1 – LA INFORMACION Y SU REPRESENTACION

1.1 Sistemas de numeración

Para empezar a comprender cómo una computadora procesa información, debemos primero entender cómo representar las cantidades. Para poder cuantificar la información se desarrollaron los **sistemas de numeración** que pueden definirse de la siguiente manera:

Un **sistema de numeración** es el conjunto de símbolos y reglas que se utilizan para la representación de los datos numéricos o cantidades.

Un sistema de numeración se caracteriza fundamentalmente por su **base,** que es el número de símbolos distintos que utiliza, y además es el coeficiente que determina cuál es el valor de cada símbolo de acuerdo a la posición que ocupa.

Los sistemas de numeración actuales son **sistemas posicionales**; el valor relativo de cada símbolo (**cifra dígito***)* está dado por su posición respecto a la coma decimal y por la base.

La notación de un número N en base B es: $N_{(B}$
Su representación en cifras, cada una de ellas Ci en el rango 0..B-1:

$$C_nC_{n-1}\ldots C_2C_1C_0.C_{-1}C_{-2}\ldots C_{-(m-1)}C_{-m}$$

Algunos ejemplos son (todos fraccionarios):

El número en el sistema decimal con los dígitos 0,1,2,3,4,5,6,7,8,9	12395.235
El número en el sistema binario con los dígitos 0,1	11001.101
El número en el sistema hexagesimal con los dígitos 0..9,A,B,C,D,E,F	19FA3.5B

Al utilizar un sistema de numeración de base B, cuando sobrepasamos una cantidad a representar, más allá de B-1, aumentamos una cifra a la izquierda y continuamos contando desde la cifra más a la derecha comenzando por cero. Repetimos este proceso para las cifras en cualquier posición.

A modo de ejemplo consideramos un sistema ternario de base 3 con las cifras 0, A, # correspondientes a las cantidades 0,1,2:

-La cantidad nula se representa por 0
-La cantidad 1 por A
-La cantidad 2 por #
-Para la cantidad 3 (1 más que 2) aumento una cifra a la izquierda de # (en este caso esa cifra es 0 y no está explícitamente representada, o sea # es equivalente a 0#) y se representa entonces por A0.
- Las cantidades 4,5,6 son respectivamente AA, A#, #0
- La cantidad 7,8,9 son respectivamente #A, ##, A00

1.1.1 EL SISTEMA DECIMAL

Desde hace bastante tiempo, el hombre ha utilizado como sistema para contar el denominado **sistema decimal**, que derivó del sistema indoarábigo: posiblemente se adoptó este sistema por contar con diez dedos en las manos.

Consta de 10 cifras: 0 1 2 3 4 5 6 7 8 9

Dado un número decimal, este se puede descomponer en función de potencias de su base, esto es como suma de potencias de 10. Por ejemplo la interpretación de las cantidades 1994 y 3.1416 será:

$1994_{(10} = 1*10^3+9*10^2+ 9*10^1+4*10^0$
$3.1416_{(10}=3*10^0+1*10^{-1}+4*10^{-2}+1*10^{-3}+6*10^{-4}$

1.1.2 EL SISTEMA BINARIO Y HEXAGESIMAL

El **sistema binario** es el sistema de numeración que utilizan los circuitos digitales que configuran el hardware de las computadoras. Un microprocesador solo entiende el "lenguaje" binario.

La base o número de símbolos que utiliza el sistema binario es 2, siendo éstos los siguientes:

0 1

Cada cifra o dígito de un número representado en el sistema binario se denomina **bit** (contracción de binary digit). Un bit representa los dos posibles valores que puede tomar el estado de un sistema con dos posibilidades, como algunas compuertas lógicas que componen los circuitos digitales. Para las medidas de cantidades de información representadas en binario se utilizan una serie de múltiplos de 2 del bit que poseen nombre propio:

NIBBLE o cuarteto.	Es el conjunto de cuatro bits (1101)
BYTE (B) u octeto.	Es el conjunto de 8 bits, o sea 2 nibbles (10011100)
KILOBYTE (KB).	Son 1024 bytes (1024*8 bits)
MEGABYTE (MB).	Son 1024 kilobytes (1024^2*8 bits)
GIGABYTE (GB).	Son 1024 megabytes (1024^3*8 bits)
TERABYTES (TB).	Son 1024 gigabytes (1024^4*8 bits)
PETABYTES (PB).	Son 1024 terabytes (1024^5*8 bits)

La razón por la que se utiliza el factor multiplicador 1024 en lugar de 1000, como sucede en el sistema decimal, es por ser la potencia de 2 más próxima a 1000, cuestión importante desde el punto de vista interno de una computadora.

$$2^{10} = 1024$$

El **byte** es considerado la unidad básica de información de almacenamiento. Los dispositivos de almacenamiento, memoria, etc. se representan como potencias de 1024 bytes, como en la tabla de arriba.

