

TEMA 16 : ELECTRONICA DIGITAL

Es aquella que se ocupa del proceso y tratamiento de señales discretas que toman un número limitado y finito de valores (generalmente dos, en cuyo caso se llaman señales BINARIAS) , en contraposición a los analógicos en los que las señales son continuas y por lo tanto pueden tomar un número infinito de valores.

1. SISTEMAS DE NUMERACIÓN Y CÓDIGOS.

Un sistema de numeración es un conjunto de reglas que permiten mediante un número limitado de símbolos representar todos los números posibles. Al número de símbolos distintos (DÍGITOS) que utiliza en la representación un sistema de numeración se le denomina BASE. A las reglas que aplicadas sobre los dígitos del número nos permiten conocer su valor es lo que se denomina CODIGO. Los sistemas de numeración utilizados son en general POSICIONALES, lo que quiere decir que el orden de magnitud o peso que tiene cada dígito se debe a la posición relativa del mismo dentro del número. El sistema de numeración empleado en la vida ordinaria es el

DECIMAL, en el cual la Base es 10 (los dígitos son 0,1,2,3,4,5,6,7,8 y 9) y en el que cada número se representa por un conjunto de dígitos ordenados cuyo valor es el de la suma de los términos del polinomio (polinomio equivalente) que tiene como coeficiente cada uno de los dígitos del número multiplicados por la potencia que resulta de elevar la base al ordinal de la posición de cada dígito empezando por 0 la primera a la izquierda de la coma (Ejemplo $128,3 = 1 \cdot 10^2 + 2 \cdot 10^1 + 8 \cdot 10^0 + 3 \cdot 10^{-1}$)

1.1. SISTEMA BINARIO.

Es aquel que utiliza únicamente 2 dígitos (0 y 1) en la representación, y tiene como base, por lo tanto, 2. En electrónica digital a la unidad MINIMA de información, que coincide por lo tanto con un dígito, se le denomina BIT.

Al bit que está situado más a la derecha, y que tiene por lo tanto menor peso se le denomina **LSB** (less significant bit o bit menos significativo), mientras que el situado mas a la izquierda es el **MSB** (more significant bit o bit más significativo). Según la base usada se distinguen :

1.1. **CODIGOS BINARIOS.** Utilizando una base de 2 dígitos (0 y 1) se pueden seguir diferentes reglas para representar un número, de donde se obtienen los distintos sistemas de codificación binarios. Los más importantes son :

CODIGO BINARIO NATURAL. Es la representación directa por un sistema equivalente al decimal, pero en base dos, del número decimal que representa el valor del mismo. Tiene la ventaja de que utiliza al MAXIMO las posibilidades de codificación con n dígitos (es decir que permite representar el máximo número posible de números con n dígitos, esto es 2^n) Para pasar un número binario natural a decimal se utiliza el desarrollo explicado más arriba (Ejemplo $1001101_{(2)} = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 64 + 1 \cdot 8 + 1 \cdot 1 = 73_{(10)}$)

Para pasar un número decimal a binario (o a cualquier otra base) se realizan divisiones sucesivas de la parte entera utilizando como divisor siempre la base y como dividendo inicialmente el número y luego los sucesivos cocientes, de manera que los restos sucesivos van formando el número en la nueva base, del menos significativo (el primer resto) al más significativo (el último COCIENTE). Las divisiones finalizan cuando se obtenga por primera vez un cociente inferior a la base, cociente que como hemos dicho constituye el dígito más significativo o de mayor peso. Si tiene parte decimal en lugar de dividir por la base se multiplica por ella, y se va colocando a la derecha de la coma la parte entera de los sucesivos productos hasta que desaparezca la parte decimal, si esto llega a producirse.

