

NOTAS DEL CURSO SOBRE MICROPROCESADORES

Por J.Aramberri, Facultad de Ciencias, Universidad del País Vasco

1. INTRODUCCION

1.1. Introducción histórica

Qué es un microprocesador?, es la parte central, el corazón de un computador en miniatura. Está fabricado sobre una pequeña pastilla de Silicio de 5 a 50 mm.

Se encarga de efectuar todos los cálculos, todas las comparaciones y controla los periféricos y la memoria del sistema. En resumen, equivale a la unión de la Unidad Aritmético-Lógica (ALU) y unidad de control (CPU) de un computador tipo Von Neuman.

Detrás de la aparición del microprocesador no existe un descubrimiento científico. Únicamente es resultado del progreso de la tecnología y también en parte de la casualidad.

Desde que apareció el primer computador, que empleaba tubos de vacío, el progreso tecnológico comenzó a acelerarse casi a la par:

- 1948: se inventó el primer transistor, de Germanio.
- 1954: se produjeron los primeros transistores sobre silicio.
- 1958: la Texas Instruments pone a punto el primer circuito y sus conexiones metálicas se realizaban por grabación sobre una pastilla de SILICIO. Se llegaban a integrar de 10 a 15 componentes en una pastilla.
- 1971: mejoran las técnicas de integración aplicando una nueva tecnología: MOS (Metal Oxide Semiconductor). Se reducía el proceso de fabricación de 130 a 140 operaciones y se ampliaba el nivel de integración: 6000 componentes en la misma pastilla.

Hoy en día, desarrollando nuevos métodos de trabajo sobre los semiconductores, se llegan a colocar del orden de 30.000 componentes sobre una pastilla de Silicio. Mañana quizás se podrá multiplicar por 5 o por 10.

Hasta 1.972 se habían realizado sobre silicio muchos circuitos diferentes (amplificadores, filtros, controladores, memorias) pero no se había pensado en representar toda la lógica de un ordenador en una pastilla.

Este año, la sociedad de Informática Datapoint hace un concurso de ofertas entre los fabricantes de componentes electrónicos. Necesita un circuito integrado "a medida" que sea capaz de controlar una pantalla de visualización.



Se presentaron dos sociedades al concurso, Texas Instruments e Intel. Después de muchos meses de esfuerzos Texas Instruments abandonó el proyecto. Pero Intel terminó el desarrollo, consiguiendo un dispositivo que casi respondía a las especificaciones solicitadas. Era un poco lento, y no podía controlar correctamente las pantallas de visualización.

Datapoint anuló el contrato e Intel se encuentra con el desarrollo hecho de un circuito integrado un tanto particular. Un circuito que podía controlar otros circuitos, que había desarrollado a petición de otra casa, y con el que no sabía que hacer. Por si acaso lo puso en el mercado, pensando que quizás ayudaría a vender el principal producto de Intel, las memorias, ya que era capaz de controlar esas memorias y otros componentes.

Pero inesperadamente este nuevo circuito, con la denominación INTEL 8008 se vendió en gran cantidad. La casa Intel se dió cuenta que entre su circuito 8008 y la lógica interna de un computador existía muy poca diferencia.

Se volvió a reunir rápidamente el equipo de investigación que había puesto a punto el 8008 con el encargo de diseñar un circuito que pudiera llevar a cabo las funciones principales de un computador (Unidad Aritmética Lógica y Unidad de Control). Al mismo tiempo, las otras casas de componentes, asombrados por el éxito del 8008, enfocaron su investigación en el mismo campo. Como resultado, al cabo de un año los primeros microprocesadores invaden el mercado:

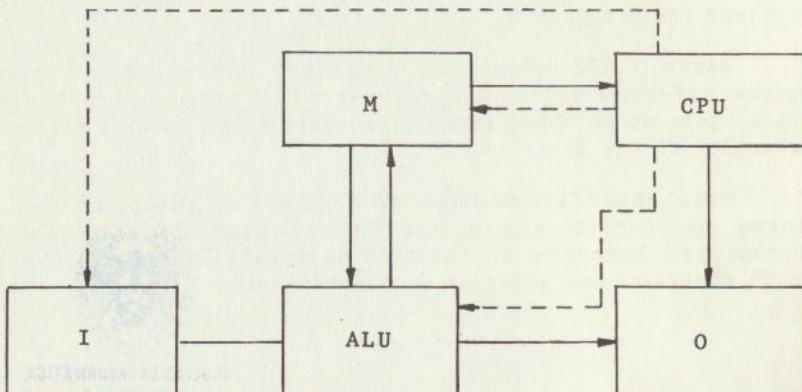
- INTEL 8080, MOTOROLA 6800, 2650 DE SIGNETIOS,
BPP58 DE ROCKARELL

La influencia del desarrollo de la tecnología, y de la casualidad, fueron en cierto modo los impulsores del fenómeno del microprocesador, que ha seguido creciendo sin cesar.

1.2. Filosofía de funcionamiento de un sistema basado en microprocesador

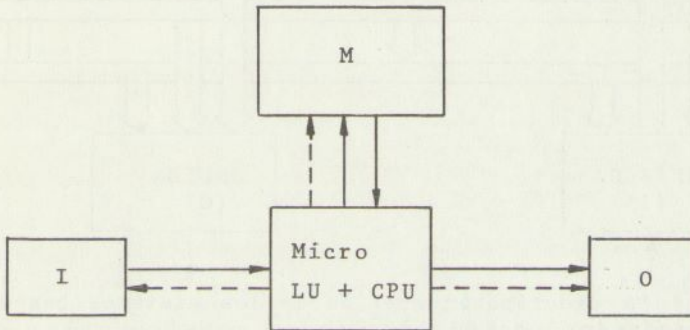
1.2.1. Arquitectura

La máquina de Von Neuman (computador convencional) tiene una serie de unidades o bloques lógicos de funcionamiento que están relacionados entre sí según el dibujo:



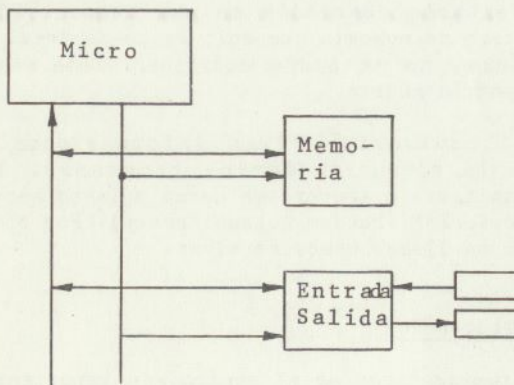
I = Input: entrada de datos
 O = Output: salida de datos
 M = Memoria de almacenamiento de programas y datos
 ALU = Arithmetic and Logic Unit: Unidad Aritmética-Lógica
 CPU = Control Process Unit: Unidad de control de procesos

Basado en el mismo principio de funcionamiento, un sistema con microprocesador engloba en dicho microprocesador los bloques unidad Aritmética Lógica (ALU) y unidad de control (CPU).



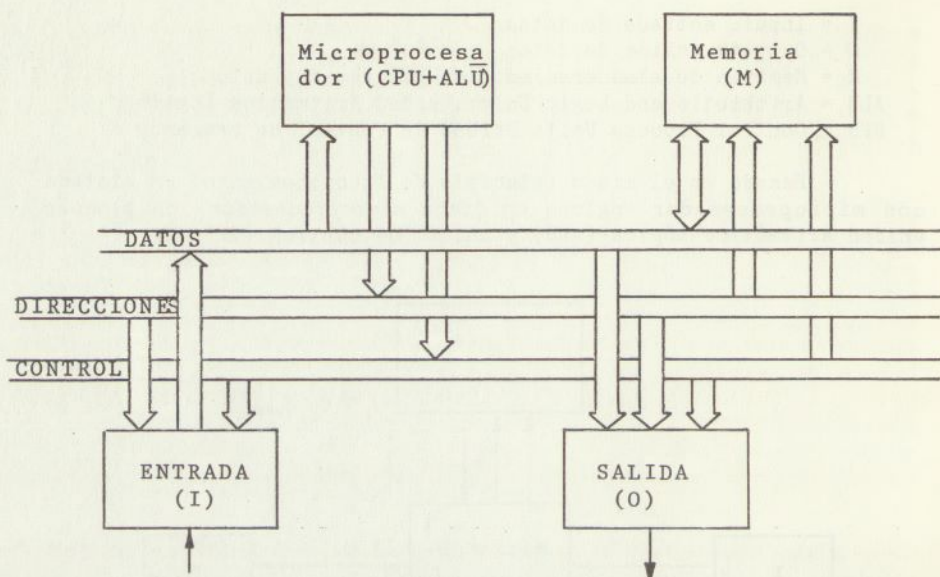
Las líneas punteadas reflejan toda una serie de señales de control, para establecer las transferencias de información en el sistema. Ya en su construcción física, se reúnen las líneas de transmisión de datos y lo mismo ocurre con las señales de control.