Un conjunto de 4 bytes (32 bits), se denomina como **palabra** (word), y a uno de 8 bytes (64 bits) como **doble palabra** (double word). Estas definiciones son utilizadas porque están presentes en el diseño de hardware actuales y en las definiciones de ciertas variables y datos de los lenguajes de programación.

Al igual que los anteriores, el **sistema hexagesimal** es posicional de base 16. Este es sistema es utilizado en los lenguajes ensambladores. Sus símbolos son los siguientes:

0 1 2 3 4 5 6 7 8 9 A B C D E F

Los símbolos de la A a la F se corresponden con los decimales del 11 al 15.

1.1.3 CONVERSION ENTRE SISTEMAS

Para convertir un número representado en un sistema de base B arbitrario en "la cantidad" o el número decimal correspondiente, esto es:

$$N_{(b} \rightarrow N_{(10}$$

Si entendemos que multiplicar una cifra en un sistema de base B por la base, implica correr la cifra una posición a la izquierda agregando un 0 en la posición original, es fácil darse cuenta que las posiciones sucesivas de las cifras en el número representan cantidades decimales correspondientes a potencias de la base elevadas a la posición:

Posición de la cifra	3	2	1	0
Nro. Decimal correspondiente	B^3	B^2	B^1	B^0

Esto nos lleva a formular un resultado conocido como el teorema fundamental de la numeración:

El Teorema Fundamental de la Numeración (TFN) dice que el valor decimal de una cantidad expresada en otros sistemas de numeración, viene dado por la fórmula:

$$\ldots+C_3*B^3+C_2*B^2+C_1*B^1+C_0*B^0+C_{-1}*B^{-1}+C_{-2}*B^{-2}+\ldots$$

donde el número en base B es $\ldots C_3C_2C_1C_0.C_{-1}C_{-2}\ldots$

Ejemplos:

$201.1_{(3} = 2*3^2+0*3^1+1*3^0+1*3^{-1} = 18+0+1+0.333 = 19.333_{(10}$

$516_{(7} = 5*7^2+1*7^1+6*7^0 = 245+7+6 = 258_{(10}$

$0.111_{(2} = 1*2^{-1}+1*2^{-2}+1*2^{-3} = 0.5+0.25+0.125 = 0.875_{(10}$

$103_{(10} = 1*10^{2}+0*10^{1}+3*10^{0} = 100+0+3 = 103_{(10}$

1.2 REPRESENTACIONES NUMERICAS

Para representar los números (decimales), existen varios esquemas que codifican esos números como binarios. Esto es necesario pues las computadoras solo entienden el lenguaje binario. Nos interesa solo entender como se representan los enteros y los reales y cuales son los rangos para estos números cuando los usamos en las variables de un programa.

1.2.1 REPRESENTACION DE NUMEROS ENTEROS

La representación más elemental de un número entero sin signo es la de un **binario puro**. En esta representación, el número decimal se obtiene de convertir el binario mediante el TFN.

Con un byte podemos representar 2^8 cantidades. Con esta codificación representamos todos los enteros positivos desde el 0 al 255.

Con un palabra (32 bits) representamos 2^{32} enteros positivos, o sea el rango que va desde 0 a 4294697295.

Para representar números enteros con signo con una palabra de 32 bits, se utiliza la siguiente codificación: el bit de más a la izquierda (posición 31) se usa para representar el signo positivo o negativo de acuerdo a si es 0 o 1. Los 31 bits restantes representan el valor o módulo del número. El rango representado es:

$$-2^{31} <= X <= 2^{31}-1 \quad o \quad -2147486648 <= X <= 2147483647$$

EL mismo argumento vale para palabras de 16 o 64 bits. Podemos construir la tabla siguiente:

Tipo	Tamaño	Rango
Short integer (sin signo)	2 bytes	0..65,535
Short integer	2 bytes	-32.768..32,767
Integer (sin signo)	4 bytes	0..4,294,697,295
Integer (con signo)	4 bytes	-2,147,486,648..2,147,486,647