CODIGO DECIMAL CODIFICADO EN BINARIO (BCD). Se codifica en binario natural cada uno de los dígitos decimales independientemente. Para codificar del 1 al 10 en binario se necesitan al menos 4 dígitos binarios ya que $10^3 = 8$ (no hay bastantes) y $10^4 = 16$ números

En cuyo caso hay 6 combinaciones que no se utilizan o no tienen significado. Por lo tanto los códigos BCD necesitan en general más dígitos binarios para representar el mismo número que el código binario natural. Las 3 variantes más usuales de códigos BCD son :

— Códigos ponderados :

— **BCD natural** : Los pesos de cada dígito son los usuales 8,4,2 y 1 ($2^3, 2^2, 2^1$ y 2^0) .
Así por ejemplo el número 9 será en BCD natural $1001 = 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1$

— **AIKEN** : Los pesos son 2,4,2 y 1 por lo que el 9 de antes será $1011 = 2 \cdot 2 + 0 \cdot 4 + 2 \cdot 2 + 1 \cdot 1$. (La razón de ser de éste código, para el que le parezca extraña, es que así se utiliza el mismo número de 0 que de 1 para representar los 10 dígitos decimales, mientras que en el anterior se utilizarán más ceros que unos)

— Códigos no ponderados :

— **Exceso a tres** : se asigna a cada número decimal el binario natural correspondiente sumándole a continuación tres. De esta forma los números van de 3 al 13 y al igual que el aiken se necesita el mismo número de ceros que de unos para representar los 10 dígitos decimales. Ejemplo, $9 = 1001 + 0011 = 1100$

CODIGO GRAY . Pertenece a la categoría de códigos reflejados. Su característica es cada número y su sucesivo se diferencian única y exclusivamente en un bit. Para ello, según el número total de dígitos a emplear, se construyen tablas sucesivas y se colocan por filas y columnas grupos que difieren entre sí en un solo dígito, y partiendo del 00...00, se numeran las celdas sucesivas de arriba abajo y de izquierda a derecha de forma serpenteante. La secuencia obtenida se puede usar para construir una nueva tabla

1.2. **OCTAL**. Es un sistema de numeración que tiene por base 8.

1.3. **HEXADECIMAL** Es un sistema de numeración en base 16. Como solo existen 10 símbolos numéricos, para representar los restantes 6 dígitos hasta 16 se utilizan las letras A=10 B=11 C=12 D=13 E=14 y F=15. La conversión de un número hexadecimal a decimal se hace por medio del polinomio equivalente igual que siempre, y para pasar de decimal a hexadecimal se utilizan las divisiones sucesivas. Es el sistema de codificación más empleado en los microprocesadores. Como 16 números se pueden representar por medio de 4 bits binarios, la conversión de binario a hexadecimal y viceversa es sencilla : se divide el número binario en grupos de 4 dígitos, y cada grupo se substituye por el correspondiente dígito hexadecimal. Es precisamente por esto que es el más usado en los ordenadores.

CODIGOS DETECTORES Y CORRECTORES DE ERROR.

2. ARITMÉTICA BINARIA.

Al igual que existen reglas para realizar las operaciones aritméticas entre números expresados en el sistema decimal, existen a su vez otras específicas para el caso en que los números se representen en código binario natural. Nos interesan la suma y la resta, ya que la multiplicación y la división pueden obtenerse a partir de aquellas.

SUMA BINARIA. Las reglas son completamente análogas al caso de los números decimales. Se colocan los números a sumar uno debajo del otro de manera que coincidan en la misma columna dígitos de igual peso. Se empieza a sumar por la derecha, de forma que $0+0=0$; $0+1=1$, $1+0=1$ y $1+1=10$. En éste último caso se coloca el 0 y el 1 situado a la derecha se **ACARREA** para sumarlo en la columna anterior siguiendo la misma regla. Si el resultado es 10 u 11, se acarrea para sumarlo a la columna anterior.

RESTA BINARIA. La resta podría hacerse de la misma forma que en decimal, pero ello obligaría a construir circuitos muy distintos para una u otra operación, por lo que se prefiere reducirlo a una suma que consiste básicamente en sumar al minuendo complementario u opuesto del sustraendo, precedidos ambos por un bit extra, denominado, bit de signo que indica con un 0 quien es el minuendo (positivo) y con un 1 el sustraendo (negativo) . Existen dos sistemas principales para obtener el complementario :

COMPLEMENTO A 2. Consiste en intercambiar los unos por ceros y los ceros por unos, sumándole un 1 al final. El número obtenido es la diferencia entre dos elevado al número de dígitos de la parte entera del número y el propio número. Así, Por ejemplo el complemento a 2 del 12 = 1100 es $-12 = 0011+1 = 0100$. Si tiene parte decimal se procede de la misma forma $12,5 = 1100,1$ es $-12,5 = 0011,0 + 1 = 0011,1$.

