

Universidad Nacional de Ingeniería
Facultad de Ciencias

Arquitectura de Computadores

El Microcontrolador 8051

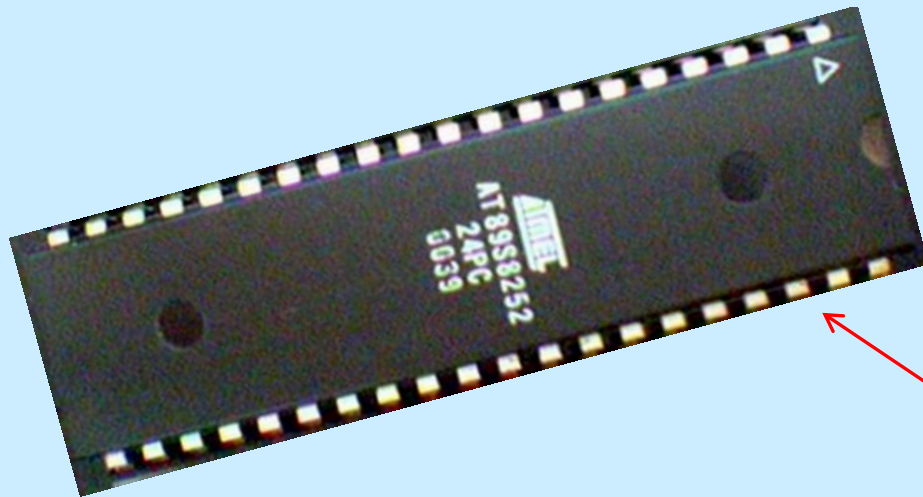
Prof: Lic. César Martín Cruz S.

2013-II

¿Qué es un microcontrolador?

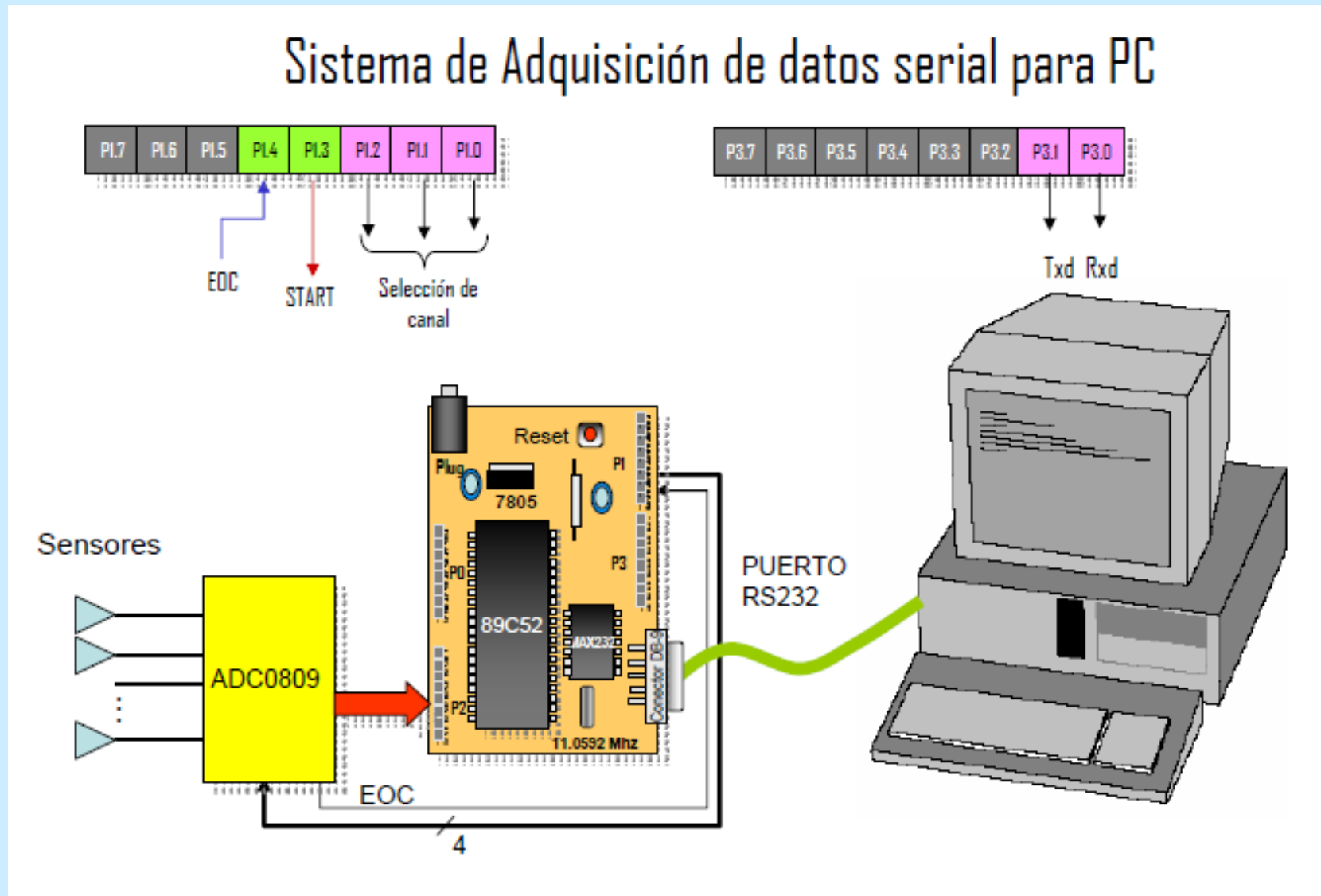
Un **microcontrolador** es un sistema económico de computador en un solo circuito integrado (chip) para hacer frente a tareas específicas, por ejemplo, mostrar información a través de LEDs, displays o controlar rutinas de ciertos dispositivos.

El conjunto más utilizado de los microcontroladores pertenecen a la familia 8051. Los Microcontroladores 8051 siguen siendo una opción preferida por una amplia comunidad de aficionados y profesionales. A través de 8051, el mundo se convirtió en testigo de la serie más revolucionaria de los microcontroladores.



El **AT89S8252** de ATMEL

Aplicación de un microcontrolador



Familia 8051

Intel fabricó el original 8051 que se conoce como MCS-51. Los otros dos miembros de la familia 8051 son los siguientes:

i. **8052** - Este microcontrolador tiene 256 bytes de RAM y 3 contadores de tiempo. Además de las características estándar del 8051, este microcontrolador tiene un adicional de 128 bytes de RAM y un temporizador. Tiene 8K bytes de ROM dentro del chip. Los programas escritos para el 8051 pueden ser utilizados en el 8052 ya que el 8051 es un subconjunto del 8052.

ii. **8031** - Este microcontrolador tiene todas las características del 8051 a excepción de que no tiene ROM.

Una ROM externa que puede ser tan grande como 64 kbytes debe ser programado y se añade a este chip para su ejecución. La desventaja de la adición de ROM externa es que dos puertos (de los 4 puertos que se tienen) se deben utilizar como buses de datos y direcciones.

Por lo tanto, sólo dos puertos quedan para las operaciones de E/S.

Existen varios microcontroladores 8051

Los 8051 utilizan varios tipos diferentes de memoria, tales como **UV-EPROM**, **Flash** y **NV-RAM**.

i. **8751** – Utiliza la versión UV-EPROM para la ROM del 8051. Este chip tiene sólo 4K bytes de UV-EPROM. Se requiere tener acceso a la EPROM y el borrador UV-EPROM (Luz ultravioleta) para borrar el contenido dentro del chip antes de que se programe de nuevo. La desventaja del uso de esta memoria es el tiempo de espera de alrededor de 20 minutos para borrar el contenido de la memoria. Debido a esta limitación, se fabrican versiones del 8051 con flash y NV-RAM.

ii. **AT89C51** de Atmel Corporation contiene el flash ROM del 8051, que se conoce popularmente como AT89C51 ('C' en el número indica CMOS). La memoria flash puede borrar el contenido en cuestión de segundos. Por lo tanto, 8751 se sustituye por AT89C51 para erradicar el tiempo de espera necesario para borrar el contenido y poder, acelerar el tiempo de desarrollo. Para construir un sistema basado en el AT89C51, es esencial contar con un quemador (programador) de ROM que soporte la memoria flash.

Serie chip	FLASH ROM	RAM	Pines E/S	Timer	Interrupciones	Vcc	Empaque_tamiento
AT89C51	4Kbytes	128 bytes	32	2	5	5V	40
AT89C52	8Kbytes	256 bytes	32	3	6	5V	40
AT89C1051	1Kbytes	64	15	1	3	3V	20
AT89C2051	2Kbytes	128	32	3	6	3V	20
AT89LV51	4Kbytes	128	32	2	5	3V	40
AT89LV52	8Kbytes	128	32	3	6	3V	40

Tabla 1.1 – Algunas versiones del 8051 de Atmel (Todos con memoria Flash ROM)

iii. **DS5000** de Dallas Semiconductor (ahora Maxim) contiene un NV-RAM como ROM del 8051, que se conoce popularmente como DS5000. El puerto serie de la PC se utiliza para cargar el programa en la ROM. La ventaja de la NV-RAM es la facilidad de borrar el contenido de un byte a la vez.

Características del 8051 original

Las principales características son:

- i.** RAM de 128 Bytes (memoria de datos).
- ii.** ROM de 4Kbytes (Memoria dentro del chip).
- iii.** Puerto Serie – Usando el UART que lo hace más simple de realizar la comunicación serie.
- iv.** Dos Temporizador/ Contador de 16 bits.
- v.** Pines de Input/output – 4 Puertos de 8 bits cada uno sobre un sólo chip.
- vi.** 5 Fuentes de Interrupción programables con niveles de prioridad.
- vii.** ALU (Arithmetic Logic Unit) de 8 bits.
- viii.** Arquitectura de memoria Harvard (memoria de datos separado de la memoria de programa (código)) – Tiene un bus de direcciones de 16 bits (para la RAM y la ROM) y 8 bits de bus de datos.
- ix.** El 8051 puede ejecutar 1 millón de instrucciones por segundo con una frecuencia de reloj de 12MHz.