1.2.2 REPRESENTACIONES EN PUNTO FLOTANTE

La representación en punto flotante surge de la necesidad de de utilizar números reales y enteros con un mayor rango que el que nos ofrece la representación anterior. Utiliza la notación científica o exponencial para un número:

$$N = mantisa*(base\ de\ exponenciación)^{exponente}$$

La base de exponenciación es 2 según el estándar IEEE 754-1985 y es la comúnmente utilizada. La representación de punto flotante se define según el tamaño de la palabra, las más comunes son las siguientes:

a) Para precisión simple (en computadoras de 32 bits).

signo	Exponente	Mantisa
31	30 23	22 0

Los números más alejados de cero representables para simple precisión son aproximadamente:

$$-10^{38} <= X <= 10^{38}$$

Los números más pequeños representables son aproximadamente:

$$|X| >= 10^{-38}$$

b) Para doble precisión (en computadoras de 64 bits).

Signo	Exponente	Mantisa
63	62 52	51 0

Los números más alejados de cero representables para simple precisión son aproximadamente:

$$-10^{308} <= X <= 10^{308}$$

Los números más pequeños representables son aproximadamente:

$$|X| >= 10^{-308}$$

1.3 CODIFICACION ALFANUMERICA

Una computadora puede trabajar internamente con un conjunto de caracteres que nos permite manejar datos, informaciones, instrucciones, etc. Este conjunto de caracteres se puede subdividir en los siguientes grupos:

- Caracteres alfabéticos: letras mayúsculas y minúsculas.
- Cifras decimales: los números del 0 al 9
- Caracteres especiales: el punto (.), el asterisco (*), órdenes de control (ACK, CR, etc.)
-

En general, cada carácter se codifica internamente en una computadora con un conjunto de 8 bits. Uno de los más utilizados es el código **ASCII** (American Standard Code for Information Interchange) extendido.

2 ARQUITECTURA DE COMPUTADORES

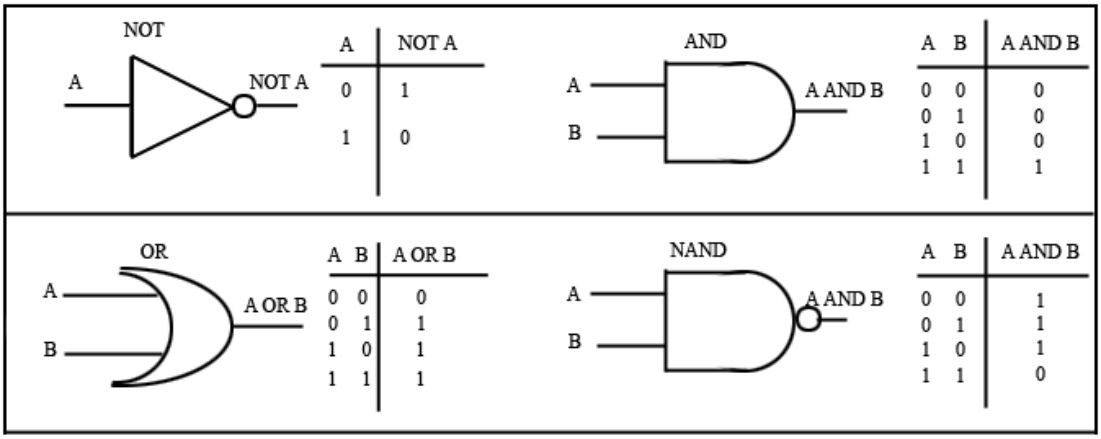
2.1 CONCEPTOS DE ELECTRÓNICA DIGITAL BÁSICA

Se puede analizar los circuitos de naturaleza electrónica con los que se construye los microprocesadores desde dos niveles: el electrónico y el lógico.

Los componentes básicos de los circuitos digitales son entre otros, resistencias, condensadores, diodos y transistores, en su mayoría implementados en **circuitos integrados**; es decir, miniaturizados e introducidos en pequeñas cápsulas.

El diseño en el **nivel electrónico** permite establecer relaciones entre las tensiones de corriente, que combinadas entre sí producen estructuras con propiedades lógicas elementales. Las señales eléctricas que circulan por los circuitos pueden utilizar distintas tensiones que se asocian con los dos valores del sistema binario.

Desde el punto de vista de la lógica; es decir en el **nivel lógico**, los elementos de los circuitos formados por componentes del nivel electrónico, están organizados en **puertas lógicas**, estructuras capaces de realizar funciones elementales de **lógica booleana**.