Para restar se les añade a los números el bit de signo y se suman, incluyendo el bit de signo. Si la suma del bit de signo ocasiona acarreo, éste se desprecia. Si el número obtenido es

positivo, es directamente la diferencia en código binario. Si es negativo (tiene bit de signo a 1) entonces el número obtenido es el complemento a 2 del resultado, por lo que si quisieramos pasarlo a decimal, sería preciso obtener el número en binario natural, restandole uno e intercambiando los unos por ceros y los ceros por unos, con lo que se podría convertir según las reglas estudiadas, bien entendido que irá precedido del signo menos.

COMPLEMENTO A 1. Consiste simplemente en intercambiar los unos por ceros y los ceros por unos en el numero original. Lo que se obtiene el resultado de restar a dos elevado al número de dígitos de la parte entera dos elevado a menos el número de dígitos de la parte fraccionaria menos el número original.

Para restar se procede de forma análoga al caso anterior, con la diferencia que si al sumar los bit de signos se ocasiona acarreo, éste se suma al resultado al final. De igual forma, si el resultado tiene el bit de signo a 1 entonces el número obtenido no es el binario natural, sino el complemento a uno del binario natural, por lo que para pasarlo a decimal es preciso desacer el complemento, es decir, intercambiar ceros y unos.

3. ÁLGEBRA DE BOOLE

El álgebra de Boole es una herramienta matemática que se desarrolló inicialmente para representar y sistematizar el pensamiento lógico humano y en el que las variables pueden tomar únicamente dos valores, verdadero o falso, y que se denominan variables booleanas, por lo que su aplicación a la electrónica digital binaria es directa, pues esos dos estados se pueden representar por 0 y 1 . Según que la forma física en la que se realice la representación de estos estados se llega a dos tipos de lógica totalmente equivalentes :

Lógica positiva cuando el 1 se representa por la presencia de tensión y el 0 por la ausencia.

Lógica negativa, cuando el 1 se representa por la ausencia de tensión y el 0 por la presencia.

3.1. FUNCIONES LÓGICAS .

Una función lógica es aquella que toma valores binarios que dependen de una expresión algebraica formada por una serie de variables binarias relacionadas entre sí por diferentes operaciones. La función lógica se puede representar así mismo mediante una TABLA DE VERDAD que es una tabla en la que se colocan por columnas todas las variables binarias que intervienen en la expresión algebraica de la función, reservando la última para los valores resultantes de la misma, y cuyas filas contienen todas las posibles combinaciones de valores binarios que pueden tomar las distintas variables binarias de la función, colocando el valor que toma la función en cada caso en la última posición de la fila. El número de filas de una tabla de verdad será por lo tanto 2^n siendo n el número de variables binarias o entradas de la función.

3.2. OPERACIONES BÁSICAS

3.2.1. Suma lógica. Es aquella operación entre dos variables binarias que dá como resultado un 1 lógico siempre que al menos una de las variables tome el valor lógico 1, o lo que es lo mismo, que tiene por resultado siempre un 1 salvo en el caso en el que las dos entradas sean 0. Corresponde a la función lógica OR y se representa por el signo + .

3.2.2. Producto lógico. Es la que dá como resultado un 1 solo cuando las dos entradas son 1, siendo 0 en todos los demás casos. Corresponde a la función lógica AND y se representa por un punto, o por la ausencia de signo entre dos variables

3.2.3. Complemento o negación. Es una operación que se realiza sobre una única variable de entrada y que dá como resultado el valor opuesto o complementario de la misma, esto es, si la variable vale 1 la negación de la misma da un 0 y viceversa. Se representa por medio de una raya sobre la variable.

3.3. POSTULADOS, PROPIEDADES Y TEOREMAS

3.3.1. Postulados

3.3.2. Propiedades

3.3.3. Teoremas

3.3.3.1. Absorción

3.3.3.2. De Morgan

4. PUERTAS LÓGICAS.

Son dispositivos electrónicos que aceptan una o varias entradas digitales binarias y que tras realizar operaciones de aritmética booleana sobre ellas producen una señal de salida determinada en función de las operaciones realizadas por la puerta con la o las entradas. Las puertas lógicas elementales son las que realizan las operaciones básicas indicadas anteriormente, esto es la suma lógica (puerta OR) , el producto lógico (puerta AND) y el complemento o negación (puerta NOT).