Diagrama de bloques del 8051/52

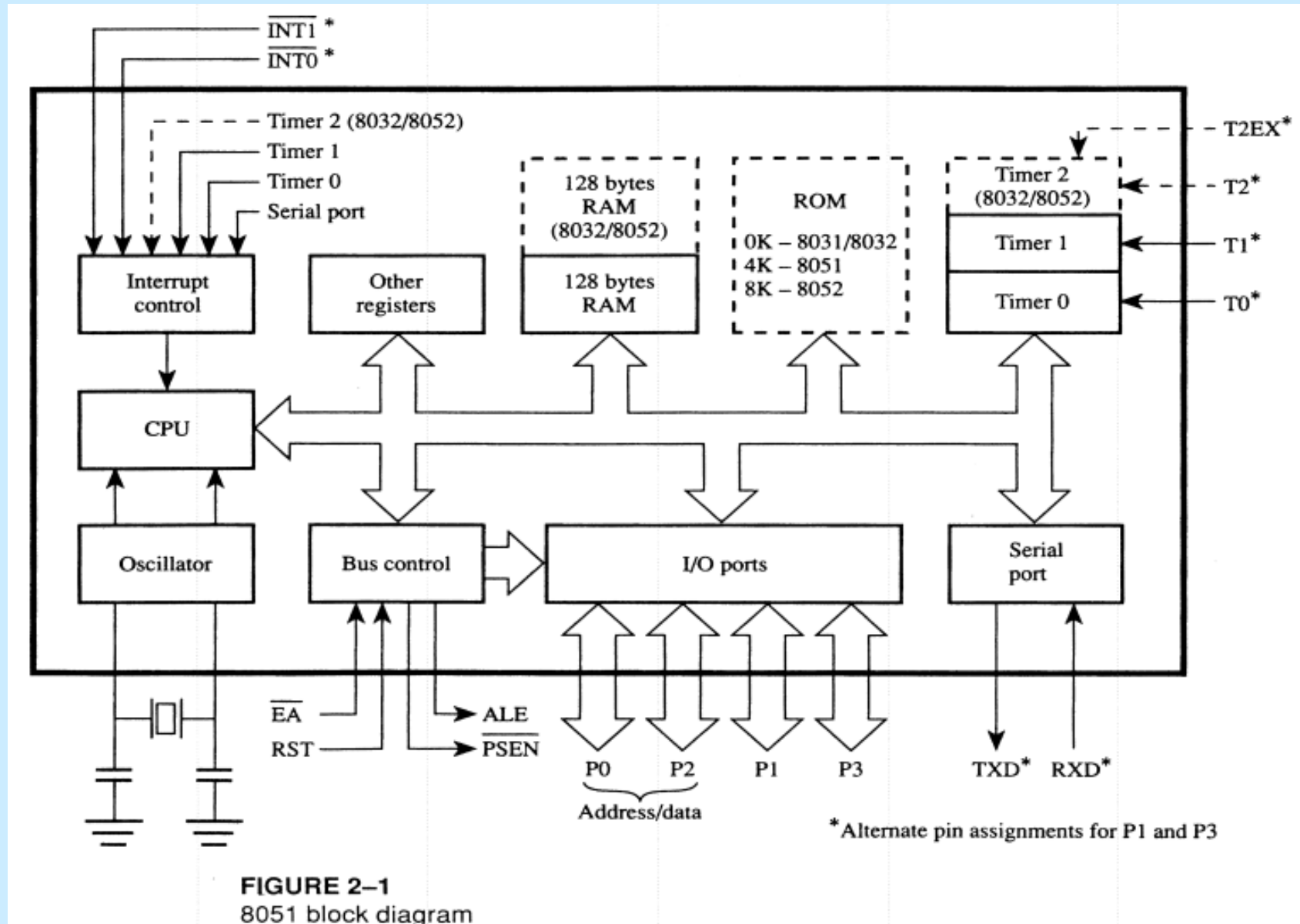


Diagrama de Pines

AT89C52

Pin	Function	Pin	Function
1	(T2) P1.0	40	VCC
2	(T2 EX) P1.1	39	P0.0 (AD0)
3	P1.2	38	P0.1 (AD1)
4	P1.3	37	P0.2 (AD2)
5	P1.4	36	P0.3 (AD3)
6	P1.5	35	P0.4 (AD4)
7	P1.6	34	P0.5 (AD5)
8	P1.7	33	P0.6 (AD6)
9	RST	32	P0.7 (AD7)
10	(RXD) P3.0	31	EA/VPP
11	(TXD) P3.1	30	ALE/PROG
12	(INT0) P3.2	29	PSEN
13	(INT1) P3.3	28	P2.7 (A15)
14	(T0) P3.4	27	P2.6 (A14)
15	(T1) P3.5	26	P2.5 (A13)
16	(WR) P3.6	25	P2.4 (A12)
17	(RD) P3.7	24	P2.3 (A11)
18	XTAL2	23	P2.2 (A10)
19	XTAL1	22	P2.1 (A9)
20	GND	21	P2.0 (A8)

AT89S52

Pin	Function	Pin	Function
1	(T2) P1.0	40	VCC
2	(T2 EX) P1.1	39	P0.0 (AD0)
3	P1.2	38	P0.1 (AD1)
4	P1.3	37	P0.2 (AD2)
5	P1.4	36	P0.3 (AD3)
6	(MOSI) P1.5	35	P0.4 (AD4)
7	(MISO) P1.6	34	P0.5 (AD5)
8	(SCK) P1.7	33	P0.6 (AD6)
9	RST	32	P0.7 (AD7)
10	(RXD) P3.0	31	EA/VPP
11	(TXD) P3.1	30	ALE/PROG
12	(INT0) P3.2	29	PSEN
13	(INT1) P3.3	28	P2.7 (A15)
14	(T0) P3.4	27	P2.6 (A14)
15	(T1) P3.5	26	P2.5 (A13)
16	(WR) P3.6	25	P2.4 (A12)
17	(RD) P3.7	24	P2.3 (A11)
18	XTAL2	23	P2.2 (A10)
19	XTAL1	22	P2.1 (A9)
20	GND	21	P2.0 (A8)

Descripción de Pines

MNEMÓNICO	CONEXIÓN	TIPO	NOMBRE Y FUNCIÓN												
P1.0-P1.7	1 - 8	E/S	<p>PUERTO 1. Es un puerto I/O bidireccional, con pull-ups internos. Puede activar hasta 4 compuertas TTL, cuando se escriben 1´s en el puerto, éste puede ser utilizado como entrada, además P1.0 y P1.1 pueden ser configurados como entradas del timer/contador 2, también recibe la dirección baja durante la programación y la verificación.</p> <table><tr><th>Pines Puerto 1</th><th>Funciones alternativas</th></tr><tr><td>P1.0</td><td>T2 (external count input to Timer/Counter 2), clock-out</td></tr><tr><td>P1.1</td><td>T2EX (Timer/Counter 2 capture/reload trigger and direction control)</td></tr><tr><td>P1.4</td><td>MOSI (usado para In-System Programming en el AT89S52)</td></tr><tr><td>P1.5</td><td>MISO (usado para In-System Programming en el AT89S52)</td></tr><tr><td>P1.6</td><td>SCK (usado para In-System Programming en el AT89S52)</td></tr></table>	Pines Puerto 1	Funciones alternativas	P1.0	T2 (external count input to Timer/Counter 2), clock-out	P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)	P1.4	MOSI (usado para In-System Programming en el AT89S52)	P1.5	MISO (usado para In-System Programming en el AT89S52)	P1.6	SCK (usado para In-System Programming en el AT89S52)
Pines Puerto 1	Funciones alternativas														
P1.0	T2 (external count input to Timer/Counter 2), clock-out														
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)														
P1.4	MOSI (usado para In-System Programming en el AT89S52)														
P1.5	MISO (usado para In-System Programming en el AT89S52)														
P1.6	SCK (usado para In-System Programming en el AT89S52)														
RST	9	E	RESET. Una entrada alta en esta línea durante dos ciclos máquina, mientras el oscilador está funcionando, detiene el dispositivo.												
P3.0-P3.7	10-17	E/S	<p>PUERTO 3. Es un puerto bidireccional con fijadores de nivel internos (PULL-UP). Cuando se escriben 1´s sobre el puerto, las líneas pueden ser utilizadas como entradas en alta impedancia.</p> <p>El puerto 3 es utilizado además, para producir señales de control de dispositivos externos tales como:</p>												

Descripción de Pines...

			Pines Puerto 3	Funciones alternativas
			P3.0(RXD)	Puerto serie de entrada.
			P3.1(TXD)	Puerto serie de salida.
			P3.2(INT0)	Interrupción externa 0.
			P3.3(INT1)	Interrupción externa 1.
			P3.4(T0)	Entrada externa timer 0.
			P3.5(T1)	Entrada externa timer 1.
			P3.6(WR)	Habilitador de escritura para memoria externa de datos.
			P3.7(RD)	Habilitador de lectura para memoria externa de datos.
XTAL2	18	S	CRISTAL 2. Es la salida del amplificador oscilador inversor.	
XTAL1	19	E	CRISTAL 1. Esta es la entrada del cristal para el circuito oscilador (generador del reloj interno) que amplifica e invierte la entrada.	
GND	20		Tierra referencia 0vs.	
P2.0-P2.7 (A8-A15)	21-28	E/S	PUERTO 2. Es un puerto I/O bidireccional, con pull-ups internos. Puede activar hasta 4 compuertas TTL, cuando se escriben 1's en el puerto, éste puede ser utilizado como entrada en alta impedancia. Como entradas, las líneas que son externamente colocadas en la posición baja, proporcionarán una corriente hacia el exterior. El puerto 2 es utilizado además, para direccionar memoria externa. Este puerto, emite el byte más alto de la dirección durante la búsqueda de datos en la memoria de programa externa y durante el acceso a memoria de datos externa que usan direccionamientos de 16 bits. Recibe también las direcciones altas durante la programación y la verificación.	