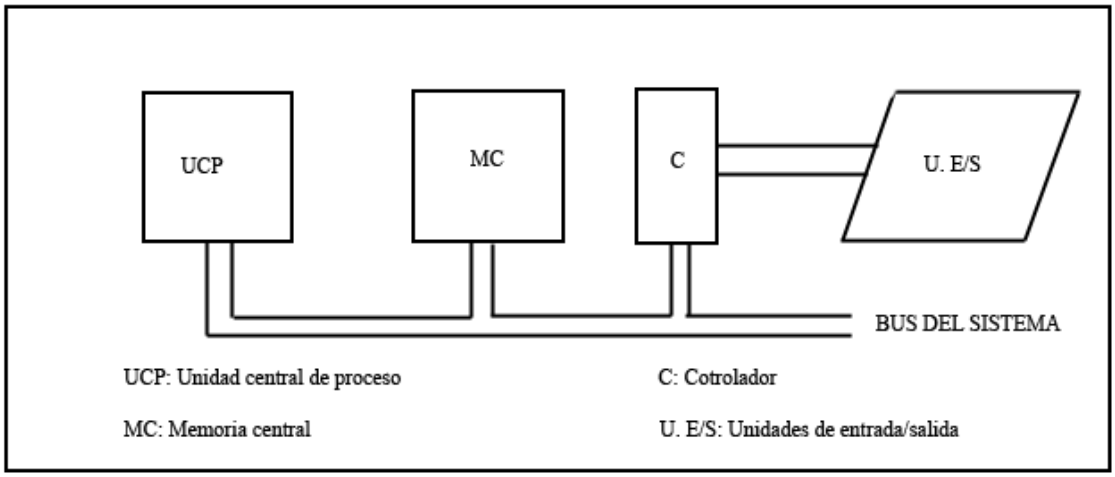


En base a estas puertas lógicas elementales se diseñan estructuras jerárquicas más complejas: sumadores, multiplicadores, funciones trigonométricas, etc.

2.2 COMPONENTES BÁSICOS DEL HARDWARE

El hardware de una computadora se puede estructurar en tres partes diferenciadas: La **Unidad Central de Proceso** (UCP) o microprocesador, la **Memoria Central** (MC) y las unidades de **Entrada/Salida** (E/S) o periféricos.

Se completa el esquema básico con el **Bus del Sistema** y los **Controladores**. El bus del sistema es un canal de comunicación entre todas las unidades. Los controladores son procesadores especializados en las operaciones de entrada/salida.



La UCP, MC, controladores y el bus del sistema, actualmente vienen integrados en una placa denominada **placa base** (motherboard).

2.3 LA UNIDAD CENTRAL DE PROCESO

La **Unidad Central de Proceso** (UCP) es el verdadero cerebro de la computadora. Su misión consiste en controlar y realizar todas las operaciones del sistema.

Físicamente está formado por circuitos de naturaleza electrónica que se encuentran integrados en una pastilla o **chip** denominada microprocesador. A su vez la UCP está compuesta por las dos siguientes unidades:

- La Unidad de Control (UC)
- La Unidad Aritmético-Lógica (UAL)

La UC es el centro nervioso, donde se controlan y gobiernan todas las operaciones. Sus elementos básicos internos son:

- Contador de Programa (CP)
- Registro de Instrucción (RI)
- Decodificador (D)
- Reloj (R)
- Secuenciador (S)

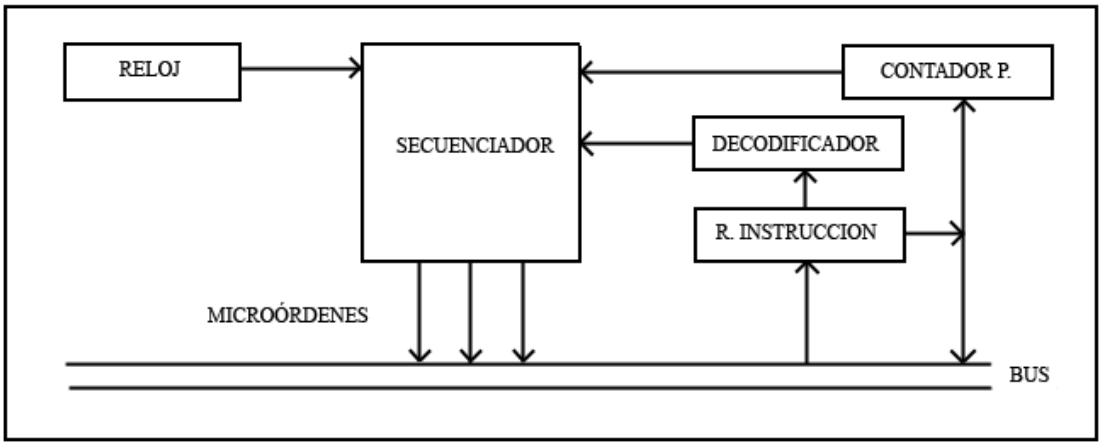
El contador de programa contiene permanentemente la dirección de memoria de la siguiente instrucción a ejecutar. Al iniciar la ejecución de un programa toma la dirección de la primera instrucción. Luego de que se recibe la instrucción en la UC, incrementa su valor a la dirección de memoria de la próxima instrucción.

