreta como llamada, haciendo sonar el timbre.

3.3. MULTIPLEXOR

Un multiplexor es un circuito para la conducción de información con 2^n entradas de datos, n entradas de selección y una salida de datos. El multiplexor conecta una de las 2^n entradas de datos a la salida. Esta entrada es seleccionada a través de las entradas de selección. También se les conoce como selectores de datos



Multiplexor 4:1

3.4. DEMULTIPLEXOR

Un demultiplexor básicamente realiza la función contraria a la del multiplexor. Está formado por una entrada de datos, n entradas de control y 2^n salidas de datos. Toma datos de una línea y los distribuye a un determinado número de líneas de salida. Por este motivo, el demultiplexor se conoce también como distribuidor de datos.

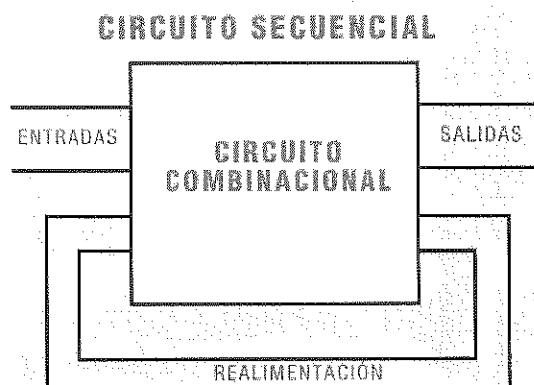
3.5. OTROS CIRCUITOS

Otros circuitos combinacionales a destacar son:

- Sumadores
- Comparadores
- PAL (Programmable Array Logic)

4. CIRCUITOS SECUENCIALES

Un circuito lógico secuencial es un circuito cuyas salidas dependen del valor que hay en las entradas en el instante actual y también de las que haya tenido en instantes anteriores. Es decir, a diferencia de los circuitos lógicos combinacionales, los circuitos secuenciales tienen memoria.



Los circuitos secuenciales se pueden clasificar en:

- **Asíncronos:** Los cambios de estado en el sistema se producen cuando cambia alguna de las entradas primarias, sin necesidad de que se active por una señal de reloj.
- **Síncronos:** Los cambios de estado en el sistema se producen cuando se recibe una señal de reloj.

4.1. BIESTABLES

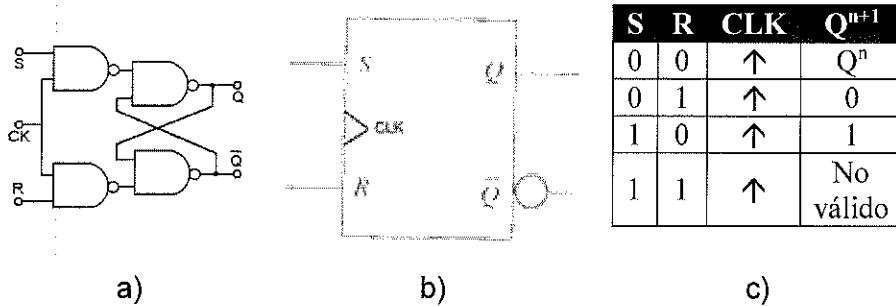
Las células elementales de memoria de los circuitos secuenciales se denominan biestables. Se caracterizan por ser capaces de adoptar dos estados estables (0 y 1), que perduran en el tiempo de un modo indefinido. Es decir, son capaces de memorizar un bit de información.

