



# TEMA 1. ARQUITECTURA DE ORDENDES



## Contenido

1. Informática e información.....	2
1.1. Simbología y codificación. ....	3
1.2. Codificación. ....	3
2. Sistemas de numeración. ....	4
2.1. Conversión de una base cualquiera a base decimal. ....	5
2.2. Conversión de una base decimal a otra cualquiera. ....	6
2.3. Conversor de Hexadecimal/Octal ....	7
2.4. Operaciones aritméticas en binario: sumas y restas. ....	9
3. Unidades de medida de la información ....	10
3.1. Conversión entre las diferentes unidades.....	11
3.2. Representación interna de la información.....	12
4. Velocidades de transferencia .....	14
5. Conceptos básicos .....	15
Hardware.....	15
Software .....	15
Firmware .....	15
6. Arquitectura de Von Neumann .....	15
7. El procesador.....	16
7.1. Unidad de Control .....	16
7.2. Unidad Aritmético-Lógica.....	18
8. Memoria principal .....	19
9. Bus del Sistema .....	20

## 1. Informática e información

La **informática** es la ciencia tecnológica que estudia el tratamiento automático y racional de la información, con el fin de obtener de ella la máxima utilidad.

La informática usa los **ordenadores** para el tratamiento y el proceso de la información.

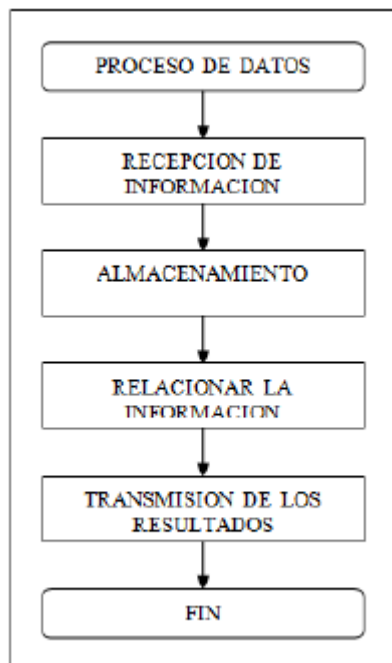
Podemos definir **Información** como la representación de hechos objetos valores ideas, etc. que permiten la comunicación entre emisor y receptor, y la adquisición del conocimiento de las cosas.

Un **sistema de comunicación** está formado por los siguientes elementos básicos:

**Emisor, fuente o transmisor:** genera o emite la Información.

**Receptor:** recibe la información.

**Medio o canal:** vía de transmisión de la información.



- Un motorista ve a lo lejos una señal de STOP. Es una información que recibe por el sentido de la vista (es una operación de lectura).
- Introduce en su memoria la señal de STOP.
- Relaciona la información obtenida con la que tenía en su memoria (código de circulación).
- El resultado de la comparación es que debe parar. Por ello transmite a sus pies y manos las órdenes necesarias para detener la moto.

### 1.1. Simbología y codificación.

- **Codificar:** Transformar unos datos a una representación predefinida y preestablecida.
- **Código:** Conjunto de condiciones y convenios que permiten transformar la información de una representación concreta a otra.
- **Un código está compuesto de:**
  - Un conjunto de reglas y convenios de transformación del alfabeto fuente.
  - Un nuevo alfabeto que sustituirá al original.
- **Código binario:** Está basado en el sistema de numeración binario, cuyos símbolos son el 0 y el 1.
- Los ordenadores trabajan con señales binarias. Por tanto, toda la información que se les proporcione debe estar codificada mediante una señal binaria. Por este motivo, los programas se traducen a un lenguaje de máquina, que está formado sólo por ceros y unos.

### 1.2. Codificación.

- **Números decimales → Binario**
- **Símbolos alfanuméricos → ASCII, EBCDIC, Unicode...**
- **Imágenes → JPEG, GIF, PNG,...**  
**Mapa de vectores: DXF, IGES, EPS, TrueType,...**
- **Sonido → WAV, MIDI, MP3,..**

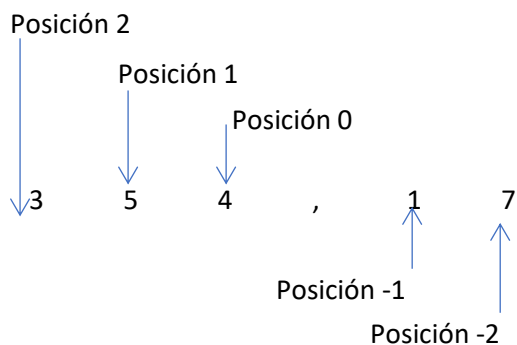
## 2. Sistemas de numeración.