El registro de instrucción contiene la instrucción que se está ejecutando en cada momento en la UCP. Esta instrucción llevará consigo el **código de operación** (CO) y los operandos o las direcciones de memoria de los mismos.

El decodificador se encarga de extraer CO de la instrucción en curso (en RI), lo analiza y emite las señales necesarias al resto de los elementos para su ejecución a través del secuenciador.

El reloj proporciona una sucesión de impulsos eléctricos o ciclos de intervalo constantes, que marcan los instantes en que han de comenzar los distintos pasos para ejecutar la instrucción. Actualmente la frecuencia de un UCP típica es del orden de pocos GHz.

En el secuenciador se generan órdenes muy elementales (microinstrucciones) que, sincronizadas por el reloj, hacen que se vaya ejecutando poco a poco la instrucción cargada en RI. Éstas microinstrucciones pasan a la UAL para efectuar el cálculo definitivo, una vez hecho esto se incrementa CP apuntando a la siguiente posición de la memoria central donde está la próxima instrucción.



En todas estas operaciones hablamos de “registros”. Los registros son áreas pequeñas de almacenamiento dentro de la UCP. Es donde de cargan como vimos las instrucciones y

los datos parciales a medida que se efectúan el proceso de lectura y ejecución de la instrucción.

Las UCPs también poseen una memoria interna llamada **memoria cache**. El tamaño actual de estas memorias es de unos pocos MB.

Una propiedad común de los programas de computadoras es el **localismo de referencia**: los mismos datos e instrucciones se acceden frecuentemente durante un intervalo de la ejecución. Es por esto que es importante mantener estos datos frecuentes en la memoria cache y en los registros, ya que su acceso para la ejecución es mucho más rápido que de memoria central.

2.4 LAS INSTRUCCIONES

Las instrucciones que es capaza de ejecutar la UCP se denominan **instrucciones de máquina**. El lenguaje que se utiliza para su codificación es el **lenguaje de máquina**.

Una instrucción de máquina tiene una estructura básica similar a la siguiente:

Código de Op.	Operando 1	Operando 2	Operando 3

El código de operación indica que operación se debe realizar. Los operandos contienen la información necesaria para poder realizar la operación. Hay instrucciones sin operando, con 1, 2 o más operandos. Se utiliza una o más palabras de máquina para codificar una instrucción. Una instrucción de máquina podría estar codificada por la palabra de 32 bits:

10000010	10000000	01111111	11111100

3 SISTEMAS OPERATIVOS Y PROGRAMAS DE COMPUTADORAS

En un sistema informático, para que el hardware o parte material pueda realizar el trabajo para el que ha sido construido, es necesario tener un conjunto de normas y órdenes que coordinen todos los procesos que se realizan. Este conjunto de órdenes se denomina **software** o parte inmaterial del sistema.

Todos los programas que conforman el software, pueden ser divididos en dos grupos bien diferenciados según su función:

- Software de sistema: compuesto por el conjunto de programas imprescindibles para el funcionamiento del hardware, facilitar el uso del sistema y optimizar sus recursos.

- Software de aplicación: Es el conjunto de programas que se desarrollan para que una computadora realice cualquier trabajo controlado por el usuario.

Un **sistema operativo** (SOP) es un conjunto de programas y funciones que controlan el funcionamiento del hardware ocultando sus detalles, ofreciendo al usuario una vía sencilla y flexible de acceso a la computadora.

Una computadora posee un conjunto de elementos que denominamos **recursos** como el procesador, la memoria interna, la entrada/salida y la información interna. Es tarea del SOP el administrar estos recursos.