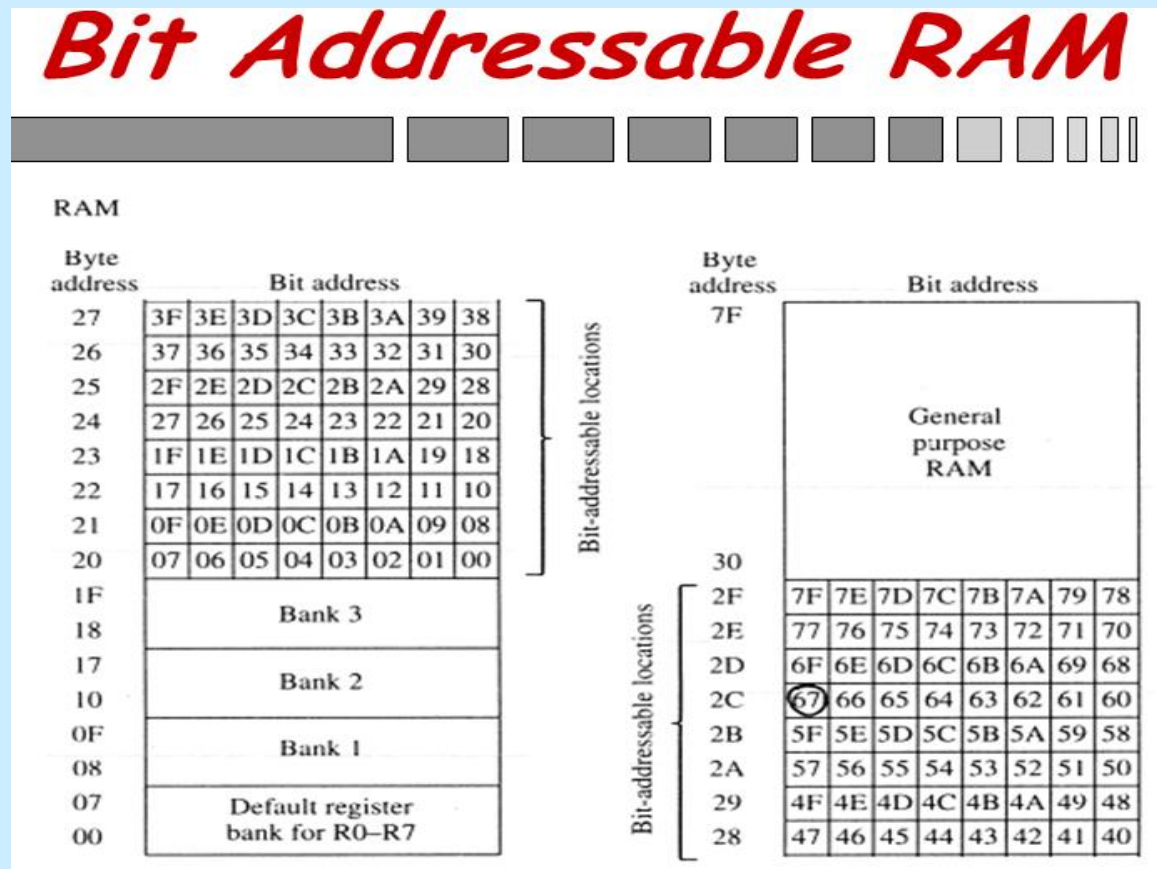
Descripción de Pines...

PSEN	29	S	PROGRAM STORE ENABLE. Habilitador de lectura para memoria de programa externa. Cuando el AT89C52 está ejecutando un código de una memoria de programa externa, PSEN es activada dos veces cada ciclo de máquina, excepto cuando se accesa a la memoria de datos externa que omiten las dos activaciones del PSEN externos. PSEN no es activado cuando se usa la memoria de programa interna.
ALE(PROG)	30	E/S	ADDRESS LATCH ENABLE. Un pulso positivo de salida, permite fijar el byte bajo de la dirección durante el acceso a una memoria externa. En operación normal. Note que un pulso del ALE es emitido durante cada acceso a la memoria de datos externos. Durante la programación este pin es la entrada del pulso de programación.
EA	31	E/S	EXTERNAL ACCESS ENABLE. Habilitar el acceso externo. EA debe ser puesto a GND con el fin de activar el dispositivo para buscar código de posiciones externas de memoria de programa a partir de 0000H hasta FFFFH. EA deberá colocarse a VCC para ejecuciones de programas internos. Este pin también recibe la programación de 12 voltios de tensión de habilitación (VPP) durante la programación de Flash ROM.
P0.0-P0.7 (AD0-AD7)	32-39	E/S	PUERTO 0. Es un puerto I/O bidireccional con salidas en colector abierto. Cuando el puerto tiene 1's escritos, las salidas están flotadas y pueden servir como entradas en alta impedancia. Puede activar hasta 8 compuertas TTL y es el que recibe los datos en la programación necesita de resistencias pull-ups en la programación y en la verificación.
Vcc	40		Se conecta a una fuente de 5 voltios.

Arquitectura de la memoria

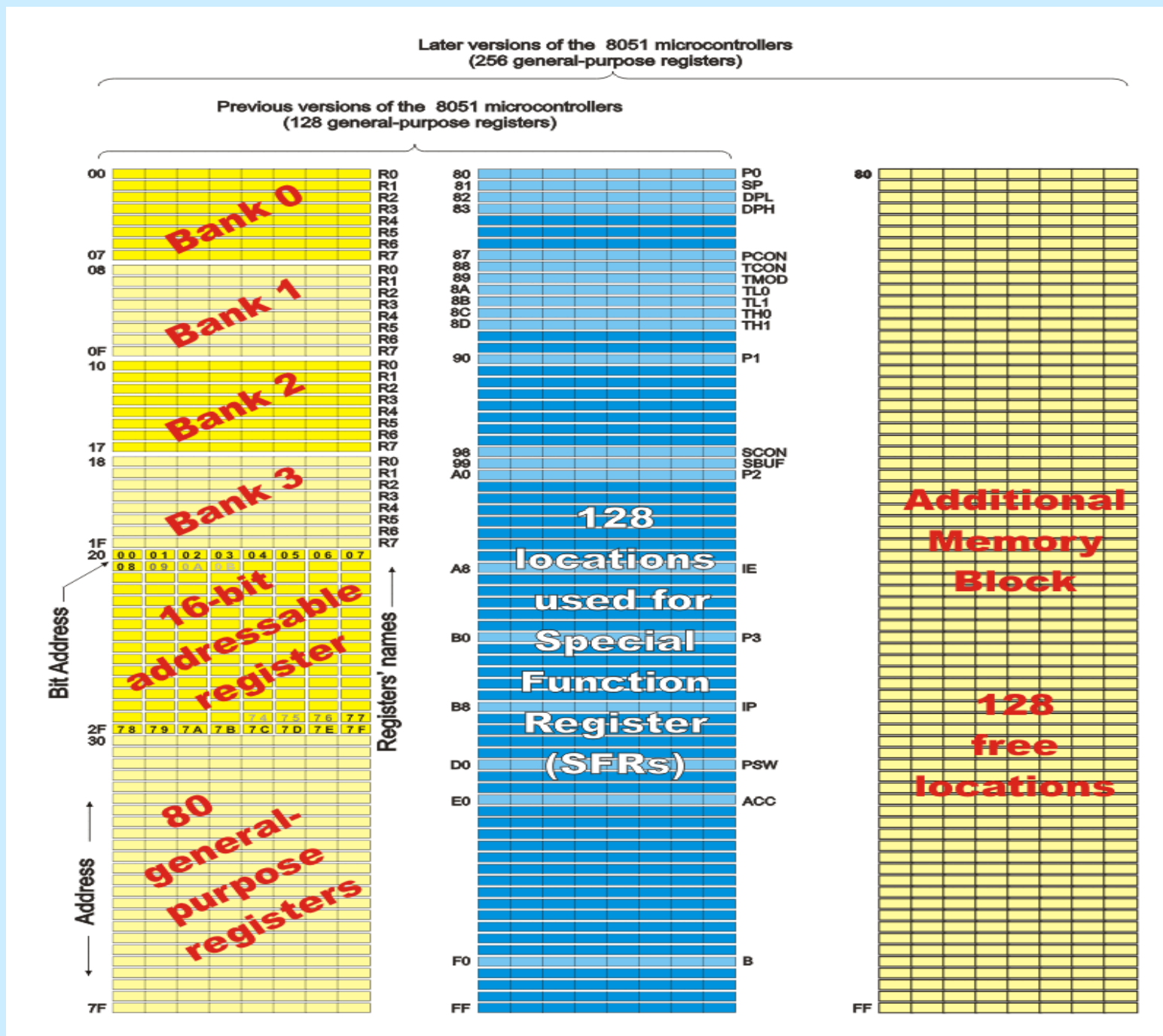
Los 4 tipos de memoria en el 8051 son:

- i. **RAM Interna.** Esta memoria es localizada desde la dirección 0 a 0xff. Las posiciones de memoria de **0x00 a 0x7F** son accedidos directamente. Los bytes desde **0x20 a 0x2F** son direccionables por bit.



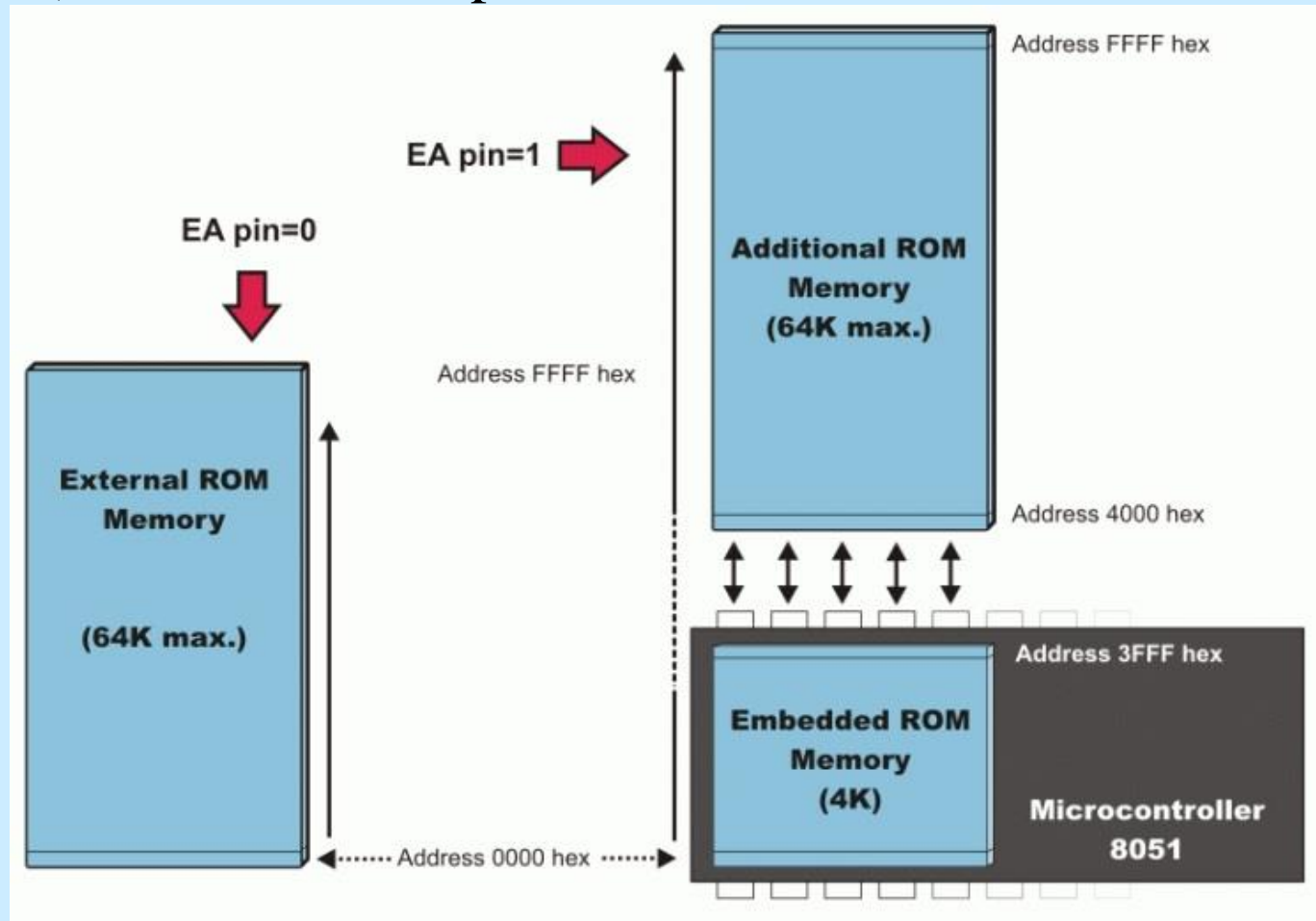
- ii. **Registros de funciones especiales** (Special Function Registers (SFR)). Localizado desde la dirección **0x80 a 0xFF** de la memoria. Las mismas instrucciones usadas para la mitad más baja de la RAM interna pueden ser usadas para el acceso de SFRs. Los SFRs son también direccionables por bit.