Es un conjunto de símbolos utilizados para la representación de cantidades, así como las reglas que rigen dicha representación.

Un elemento importante en el sistema de numeración es la **base**, que determina el número de símbolos para representar los números. Así, el sistema más utilizado diariamente es el **sistema decimal**, que utiliza **10** caracteres. En la siguiente tabla se muestra los diferentes sistemas de numeración más representativos.

Sistemas	Base	Dígitos
Binario	2	0 1
Octal	8	0 1 2 3 4 5 6 7
Decimal	10	0 1 2 3 4 5 6 7 8 9
Hexadecimal	16	0 1 2 3 4 5 6 7 8 9 A B C D E F Siendo las letras la representación de los valores comprendidos entre el 10 y el 15 del sistema decimal.

Es importante destacar que en todos los sistemas tiene importancia la posición del dígito frente a la coma decimal:



Con tantos sistemas de numeración, nosotros podemos representar un número cualquiera en todos ellos, de tal manera que por ejemplo en octal tenga un valor y al pasarlo a decimal tenga otro. ¿Cómo pasamos un número de un sistema a otro? Mediante la conversión entre sistemas.

## 2.1. Conversión de una base cualquiera a base decimal.

Mediante el teorema fundamental de la numeración podemos convertir una cantidad expresada en cualquier base a base decimal:

$$\text{Número decimal} = \sum (\text{dígito})_i \times (\text{base})^i$$

Siendo  $i$  la posición del dígito respecto a la coma decimal.

Ejemplos:

**El número  $23'5_8$  que está en octal lo vamos a pasar a decimal:**

$$2 \times 8^1 + 3 \times 8^0 + 5 \times 8^{-1} = 16 + 3 + 0'125 = 19'625_{10}$$

**El número  $101'01_2$  que está en binario lo vamos a pasar a decimal:**

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 4 + 0 + 1 + 0 + 0.25 = 5'25_{10}$$

**El número  $B3'F_{16}$  que está en hexadecimal lo vamos a pasar a decimal:**

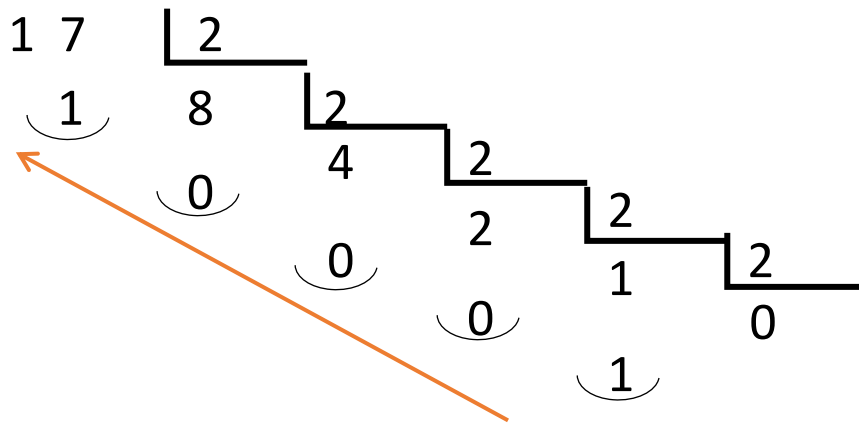
$$11 \times 16^1 + 3 \times 16^0 + 15 \times 16^{-1} = 176 + 3 + 0'9375 = 179'9375_{10}$$

## 2.2. Conversión de una base decimal a otra cualquiera.

La técnica más utilizada para convertir un número de decimal a otro sistema es multiplicar su parte decimal y dividir su parte entera por la base del sistema al que queremos convertir el número.

Vamos a convertir el número  $17,125_{10}$  a binario.

Lo primero es coger la parte entera (17) y dividirla por la base del sistema al que queremos convertir. En este caso al ser binario, lo dividimos por 2 (si fuera a octal por 8; hexadecimal por 16). Hay que ir dividiendo el cociente una y otra vez hasta que sea menor que el divisor (en nuestro caso 2)



La parte entera (17) en binario será los valores de los restos en orden inverso.