Los biestables se dividen en dos categorías:

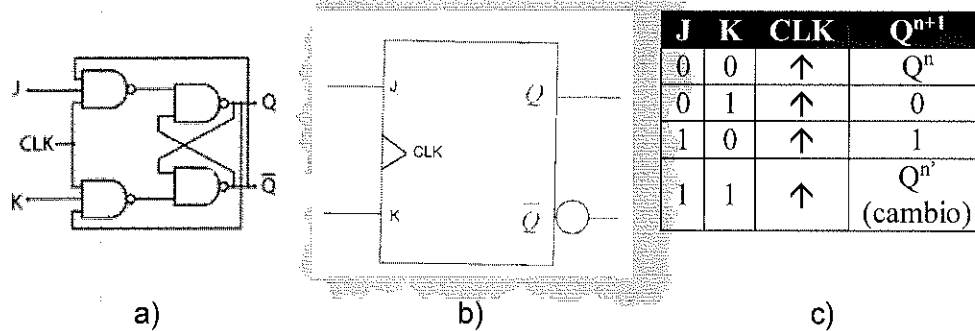
- A. **Latches:** son biestables asíncronos.
- B. **Flip-flops:** son biestables síncronos.

A continuación, se detalla la implementación (a), el símbolo lógico (b) y la tabla de verdad (c) de los principales flip-flops:

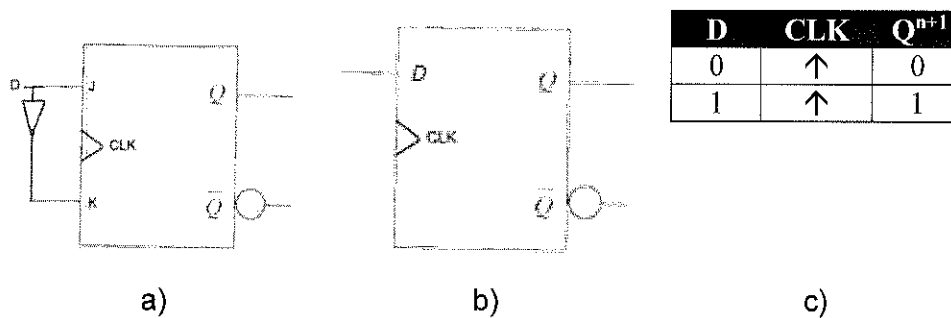
- Flip-Flop S-R disparado por flanco de subida:



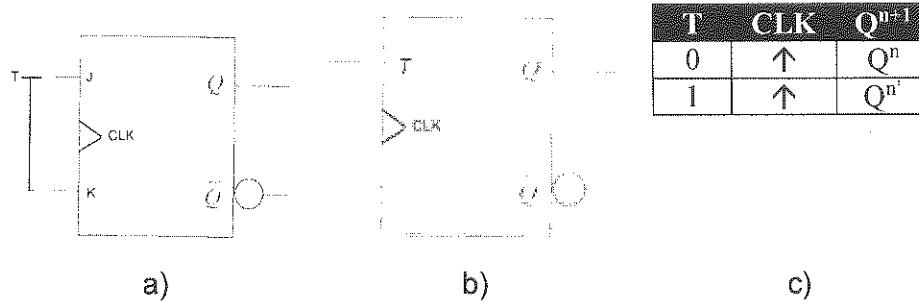
- Flip-Flop J-K disparado por flanco de subida:



- Flip-Flop D disparado por flanco de subida:



- Flip-Flop T disparado por flanco de subida:



4.2. CONTADORES

Los flip-flops pueden conectarse entre sí para realizar funciones de recuento. A esta combinación de flip-flops se le denomina contador.

Dependiendo del modo en que se aplique la señal de reloj, se distinguen:

- Contadores asíncronos
- Contadores síncronos

Dentro de cada una de estas dos categorías, los contadores se clasifican por el tipo de secuencia, por el número de estados, por el número de flip-flops del contador, etc.

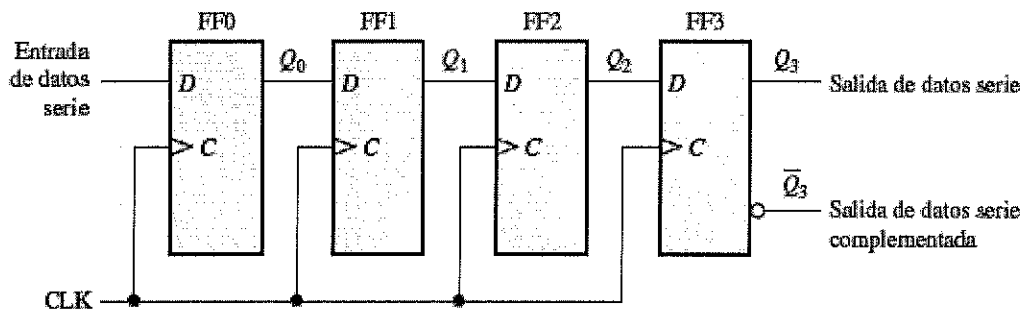
Se usan habitualmente en circuitos temporizadores, sistemas de control, divisores de frecuencia, etc.