<i>Bit Addressable RAM</i>															
Byte address	Bit address								Byte address	Bit address					
98	9F	9E	9D	9C	9B	9A	99	98	SCON	FF					
										F0	F7	F6	F5	F4	F3
											F2	F1	F0		B
90	97	96	95	94	93	92	91	90	P1	E0	E7	E6	E5	E4	E3
											E2	E1	E0		ACC
8D	not bit addressable								TH1						
8C	not bit addressable								TH0	D0	D7	D6	D5	D4	D3
8B	not bit addressable								TL1		D2	-	D0		PSW
8A	not bit addressable								TL0	B8	-	-	-	BC	BB
89	not bit addressable								TMOD		BA	B9	B8		IP
88	8F	8E	8D	8C	8B	8A	89	88	TCON	B0	B7	B6	B5	B4	B3
87	not bit addressable								PCON		B2	B1	B0		P3
										A8	AF	-	-	AC	AB
83	not bit addressable								DPH		AA	A9	A8		IE
82	not bit addressable								DPL	A0	A7	A6	A5	A4	A3
81	not bit addressable								SP		A2	A1	A0		P2
80	87	86	85	84	83	82	81	80	P0	99	not bit addressable				
															SBUF



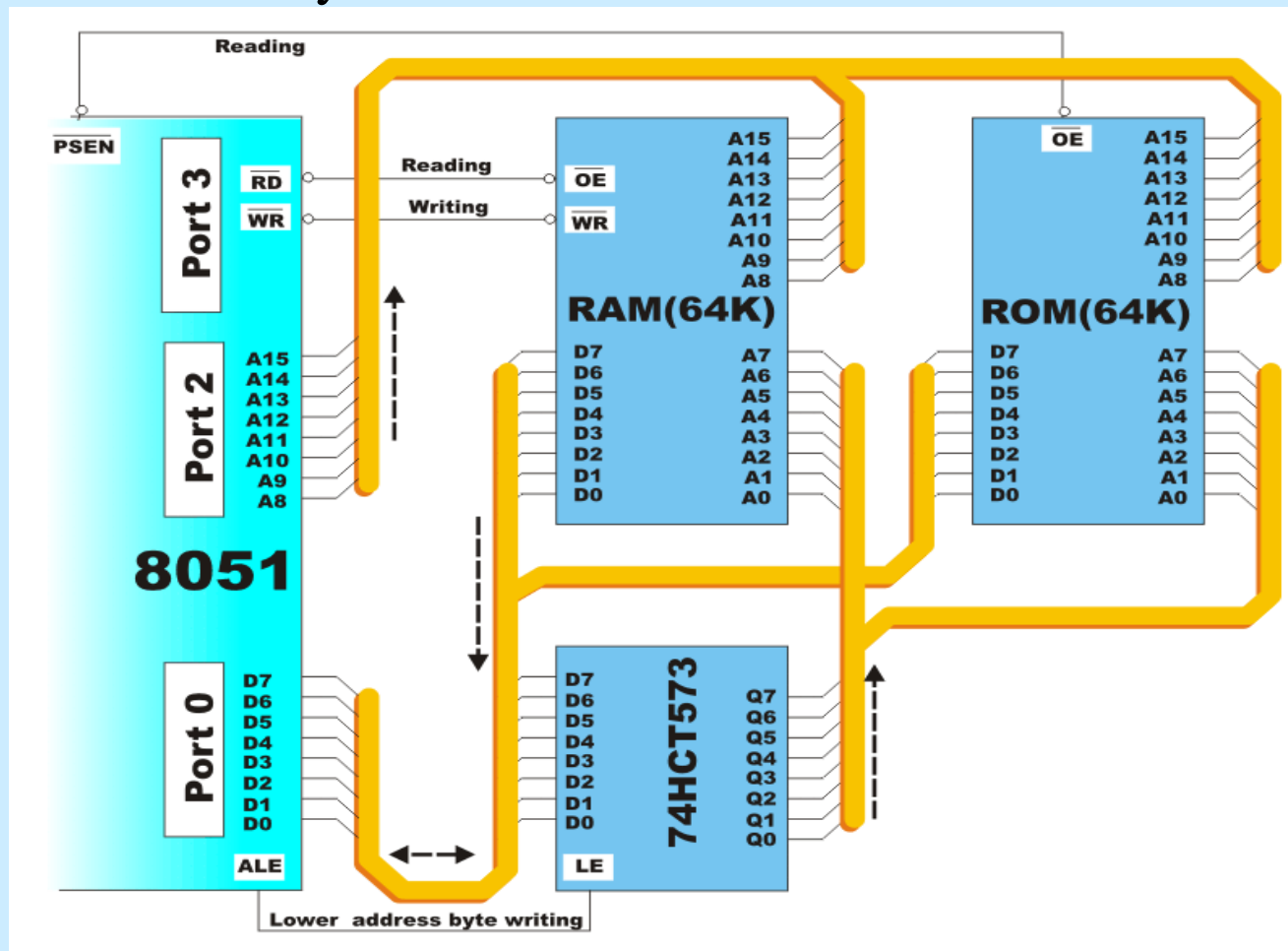
iii. Memoria de Programa. Es la memoria de sólo lectura. Con la ayuda del **registro** de función especial de 16 bits **DPTR**, esta memoria puede también acceder a las tablas de constantes.

iv. Memoria de datos externa. La Instrucción **MOVX** (Move External) debe ser usado para acceder la memoria de datos externa.