Existen así mismo otras 4 puertas lógicas muy empleadas, que se denominan puertas lógicas derivadas y que realizan funciones lógicas muy sencillas, pero que no son elementales pues se pueden obtener a partir de las básicas, pero que son muy utilizadas e interesantes :

- Puerta NOR : Combinación de una función OR y una NOT. La salida solo es 1 cuando ambas entradas son 0
- Puerta NAND : Combinación de una función AND y una NOT. La salida solo es 0 cuando ambas entradas son 1
- Puerta XOR (OR exclusivo) La salida solo es 1 cuando es 1 una y una sola de las entradas, siendo 0 cuando las 2 entradas son iguales (o las dos 0 o las dos 1). Esta operación tiene signo propio, pero se puede poner en función de las operaciones básicas de la siguiente manera :

$$a \oplus b = a\bar{b} + \bar{a}b$$

- Puerta XNOR (NOT OR exclusivo) La salida solo es 0 cuando es 0 una de las entradas y solo una. Si las dos son 0 o 1 entonces da 1.

5. OBTENCIÓN DE LA FUNCIÓN LÓGICA A PARTIR DE LA TABLA DE VERDAD

6. SIMPLIFICACIÓN DE FUNCIONES

6.1. Método de Karnaugh

7. IMPLEMENTACIÓN DE FUNCIONES CON PUERTAS NAND O NOR

TEMA 16 (bis) : CIRCUITOS COMBINACIONALES Y SECUENCIALES.

1. CIRCUITOS COMBINACIONALES

Son aquellos en los que en cada instante, el estado lógico de las salidas depende únicamente de la combinación de estados de sus entradas

1.1. CONVERSIÓN DE CÓDIGO :

Implementan una función lógica que pasa una cierta información de un código a otro.

1.1.1. **CODIFICADORES** : Tienen 2^n entradas y n salidas, de forma que la activación de una de las entradas (información sin codificar) produce como salida el número de orden (en decimal) de la entrada en código binario natural de n bits. Un ejemplo es el circuito que traduce la pulsación de una tecla en un número binario. Hay dos tipos

1.1.1.1. **Codificadores sin prioridad** : No puede activarse más que una entrada al mismo tiempo, no se emplean

1.1.1.2. **Codificadores con prioridad** : En caso de activarse simultáneamente varias entradas, se obtiene en la salida la de mayor valor.

1.1.2. **DECODIFICADORES** : Tienen n entradas y 2^n salidas. Su función es la inversa del anterior. Cuando se presenta en la entrada un número binario de n bits, se activa una o varias salidas de acuerdo a una determinada función lógica de conversión : Hay muchos tipos, pero destacaremos 2

1.1.2.1. **Salida 1 de N** : Cada número binario representado en las entradas, activa una sola salida cuyo número de orden corresponde a dicho número en decimal. Una aplicación es el caso del accionamiento de un teletipo o impresora a la salida de un ordenador

1.1.2.2. **Decodificador BCD-7 Segmentos** Se utiliza para representar un número decimal de 0 a 9 en un display de 7 segmentos a partir de su correspondiente entrada en binario.

1.1.3. que tiene como número de orden

1.1.4. CONVERTIDORES DE CÓDIGO

1.2. SELECCIÓN DE CANAL

1.2.1. **MULTIPLEXADORES (MUX)**: Encaminan su única entrada hacia una de las 2^n salidas que posee de forma que su número de orden coincide con el valor numérico de una variable binaria que llega a través de n entradas de control

1.2.2. **DEMULTIPLEXADORES** Poseen 2^n entradas o selecciones, 1 salida única y n entradas de control o selección.

1.3. CIRCUITOS ARITMÉTICOS

1.3.1. SUMADOR

- Semisumador : Es el que realiza la suma de dos números binarios de un solo bit.
- Sumador total : Es el que realiza la suma de números binarios de varios bits. Resulta de modificar el anterior teniendo en cuenta la repercusión que en la suma de cada bit tienen los acarros de las sumas de los anteriores.