Así llegamos a un esquema gráfico más sencillo:



datos control
(direcciones y control)

Detallando más en profundidad las relaciones entre estas unidades llegamos a la figura:



Esta descripción gráfica de los sistemas basados en microprocesador nos dá idea de su construcción, de su "Arquitectura". Es lo que se llama el "HARDWARE".

Los programas, la parte no material, que residen en la memoria del sistema constituyen el "SOFTWARE".

1.2.2. La memoria

La memoria es el centro que contiene almacenada esta información. La capacidad total de la memoria es variable, pero no infinita. Existen informaciones que deben permanecer almacenadas invariablemente, por ejemplo, ciertos programas. Estas informaciones estarán guardadas en una memoria ROM (Read Only Memory). Es un tipo de memoria que solo se puede leer. Su contenido es indestructible, no se puede modificar. Esta memoria fija se llama también memoria muerta.

Por el contrario, otras informaciones deben poder modificarse en la ejecución de los programas. Todas estas informaciones fugitivas o temporales serán primero escritas y luego borradas en memoria RAM (Random Access Memory). Por oposición a las memorias muertas se llaman memorias vivas.

1.2.3. El microprocesador

El microprocesador es el centro más importante de todo el sistema.

- En el interior del microprocesador es donde se ejecutan realmente las operaciones aritméticas y lógicas (ALU). También extrae las instrucciones del programa de la memoria donde están almacenadas. Para esto es necesario que existan hilos conductores entre el microprocesador y

las memorias donde estan almacenados los datos e instrucciones.

- Estas líneas reciben el nombre de BUS.

Existen tres buses fundamentales: - el Bus de datos (data bus)
 - el Bus de direcciones (address bus)
 - el Bus de control (control bus)

El bus de direcciones es normalmente unidireccional, es decir, las senales circulan en el sentido microprocesador --- Memoria.

El bus de datos es bidireccional, es decir, que las informaciones pueden circular en ambos sentidos:

Memoria --- microprocesador cuando se extrae una instrucción o un operando de memoria, y microprocesador --- memoria cuando se guarda en memoria, y microprocesador --- memoria cuando se guarda en memoria RAM el resultado de una operación efectuada en la unidad central.

El bus de control transmite las senales de activación y desactivación de los órganos del sistema: Lectura de memoria, escritura en memoria, orden de leer datos por un periférico, o de escribir ... También lleva unas senales de reloj (clock) que se obtienen de un cristal de cuarzo y se utilizan para sincronizar las distintas fases de las instrucciones.

Todas las informaciones del microprocesador estan representadas en sistema binario, que se adapta bien a las senales electricas que tienen dos estados, todo o nada. La información elemental del sistema binario se llama bit (contracción de Binary Digit).

Una palabra o unidad de información está compuesta de un cierto número de bits, número fijo para cada sistema.

Los microprocesadores utilizan por lo general palabras de 4, 8 y 16 bits. La mayor parte actualmente estan estructurados con palabras de 8 bits. Una palabra de 8 bits se denomina Byte y también octeto

1.2.4. Los perifericos de entrada salida (I/O)

Los periféricos de Entrada Salida (I/O) son los órganos que permiten la comunicación con el exterior. Como el microprocesador trabaja con información binario y nosotros empleamos otra representación de la información, necesitamos adaptar ambas representaciones por medio de una Interface. También es necesaria la interface para sincronizar el microprocesador y los perifericos. El microprocesador es mucho más rápido que los dispositivos de entrada salida, y es preciso sincronizar las velocidades de ambos.

Por ejemplo, escribir una letra por un teletipo puede tardar 100 milisegundos (10 caracteres por segundo). En ese tiempo un microprocesador puede ejecutar 50.000 instrucciones.

2. TECNOLOGIA DE LOS MICROPROCESADORES

2.1. Introducción

El microprocesador ha nacido gracias al desarrollo de las tecnologías de difusión aplicadas a los circuitos integrados LSI.

Las tecnologías más utilizadas son sin duda alguna las de tipo MUS (N-MOS o P-MOS) o C-MOS. Permiten realizar sumas en tiempos del orden de 1 a 2 microsegundos. La tecnología N-MOS es un poco más rápida que la P-MOS. Los portadores móviles N son menos pesados que los portadores P.

Los consumos de potencia durante el funcionamiento son muy débiles, y la superficie ocupada por un transistor es mínima. Así se pueden integrar, sin peligro de un calentamiento excesivo, todas las funciones de un computador en una pastilla de Silicio de algunos mm^2 de superficie.

Algunos intentos de emplear tecnología TTL/S (schottky) permiten diseñar microprocesadores con mayor velocidad de ejecución de las operaciones. Una suma en 1/10 de microsegundo. Pero esta tecnología consume bastante potencia en funcionamiento, y es preciso "repartir" el microprocesador en varios circuitos integrados (por lo menos 2). Por estas razones esta tecnología está prácticamente abandonada.

Para obtener velocidades de ejecución más elevadas, se comienza a utilizar la tecnología ECL (Emitter Coupled Logic). La representación de las informaciones lógicas no corresponden a estados de saturación o corte de un transistor, sino que se produce antes. De este modo la velocidad de conmutación (cambio de estado) es bastante mayor.

Algunos microprocesadores emplean tecnología I^2L (Integrated Injection Logic). Tiene una velocidad media de ejecución, pero la potencia consumida es muy pequeña y el proceso de fabricación es sencillo.

Otra tecnología empleada también en circuitos integrados es la SOS (Silicon on Sapphire), en la que el sustrato donde se construye el circuito es un material aislante. Alcanza velocidades similares a la tecnología TTL/S con un consumo de potencia parecido a la de los MOS.

2.2. Proceso de fabricación de un transistor MUS (P)

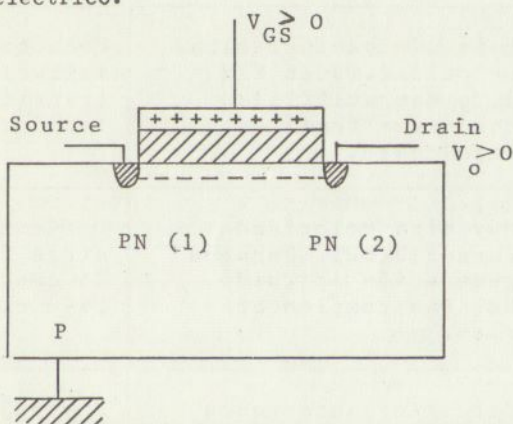
Se dispone en un sustrato de Silicio monocristalino dopado con impurezas P. Vamos a fabricar un transistor MOS canal

N. Se coloca sobre el sustrato un dieléctrico, constituido por óxido de Silicio (SiO_2). Sobre este óxido se deposita un contacto que formará la GATE del transistor y que será de aluminio o de silicio policristalino.

El conjunto del sustrato dieléctrico y puerta actúa como un condensador; si se aplica sobre la GATE una tensión V_{GS} 0 suficientemente grande se observa:

- Sobre el sustrato P. cara al dieléctrico, cargas eléctricas negativas (electrones N).
- Sobre la Gate, cara al dieléctrico, cargas positivas P.

Si además del condensador, se hace difusión de dos uniones PN junto al dieléctrico.



Llamamos a PN(1) Source y a PN(2) Drain (fuente y drenaje). Polarizando positivamente drain con respecto a Source y al sustrato, se creará una corriente de drenaje proporcional a la tensión V_{GS} , mayor es la carga del condensador MOS y la tensión positiva de drain puede captar más portadores N.

De una forma similar se construyen los transistores MOS de canal P, utilizando un sustrato dopado N.