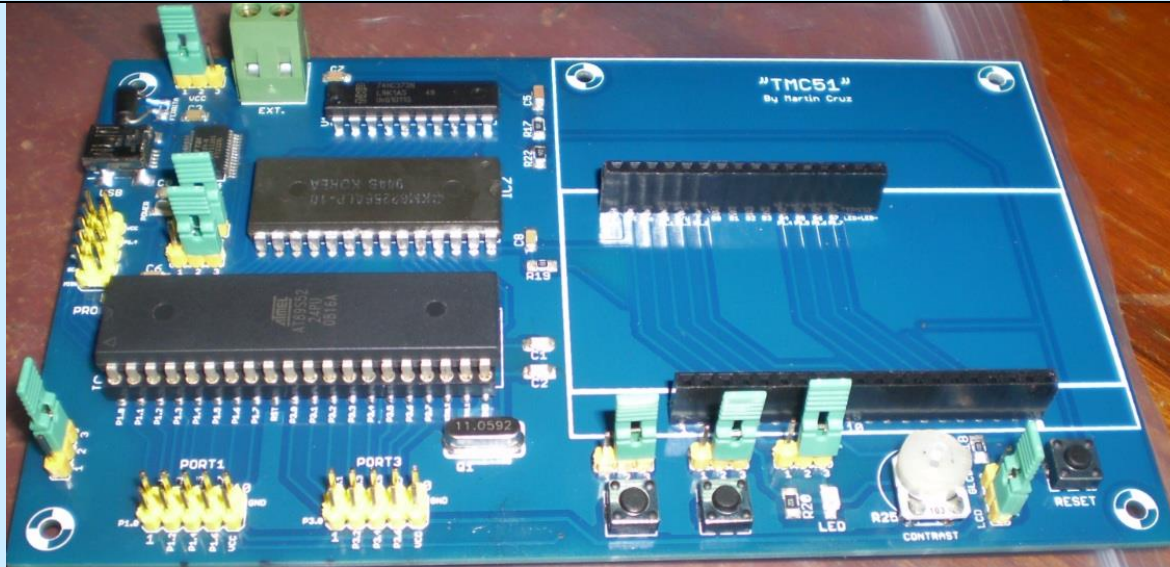


Expansión de memoria

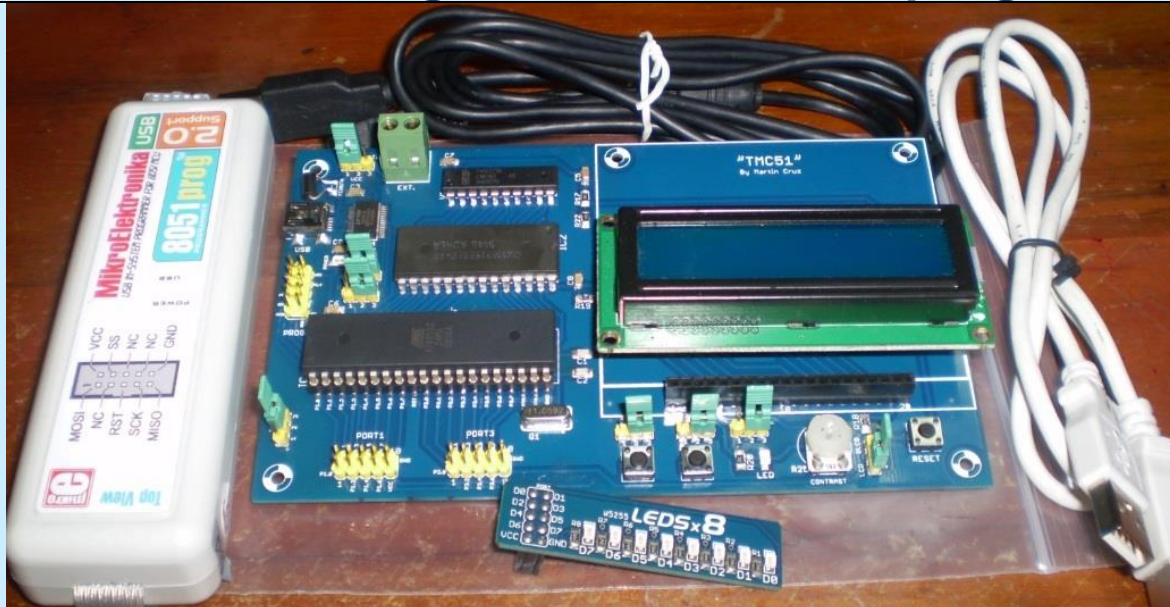
En el caso de que la memoria (RAM o ROM) que se encuentra dentro del microcontrolador no sea suficiente, es posible añadir dos chips de memoria externa con capacidad de **64 KB** cada uno. Los puertos **P0**, **P2** y **P3** se utilizan para su direccionamiento y transmisión de datos.



El Módulo entrenador de la familia 8051("TMC51")



La Tarjeta electrónica con algunos accesorios: programador, LEDs,etc



The diagram illustrates a complete hardware setup for an AT89C52 microcontroller. Key components include:

- AT89C52 (U1):** The central microcontroller unit.
- Crystal Oscillator:** Consists of a crystal (X1) and two matching capacitors (C1, C2).
- Power Supply:** A 5V supply (Vcc=5v) is provided to the microcontroller and other components.
- Reset Circuit:** Utilizes resistors (R1, R9) and capacitors (C3) to manage the RST pin.
- Data Memory:** A 62256 EPROM (U2) provides non-volatile storage, interfaced via its address (A0-A14), data (D0-D7), and control (CE, WE, OE) lines.
- Control Logic:** Two 74HC00 NAND gates (U4:B, U4:C) process signals from the microcontroller to enable the EPROM.
- Buffering:** A 74HC373 (U3) acts as an 8-bit data buffer between the microcontroller and the EPROM.

El Trainer Module of Cruz para la familia 8051(“TMC51”)

Es una tarjeta electrónica con un microcontrolador AT89C52 o AT89S52 que consiste de hardware y firmware. El **firmware** para este tipo de sistemas de propósito general es usualmente un **programa monitor** que permite a los usuarios inspeccionar y modificar los atributos del sistema tal como la memoria y los puertos. Además, un programa monitor debe permitir cargar y ejecutar otros programas aplicativos. Una vez que el programa aplicativo ha sido completamente desarrollado y probado, puede ser grabado en un Flash ROM y el sistema con un microcontrolador puede ser usado como un sistema embebido.

Se utiliza como oscilador el 11.0592 Mhz que permiten sean generados populares velocidades de comunicación en baudios.

Cuando se presiona el botón de RESET este pin se conecta a VCC, reseteando (volviendo a reiniciar) el módulo.

En el “TMC51”...

El Puerto 0 y el puerto 2 son usados para acceso de memoria externa. El Puerto 0 primero emite el byte bajo de la dirección. La señal ALE es usado para enganchar el byte bajo de la dirección. Luego, el Puerto 0 emite o recibe el byte de datos.

Un chip el 74LS373 tipo octal latch es usado para extraer el byte bajo de dirección. Notar que la salida del latch es siempre habilitado.

La memoria de datos externa y la de códigos se superponen. Esto se logra haciendo un “AND” de las señales PSEN y RD. La señal de salida del AND es activado (puesto a bajo) si las señales PSEN or RD son bajos. Dos compuertas de un chip 7400 son usados para generar esta señal.

REGISTROS DE FUNCIONES ESPECIALES

Son los registros con los cuales se puede controlar en su totalidad al AT89C52.

Símbolo	Nombre	Dirección
ACC	Acumulador	0E0H
B	Registro B	0F0H
PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer(16 bits)	82H,83H
DPL	Byte bajo del Data Pointer	82H
DPH	Byte alto del Data Pointer	83H
IP	Prioridad de interrupciones	0B8H
IE	Habilitador de interrupciones	0A8H
TMOD	Modo de control del Timer/Contador	89H
TCON	Control del Timer/Contador	88H
TH0	Byte alto del Timer/Contador 0	8CH
TL0	Byte bajo del Timer/Contador 0	8AH
TH1	Byte alto del Timer/Contador 1	8DH

REGISTROS DE FUNCIONES ESPECIALES

Símbolo	Nombre	Dirección
TL1	Byte bajo del Timer/Contador 1	8BH
SCON	Control del puerto serie	98H
SBUF	Buffer de datos del puerto serie	99H
PCON	Control de potencia	87H
P0	Puerto 0	80H
P1	Puerto 1	90H
P2	Puerto 2	0A0H
P3	Puerto 3	0B0H

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

↑ Bit-addressable Registers

SFRS (SPECIAL FUNCTION REGISTERS)

REGISTRO A(Acumulador)

Este registro es de los más importantes, puesto que es utilizado como un registro procesador y en torno a él se realizan la mayoría de las operaciones, tanto aritméticas como lógicas, es por esto que en su arquitectura es el registro más complicado. Es importante mencionar que después de realizada una operación aritmética, el resultado de esta operación aparecerá en el acumulador.

REGISTRO B

El registro B es usado durante las operaciones de multiplicación y división, por lo que se toma como un acumulador auxiliar. Pero además, puede ser utilizado como cualquier otro registro de propósito general.

STACK POINTER

El Stack Pointer es un registro de 8 bits. Contiene la dirección de memoria RAM interna, en la que se guardará el dato al utilizar la instrucción PUSH o en su defecto, de donde el dato contenido en esta dirección, será obtenido al ejecutar la instrucción POP. Después del reset el SP apunta a la dirección 07H, por lo que la primera dirección disponible, por default, será la dirección 08H (R0 del segundo banco de registros). En todas las llamadas a subrutina o interrupción es usado recuperando los datos almacenados con las instrucciones de retorno RET y RETI.

DATA POINTER

El Data Pointer (DPTR) consiste en un byte alto (DPH) y un byte bajo (DPL). Está diseñado para retener direcciones de 16 bits. Puede ser manipulado como un registro de 16 bits, o como dos registros independientes de 8 bits. Este registro se utiliza para mover datos hacia y desde la memoria ROM interna utilizando 16 bits de direccionamiento.

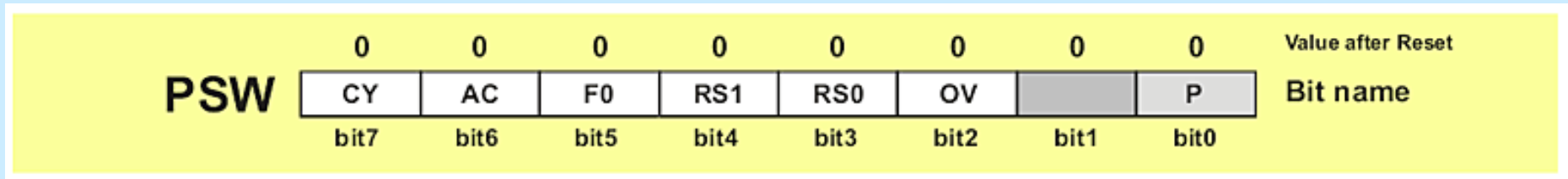
PUERTOS 0 AL 3

P0, P1, P2 y P3 son los latches de los SFR correspondientes a los Puertos 0, 1, 2 y 3 respectivamente. Escribir un 1 en un bit de un puerto (P0, P1, P2 o P3) causa la correspondiente salida de un estado alto por el pin del puerto. Escribir un cero causa la salida de un estado bajo por el mismo.

BUFFER DE DATOS SERIE

El buffer serie son, en realidad, dos registros separados, un buffer para transmitir y un buffer para recibir. Cuando un dato es movido a SBUF, este va al buffer para transmitir y es usado para la transmisión serial. Moviendo un dato al SBUF es como inicia la transmisión serie. Cuando un dato es movido desde el SBUF, viene del buffer de recepción.

Registro de Program Status Word (PSW)



Contiene varios bits de estado que reflejan el estado corriente del CPU. Este registro contiene el bit Carry, Auxiliary Carry, dos bits de selección del banco de registros, flag Overflow, bit de paridad y flag de estado definible por el usuario.

- **P (bit de Paridad).** Si un número almacenado en el acumulador es par entonces este bit será automáticamente puesto a (0), de otro modo será (1). Es usado principalmente durante la transmisión y recepción de datos vía comunicación serie.
- **Bit 1.** Este bit está destinado a ser utilizado en versiones futuras de microcontroladores.

- **OV Overflow** ocurre cuando el resultado de una operación aritmética es más grande que 255 y no puede ser almacenado en un registro. La condición de Overflow causa que el bit OV se fije a (1). De otro modo, se pondrá a (0).

- **Bits RS0, RS1 de selección del banco de registros.** Estos dos bits son usados para seleccionar uno de los cuatro bancos de registros de la RAM.

RS1	RS0	SPACE IN RAM
0	0	Bank0 00h-07h
0	1	Bank1 08h-0Fh
1	0	Bank2 10h-17h
1	1	Bank3 18h-1Fh

-**F0 (Flag 0).** Este es un bit de propósito general disponible para uso.

-**AC (Flag Auxiliary Carry)** es usado para operaciones **BCD** solamente.

-**CY (Flag Carry)** es el (noveno) bit usado para todas las operaciones aritméticas e instrucciones de tipo “shift”.

Si se desea programar ó modificar este registro, se deberá utilizar la instrucción:

MOV PSW,#DATO

donde DATO es la palabra de control en hexadecimal generada a partir de acomodar 1's y 0's en el registro PSW. Si es necesario modificar un solo bit de este registro, se utilizan las siguientes instrucciones:

SETB PSW.x PSW.x = 1

CLR PSW.x PSW.x = 0

Donde: x = número de bit del registro PSW.