**1 0 0 0 1**

Lo siguiente es convertir la parte decimal del número (0,125). Para ello iremos multiplicando la cantidad por la base (2). Sobre la parte decimal del resultado volvemos a multiplicar por la base, y así una y otra vez hasta que o bien en una de las multiplicaciones el resultado no tenga parte decimal (es decir, sólo tenga parte entera) o veamos que se produce un ciclo de resultados (es decir, un bucle infinito).

$$0,125 \times 2 = 0,250$$

$$0,250 \times 2 = 0,5$$

$$0,5 \times 2 = 1$$

El número convertido será los dígitos de las partes enteras del resultado de las multiplicaciones de arriba abajo.

**0 0 1**

Por tanto, el número  $17,125_{10}$  en decimal equivale al  $10001'001_2$  en binario.

### 2.3. Conversor de Hexadecimal/Octal

Cuando tenemos un dígito a octal y lo queremos pasar a hexadecimal o viceversa, primero lo convertiremos a binario y después a hexadecimal u octal. Por tanto, el convertir a binario nuestro número será un paso intermedio.

Tabla de equivalencia de los números de octal a binario:

Dígito octal	Dígito binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Tabla de equivalencia de los números de hexadecimal a binario:

Dígito hexadecimal	Dígito binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111



$0110'010101000 \rightarrow 6'54_{16}$

## 2.4. Operaciones aritméticas en binario: sumas y restas.

1. *Suma:*

Al igual que en la decimal, hay que tener en cuenta que influye la posición del número respecto a la coma decimal. De tal manera que el primer dígito a la izquierda de la coma representa las unidades, el segundo las decenas, el tercero las centenas, etc. Cuando sumas un número y no cabe en esa unidad, se pasa el sobrante a la siguiente unidad superior (es decir, de las unidades a las decenas, de las decenas a las centenas y así sucesivamente).

Hay que tener en cuenta las tablas de la suma:

<u>Tabla del 0</u>	<u>Tabla del 1</u>
$0 + 0 = 0$	$1 + 0 = 1$
$0 + 1 = 1$	$1 + 1 = 0$ y me llevo 1 a la siguiente unidad

$$\begin{array}{r}
 111 \\
 11001 \\
 +10011 \\
 \hline
 101100
 \end{array}$$

2. *Resta:*

Como sucede en la suma, hay que tener en cuenta la posición de los dígitos respecto a la coma decimal. Cuando el dígito superior es menor que el dígito inferior, se produce un acarreo de tal manera que traes una unidad del dígito superior.

Hay que tener en cuenta las tablas de la resta:

<u>Tabla del 0</u>	<u>Tabla del 1</u>
$0 - 0 = 0$	$1 - 0 = 1$
$0 - 1 = \text{no cabe (me llevo 1 de la unidad superior)}$	$1 - 1 = 0$

$$\begin{array}{r} \phantom{0}1\phantom{0}1\phantom{0}1\phantom{0} \\ -1\phantom{0}0\phantom{0}1\phantom{0}1 \\ \hline 0\phantom{0}1\phantom{0}1\phantom{0}0 \end{array}$$