4.3. REGISTROS

Los registros se utilizan principalmente para almacenar (registros de almacenamiento) y desplazar datos digitales (registros de desplazamiento).

Atendiendo a la forma en que se almacenan y se desplazan los datos, se pueden clasificar en:

- Registros de desplazamiento con entrada y salida serie



- Registros de desplazamiento con entrada serie y salida paralelo
- Registros de desplazamiento con entrada paralelo y salida serie
- Registros de desplazamiento con entrada y salida paralelo
- Registros de desplazamiento bidireccionales

5. CIRCUITOS INTEGRADOS COMERCIALES

Existe una gran cantidad de circuitos integrados comerciales que implementan los circuitos lógicos vistos a lo largo del tema.

A continuación, se listan algunos a modo de ejemplo:

- 74LS08 Cuádruple AND de 2 entradas
- 74HC153 Multiplexor 4 a 1
- 74HC147 Codificador Decimal/BCD
- 74HC163 Contador binario síncrono de 4 bits

6. RECURSOS Y HERRAMIENTAS EDUCATIVAS DE INTERÉS

Para el diseño y simulación de circuitos lógicos digitales podemos hacer uso de herramientas software como:

- LogicWorks
- Orcad
- MPLAB X

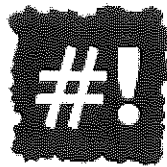
7. CONCLUSIÓN

En el presente tema se ha presentado una visión global de la lógica de circuitos describiendo los circuitos combinacionales y secuenciales, que son la base sobre la que se materializa cualquier sistema digital, desde los más simples hasta los más complejos y sofisticados, como por ejemplo los ordenadores.

Los circuitos combinacionales y secuenciales pueden construirse con diferentes tecnologías de fabricación. Así, al conjunto de componentes lógicos fabricados con la misma tecnología se le denomina familia lógica. De este modo, según los dispositivos semiconductores en lo que se basan podemos distinguir las familias bipolares (TTL, ECL, RTL y DTL) y las familias MOS (NMOS, CMOS y PMOS).

8. BIBLIOGRAFÍA

- Floyd, T. **Fundamentos de sistemas digitales**. Editorial Prentice Hall.
- Tocci, Ronald J. **Sistemas digitales: Principios y aplicaciones**. Editorial Prentice Hall.
- Prieto, A. **Introducción a la informática**. Editorial McGraw-Hill.
- www.designworkssolutions.com (LogicWorks)
- www.orcad.com (Orcad)
- www.microchip.com/mplab (MPLAB)



Preparador Informática

www.preparadorinformatica.com

TEMA 10. INFORMÁTICA / S.A.I.

**REPRESENTACIÓN INTERNA
DE LOS DATOS**

TEMA 10 INF / SAI: REPRESENTACIÓN INTERNA DE LOS DATOS

1. INTRODUCCIÓN

2. REPRESENTACIÓN INTERNA DE LOS DATOS

2.1. REPRESENTACIÓN DE INSTRUCCIONES

2.2. REPRESENTACIÓN DE TEXTOS

2.3. REPRESENTACIÓN DE SONIDOS

2.4. REPRESENTACIÓN DE IMÁGENES

2.5. REPRESENTACIÓN DE NÚMEROS

2.5.1. NÚMEROS ENTEROS

2.5.2. NÚMEROS REALES

3. RECURSOS Y HERRAMIENTAS EDUCATIVAS DE INTERÉS

4. CONCLUSIÓN

5. BIBLIOGRAFÍA

Preparador Informática

1. INTRODUCCIÓN

Un computador es una máquina que procesa, memoriza y transmite información. Esta información se da en la forma de datos, que son conjuntos de símbolos utilizados para expresar o representar un valor numérico, un hecho, un objeto o una idea; en la forma adecuada para ser objeto de tratamiento.