2.3. Clasificación de microprocesadores según tecnologías

Sobre estas tecnologías se han fabricado una serie de microprocesadores, la mayor parte de ellos en MOS

N - MOS:	8080 Intel
	6800 Motorola
	F.8 Fairchild
	2650 Signetics
P - MOS:	8008 Intel
	PPS8 Rockwell
	IMP8 National s/c
	5065 Mostek
C - MOS:	COSMAC RCA
	IM 100 INTERSIL
ECL:	M 10800 MOTOROLA

VENTAJAS Y DESVENTAJAS DE LAS DISTINTAS FAMILIAS LOGICAS

FAMILIA	VENTAJAS	DESVENTAJAS
RTL	Baja disipación	Márgenes de ruido insuficientes. Baja velocidad
DTL	Baja disipación	Márgenes de ruido pequeños. Baja velocidad
TTL	Baja disipación. Alta velocidad. Buen FAN OUT. Compatibilidad con otras familias. Bajo coste.	Poca tolerancia en alimentación. Sensible a transitorios en alimentación.
ECL	Muy alta velocidad. Buen FAN OUT. Pequeña generación de ruido. Salidas complementarias.	Difícil interfase con otras familias. Elevado coste. Requiere montaje cuidadoso.
MOS	Circuitos integrados complejos. Bajo coste por función. Menor número de partes a montar, probar e inspeccionar. Bajo consumo por función. Diseño bajo pedido.	Muy baja velocidad. Requiere varias alimentaciones.
CMOS	Muy bajo consumo. Insensible a variaciones de alimentación. Alta inmunidad al ruido. Compatible con otras familias.	Baja velocidad. Sensible a danos por electricidad estática.
I ² L	Bajo producto velocidad x disipación. Alta densidad.	

FAMILIAS LOGICAS COMPLETAS

FAMILIA	TIPO DE PUERTA BASICA	TIEMPO DE PROPAGACION POR PUERTA (ns)	DISIPACION POR PUERTA (mW)	MARGEN DE RUIDO (V)	RUIDO GENERALIZADO	FAN OUT	FRECUENCIA DE RELOJ (MHz)	TENSION DE ALIMENTACION	COSTE POR FUNCION
RTL	NOR	50	10	0,2	MEDIO	4	8	3	MEDIO
DTL	NAND	25	15	0,7	MEDIO	8	12-30	5	MEDIO
TTL	NAND	10	15	0,4	ALTO	10	15-30	5	BAJO
ECL	OR-NOR	2	50	0,4	MEDIO	25	400	-5,2	ALTO
C-MOS	NOR y NAND	30	0,05	0,45V _{DD}	BAJO-MEDIO	100	5	4,6 a 16	MEDIO

TECNOLOGIAS UTILIZADAS EN LA FABRICACION DE MICROPROCESADORES

TECNOLOGIA	FAMILIAS	RETARDO PROPAGACION PUERTA	RETARDO DISIPACION (PJ)	TIEMPO DE CICLO INS-TRUCCION	4 PUERTAS			DIFUSIONES
					CAMPO	SUPERFICIE (minch) ² *	MASCARAS	
MOS	CANAL P 1972	1 μ s (tip 100ns)	50	1,462.5 μ s	2	10.0	4	1
	CANAL N PUERTA DE SILICIO 1973	10 100 ns	10	0.36 μ s	2	9.6	7	3
	CMOS 1974	100 ns (tip 50)	4	7.56 μ s	3	46.0	6	3
BIPOLAR	TTL SCHOTKY BAJA DISIPACION (LSTTL) 1974	10.25 ns	10.20	100.200 ns	3	19,8	7	4
	IMPLANTACION IONICA 1975	10.25 ns	0,11	100.830 ns	1	4.8	4	2
	ECL 1975	1.10 ns	50	55 ns	3	31	7	4

* 1 minch = 25.4 μ

3. SISTEMAS DE NUMERACION Y CODIGOS

3.1. Introducción

Es fundamental en el uso de los microcomputadores el conocimiento de sistemas de numeración. En un microprocesador clásico vamos a tener 8 líneas de datos y 10 líneas de direcciones, que nos presentan respectivamente números binarios de 8 y 16 dígitos binarios (bits). La memoria y los registros también van a contener una información binaria sobre la que hemos de representar números y caracteres.

Por otra parte mucha información sobre los microprocesadores emplea notación octal o hexadecimal, por razones de abreviar en la representación de datos, por lo que nos será imprescindible conocerlos.

3.2. Sistema decimal

Es el empleado normalmente en la actualidad en nuestros cálculos manuales. Tiene diez dígitos (0 - 9), que en principio se asociaron a los dedos de las manos. El sistema decimal es el típico sistema de notación posicional. La posición de cada dígito determina su valor o peso.

$$3 \ 4 \ 5 \ ' \ 2 \ 8$$

es la forma abreviada de

$$(3 \cdot 10^2) + (4 \cdot 10^1) + (5 \cdot 10^0) + (2 \cdot 10^{-1}) + (8 \cdot 10^{-2})$$

La posición de cualquier dígito determina la potencia de 10 por la que se multiplica, la raíz o base del sistema de numeración 10, y se cumple que:

- 1/ El número de dígitos es igual a la base.
- 2/ El dígito más grande es menor que la base.
- 3/ Cada dígito está implícitamente multiplicado por la potencia de la raíz que corresponde a su posición.

3.3. Sistema binario

En las calculadoras digitales, los registros y la memoria almacenan números, que están representados en base 2 (binario).

Hay dos dígitos binarios 0 y 1.

Para convertir números de base 10 a base 2 se emplea el método conocido como división repetida, es:

39	:	2	=	19	resto	1	LSB
19	:	2	=	9	resto	1	↑
9	:	2	=	4	resto	1	
4	:	2	=	2	resto	0	
2	:	2	=	1	resto	0	
1	:	2	=	0	resto	1	

LSB = Least Significant bit (bit menos significativo)
 MSB = Most Significant bit (bit más significativo)

$$100\ 111_2 = 39_{10}$$

En general, para convertir un número de base 10 a cualquier base se utiliza el método de las divisiones sucesivas.

Para recorrer el camino inverso, de cualquier base a decimal, nos basamos en el significado de la notación posicional.

$$A_n A_{n-1} \dots A_1 A_0(B) = \\ = (A_n \cdot B^n) + (A_{n-1} \cdot B^{n-1}) + \dots + A_1 B^1 + A_0 B^0$$

3.4. Sistema octal

Otro sistema de numeración muy empleado en las calculadoras digitales es el sistema octal (base 8). Tiene ocho dígitos, del 0 al 7.

Sirve para representar de forma abreviada números binarios que pueden tener una longitud muy grande en bits.

EJEMPLO:

Para el mismo ejemplo anterior 39_{10}

lo pasamos a octal

$$\begin{array}{r} 39 : 8 = 4 \quad \text{resto } 7 \uparrow \\ 4 : 8 = 0 \quad \text{resto } 4 \uparrow \end{array}$$

$$100111_2 = 39_{10} = 47_8 = \begin{array}{|c|} \hline 100 \\ \hline \end{array} 111$$

Pasando los dígitos octales 4 y 7 a binario tenemos

$$\begin{array}{r|l} 4 : 2 = 2 \text{ resto } 0 \uparrow & 7 : 2 = 3 \text{ resto } 1 \uparrow \\ 2 : 2 = 1 \text{ resto } 0 \uparrow & 3 : 2 = 1 \text{ resto } 1 \uparrow \\ 1 : 2 = 0 \text{ resto } 1 \uparrow & 1 : 1 = 0 \text{ resto } 1 \uparrow \end{array}$$

Vemos que es posible representar de forma abreviada números binarios en base octal y viceversa con un método de conversión directo.

Para la unidad de información empleada en microprocesadores, que es de 8 bits, necesitamos hasta tres dígitos octales. Esto se debe a que cada dígito octal representa tres dígitos binarios.

3.5. Sistema hexadecimal

Utilizando la base 16 como sistema de numeración, se necesitan 16 dígitos. Cogemos los dígitos decimales del 0 al 9 y las letras de A a F que correspondan a los dígitos 11_{10} a 15_{10} .

El sistema Hexadecimal se emplea por las mismas razones que el octal. Al ser la base una potencia de 2, la conversión Hexadecimal-binario y viceversa es inmediata. Cada dígito Hexadecimal corresponde a 4 dígitos binarios. Un registro de 8 bits o byte se puede representar como dos dígitos hexadecimales. Es un método muy utilizado.

$$\begin{array}{r}
 \text{Es:} \qquad 92 : 16 = 5 \text{ resto } 12 = C \uparrow \quad 92(10 = 50(16) \\
 \qquad \qquad 5 : 16 = 0 \text{ resto } 5 \quad \uparrow \\
 \\
 \qquad \qquad 92 : 2 = 46 \text{ resto } 0 \quad \uparrow \\
 \qquad \qquad 46 : 2 = 23 \text{ resto } 0 \\
 \qquad \qquad 23 : 2 = 11 \text{ resto } 1 \\
 \qquad \qquad 11 : 2 = 5 \text{ resto } 1 \\
 \qquad \qquad 5 : 2 = 2 \text{ resto } 1 \\
 \qquad \qquad 2 : 2 = 1 \text{ resto } 0 \\
 \qquad \qquad 1 : 2 = 0 \text{ resto } 1
 \end{array}$$

Resumiendo, la notación Hexadecimal se emplea porque cada dígito hexadecimal es una representación de 4 bits. Es muy sencillo hacer conversiones inmediatas de hexadecimal a binario y viceversa.