### 3. Unidades de medida de la información

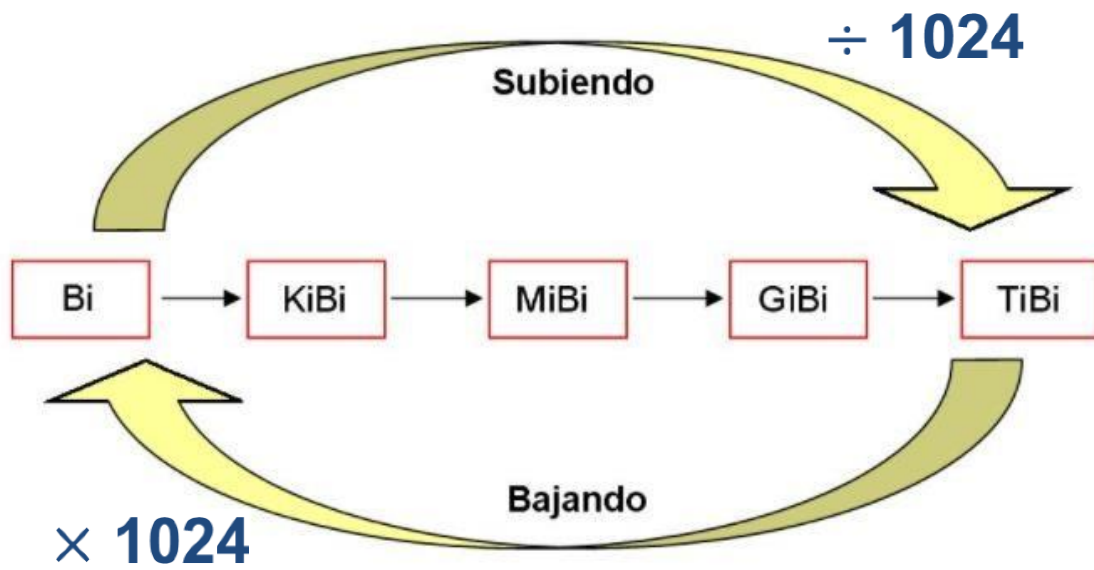
El **sistema binario**, se basa en la representación de cantidades utilizando los dígitos 1 y 0. Cada dígito de un número representado en este sistema se denomina bit.

- Se suelen utilizar con nombre propio determinados conjuntos de dígitos binarios:
- Cuatro bits: Se denominan cuarteto o nibble (1001).
- Ocho bits: Octeto o byte (10010110).
- Kibibyte y kilobyte no son sinónimos, pero suelen usarse incorrectamente como si lo fueran:
- 1 kibibyte =  $1024\text{ B} = 2^{10}\text{ bytes}$ .
- 1 kilobyte =  $1000\text{ B} = 10^3\text{ bytes}$ .

Nombre (símbolo)	Sistema Internacional de Unidades (SI) Estándar (uso decimal)	Prefijo binario (uso binario)	Nombre (símbolo)
Kilobyte (kB)	$1000^1 = 10^3\text{ bytes}$	$1024^1 = 2^{10}\text{ bytes}$	Kibibyte (kib)
Megabyte (MB)	$1000^2 = 10^6\text{ bytes}$	$1024^2 = 2^{20}\text{ bytes}$	Mebibyte (Mib)
Gigabyte (GB)	$1000^3 = 10^9\text{ bytes}$	$1024^3 = 2^{30}\text{ bytes}$	Gibibyte (Gib)
Terabyte (TB)	$1000^4 = 10^{12}\text{ bytes}$	$1024^4 = 2^{40}\text{ bytes}$	Tebibyte (Tib)
Petabyte (PB)	$1000^5 = 10^{15}\text{ bytes}$	$1024^5 = 2^{50}\text{ bytes}$	Pebibyte (Pib)
Exabyte (EB)	$1000^6 = 10^{18}\text{ bytes}$	$1024^6 = 2^{60}\text{ bytes}$	Exbibyte (Eib)
Zettabyte (ZB)	$1000^7 = 10^{21}\text{ bytes}$	$1024^7 = 2^{70}\text{ bytes}$	Zebibyte (Zib)
Yottabyte (YB)	$1000^8 = 10^{24}\text{ bytes}$	$1024^8 = 2^{80}\text{ bytes}$	Yobibyte (Yib)

Para almacenar los caracteres que forman el alfabeto se utilizan los códigos de E/S que traducen la información o los datos que nosotros podemos entender a una representación que la máquina puede interpretar y procesar. Los códigos estandarizados que se utilizan son el BCD, EBCDIC, ASCII y Unicode.

### 3.1. Conversión entre las diferentes unidades.



### 3.2. Representación interna de la información.

#### Código ASCII estándar de 7 bits y 128 caracteres (del 0 al 127)

	000 0	001 1	010 2	011 3	100 4	101 5	110 6	111 7
0000 0	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
0001 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0010 2	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
0011 3	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
0100 4	EOT 4	DC4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
0101 5	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
0110 6	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
0111 7	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1000 8	BS 8	CAN 24	( 40	8 56	H 72	X 88	h 104	x 120
1001 9	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1010 A	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1011 B	VT 11	ESC 27	+ 43	; 59	K 75	[ 91	k 107	{ 123
1100 C	FF 12	FS 28	' 44	< 60	L 76	\ 92	l 108	 124
1101 D	CR 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125
1110 E	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1111 F	SI 15	US 31	/ 47	? 63	O 79	- 95	o 111	DEL 127