La ejecución de un programa en un computador implica la realización de unos tratamientos con unos datos. Para que el computador pueda ejecutar un programa es necesario darle dos tipos de informaciones: las **instrucciones** que forman dicho programa y los **datos** con los que debe operar. La elección de qué tipos de datos e instrucciones se pueden utilizar, así como qué representación van a tener, son las decisiones más importantes y decisivas que deben tomar los diseñadores de un nuevo computador, ya que son clave para poder diseñar sus componentes estructurales.

El presente tema está dedicado a estudiar los aspectos relacionados con la representación interna de los datos en los computadores, describiendo las diferentes codificaciones atendiendo al tipo de dato a representar (datos alfanuméricos, datos numéricos, etc.)

2. REPRESENTACIÓN INTERNA DE LOS DATOS

Relacionado con cómo se representa la información con la que trabaja un computador podemos considerar que existen dos niveles en la representación de la información:

- Nivel de **representación externa**: usada por las personas e inadecuada para el ordenador.
- Nivel de **representación interna**: adecuada al ordenador y no inteligible directamente para las personas.

A continuación, se describe cómo la información expresada en el nivel de representación externa es representable en los computadores (representación interna). Este paso de una representación a otra se denomina codificación y el proceso inverso decodificación.



Los computadores internamente representan la información usando una lógica binaria, es decir, su alfabeto tiene únicamente dos símbolos: 0 y 1. Esto se debe a que son más fiables y más sencillos de construir al contemplar únicamente dos valores.

Dependiendo del tipo de información que se represente se utilizarán diferentes representaciones. Se pueden diferenciar los siguientes tipos de representaciones:

- Representación de instrucciones
- Representación de textos
- Representación de sonidos
- Representación de imágenes
- Representación de datos numéricos

2.1. REPRESENTACIÓN DE INSTRUCCIONES

Las instrucciones de un programa en lenguaje de alto nivel se dan en forma de texto, representándose por medio de un código de E/S, y el traductor se encarga de transformarlas en instrucciones máquina, que son patrones de bits determinados por el repertorio de instrucciones del procesador.

En general, una instrucción máquina se divide en dos campos:

CÓDIGO OPERACIÓN	DATOS DE LA OPERACIÓN
------------------	-----------------------

- **Código de operación:** indica la operación a realizar.
- **Datos de la operación:** contiene los datos para realizar la operación (operandos). En función del modo en que se incluyen los operandos en la instrucción podemos hablar de cinco tipos de direccionamiento:
 - Implícito
 - Inmediato
 - Directo
 - Indirecto
 - Relativo

2.2. REPRESENTACIÓN DE TEXTOS

La información se suele introducir en el computador utilizando el lenguaje escrito cuyas unidades básicas son caracteres. Los caracteres se suelen agrupar en 5 grupos:

- **Caracteres alfabéticos:** corresponde a las letras mayúsculas y minúsculas del abecedario inglés.

$a, b, \dots, z, A, B, \dots, Z$

- **Caracteres numéricos:** corresponde a las diez cifras decimales.

$0, 1, \dots, 9$

- **Caracteres especiales:** son símbolos ortográficos y matemáticos no incluidos en los grupos anteriores.

$), (, !, \%, +, -, \dots, \text{etc.}$

- **Caracteres gráficos:** son símbolos con los que se pueden representar figuras, dibujos, etc.

♦♥♠♣

- **Caracteres de control:** se corresponden con las órdenes de control (carácter de salto de página, comienzo de línea, etc.)

Debido a que en la representación interna de la información en los computadores solo disponemos de ceros y unos, la comunicación hombre-máquina se debe establecer a través de una correspondencia entre el conjunto de todos los caracteres, $\alpha = \{a, \dots, z, A, \dots, Z, 0, \dots, 9\}$, y un conjunto $\beta = \{0, 1\}^n$.