3.6. Complemento a 2

Hasta ahora hemos representado números naturales, es decir, enteros y positivos. En los registros de nuestro microprocesador solo vamos a tener en realidad números enteros y positivos.

Pero nos interesa representar también números enteros negativos, y para ello debemos adoptar una convención. La más utilizada es la de representar los números negativos en "complemento a 2".

La unidad de información (byte) tiene ocho bits. De los 256 estados que se pueden representar con 8 bits reservaremos la mitad para números positivos y la otra mitad para negativos.

Por convenio, el bit de la izquierda será el bit de signo; si es cero, el número es positivo. Pero si es uno, el número es negativo y su valor se considera que es -2^{n-1} (n un byte : n=8). Los siete bits restantes siempre representan una cantidad positiva.

EJEMPLO:

$$\begin{array}{r}
 67(10 = 0 \mid 1000011 \\
 -1(10 = 1 \mid 1111111 \quad \swarrow +127=-1 \\
 \quad \quad \swarrow -128 \\
 -25(10 = 1 \mid 1100111 \quad \swarrow +103=-25 \\
 \quad \quad \swarrow -128
 \end{array}$$

CONVERSION CHART

<u>Decimal</u>	<u>Octal</u>	<u>Hexadecimal</u>	<u>Binary</u>
0	0	0	0000 0000
1	1	1	0000 0001
2	2	2	0000 0010
3	3	3	0000 0011
4	4	4	0000 0100
5	5	5	0000 0101
6	6	6	0000 0110
7	7	7	0000 0111
8	10	8	0000 1000
9	11	9	0000 1001
10	12	A	0000 1010
11	13	B	0000 1011
12	14	C	0000 1100
13	15	D	0000 1101
14	16	E	0000 1110
15	17	F	0000 1111
16	20	10	0001 0000
17	21	11	0001 0001
18	22	12	0001 0010
19	23	13	0001 0011
20	24	14	0001 0100
21	25	15	0001 0101
22	26	16	0001 0110
23	27	17	0001 0111
24	30	18	0001 1000
25	31	19	0001 1001
26	32	1A	0001 1010
27	33	1B	0001 1011
28	34	1C	0001 1100
29	35	1D	0001 1101
30	36	1E	0001 1110
31	37	1F	0001 1111
32	40	20	0010 0000
33	41	21	0010 0001
34	42	22	0010 0010
35	43	23	0010 0011
36	44	24	0010 0100
37	45	25	0010 0101
38	46	26	0010 0110
39	47	27	0010 0111
40	50	28	0010 1000

Hay una regla muy sencilla para obtener el complemento a dos de un número, y sirve para números positivos y para números negativos representados en complemento a 2.

- 1° complementar bit a bit el número dado, incluido el bit de signo.
- 2° sumar una unidad de orden inferior.

ES:	25 ₍₁₀₎ = 0	0011001 ₍₂₎	
complemento bit a bit	1	1100110	←--Equivale a: restar 128 (bit signo) calcular 127-n (bit7-0)
Suma 1		+ 1	: sumar 1 (bit 7-0)
	1	1100111	-128 + (127-n) + 1 = -n

Este método es usado generalmente en los microprocesadores, ya que es fácil de implementar por Hardware.

Es de destacar que el cero se considera positivo, y que el rango de número enteros es de +127 a 0 y de -1 a -128.

3.7. Aritmetica en complemento a dos

En todas las operaciones aritméticas sobre el microprocesador es preciso controlar por software que el resultado de la operación no salga del rango permitido (+127, -128).

Vamos a hacer unos ejemplos con 5 bits (rango 15 a -16). Existen unos bits indicando las posibles llevadas (carry) en el bit de signo (C_s) y sucesivo (C_{s+1})

<u>SUMA</u>	C_{s+1}	C_s			
	0	0			
			0, 0111	+ 7	
			1, 1000	- 8	
			1, 1111	- 1	correcto
 <u>RESTA</u>	 C_{s+1}	 C_s			
	1	1			
			1, 1001	- 7	
			1, 1000	- 8	
			1, 0001	-15	correcto
 <u>SUMA</u>	 C_{s+1}	 C_s			
	0	1			
			0, 1100	12	
			0, 1110	14	
			1, 1010	+26	incorrecto sale del rango
 <u>RESTA</u>	 C_{s+2}	 C_s			
	1	0			
			1, 0100	-12	
			1, 0011	-14	
			0, 0110	-26	incorrecto sale del rango

Se puede comprobar que el resultado es incorrecto si:

$$C_{S+1} \text{ es distinto que } C_S$$

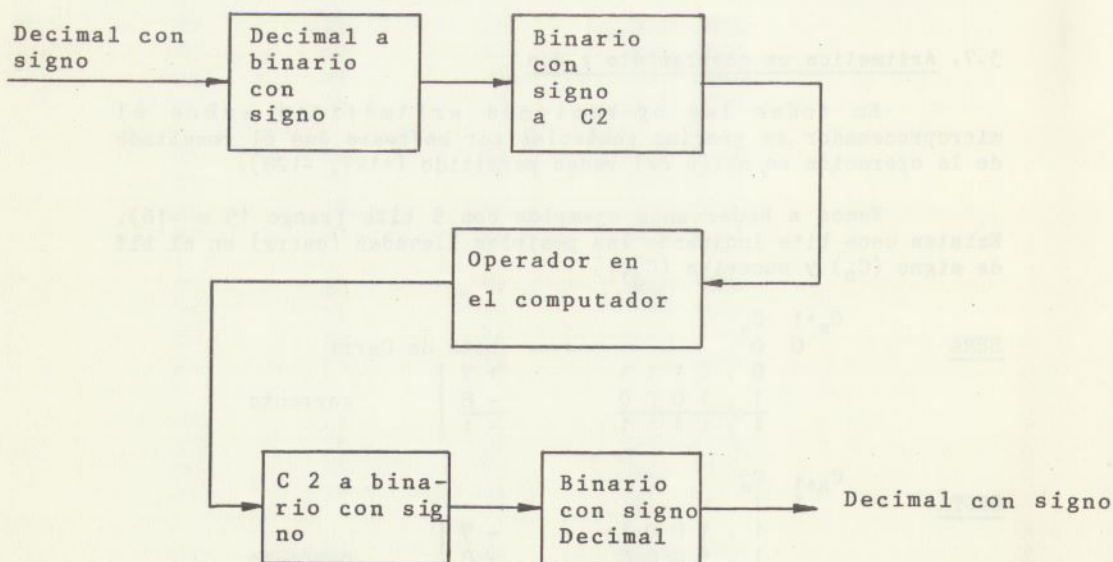
En algunos microprocesadores será preciso comprobar directamente sobre estos C_S y C_{S+1} , y en otros se genera automáticamente una señal de error (OVERFLOW).

$$\text{OVERFLOW} = C_S (+) C_{S+1}$$

Con la utilización del complemento a 2, se puede emplear el mismo circuito para adiciones y subtracciones.

3.8. Codificación B C D

El sistema binario es el más adecuado para los microprocesadores. Sin embargo, la gente está más acostumbrada a emplear números decimales. El método ideal es realizar las operaciones en el computador en binario, y convertir el resultado a decimal para enseñárselo al operador. La conversión del binario a decimal y viceversa, aunque sencilla, gasta tiempo de cálculo. El proceso es el siguiente:



Este proceso de conversión a veces no es rentable y los números se almacenan y se tratan guardando codificado cada dígito decimal según un cierto código. Este sistema es llamado BCD (binary coded decimal). El peso-posición se mantiene, pero el dígito se representa por una combinación de ceros y unos.

Para cada dígito se emplean 4 bits (16 estados), por lo que existen un gran número de métodos para codificarlos (solo se necesitan 10 estados).

Table 3.9 BCD Codes

Decimal	8421 BCD	2'421 BCD	Excess-3	Gray Code
0	0000	0000	0011	0000
1	0001	0001	0100	0001
2	0010	0010	0101	0011
3	0011	0011	0110	0010
4	0100	0100	0111	0110
5	0101	1011	1000	0111
6	0110	1100	1001	0101
7	0111	1101	1010	0100
8	1000	1110	1011	1100
9	1001	1111	1100	1000

El más común es el de representar cada dígito por su binario equivalente de 4 bits.

POR EJEMPLO: 5 9 decimal
 0101 1001 B C D

Existen en algunos microprocesadores instrucciones que facilitan la operación con números codificados en BCD.

Los otros códigos BCD se emplean por razones de mejorar la detección de errores y de operaciones. En el 2421BCD al dar peso 2 al dígito binario de peso normal 8 obtenemos unas configuraciones simétricas. Lo mismo ocurre en el código de Exceso de tres. Al complementar cada bit de código BCD obtenemos el denominado complemento a 9 de un dígito, que facilita la substracción. El código Gray solo modifica un bit para pasar de un dígito al siguiente. Este código es muy útil para para ciertos conversores Analógicos digitales, para detectar errores.