Registros de los Timers

El AT89C52 cuenta con tres registros timers (T0, T1 y T2), los cuales a su vez, están compuestos de dos registros de 8 bits cada uno, los registros pares TH0, TL0, TH1, TL1, TH2 y TL2 respectivamente.

PROGRAMACION DEL MICROCONTROLADOR 8051

LENGUAJE ASSEMBLER

Un programa en lenguaje assembler es un conjunto de instrucciones que se pueden convertir en un programa ejecutable en lenguaje de máquina.

Esta conversión se realiza utilizando un programa “**ensamblador**”.

Los instrucciones se dividen en tres categorías:

- 1) **Pseudoinstrucciones (Directivas).** Se emplean para proporcionar información con el fin de convertir el programa a una versión de lenguaje de máquina.
- 2) **Descriptores de Datos.** Utilizados para definir valores constantes y reservar posiciones de memoria.
- 3) **Instrucciones Ejecutables.** Equivalentes a las instrucciones en lenguaje de máquina.

Línea de instrucción en un programa en lenguaje assembler:

Campo Etiqueta.	:	Campo Operación		Campo Operando	;	Campo Comentario
-----------------	---	-----------------	--	----------------	---	------------------

EJEMPLO DE UN PROGRAMA EN LENGUAJE ASSEMBLER

			Program header with basic data
			;Example of a program which generates a sequence of pulses with ;the frequency of 1KHz. The output pin is P0.1. ;Quartz Crystal=12MHz ;Version: 1.0, Date: 5th of May 2003, Author: John Smith
Directives	\$MOD8253		;Program is written for 8253 MCU
	DSEG		;Next segment refers to internal RAM
	Var1	ORG 20h	
	STATE	DS 1	;Byte at location 20h is reserved
CSEG	OUTPUT	BIT Var1.0	;Bit "STATE" is assigned an address
		BIT P1.0	;Bit "OUTPUT" is assigned an address
			;Next segment refers to program memory
			;Program starts at address 0000h
Labels		ORG 0h	
		AJMP START	;Jump to the lable "START"
		ORG 0Bh	;Timer T0 interrupt-vector address
		AJMP INTERRUPT	;Jump to the lable "INTERRUPT"
START		MOV IE,#82h	;Interrupt enabled on Timer T0 overflow
		MOV TMOD,#01	;16-bit Timer
		MOV TH0,#FEh	;Starting value of Timer is FE0Ch
		MOV TL0,#0Ch	
LOOP		SETB STATE	;Bit "STATE" is set
		SETB TR0	;Timer T0 starts operating
		NOP	
		SJMP LOOP	;Program remains in endless loop
INTERRUPT		CLR TR0	;Timer must be stopped before overflow
		MOV TH0,#FEh	;Timer T0 starting value is rewritten
		MOV TL0,#0Ch	
		SETB TR0	;Timer starts recounting
		CPL STATE	;Current state is complemented
		MOV C,STATE	;Bit"STATE" is copied to C bit
		MOV OUTPUT,C	;C bit is copied to bit "OUTPUT"
		RETI	;Return from interrupt routine
END			
Instructions (mnemonics) (operands)			Comments

MODOS DE DIRECCIONAMIENTO

- Son ocho modos de direccionamiento disponibles para el 8051.
- Los diferentes modos de direccionamiento determinan como el byte del operando es seleccionado.

Addressing Modes	Instruction
Register	MOV A, B
Direct	MOV 30H,A
Indirect	ADD A,@R0
Immediate Constant	ADD A,#80H
Relative*	SJMP +127/-128 of PC
Absolute*	AJMP within 2K
Long*	LJMP FAR
Indexed	MOVC A,@A+PC

* Related to program branching instructions

Direcccionamiento por Registro

- La instrucción en este modo involucra la transferencia de información entre registros. El acumulador es el registro A.

- Ejemplo:

MOV R0, A

Direcccionamiento Directo

- En este modo se especifica el operando como la dirección de memoria (típicamente especificado en formato hexadecimal) o dando su nombre abreviado (por ejemplo P3).
- Usado para acceder a registros SFR.
- Ejemplos:

MOV A, P3 ; Transfiere el contenido del Puerto3 al acumulador.

MOV A, 20H ; Transfiere el contenido de la Posición 20H al acumulador.

Direcccionamiento Indirecto

- Este modo usa un puntero para sostener la dirección efectiva del operando.
- Solamente los registros R0, R1 y DPTR pueden ser usados como los registros punteros.
- Ejemplos:

MOV @R0, A

**;Almacena el contenido del
;Acumulador en la posición de
;memoria apuntada por el
;contenido del registro R0. R0
;puede tener una dirección de 8 bits.**

MOVB A, @DPTR

**;Transfiere el contenido de
;la posición de memoria
;apuntada por DPTR al
;acumulador. DPTR puede
;tener una dirección de 16 bits.**

DIRECCIONAMIENTO INMEDIATO DE CONSTANTE

- Este modo usa una constante de 8 o 16 bits como el operando fuente.

- Ejemplos:

ADD A, #030H	;Añade un valor de 8 bits ;30H al registro acumulador.
MOV DPTR, #0FE00H	;Mueve una constante de 16 ;bits al registro DPTR.

DIRECCIONAMIENTO RELATIVO

- Este modo es usado con algún tipo de instrucciones de salto, como SJMP (short jump) y saltos condicionales como JNZ.
- La dirección de destino debe estar dentro del rango -128 y +127 bytes desde la dirección de la instrucción corriente.

- Ejemplo:

GoBack:	DEC A	;Decremento de A
	JNZ GoBack	;Si A no es cero, regresa a GoBack

DIRECCIONAMIENTO ABSOLUTO

- Dos instrucciones asociados con este modo de direccionamiento son ACALL y AJMP.
- Estos son instrucciones de 2 bytes donde la dirección absoluta de 11 bits es especificado como el operando.
- Ejemplo:
**ACALL PORT_INIT ;POR_INIT debe ser
;localizado dentro de los 2kbytes.**

DIRECCIONAMIENTO LARGO

- Este modo de direccionamiento es usado con las instrucciones LCALL y LJMP.
- Es una instrucción de 3 bytes y los últimos 2 bytes especifican una posición de destino de 16 bits.
- El programa siempre regresará a la misma posición no importa donde el programa estaba previamente.
- Ejemplo:
**LCALL TIMER_INIT ;TIMER_INIT es una dirección
;de 16 bits.**

DIRECCIONAMIENTO INDEXADO

- El direccionamiento indexado es útil cuando hay una necesidad de recuperar datos desde una tabla.
- Un registro de 16 bits (puntero de datos) sostiene la dirección de base y el acumulador sostiene un desplazamiento de 8 bits o valor de índice.
- La suma de estos dos registros forma la dirección efectiva para una instrucción JMP o MOVC.
- Ejemplo:

MOV A, #08H	;Offset de inicio de tabla
MOV DPTR, #01F00H	;Dirección de inicio de la tabla
MOVC A,@A+DPTR	;Consigue el valor de la tabla
	;en la dirección de inicio + offset y
	;lo pone en A.

TIPOS DE INSTRUCCIONES

❖ **Las instrucciones del 8051 están divididas en cinco grupos:**

- **Aritméticas**
- **Lógicas**
- **Transferencia de datos**
- **Booleanas**
- **De salto**

Listado de operandos y su significado dentro del conjunto de instrucciones del 8051

- **A** - acumulador;
- **Rn** - es uno de los registros de trabajo (R0-R7) dentro del banco activo de la memoria RAM;
- **Direct** - es cualquier dirección de 8-bits de la RAM;
- **@Ri** - es la posición de la RAM interna o externa direccionado por el registro R0 o R1;
- **#data** - es una constante de 8-bits incluido en la instrucción (0-255);
- **#data16** - es una constante de 16-bits incluida como bytes en la instrucción (0-65535);
- **addr16** - es una dirección de 16-bits;
- **addr11** - es una dirección de 11-bits;
- **rel** - es la dirección de una posición de memoria (desde -128 a +127);
- **bit** - es cualquier bit direccionable como pines I/O, control o bit de estado;
- **C** - es el flag carry del registro de estado (registro PSW).

INSTRUCCIONES ARITMETICAS

ARITHMETIC INSTRUCTIONS		
Mnemonic	Description	Byte
ADD A,Rn	Adds the register to the accumulator	1
ADD A,direct	Adds the direct byte to the accumulator	2
ADD A,@Ri	Adds the indirect RAM to the accumulator	1
ADD A,#data	Adds the immediate data to the accumulator	2
ADDC A,Rn	Adds the register to the accumulator with a carry flag	1
ADDC A,direct	Adds the direct byte to the accumulator with a carry flag	2
ADDC A,@Ri	Adds the indirect RAM to the accumulator with a carry flag	1
ADDC A,#data	Adds the immediate data to the accumulator with a carry flag	2
SUBB A,Rn	Subtracts the register from the accumulator with a borrow	1
SUBB A,direct	Subtracts the direct byte from the accumulator with a borrow	2
SUBB A,@Ri	Subtracts the indirect RAM from the accumulator with a borrow	1
SUBB A,#data	Subtracts the immediate data from the accumulator with a borrow	2
INC A	Increments the accumulator by 1	1
INC Rn	Increments the register by 1	1
INC Rx	Increments the direct byte by 1	2
INC @Ri	Increments the indirect RAM by 1	1
DEC A	Decrements the accumulator by 1	1
DEC Rn	Decrements the register by 1	1
DEC Rx	Decrements the direct byte by 1	1
DEC @Ri	Decrements the indirect RAM by 1	2
INC DPTR	Increments the Data Pointer by 1	1
MUL AB	Multiplies A and B	1

INSTRUCCIONES LOGICAS

LOGIC INSTRUCTIONS		
Mnemonic	Description	Byte
ANL A,Rn	AND register to accumulator	1
ANL A,direct	AND direct byte to accumulator	2
ANL A,@Ri	AND indirect RAM to accumulator	1
ANL A,#data	AND immediate data to accumulator	2
ANL direct,A	AND accumulator to direct byte	2
ANL direct,#data	AND immediate data to direct register	3
ORL A,Rn	OR register to accumulator	1
ORL A,direct	OR direct byte to accumulator	2
ORL A,@Ri	OR indirect RAM to accumulator	1
ORL direct,A	OR accumulator to direct byte	2
ORL direct,#data	OR immediate data to direct byte	3
XRL A,Rn	Exclusive OR register to accumulator	1
XRL A,direct	Exclusive OR direct byte to accumulator	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	1
XRL A,#data	Exclusive OR immediate data to accumulator	2
XRL direct,A	Exclusive OR accumulator to direct byte	2
XORL direct,#data	Exclusive OR immediate data to direct byte	3
CLR A	Clears the accumulator	1
CPL A	Complements the accumulator (1=0, 0=1)	1
SWAP A	Swaps nibbles within the accumulator	1
RL A	Rotates bits in the accumulator left	1
RLC A	Rotates bits in the accumulator left through carry	1
RR A	Rotates bits in the accumulator right	1