Esta correspondencia establece lo que se conoce como código de entrada/salida. Al proceso de asignar a cada uno de los caracteres una secuencia de ceros y unos se le denomina codificación. En principio, esta correspondencia es arbitraria, pero existen códigos de E/S normalizados que son utilizados por diferentes diseñadores de computadores.

Algunos de los códigos entrada/salida más utilizados son:

- **EBCDIC** (*Extended Binary Coded Decimal Interchange Code*): utiliza 8 bits para representar cada carácter, por lo que puede representar hasta 256 caracteres.
- **ASCII** (*American Standard Code for Information Interchange*): su versión básica utiliza 7 bits y es capaz de representar hasta 128 caracteres. Existen versiones ampliadas que utilizan 8 bits. Entre ellas se encuentran los códigos ISO 8859-n, donde n es un número que identifica el juego de los nuevos caracteres introducidos. Por ejemplo, la **ISO 8859-1** (Latín-1) se proyectó para América y Europa occidental para incluir las vocales con tilde y otras letras latinas no usadas en los países anglosajones.
- **UNICODE**: creado en 1991 con el objetivo de solucionar los problemas relacionados con las limitaciones de los códigos existentes hasta ese momento. Existen diferentes formas de codificar Unicode: UTF-8, UTF-16 y UTF-32. Esta última utiliza todos los puntos posibles de Unicode usando 32 bits (4 bytes por carácter). UTF-16 y UTF-8 son codificaciones de longitud variable. Esto significa que, si un carácter se puede representar con un sólo byte, UTF-8 empleará sólo un byte. Si requiere dos bytes, usará dos, y así sucesivamente.

Preparador Informática

2.3. REPRESENTACIÓN DE SONIDOS

Una señal de sonido se capta por medio de un micrófono que produce una señal analógica. Para poder almacenarla y procesarla utilizando técnicas digitales se realiza un proceso de muestreo en el intervienen dos parámetros fundamentales:

- **Frecuencia de muestreo**: cuanto mayor sea la frecuencia a la que se muestrea mayor será la calidad del sonido grabado.
- **Precisión**: es el número de bits con el que se representa cada muestra.

Una vez obtenidos los valores binarios de las muestras hay que codificarlos de acuerdo con un determinado formato.

Hay diversos formatos de codificación, entre los que se encuentran:

- **PCM** (Pulse Code Modulation)



- DPCM (Differential Pulse Code Modulation)
- ADPCM (Adaptive Differential Pulse Modulation)
- μ - law
- MPEG
- Etc.

Cuanto mayor sea la frecuencia y la precisión mayor será la calidad del sonido, pero también será mayor el espacio de almacenamiento ocupado.

2.4. REPRESENTACIÓN DE IMÁGENES

Existen dos formas básicas de representar las imágenes:

A. Mapas de bits

La imagen está compuesta por puntos (píxeles), y a cada uno de ellos se le puede asociar su nivel de color. Para codificar y almacenar la imagen hay que tener en cuenta dos factores: número de píxeles y nivel de color asociado a cada píxel. La calidad de una imagen viene determinada por la resolución de la imagen (número de píxeles por línea x número de píxeles por columna).

Algunos de los principales formatos son: BMP, TIFF, JPEG, GIF, PNG, etc.

Preparador Informática

B. Mapas de vectores:

La imagen se descompone en una colección de objetos tales como líneas, polígonos y textos con sus respectivos atributos (grosor, color, etc.). Se utilizan en aplicaciones de diseño asistido por ordenador (CAD). Como desventaja presenta que su fidelidad respecto de la imagen real es inferior a la lograda con mapas de bits. Algunos de los principales formatos son: IGES, DxF, EPS, TrueType, etc.

2.5. REPRESENTACIÓN DE DATOS NUMÉRICOS

Si se va a realizar alguna operación aritmética, la representación de los datos numéricos como textos es inapropiada.