En cada byte (8 bits) se pueden almacenar dos dígitos de un número en BCD. De las 256 posibles configuraciones solo se emplean 100, pero se eliminan las conversiones a binario y viceversa.

3.9. Código ASC II

Para representar caracteres alfanuméricos se les asigna a cada uno de ellos una configuración binaria. El código alfanumérico más empleado es el código ASCII (American Standar Code for Information Interchange), y se usa en casi todos los microprocesadores, en la versión de 7 bits de código más uno de paridad para cada carácter; en total un byte.

Table 3.10 OCTAL VALUES FOR 7-BIT AND 6-BIT ASCII CODES

Printing Character	7-Bit ASCII	6-Bit ASCII	Printing Character	7-Bit ASCII	6-Bit ASCII
()	100	00	(Space)	040	40
A	101	01	!	041	41
B	102	02	"	042	42
C	103	03	#	043	43
D	104	04	\$	044	44
E	105	05	%	045	45
F	106	06	&	046	46
G	107	07	'	047	47
H	110	10	(050	50
I	111	11)	051	51
J	112	12	.	052	52
K	113	13	+	053	53
L	114	14	,	054	54
M	115	15	-	055	55
N	116	16	.	056	56
O	117	17	/	057	57
P	120	20	0	060	60
Q	121	21	1	061	61
R	122	22	2	062	62
S	123	23	3	063	63
T	124	24	4	064	64
U	125	25	5	065	65
V	126	26	6	066	66
W	127	27	7	067	67
X	130	30	8	070	70
Y	131	31	9	071	71
Z	132	32	:	072	72
[133	33	;	073	73
/	134	34	<	074	74
]	135	35	=	075	75
†	136	36	>	076	76
+	137	37	?	077	77
Null	000				
Hort. Tab	011				
Line Feed	012				
Vertical Tab	013				
Form Feed	014				
Carriage Return	015				
Tabout	177				

3.10. Ejercicios1) Convertir

- a) $10111010_{(2)}$ a decimal
- b) $47_{(10)}$ a binario
- c) $87_{(10)}$ a octal
- d) $DABA_{(16)}$ a decimal

2) Convertir el número binario 10100111 a:

- a) octal
- b) hexadecimal
- c) decimal

3) Convertir a binario:

- a) $275_{(8)}$
- b) $3F_{(16)}$
- c) $98_{(16)}$

4) Poner el complemento a 2 (en 8 bits)

- a) -24
- b) $-107_{(10)}$

5) Usando complemento a 2 (en 8 bits) calcular en binario

- a) $86_{(10)}$ -24
- b) $15_{(10)}$ $-107_{(10)}$

6) Dar la representación en 8421 BCD de

- a) 7290
- b) 5164
- c) 3594

EL MICROPROCESADOR

1. DESCRIPCION INTERNA DE UN MICROPROCESADOR

El microprocesador va a tratar las informaciones aritméticas o lógicas según las instrucciones de un programa almacenado en memoria. El juego de instrucciones que interpreta cada microprocesador personaliza cada microprocesador.

1.1. Características

Desde el punto de vista de funcionamiento vamos a describir un microprocesador modelo, que se ajusta a la mayor parte de los microprocesadores actuales:

- 1) LONGITUD DE PALABRA: 8 bits. Esta longitud es la misma para las instrucciones que para los operandos.
- 2) RANGO DE DIRECCIONAMIENTO: 16 bits. Puede direccionar hasta 2^{16} palabras = 64 k palabras, que también se llaman espacio de direcciones.
- 3) LONGITUD DE LA INSTRUCCION: 8, 16 ó 24 según el tipo de instrucción.
El código de operación ocupa 8 bits (un byte).
La dirección del operando o el mismo operando pueden ocupar a continuación 8 ó 16 bits (1 ó 2 bytes).

Examinaremos los tipos de direccionamiento más adelante.

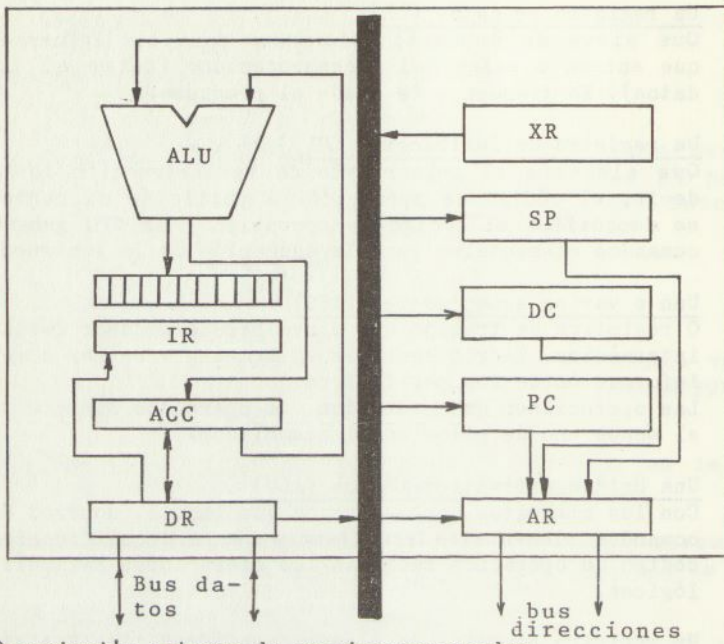
1.2. Registros internos

Un microprocesador está compuesto por los siguientes circuitos:

- Un contador de programa PC
Almacena la dirección de memoria central correspondiente a la instrucción en ejecución. Si el rango de direcciones es de 16 bits, este contador será por hipótesis de 16 bits, normalmente las instrucciones están en secuencia natural en memoria, y al final de una instrucción el PC se incrementa en una unidad para apuntar a la instrucción siguiente.
- Un registro de dirección del operando DC (Data counter)
Donde se genera la dirección de memoria o la que hay que ir a tomar el operando.
Se llama DC por similitud con el PC.
- Un registro externo de direcciones (AR) (address Register)
A través del cual se emiten al bus de direcciones y en suma a la memoria central y a los periféricos las senales correspondientes para seleccionar un dispositivo.
También es transparente desde el programa.

- Un registro de datos (DR)
Que sirve de depósito intermedio para las informaciones que entran o salen del microprocesador (datos al bus de datos). Es transparente desde el programa.
- Un registro de Instrucción (RI)
Que almacena el primer byte de la instrucción leída, es decir, el código de operación. A partir de su contenido, se decodifica el código de operación y la CPU genera los comandos elementales para la ejecución de la instrucción.
- Uno o varios acumuladores (ACC)
0 registros de trabajo que sirve para almacenar resultados intermedios, leer o escribir en memoria y enviar o recibir información de los periféricos.
Las operaciones que necesitan dos operandos siempre tienen al menos uno de ellos en el acumulador.
- Una Unidad Aritmético-Lógica (ALU)
Con los circuitos combinatorios que bajo el control de los comandos elementales resultantes de la decodificación del código de operación realizan las operaciones aritméticas y lógicas.
- Un registro especial de condiciones (Condition Register)
Almacenan los marcadores que indican, después de la ejecución de cada instrucción, el estado interno del microprocesador. Estos registros de condiciones suelen ser testeables por programa, por instrucciones de ruptura de secuencia condicional.
Los más habituales son:
 - Resultado cero en el acumulador (Z)
 - Signo del resultado (S) (también MSB)
 - Paridad del resultado (E) (también LSB)
 - Llevada en operación aritmética (C) (carry)
 - Superación del rango permitido (O) (overflow)
 - Estado del sistema de Interrupciones (I)
- Un registro apuntador de Pila (PS. pointer Stack)
Para manejar zonas de memorias como estructuras LIFO (Stack o pila). La pila es una estructura de datos que presenta muchas ventajas a la hora de la programación.
- Un registro índice (XR) (index Register):
En cierto modo de direccionamiento (direccionamiento indexado) sirve para construir la dirección del operando, añadiendo un desplazamiento al valor almacenado en el registro índice. Tiene 16 bits. Es modificable por programa. (Ver cuadro n° 1)
Entre todos estos registros o circuitos se establece una comunicación, bien a través de una serie de líneas comunes o bus interno, o con caminos particulares que los enlazan.

CUADRO 1



1.3. Descripción externa de un microprocesador

Como para la descripción interna, también ahora elegimos un procesador modelo, al que se ajustan más o menos los microprocesadores más conocidos.