INSTRUCCIONES DE TRANSFERENCIA DE DATOS

DATA TRANSFER INSTRUCTIONS		
Mnemonic	Description	Byte
MOV A,Rn	Moves the register to the accumulator	1
MOV A,direct	Moves the direct byte to the accumulator	2
MOV A,@Ri	Moves the indirect RAM to the accumulator	1
MOV A,#data	Moves the immediate data to the accumulator	2
MOV Rn,A	Moves the accumulator to the register	1
MOV Rn,direct	Moves the direct byte to the register	2
MOV Rn,#data	Moves the immediate data to the register	2
MOV direct,A	Moves the accumulator to the direct byte	2
MOV direct,Rn	Moves the register to the direct byte	2
MOV direct,direct	Moves the direct byte to the direct byte	3
MOV direct,@Ri	Moves the indirect RAM to the direct byte	2
MOV direct,#data	Moves the immediate data to the direct byte	3
MOV @Ri,A	Moves the accumulator to the indirect RAM	1
MOV @Ri,direct	Moves the direct byte to the indirect RAM	2
MOV @Ri,#data	Moves the immediate data to the indirect RAM	2
MOV DPTR,#data	Moves a 16-bit data to the data pointer	3
MOVC A,@A+DPTR	Moves the code byte relative to the DPTR to the accumulator (address=A+DPTR)	1
MOVC A,@A+PC	Moves the code byte relative to the PC to the accumulator (address=A+PC)	1
MOVX A,@Ri	Moves the external RAM (8-bit address) to the accumulator	1
MOVX A,@DPTR	Moves the external RAM (16-bit address) to the accumulator	1
MOVX @Ri,A	Moves the accumulator to the external RAM (8-bit address)	1
MOVX @DPTR,A	Moves the accumulator to the external RAM (16-bit address)	1
PUSH direct	Pushes the direct byte onto the stack	2
POP direct	Pops the direct byte from the stack/td>	2

INSTRUCCIONES BOOLEANAS

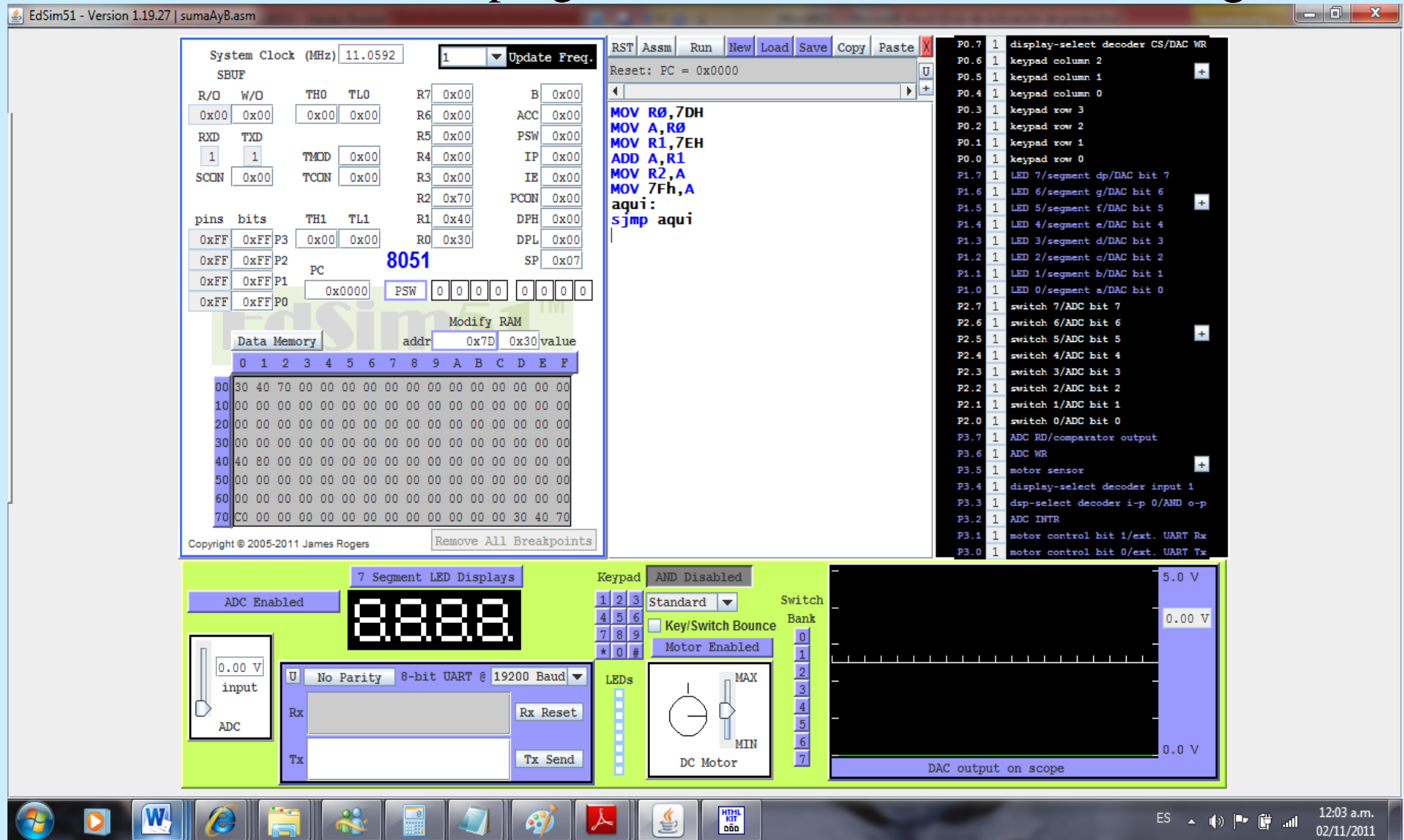
BIT-ORIENTED INSTRUCTIONS		
Mnemonic	Description	Byte
CLR C	Clears the carry flag	1
CLR bit	Clears the direct bit	2
SETB C	Sets the carry flag	1
SETB bit	Sets the direct bit	2
CPL C	Complements the carry flag	1
CPL bit	Complements the direct bit	2
ANL C,bit	AND direct bit to the carry flag	2
ANL C,/bit	AND complements of direct bit to the carry flag	2
ORL C,bit	OR direct bit to the carry flag	2
ORL C,/bit	OR complements of direct bit to the carry flag	2
MOV C,bit	Moves the direct bit to the carry flag	2
MOV bit,C	Moves the carry flag to the direct bit	2

INSTRUCCIONES DE SALTO

BRANCH INSTRUCTIONS		
Mnemonic	Description	Byte
ACALL addr11	Absolute subroutine call	2
LCALL addr16	Long subroutine call	3
RET	Returns from subroutine	1
RETI	Returns from interrupt subroutine	1
AJMP addr11	Absolute jump	2
LJMP addr16	Long jump	3
SJMP rel	Short jump (from -128 to +127 locations relative to the following instruction)	2
JC rel	Jump if carry flag is set. Short jump.	2
JNC rel	Jump if carry flag is not set. Short jump.	2
JB bit,rel	Jump if direct bit is set. Short jump.	3
JBC bit,rel	Jump if direct bit is set and clears bit. Short jump.	3
JMP @A+DPTR	Jump indirect relative to the DPTR	1
JZ rel	Jump if the accumulator is zero. Short jump.	2
JNZ rel	Jump if the accumulator is not zero. Short jump.	2
CJNE A,direct,rel	Compares direct byte to the accumulator and jumps if not equal. Short jump.	3
CJNE A,#data,rel	Compares immediate data to the accumulator and jumps if not equal. Short jump.	3
CJNE Rn,#data,rel	Compares immediate data to the register and jumps if not equal. Short jump.	3
CJNE @Ri,#data,rel	Compares immediate data to indirect register and jumps if not equal. Short jump.	3
DJNZ Rn,rel	Decrements register and jumps if not 0. Short jump.	2
DJNZ Rx,rel	Decrements direct byte and jump if not 0. Short jump.	3
NOP	No operation	1

PRUEBAS DE PROGRAMAS USANDO EL LENGUAJE ENSAMBLADOR DEL 8051

Para ello se va utilizar el programa simulador **EdSim51**. Ver figura:



Ejemplo 1

Dos enteros A=11h y B=16h son almacenados en formato complemento a dos en las posiciones de memoria 40h y 41h.

Hacer un programa que encuentre el valor de la suma de X y Y.

Guardando el resultado en la posición 42h.

Solución:

;El número X=11h es almacenado en 40h

;El número Y=16h es almacenado en 41h

;El resultado será almacenado en 42h

org 0000h

Inicio:

mov R0,40h ;muevo el contenido de 40h a R0

mov R1,41h ;muevo el contenido de 41h a R1

mov A,R0 ;muevo el contenido de R0 al acumulador A

add A,R1 ;sumo R0 y R1 y lo guardo en el acumulador A

mov 42h,A ;almaceno el valor del acumulador A en 42h

Aquí:

sjmp \$;el control se queda dando vueltas aquí.

Ejemplo 2

Hacer un programa que encuentre el resultado de “A-2”, donde A es un número en formato complemento a 2 de 8 bits y que el resultado lo guarde en 52h.

Solución:

;Programa que realiza la operación A-2
;El número A=13h es almacenado en 50h
;El resultado será almacenado en 52h

org 0000h

Inicio:

mov A,50h ;muevo el contenido de 50h al Acumulador
subb A, #2 ;se resta A menos 2 y lo guardo en A
mov 52h,A ;almaceno el valor de A en 52h

Aquí:

sjmp Aquí ;el control se queda dando vueltas aquí.

Ejemplo 3

Hacer un programa que encuentre el resultado de $X + Y - Z$, donde X, Y y Z son números en formato complemento a 2 de 8 bits.

Usar las memorias 50h para X, 51h para Y y 52h para Z. El resultado que lo guarde en 53h.