En este caso, los computadores utilizan representaciones específicas para valores numéricos.

Hay dos formas de representar los datos numéricos:

- Datos de tipo entero.
- Datos de tipo real.

2.5.1. NÚMEROS ENTEROS

Para la representación de números enteros se distinguen dos formas de representarlos en el interior del computador: representación binaria y representación BCD, distinguiéndose dentro de cada una de ellas varios métodos distintos de codificación.

1) Representación binaria

- *Enteros sin signo (valor absoluto)*: todos los bits del dato representan el valor del número expresado en binario natural. Con n bits se pueden representar 2^n números.
- *Enteros con signo*
 - Signo y magnitud: se reserva el bit más significativo para indicar el signo (0 para positivos y 1 para negativos), y el resto de bits indican la magnitud.
 - Complemento a 1: la representación de un número positivo en complemento a uno es igual que con el sistema signo-magnitud, mientras que la representación de un número negativo consiste en invertir los bits del valor positivo.
 - Complemento a 2: la representación de un número positivo en complemento a dos es igual que con el sistema signo-magnitud, mientras que la representación de un número negativo se calcula obteniendo en primer lugar el complemento a 1 y sumándole 1 al resultado obtenido.

- o Sesgada (o en exceso): todos los valores se representan sumándole un sesgo o exceso. Se suele tomar como sesgo $S = 2^{n-1}$

		(1 bit)	(n-1 bits)
Signo y magnitud:		0/1	Valor absoluto de N
Complemento a 1:	$N > 0$	0	Valor absoluto de N
	$N < 0$	1	Complemento a 1 de N
Complemento a 2:	$N > 0$	0	Valor absoluto de N
	$N < 0$	1	Complemento a 2 de N
Sesgado:	$S = 2^{n-1}$		$N + S$

Valores máximos y mínimos que pueden representar con n bits cada representación:

		(1 bit)	(n-1 bits)
Signo y magnitud:	$2^{n-1} - 1$	0	11111...1111
	$-(2^{n-1} - 1)$	1	1111...1111
Complemento a 1:	$2^{n-1} - 1$	0	11111...1111
	$-(2^{n-1} - 1)$	1	1000...0000
Complemento a 2:	$2^{n-1} - 1$	0	1111...1111
	-2^{n-1}	1	1000...0000
Sesgado:	$2^{n-1} - 1$		1111...1111
	-2^{n-1}		0000...0000

Ejemplo: Números enteros representables con $n=4$

Nº decimal	Sin signo	Signo y magnitud	Complemento a 1	Complemento a 2	Sesgado
7	111	0111	0111	0111	1111
6	110	0110	0110	0110	1110
5	101	0101	0101	0101	1101
4	100	0100	0100	0100	1100
3	011	0011	0011	0011	1011
2	010	0010	0010	0010	1010
1	001	0001	0001	0001	1001
0	000	0000	0000	0000	1000
-0		1000	1111		
-1		1001	1110	1111	0111
-2		1010	1101	1110	0110
-3		1011	1100	1101	0101
-4		1100	1011	1100	0100
-5		1101	1010	1011	0011
-6		1110	1001	1010	0010
-7		1111	1000	1001	0001
-8				1000	0000

2) Representación BCD

Se representan codificando individualmente cada cifra decimal en binario con 4 bits. Existen 3 codificaciones posibles:

- BCD natural: los pesos de cada posición son: 8-4-2-1
- BCD Aiken: los pesos de cada posición son: 2-4-2-1
- BCD exceso a 3: se codifica en BCD natural y se suma 3 al resultado.