Desde el punto de vista de funcionamiento, existen una serie de conexiones externas que permiten al microprocesador interactuar con los demás dispositivos del sistema. Estas líneas se denominan bus del microprocesador.

Desde el punto de vista físico, el microprocesador será una "cucaracha" con un número de patas de 28 a 40.

- a) Las líneas de BUS sirven de medio de conexión de los elementos del sistema. En principio, se consideran como líneas dispuestas en paralelo a las que se pueden unir las distintas pantallas.

Siempre se clasifican en tres grupos

- Bus de direcciones: normalmente 16 ($A_{15}-A_0$). Permiten direccionar una posición de memoria o un periférico. Tienen tres posibles estados "0", "1" y alta impedancia.
- Bus de datos: bidireccionales, tres estados. Son ocho (D_7-D_0).
- Bus de control: Reune todas las señales de órdenes y de estados (de periféricos, etc.).

Las señales más corrientes en el bus de control son:

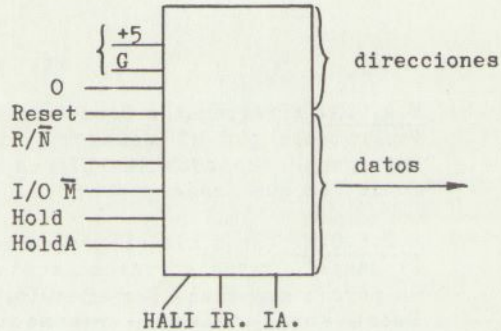
- 1º) Orden de lectura o escritura R/W.
- 2º) Operación con memoria o con entrada salida I/O M.
- 3º) Petición de interrupción (I.R.).
- 4º) Reconocimiento de interrupción (I.A.).

- 5). Petición de inhibición (HOLD) (I) para D.M.A.
- 6). Reconocimiento de inhibición (HOLDA)
- 7). Comandos manuales - Parada (Halt)
- Reinicio (Reset)
- 8). Reloj (clock)

b) Caja del microprocesador

40 patas Dual IN-LINE Package
(pins)

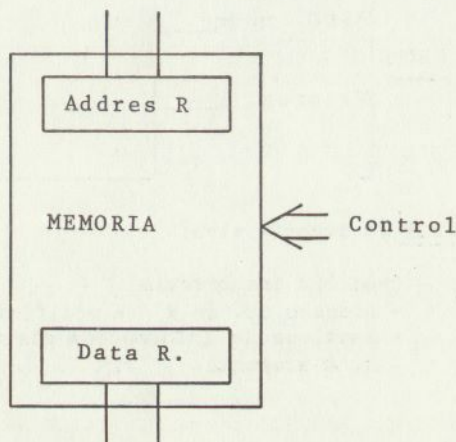
Alimentación: 5 voltios



2. MEMORIAS

Vienen caracterizadas por distintas dimensiones o características.

- Talla de almacenamiento: número de bits de direccionamiento.
- Organización: palabras de 1 bit, 4 bits, 8 bits, aunque siempre haremos módulos de 8 bits.
- Tiempo de Lectura / Escritura: 20 ns --- 700 ns.
- Construcción física: ROM, PROM, EPROM
RAM dinámicas, estáticas.



2.1. Memoria ROM

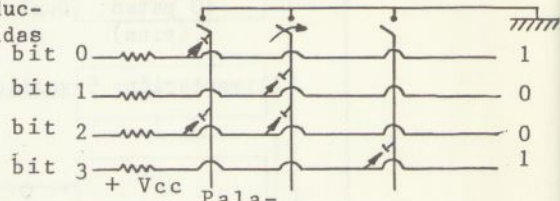
Exclusivamente destinada a contener las instrucciones del programa. Permite únicamente la lectura.

Puede ser de tres tipos:

- a) ROM (escrita por el fabricante)

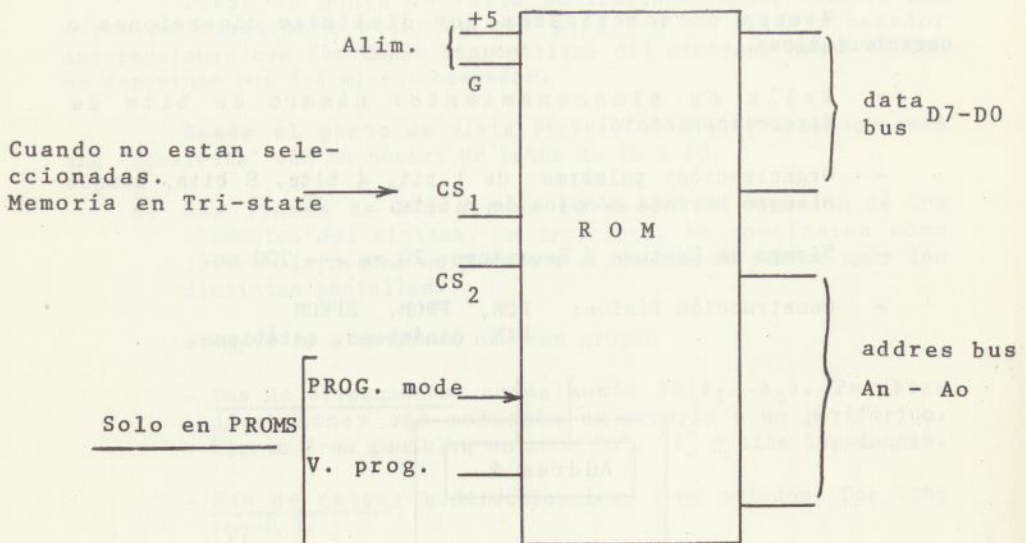
Un conjunto de líneas conductoras que pueden estar unidas en los cruces por diodos.

Principio de diodos



- b) P.R.O.M. (Programmable Read-Only memory) Programable por el usuario. Mismo principio anterior. Entregada con todos los diodos en cada cruce, el usuario funde los que desea.

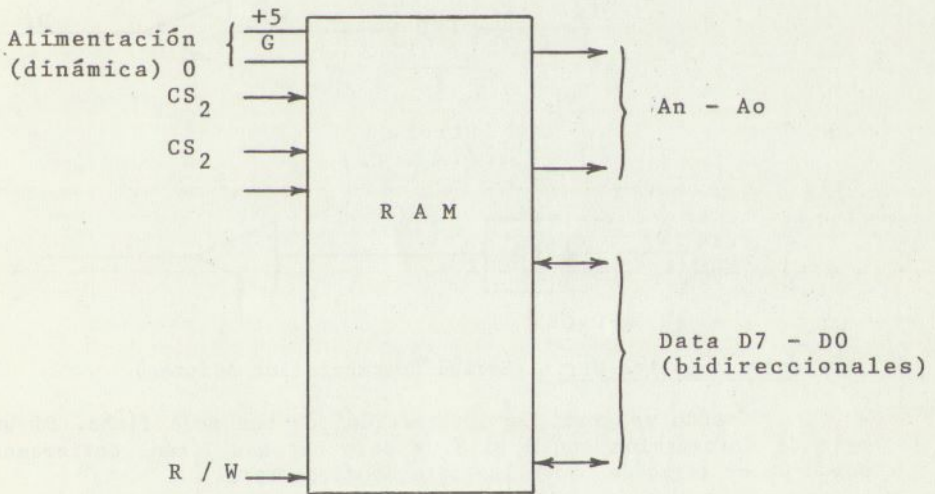
- c) E.P.R.O.M. (Erasable, Reprogrammable ROM) El usuario puede borrar el contenido de dicha memoria por un método especial. Por ejemplo, rayos ultravioletas (UV). Estas son fácilmente reconocibles por la ventana de cristal de cuarzo. Están fabricadas en tecnología MOS (transistores). Existen ya de talla de 16 k bytes y se esperan ya las de 32 k bytes. Todas estas memorias tienen una configuración similar.



2.2. Memoria RAM (o memoria viva)

- RAM estática:
- cada bit una báscula
 - ninguna op. de R / W modifica el contenido
 - mantiene la información mientras la alimentación esté presente.

- RAM dinámica: - Cada bit es un condensador elemental cuyo contenido dura unos milisegundos.
- Necesitan una operación suplementaria de "Refresco" destinada a recargar los condensadores de estado lógico 1.
 - Son más complicadas de manejar, aunque tienen mucha capacidad de integración.



3. DISPOSITIVOS DE ENTRADA/SALIDA

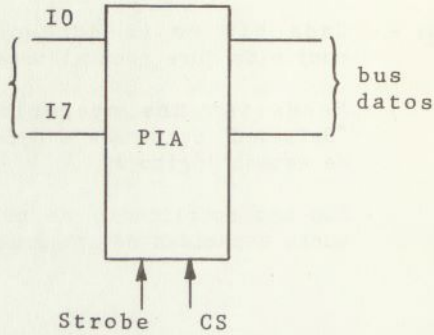
3.1. Entrada salida en paralelo

Consiste en una transferencia de un dato de 8 bits, a través del bus de datos, entre el microprocesador y un periférico.