1.3.2. RESTADOR

2. CIRCUITOS SECUENCIALES

- Son aquellos cuya salida depende no solo de la combinación de valores de las entradas, sino también de la salida anterior, en otras palabras tienen memoria del estado anterior
- Todos los circuitos secuenciales se realizan a partir de circuitos biestables (o flip-flops) implementados con las puertas lógicas estudiadas hasta ahora, es decir, circuitos que pueden encontrarse en dos estados diferentes que se mantienen indefinidamente hasta que se aplique una nueva entrada o activación. Se dividen en

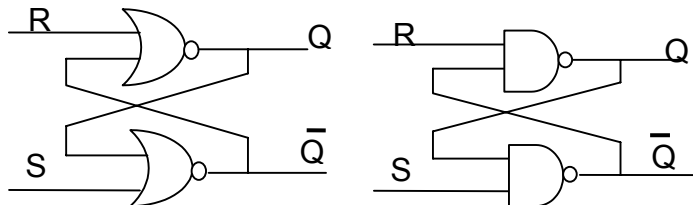
2.1. ASÍNCRONOS :

Son aquellos en los que el cambio de estado se produce inmediatamente después de aplicarles las entradas, con el tiempo de retardo característico que tenga el circuito, de manera que si hay varios circuitos, cada uno cambia a su aire pudiendo llegar una señal nueva a uno de ellos antes de que haya terminado de procesar la anterior. Los principales tipos son

2.1.1. **BIESTABLE R-S** (Reset-Set) Es el biestable básico. Tiene dos entradas (R y S) y dos salidas (Q y Not Q) La activación de la entrada R (Reset a 1) produce un 0 a la salida, mientras que la activación de la S (Set a 1) produce un 1 a la salida. Si las dos entradas son 0, la salida no cambia (es cero o uno según fuera en el estado anterior. El inconveniente es que si se activan las dos, la salida es indeterminada, dependiendo del tipo de puerta lógica con la que se implemente el biestable :

- Borrado prioritario : R y S a 1 implica 0 a la salida. Con puertas NOR
- Inscripción prioritaria : R y S a 1 implica 1 a la salida. Con puertas NAND

Esta implementación del biestable no es imprescindible recordarla, ya que se puede



obtener a partir de la tabla de verdad de la función lógica.

R-S			
R	S	Q_T	Q_{T+1}
0	0	X	X
0	1	X	1
1	0	X	0
1	1	?	?

J-K			
J	K	Q_T	Q_{T+1}
0	0	X	X
0	1	X	1
1	0	X	0
1	1	0	1
1	1	1	0

T			
T	Cik	Q_T	Q_{T+1}
0	0	X	X
0	1	X	X
1	0	X	X
1	1	0	1
1	1	1	0

D			
D	Cik	Q_T	Q_{T+1}
0	0	X	X
0	1	X	0
1	0	X	X
1	1	X	1

2.1.2. **BIESTABLE J-K**. Es un biestable R-S en el que se ha eliminado el defecto de funcionamiento cuando las dos entradas son 1, de manera que aquí con J y K simultáneamente a 1 el biestable cambia al estado opuesto al anterior.

2.1.3. **BISTABLE P-Q**

2.1.4. **BIESTABLE T** (Toggle) Tiene una entrada única, De forma que si es un 1 se produce un cambio de estado (si era 0 pasa a 1 y si era 1 pasa a 0) y si es un cero no cambia. Este tipo de biestables no se fabrican como tal, ya que se obtienen fácilmente a partir de los J-K sin mas que unir las dos entradas (J y K) en una sola (T) Tienen una aplicación inmediata para fabricar contadores.

2.2. SÍNCRONOS

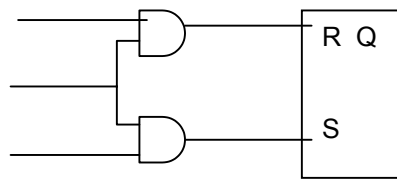
Son aquellos en los que no se produce el cambio de estado hasta que se recibe una señal de reloj, aunque cambien los valores de las entradas. De esta forma se consigue que todos los biestables cambien a la vez, ajustando el periodo entre impulsos de la señal de reloj al tiempo de retardo del circuito mas lento. Los biestables asíncronos vistos, se pueden sincronizar añadiendo una entrada de reloj (Clk)

2.2.1. ACTIVADOS POR NIVEL

El biestable funciona durante todo el tiempo que la señal de reloj está activa (nivel alto o nivel bajo según sea lógica positiva o negativa). El resto del tiempo se mantiene la señal anterior

2.2.1.1. BIESTABLE R-S SINCRONO

A partir del biestable R-S asíncrono se obtiene el síncrono sin más que añadir una nueva entrada, la de reloj, que se combina con las variables R-S mediante puertas AND



2.2.1.2. BIESTABLE D (Delay)

El biestable D tiene una entrada, una salida y una entrada de reloj. En cada impulso de reloj el biestable D da salida al valor almacenado en el ciclo anterior y se carga con el valor de la entrada.