HOLA

H = 1001000

O = 1001111

L = 1001100

A = 1000001

# HOLA

1001000100111110011001000001

Tabla ASCII estendida de 8 bits y 256 caracteres (0 al 255)

128	Ç	144	É	160	á	176	⋯	193	⌣	209	⌣	225	ß	241	±
129	ü	145	æ	161	í	177	⋯	194	⌣	210	⌣	226	Γ	242	≥
130	é	146	Æ	162	ó	178	■	195	⌣	211	⌣	227	π	243	≤
131	â	147	ô	163	ú	179		196	—	212	⌣	228	Σ	244	∫
132	ä	148	ö	164	ñ	180	†	197	†	213	⌣	229	σ	245	∫
133	à	149	ò	165	Ñ	181	‡	198	‡	214	⌣	230	μ	246	÷
134	â	150	û	166	ª	182	‡	199	‡	215	‡	231	τ	247	≈
135	ç	151	ù	167	º	183	⌣	200	⌣	216	‡	232	Φ	248	°
136	ê	152	—	168	¿	184	‡	201	⌣	217	‡	233	⊗	249	·
137	ë	153	Ö	169	—	185	‡	202	⌣	218	⌣	234	Ω	250	·
138	è	154	Û	170	¬	186	‡	203	⌣	219	■	235	δ	251	√
139	ï	156	£	171	½	187	‡	204	‡	220	■	236	∞	252	—
140	î	157	¥	172	¾	188	‡	205	=	221	■	237	φ	253	²
141	ì	158	—	173	¡	189	‡	206	‡	222	■	238	ε	254	■
142	Ä	159	ƒ	174	«	190	‡	207	⌣	223	■	239	∩	255	
143	Å	192	Ł	175	»	191	‡	208	⌣	224	α	240	≡		

Source: [www.asciitable.com](http://www.asciitable.com)

## Código UNICODE

Fueron creados en 1991 para tener códigos alfanuméricos estándar, comunes en todas las regiones. Se utiliza el mismo código Unicode para idiomas chino, árabe, etc. Como máximo necesitan 32 bits. Los primeros 7 bits permiten la compatibilidad con ASCII.

con 1 byte: se puede representar el código US-ASCII.

Con 2 bytes: caracteres latinos y alfabetos árabes, griego, cirílico, armenio, hebreo, sirio y thaana.

Con 3 bytes: resto de caracteres utilizados en todos los lenguajes.

Con 4 bytes: caracteres gráficos y poco comunes.

#### 4. Velocidades de transferencia

Se suelen medir en bits por segundo.

### Bit vs Byte



Cuota Mensual:	Tráfico incluido:	Velocidad Bajada / subida:
<b>15 €</b>	<b>Ilimitado</b>	<b>3 Mbps / 1,4 Mbps</b>

GRUPO  
ADSL  
ZONE



$$\frac{51347 \text{ kbps}}{8 \text{ bits/byte}} = 6.418,375 \text{ kbps} = \mathbf{6,2679 \text{ MBps}}$$

## 5. Conceptos básicos

### Hardware

Es la parte física del equipo.

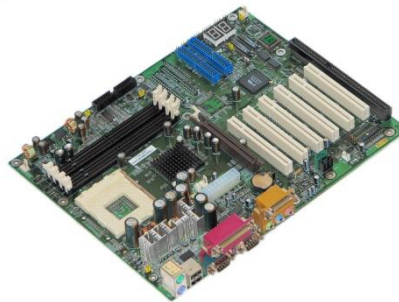
### Software

Conjunto de programas y aplicaciones formado por instrucciones y rutinas utilizados para el control del sistema

### Firmware

De más reciente aparición (en términos relativos), es una mezcla de los dos anteriores. Es un conjunto de rutinas básicas de configuración (elementos software) guardadas en un elemento hardware (un ejemplo podría ser la **BIOS** de nuestro equipo)

La **BIOS** acrónimo de (Binary Input Output System) se encuentra en todos los PCs. contiene las instrucciones más elementales para que puedan funcionar y desempeñarse adecuadamente, pueden incluir rutinas básicas de control de los dispositivos. Si tu equipo no la tuviera ni siquiera podrías empezar a usarlo.



## 6. Arquitectura de Von Neumann

Un ordenador se compone básicamente de un procesador, una memoria principal y periféricos.