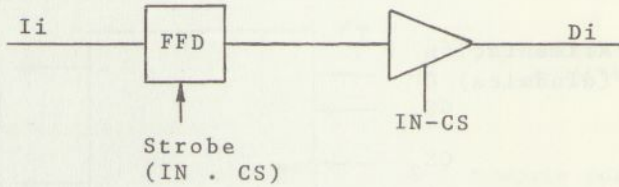
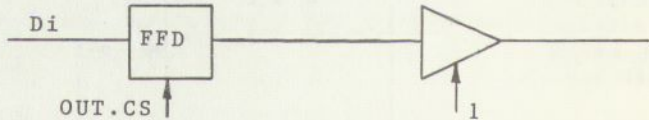
El sistema más elemental actúa como un conjunto de registro y puerta tri-state. (ver ejemplo pag. 26).

Existen circuitos más complejos, con hasta dos y tres puertas que se pueden programar como de entrada o de salida, y se encargan incluso de señalar por medio de un canal de interrupción cuando están de nuevo disponibles.

EJEMPLO:

Entrada

por cada línea

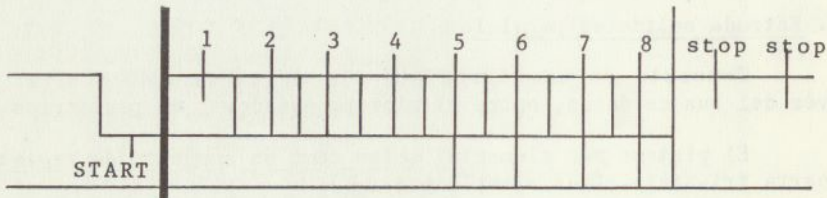
Salida

3.2. Entrada salida serie (Serial Communication Adapter)

Cuando se envía la información por una sola línea. Si un byte de información con 8 bits, y solo hay una línea, enviaremos según un criterio de orden los bits de dicho byte.

La transmisión puede ser:

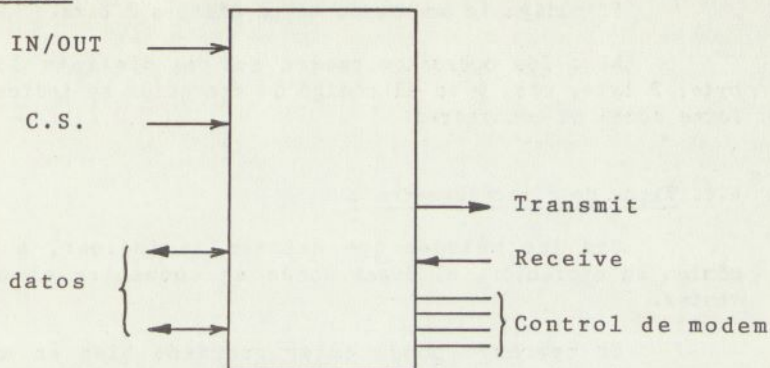
* ASINCRONA: Los bits se envían a una determinada frecuencia, precedidos de una señal de START y una o varias de STOP. Es carácter a carácter.



* SINCRONA: Transmisión por bloques de caracteres.

La transmisión serie emplea menos líneas que una transmisión en paralelo. Pero es preciso convertir el dato del bus de datos en una serie de pulsos.

EJEMPLO:

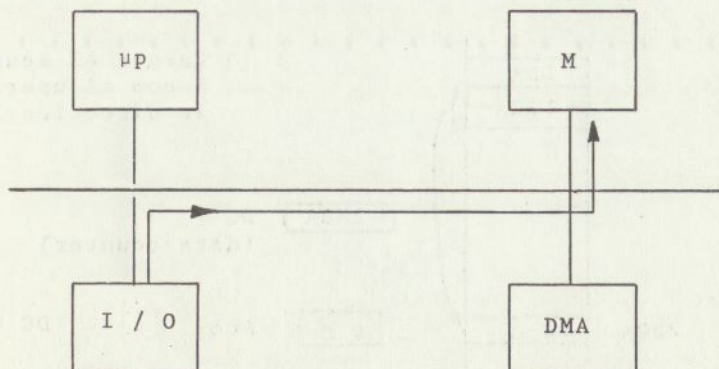
Entrada y salida

Se puede emplear ventajosamente para largas distancias (menos cableado e incluso a través de línea telefónica o canal de radio). Se emplean aparatos moduladores-demoduladores (MODEM) que se controlan con señales de esta misma pastilla. Son necesarios para enviar la información en forma diferente a pulsos eléctricos, que tienen poco alcance y pueden ser perturbados por ruido eléctrico.

CIRCUITO CONTROLADOR DMA

Es otra forma de hacer transferencias de información por bloques (de memoria a memoria o de tipo memoria-periférico).

Tiene la ventaja de evitar el paso por el μ -procesador, con lo que se acelera las transferencias de información.

4. FUNCIONAMIENTO DEL MICROPROCESADOR

Cada instrucción de un programa nos va a indicar dos cosas:

- en un código de operación, qué es lo que tratamos de hacer.

- en un campo de operandos, sobre que datos actua la operación.

El código de operación ocupa siempre 8 bits.

El o los operandos pueden ser de distinta longitud (1 byte, 2 byte, etc) y en el código de operación se indica en cierta forma donde se encuentran.

4.1. Tipos de direccionamiento

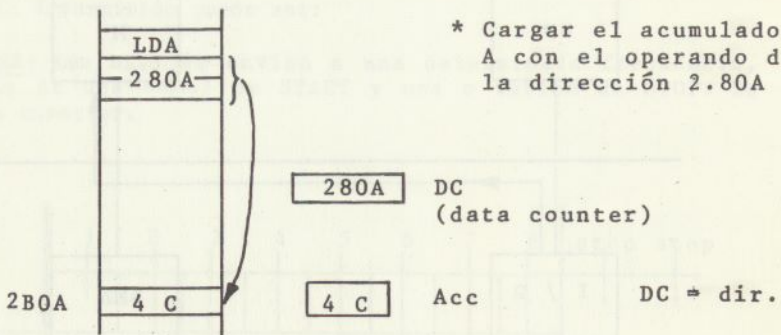
Son los métodos que existen de indicar, a partir del código de operación, el lugar donde se encuentra el operando a tratar.

Un operando puede estar guardado bien en un registro especial de trabajo o bien en memoria.

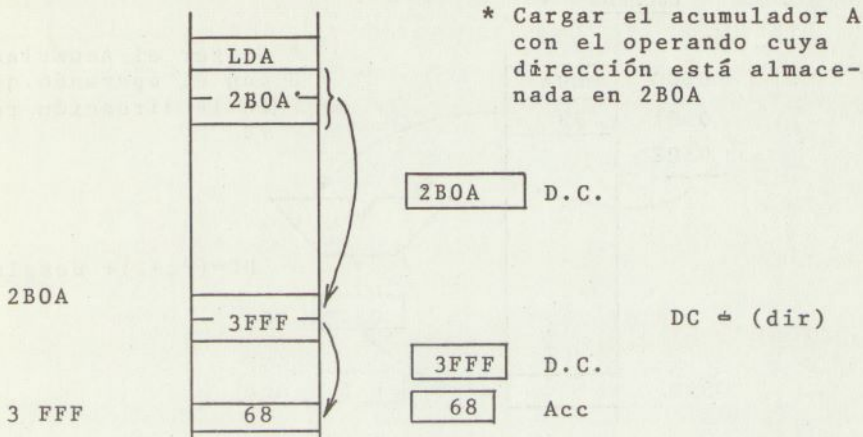
- a) Direccionamiento implícito. Cuando el operando se encuentra en un registro, al que expresamente hace referencia la instrucción;
EJEMPLO: Complementar el Acumulador

C M A

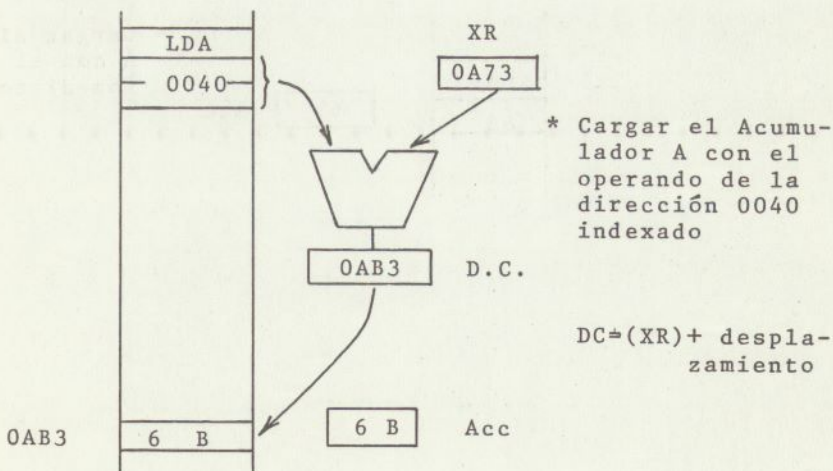
- b) Direccionamiento directo. Después del código de operación viene la dirección del operando.
 Si la dirección del operando viene en un solo byte, se dice que es un direccionamiento directo a página base es decir, a las 256 primeras posiciones de memoria.
 Si la dirección viene en dos bytes, es un direccionamiento directo extendido, y puede apuntar a cualquier posición de memoria.
EJEMPLO:



- c) Direccionamiento indirecto. Para conocer la dirección absoluta del operando es necesaria una lectura intermedia de la memoria. La dirección de memoria apuntada detrás del código de operación no contiene el operando, sino la dirección donde está el operando.