2.2.1.3. BIESTABLE Latch D (Cerrojo) Es una variante del biestable D, en el que en lugar de una entrada de reloj tiene una entrada controlable, denominada "Enable", que permite actuar al biestable solo cuando su valor es 1, actuando como un "cerrojo" cuando la variable Enable es 0.

2.2.2. ACTIVADOS POR FLANCO

El biestable ejecuta su función sólo durante el tiempo que dura la transición de la señal de reloj de nivel bajo a nivel alto (por flanco de subida) o de nivel alto a nivel bajo (por flanco de bajada). La razón de su existencia es que en el caso de los activados por nivel, puede ocurrir que la combinación de entradas cambie por segunda vez durante el tiempo que dura la señal de reloj, si el tiempo de retardo del biestable anterior es menor que la duración del impulso de reloj, con lo que se perdería el primer cambio y el sistema se desincronizaría. En cambio, al ser el tiempo de subida o bajada muy corto, no es posible que ningún biestable tenga un tiempo de retardo inferior.

Con cualquiera de los biestables síncronos vistos, activados por nivel se pueden obtener los correspondientes activados por flanco sin mas que realizar una de las siguientes transformaciones

2.2.2.1. MASTER-SLAVE

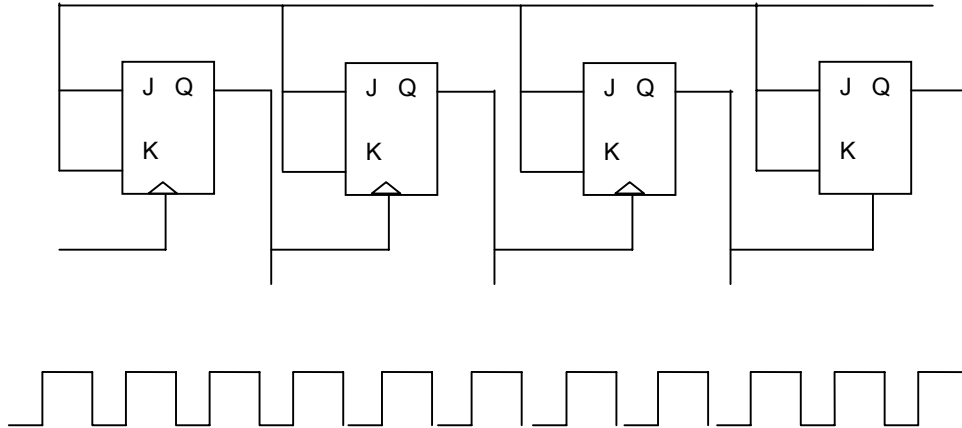
2.2.2.2. EDGE-TRIGGERED

2.3. APLICACIONES .

2.3.1. REGISTROS DE DESPLAZAMIENTO. Son dispositivos constituidos por una cadena de biestables que almacenan información binaria de manera que la transfieren de un biestable al siguiente en cada impulso del reloj.

2.3.2. REGISTROS DE MEMORIA

2.3.3. CONTADORES. Son dispositivos digitales que tienen una entrada de reloj y N salidas en las que representara en un determinado código, el número de impulsos de reloj que ha recibido el dispositivo.



TEMA 16 : EL ORDENADOR COMO SISTEMA DE CONTROL.

1. ESTRUCTURA FUNCIONAL DE LOS ORDENADORES

De forma general, podemos decir que un ordenador consta de 3 bloques :

- **Unidad de entrada** : Está formada por los dispositivos que introducen en el ordenador la información (instrucciones y datos) transformándola en señales binarias de naturaleza eléctrica. Ejemplo de ellas son el teclado, el ratón, el Scanner, el modem, lectores de códigos de barras o de tarjetas o cinta perforada, los lectores de disquetes o CD-ROM
- **Unidad de salida**. Es el dispositivo que comunica al exterior los resultados de los programas ejecutados. Transforman la información eléctrica binaria en otro tipo de señal, generalmente caracteres impresos o visualizables. Ejemplos son el monitor, la impresora, el plotter, un actuador de cualquier naturaleza, una perforadora de tarjetas o cinta, un modem.
- **Unidad de proceso**. Es la encargada de realizar sobre los datos las operaciones indicadas por las instrucciones de los programas para obtener los resultados deseados. Es el verdadero corazón del ordenador