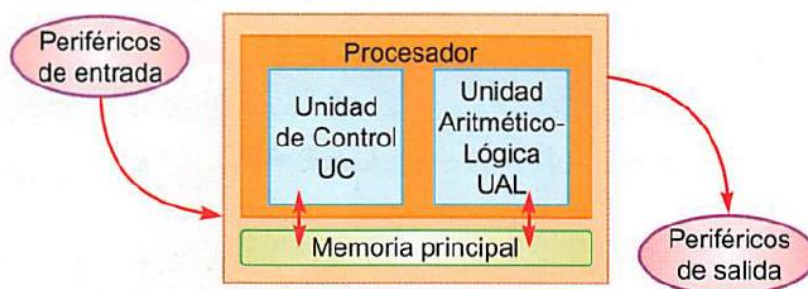


Figura 1.1. Unidad Central de Proceso (UCP).

El procesador se encarga de ejecutar los programas y realizar los cálculos matemáticos necesarios para que un programa funcione.

La memoria almacena el programa que se ejecuta y los datos que este necesita.



## 7. El procesador

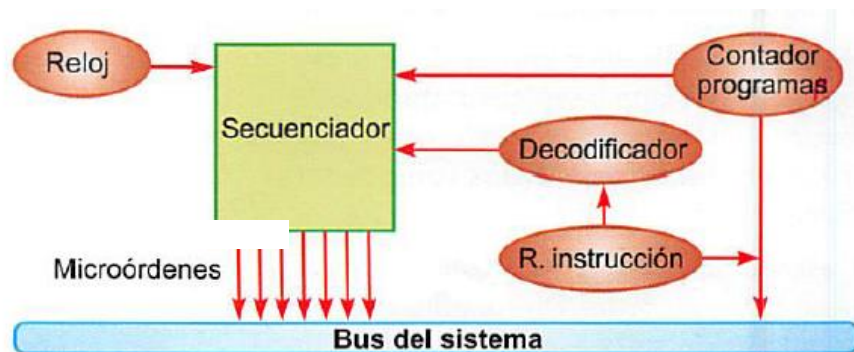
El procesador es la parte principal del ordenador. En él se ejecutan los programas y se realizan los cálculos necesarios para su funcionamiento.

### 7.1. Unidad de Control

Es el “cerebro” del ordenador gobernando el funcionamiento global del sistema. Recibe las órdenes del programa que se está ejecutando y las encadena, las interpreta y las transforma enviando las órdenes precisas al resto de los elementos del ordenador con el fin de que cada uno actúe en el momento y en la forma precisa.

Las funciones de la Unidad de Control son:

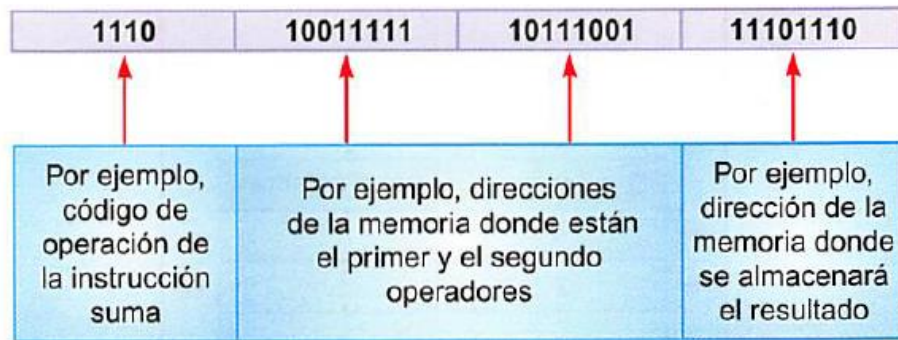
- Analiza e interpreta instrucciones del programa que se está ejecutando.
- Controla los demás componentes físicos del ordenador mediante órdenes dirigidas a estos componentes.
- Atiende y decide sobre posibles interrupciones que se pueden producir en el proceso (por ejemplo, la impresora se queda sin papel y envía una señal).