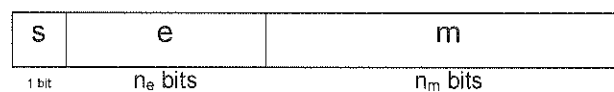
Número Decimal	BCD natural	BCD Aiken	BCD exceso a 3
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	0101	1000
6	0110	0110	1001
7	0111	0111	1010
8	1000	1110	1011
9	1001	1111	1100

2.5.2. NÚMEROS REALES

Para la representación de números reales, se utiliza la notación en coma flotante de la siguiente forma: $N = M \times B^E$ donde M: mantisa, B: base, E: exponente

Ejemplo: $N = 1,07653 \times 10^{12}$

En la actualidad, la mayoría de fabricantes de computadores siguen la representación normalizada IEEE754 para la representación de números reales, cuyo esquema es el siguiente:



Donde el significado de cada campo es:

Campo del signo (s)

$s=0$, si $N > 0$

$s=1$, si $N < 0$

Campo del exponente (e)

El exponente se almacena en la forma de entero sesgado, donde $S = 2^{n_e-1} - 1$ y

$e = E + S$

Campo de la mantisa (m)

Cuando el 1 más significativo de la mantisa se encuentra en la posición de las unidades el número se dice que está normalizado.

El campo de la mantisa solo almacena la parte fraccionaria del número normalizado, puesto que el 1 siempre estará (se dice que está implícito).

Existen varios formatos de IEEE 754:

- ✓ De 32 bits:
 - 1 bit para el signo, 8 bits para el exponente, 23 bits para la mantisa
- ✓ De 64 bits:
 - 1 bit de signo, 11 bits para el exponente, 52 bits para la mantisa
- ✓ De 128 bits:
 - 1 bit de signo, 15 bits para el exponente, 112 bits para la mantisa

Ejemplo: Representar el número real $A=87$ en el formato IEEE754 simple precisión (32 bits)

1º Pasamos el número decimal a binario natural:

$$A = 87_{10} = 1010111_2$$

2º Normalizamos el número:

$$A = 1010111_2 = 1,010111 \times 2^6$$

De modo que tenemos:

- Signo del n° : positivo
- Mantisa binaria: $M = 1,010111$
- Exponente con base binaria: $E = 6$

3º Obtenemos el exponente sesgado

$$A = 1,010111 \times 2^6$$

$$e = E + S = 6 + 127 = 133_{10}$$

Pasamos el exponente sesgado a binario

$$e = 133_{10} = 10000101_2$$

4º Formamos el número en la representación IEEE754

$$A = 1,010111 \times 2^6$$

- Como $A > 0$, $s = 1$
- Como $M = 1,010111$; $m = 010111$
- Exponente sesgado: $e = 10000101$

0	1000 0101	010 1110 0000 0000 0000 0000
s	e	m
1 bit	8 bits	23 bits

3. RECURSOS Y HERRAMIENTAS EDUCATIVAS DE INTERÉS

– **IEEE-754 Analysis**

Es una aplicación web desarrollada por el Queens College (Universidad de Nueva York) que permite examinar las relaciones entre los números binarios y decimales y los tres formatos de números descritos por el estándar de punto flotante IEEE-754-2008.

Sitio web: <https://babbage.cs.qc.cuny.edu/ieee-754/>

– **HexEd.it**

Es una aplicación web que permite subir archivos de texto plano guardados con diferentes formatos y muestra la codificación empleada para almacenar cada carácter

Sitio web: <https://hexed.it>

4. CONCLUSIÓN

En este tema se ha presentado una visión global de los aspectos fundamentales relacionados con la representación interna de los datos en un computador, ya que son clave para poder diseñar sus componentes estructurales.

Se han introducido y descrito los diferentes tipos de representaciones atendiendo al tipo de dato que se puede representar (representación de textos, sonidos, imágenes, datos numéricos)

5. BIBLIOGRAFÍA

- Prieto A., y otros. *Introducción a la informática*. Editorial McGraw-Hill
- De Miguel Anasagasti, P. *Fundamentos de los computadores*. Editorial Paraninfo.
- García de Sola, J. F. *Estructura de la información*. Editorial McGraw-Hill
- http://atc.ugr.es/APrieto_videoclases Departamento de Arquitectura y Tecnología de Computadores. Universidad de Granada.
- www.unicode.org (Consortio Unicode)
- <https://hexed.it>
- <https://babbage.cs.qc.cuny.edu/ieee-754/>