2. HARDWARE Y SOFTWARE

Todos los elementos necesarios para el proceso de la información en un ordenador se dividen tradicionalmente en dos grupos atendiendo a su naturaleza :

2.1. HARDWARE (“Lo duro”)

Es la parte física del ordenador, constituida por todos aquello que es tangible, cables, chips, tarjetas de circuito impreso, fuente de alimentación, impresoras, monitores, teclados, ratones, etc.....

2.2. SOFTWARE (“Lo etéreo”)

Es la parte lógica intangible del ordenador, constituida por la información y los programas, las instrucciones y los datos, los algoritmos etc... No confundir conceptualmente con el soporte físico (disquetes, CD-ROM, memorias, etc...) sobre el cual forzosamente debe existir (el programa es el mismo independientemente que se encuentre escrito en un papel, en un disquete magnético o un disco óptico, en lenguaje fortran, pascal, C o ensamblador). Se divide en :

- Software de tratamiento
 - De programación
 - De aplicación
- Software de diagnóstico y mantenimiento

3. CPU.

La CPU (Central Processing Unit o Unidad Central de Proceso). Es el verdadero “cerebro” del ordenador, ya que es la que toma los datos e instrucciones, los examina, ejecuta las instrucciones, obtiene los resultados y los envía a la salida. Se compone de :

- Unidad de Control : Extrae las instrucciones de la memoria principal y determina su tipo
- Unidad Aritmético-Lógica (ALU) es la unidad operativa que realiza las operaciones indicadas por las instrucciones con los datos. Las operaciones aritméticas que puede realizar son la suma y la resta, y las lógicas AND, OR y NOT.
- Registros de trabajo : Un registro es una pequeña memoria de altísima velocidad en la que se almacenan temporalmente los datos que intervienen en una instrucción, los resultados parciales de las operaciones o alguna información de control del programa. Los más importantes son :
 - Contador de programa. Indica el número de instrucción que debe ejecutarse a continuación.
 - Registro de Instrucción.(RI) Contiene la información de la instrucción que se está ejecutando en este momento.

El proceso de trabajo de cada una de las partes de la CPU no es independiente, sino que están interrelacionados de la siguiente forma :

- 1º. La unidad de Control extrae de la memoria la siguiente instrucción indicada por el contador de programa y la almacena en el Registro de Instrucción
- 2º. Se incrementa el contador de programa de modo que señale a la siguiente instrucción
- 3º. Se determina el tipo de instrucción que se acaba de extraer (cada instrucción tiene un número de código que corresponde al juego de instrucciones que es capaz de interpretar la CPU)
- 4º. Si la instrucción requiere datos de la memoria, se determina la dirección en la que están situados
- 5º. Extrae de la memoria dichos datos y los carga en registros de trabajo
- 6º. Se ejecuta la instrucción utilizando la Unidad aritmético-lógica
- 7º. Se almacenan los resultados en el lugar apropiado (memoria o dispositivo de salida)

4. MEMORIA

La memoria es la parte del ordenador donde se almacenan los programas y datos en forma binaria. La unidad más pequeña de información binaria es el **bit** (un 0 o un 1). La memoria puede considerarse constituida idealmente como una tabla de celdas o un casillero, de forma que cada una de las celdas se puede almacenar un bit y está identificada por una dirección numérica.

- **Byte** : No obstante, como los humanos codificamos la información utilizando básicamente letras, dígitos y algunos símbolos, es necesario utilizar varios bits para poder representar los caracteres alfanuméricos, por lo que se ha establecido agrupaciones de 8 bits, denominadas bytes que permiten codificar hasta 256 signos distintos (código ASCII).
- **Palabra** : Según la naturaleza de cada ordenador, los bytes se agrupan formando palabras (8, 16, 32 o 64 bits) que representa el número de bits que puede procesar simultáneamente el ordenador, siendo más rápido y eficiente cuanto mayor sea el tamaño de la palabra que puede procesar.

La memoria se divide en dos tipos principales

- Memoria Central o Principal
- Memoria Auxiliar o Secundaria

5. MICROPROCESADOR

6. AUTÓMATAS