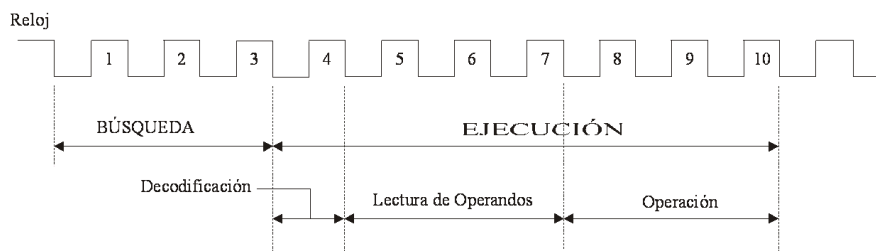
**Figura 1.2.** Esquema básico de la Unidad de Control.

Partes de la unidad de Control.

- **Registro Contador de Programa.** Contiene la dirección de memoria donde se encuentra la siguiente instrucción que se va a ejecutar.
- **Registro de instrucción.** Contiene la instrucción que se está ejecutando en ese momento.



- **Decodificador.** Se encarga de analizar y extraer el código de la instrucción.
- **Secuenciador.** Envía una serie de microórdenes al resto de elementos que, sincronizadas con los pulsos de reloj, hace que se ejecute paso a paso la instrucción que está en el registro de instrucción.
- **Reloj.** Indica en qué momento debe comenzar una determinada operación y en qué momento debe finalizar. Para esto emite impulsos (pulsos de reloj) a intervalos fijos de tiempo que sirven como referencia al resto de los componentes. (Sin entrar en detalles, estos elementos suelen estar formados por una circuitería cuyo elemento principal es un Cristal de Cuarzo).

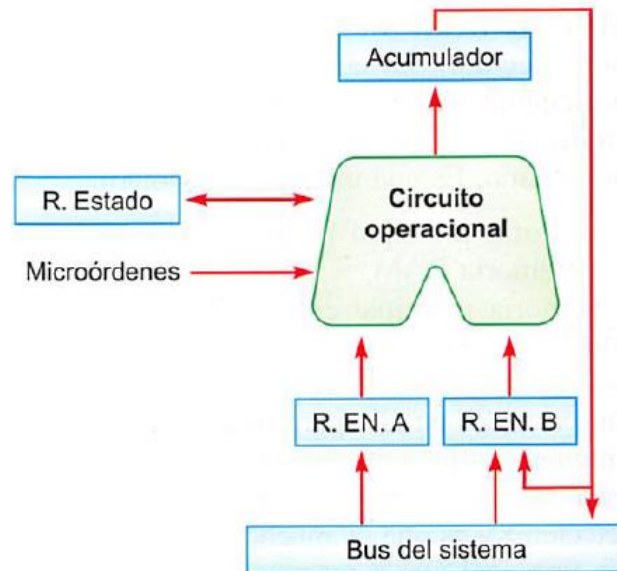


Representación de las fases de una instrucción en función del reloj del sistema.

## 7.2. Unidad Aritmético-Lógica

Lleva a cabo las operaciones que puede realizar el ordenador. Pueden ser aritméticas (sumas, productos, etc.) y lógicas (operaciones de comparación).

La mayoría de las operaciones aritméticas de la UAL son sumas, y el resto de operaciones las efectúa en base a la suma. La UAL trabaja a cientos de miles de operaciones por segundo (MIPS).



**Figura 1.4.** Esquema básico de la UAL.

- **Circuito operacional.** Se encarga de la realización de operaciones con los datos suministrados por registro de entrada. Dispone de unas entradas (microórdenes) para seleccionar el tipo de operación en cada momento.
- **Registros de entrada en A y en B.** Son utilizados para almacenar los datos que intervienen en la instrucción por parte del Circuito operacional.
- **Registro de Estado.** En él queda constancia de algunas condiciones que se dieron en la última operación realizada y que se tendrá que tener en cuenta en otras operaciones posteriores.

**Registro Acumulador.** En él se depositan los resultados de las operaciones llevadas a cabo por el circuito operacional.

## 8. Memoria principal

También conocida como memoria RAM o memoria interna.

La memoria RAM (Random Access Memory o memoria de Acceso Aleatorio) es una memoria volátil y de tiempo de acceso muy rápido. Al ser volátil la información sólo se almacena mientras esté conectada a la alimentación de energía. Una vez apagada la alimentación, la información desaparece.

Decir que también existe la memoria ROM (Read Only Memory o Memoria de Sólo Lectura). Es de acceso aleatorio y se utiliza principalmente para almacenar datos básicos y de configuración del ordenador, como es el arranque del mismo. Es una memoria no volátil.