Solución:

;Programa que realiza la operación $X + Y - Z$

;El número X=15h es almacenado en 50h

;El número Y=3h es almacenado en 51h

;El número Z=3h es almacenado en 52h

org 0000h

Inicio:

mov R0,50h ;muevo el contenido de 50h a R0

mov R1,51h ;muevo el contenido de 51h a R1

mov R2,52h ;muevo el contenido de 52h a R2

mov A,R0 ;muevo el contenido de R0 al Acumulador

add A,R1 ;sumo el acumulador "A" y R1 y lo guardo en A

subb A,R2 ;resto el acumulador "A" y R2 y lo guardo en A

mov 53h,A ;almaceno el valor del acumulador "A" en 53h

Aqui:

sjmp Aqui ;el control se queda aquí dando vueltas.

Ejercicio 1

Hacer un programa que encuentre el resultado de $X - Y + Z$, donde X, Y y Z son números en formato complemento a 2 de 8 bits.

Usar las memorias 50h para X, 51h para Y y 52h para Z. El resultado que lo guarde en 53h y lo muestre en el puerto P1 (LEDS).

Ejemplo 4

Hacer un programa que encuentre el resultado de **(A03C) OR (1BBF)**.
Usar las posiciones de memorias 70h Y 71h para guardar el resultado.
El Byte más significativo guardar en 70h.

Solución:

;Programa que realiza la operación **(A03C) OR (1BBF)**

;El número A0 es almacenado en 50h

;El número 3C es almacenado en 51h

;El número 1B es almacenado en 60h

;El número BF es almacenado en 61h

;El resultado será almacenado en 70h y 71h

org 0000h

Inicio:

mov R0,50h ;muevo el contenido de 50h a R0

mov R1,51h ;muevo el contenido de 51h a R1

mov A,R0 ;muevo R0 al acumulador

orl A,60h ;or lógico entre A y el contenido de la memoria 60h

mov 70h,A ;muevo el contenido de A a 70h

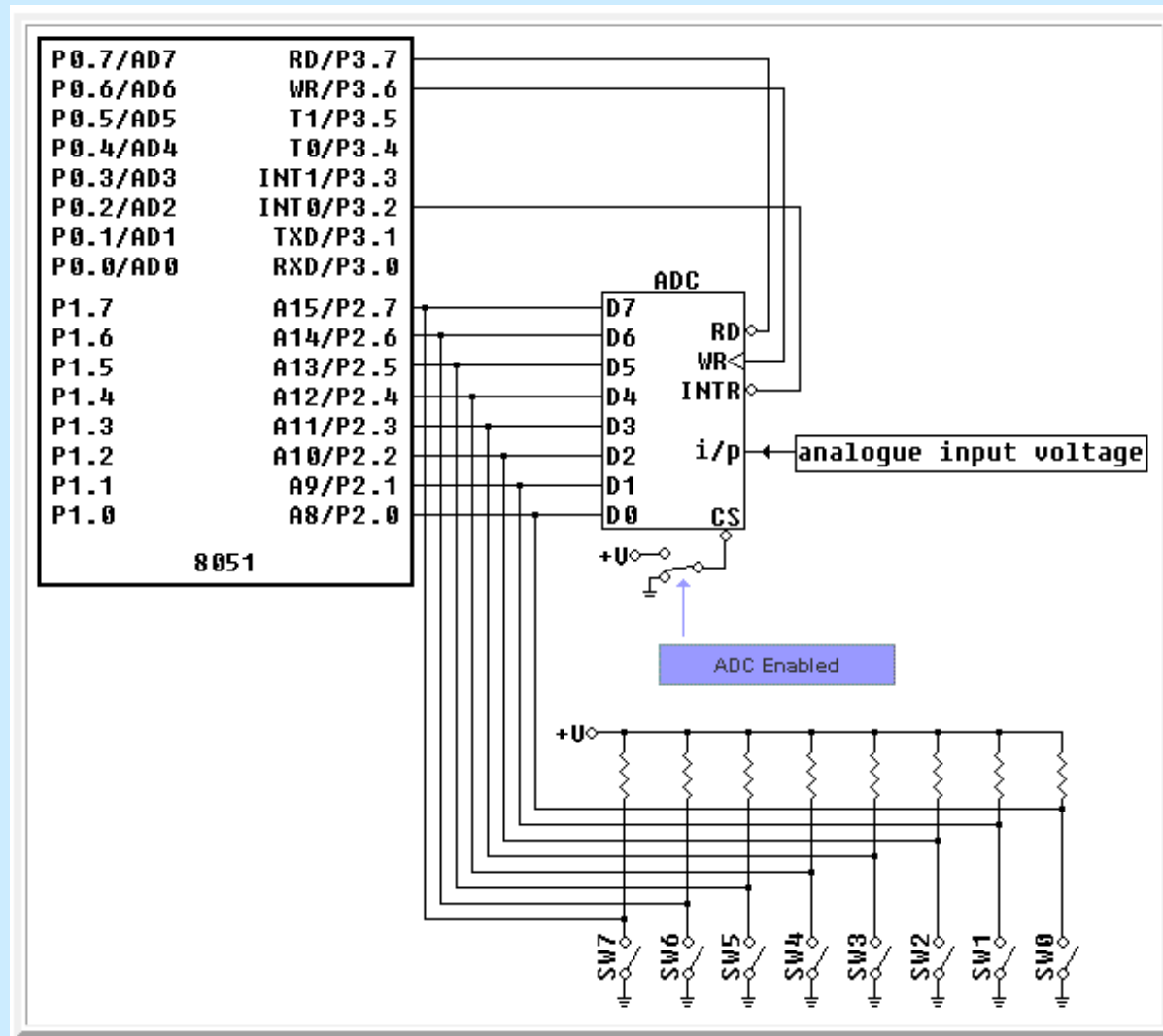
mov A,R1	;muevo R1 a A
orl A,61h	;or lógico entre A y el contenido de la memoria 61h
mov 71h,A	;almaceno el valor de A en 71h

Aqui:

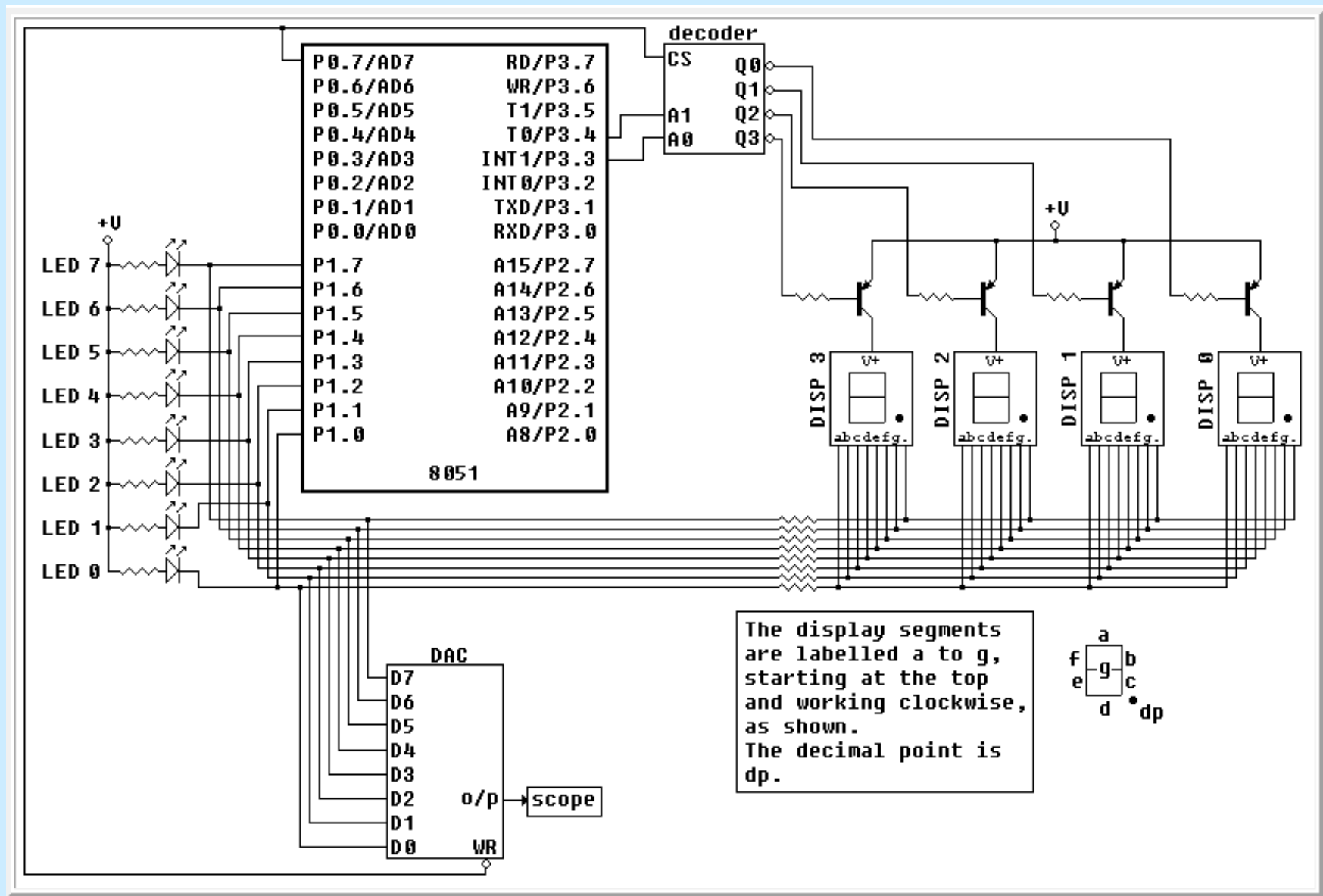
sjmp Aqui	;el control se queda dando vueltas aquí.
------------------	--

DIAGRAMA DE CONEXIONES CON EL 8051

Banco de switches (interruptores) y ADC(convertidor análogo a digital)



Banco de Leds, displays de 7 segmentos y convertidor digital a análogo(DAC)



Ejercicio 2

Hacer un programa que muestre una **barra de progreso**. Utilizando los Leds conectados al puerto P1.

Ejercicio 3

Hacer un programa que utilizando 2 switches conectados a P2.0 y P2.1 seleccione dos juegos de luces diferentes.

LIBRO RECOMENDADO: Con muchos programas y ejercicios.

"Programando el Microcontrolador 8051 en el simulador EDSIM51 y el Módulo TMC51" de César Martín Cruz Salazar

PRECIO:S/. 35