EJEMPLO:d) Direccionamiento indexado

Se obtiene la dirección del operando anadiendo al registro índice un desplazamiento, indicado después del código de operación.

EJEMPLO:

También se pueden combinar el direccionamiento indexado junto con el indirecto, serán:

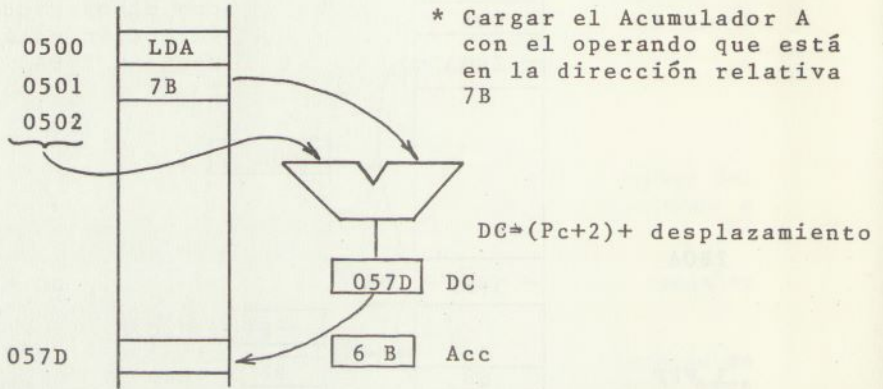
Preindexado = indexado + indirecto

Postindexado = indirecto + indexado

e) Direccionamiento relativo

La dirección del operando que va expresada a continuación del código de operación toma como origen relativo el valor del contador de programa.

EJEMPLO:

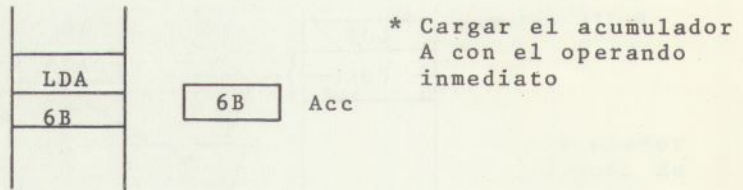


Si en un programa solo hay direccionamientos relativos, tendremos código independiente de la posición (PIC).

f) Direccionamiento inmediato

Cuando el operando está inmediatamente después del código de operación.

EJEMPLO:



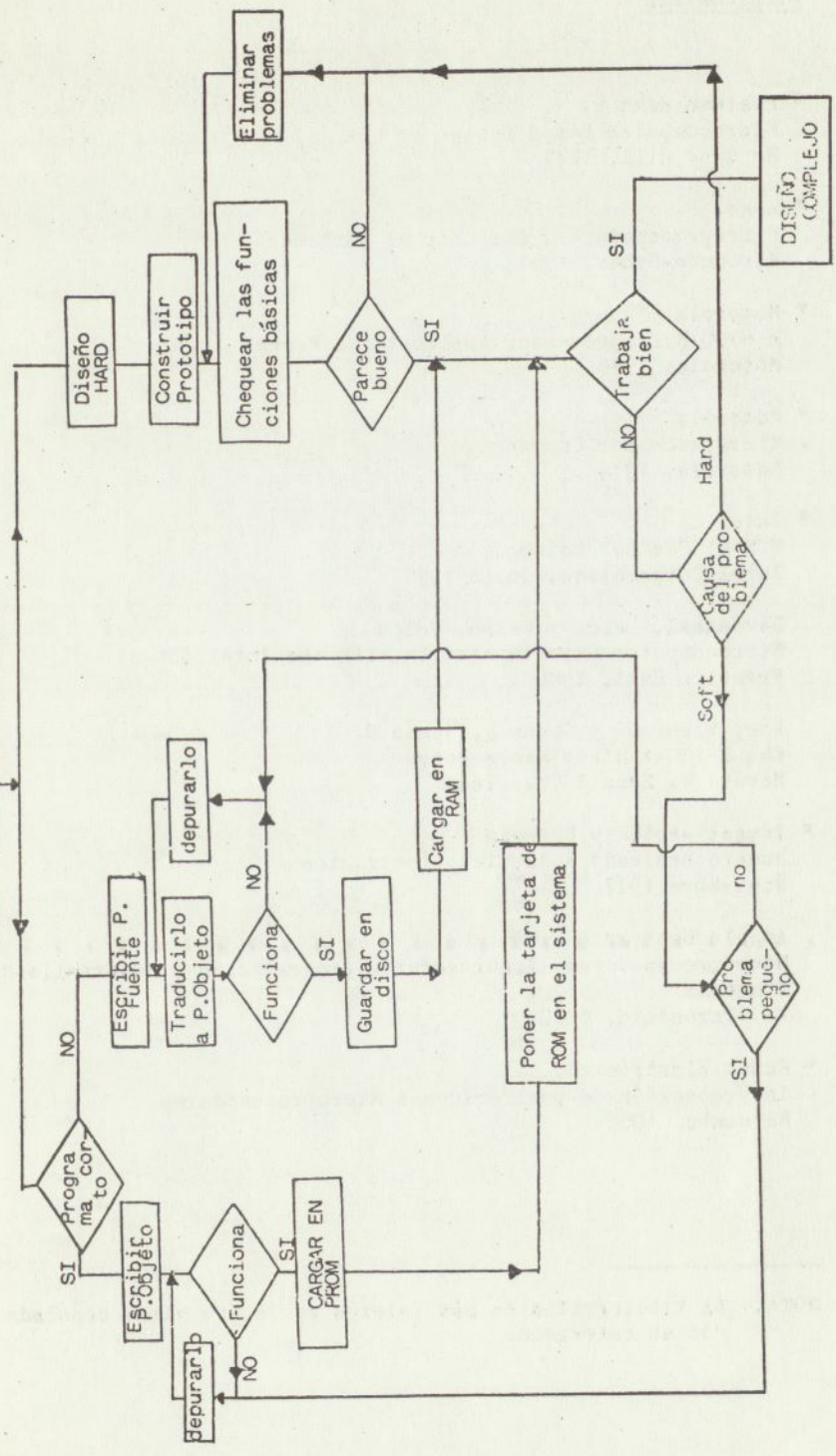
DISEÑO DE UN SISTEMA BASADO EN MICROPROCESADOR

(Tantos sensores, tanta velocidad de testeo, etc.)

(Por microprocesador, cuanta memoria) ...

Decidir las especificaciones del sistema

Decidir el compromiso Hard - Soft



BIBLIOGRAFIA

- * Peatman John B.
Microcomputer Based Design
Mc Graw Hill, 1977
- * Zaks, R.
Microprocesadores, del chip al sistema
Marcombo-Sybex, 1981
- * Motorola
M 6800 Microprocessor Applications Manual
Motorola, 1975
- * Motorola
Microprocessor Course
Motorola, 1979
- * Intel
MCS-85 Seminar Notebook
Intel Corporation, Julio 1977
- Leventhal, Lance y Walsh, Colin
Microcomputer Experimentation with the Intel SDK-85
Prentice Hall, 1980
- Poe, Elmer C. y Goodwin, James C.
The S-100 & other Micro buses
Howard W. Sams & Co., Inc.
- * Investigación y Ciencia
Numero dedicado a la Microelectronica
Noviembre 1977
- Angulo Usategui, S.M.
Microprocesadores: Arquitectura, programación y desarrollo de
Sistemas
Ed. Paraninfo, 1980
- * Mundo Electrónico
Interconexión de periféricos a Microprocesadores
Marcombo, 1980

NOTA.- La bibliografía de más interés es la que viene señalada con un asterisco.