La memoria principal es parte fundamental del ordenador, ya que tanto los programas como los datos que queremos que se ejecuten en el procesador deben estar cargados en ella.

Como ventaja respecto a la memoria secundaria decir que es mucho más rápida que ésta última, pero como contrapartida es volátil mientras que la secundaria es no volátil.

La memoria principal es como una matriz donde en cada celda se almacena un bit (dígito binario), es decir, un cero o un uno.

La capacidad mide la cantidad de información que podremos almacenar y/o que podremos procesar

La capacidad mínima de almacenamiento es el **byte** (8 bits).

Un byte es el número mínimo de bits necesarios para representar un carácter.

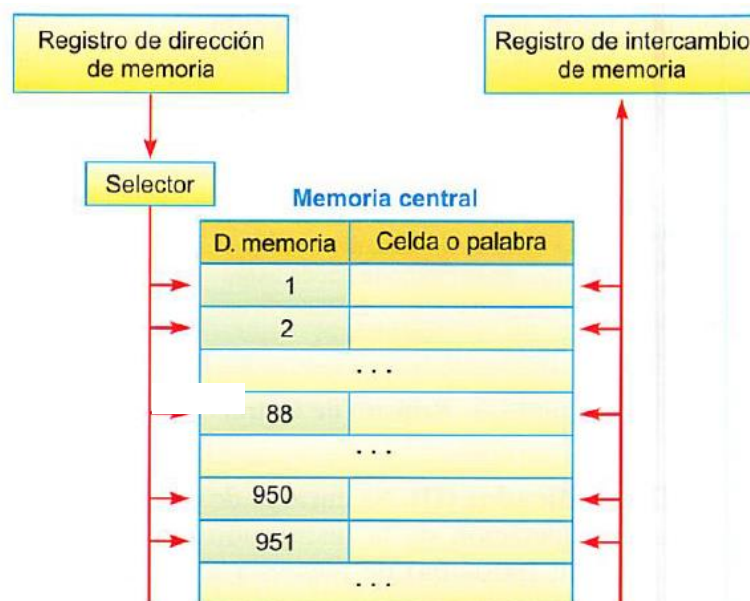


Figura 1.5. Esquema básico de la memoria principal.

- **Registro de Dirección de Memoria.** Contiene la dirección de la celda que se trata de seleccionar de la memoria bien para leer de ella o bien para escribir en ella.
- **Registro de Intercambio de Memoria.** En él se deposita el contenido de una celda de memoria que ha sido seleccionada en una operación de lectura, o bien la información que contiene se deposita en una celda de memoria en una operación de escritura. En definitiva, es por donde entran y salen los datos de la memoria.

El tamaño del RIM es un parámetro típico en las memorias, también es llamado ancho de palabra. Es un número de bits, normalmente múltiplo de 8. En cada lectura o escritura se leerá o se escribirá en la memoria ese conjunto de bits.

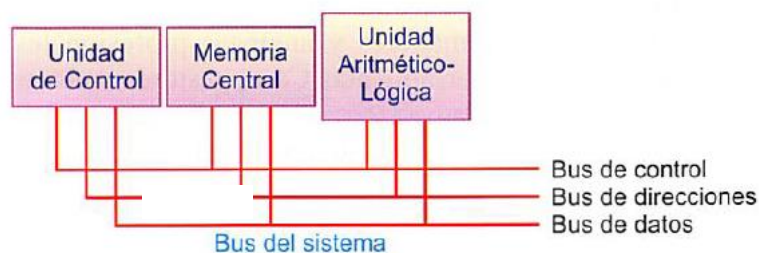
- **Selector de Memoria.** Es el encargado de conectar la celda de memoria, cuya dirección está en el RDM, con el RIM para la transferencia de los datos en un sentido o en otro, siempre que se produce una orden de lectura o escritura.

## 9. Bus del Sistema

Conjunto de circuitos que permite la comunicación entre UCP y el resto de unidades del ordenador. La transmisión la realiza en paralelo, es decir, enviando un número de bit de forma simultánea entre dos unidades.

En el bus del sistema se diferencian:

- **Bus de datos:** Se envían los datos a tratar.
- **Bus de control:** Se envían todas las instrucciones.
- **Bus de direcciones:** Se envían todas las direcciones de memoria.



**Figura 1.6.** Esquema básico del bus del sistema.