

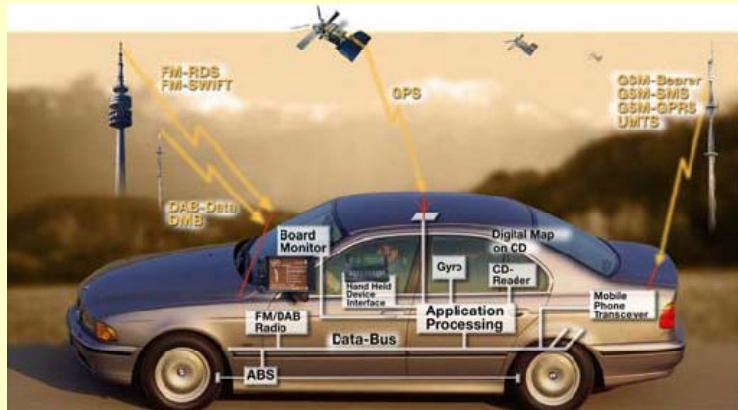
## Microcontrolador

- MCU
  - Microcomputadora en un solo circuito que cumple funciones de CONTROL

## Sistemas Empotrados (embedded)

- Basados en componentes programables (ej. Microcontroladores, DSPs....)
- Son generalmente sistemas reactivos de tiempo real:
  - “Reaccionan” a eventos externos
  - Mantienen interacción permanente.
  - Están continuamente funcionando.
  - Están sujetos a restricciones externas de tiempo
- Realizan varias tareas concurrentemente.

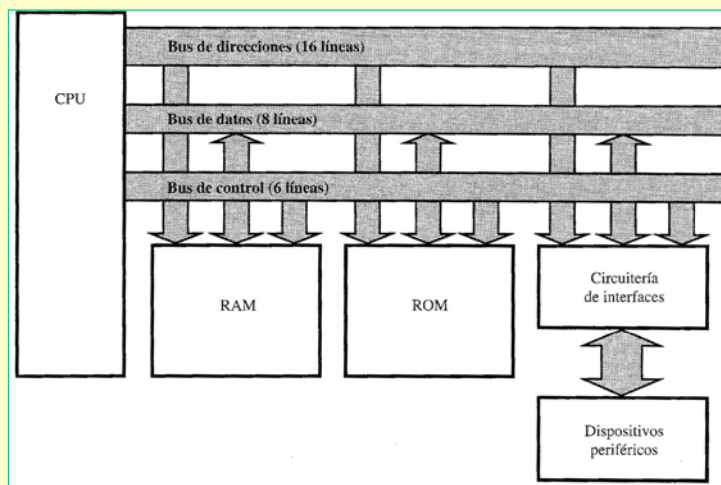
## Un ejemplo completo de SS.EE.



Microcontroladores - Parte 1

3

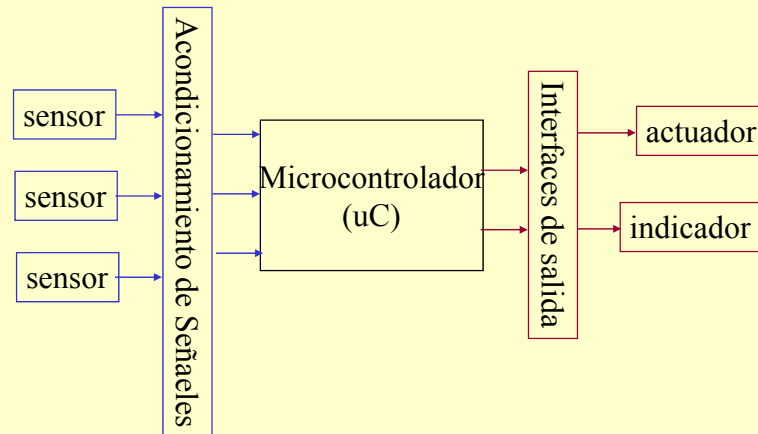
## Microcontrolador



Microcontroladores - Parte 1

4

## Sistema de Adquisición de datos



Microcontroladores - Parte 1

5

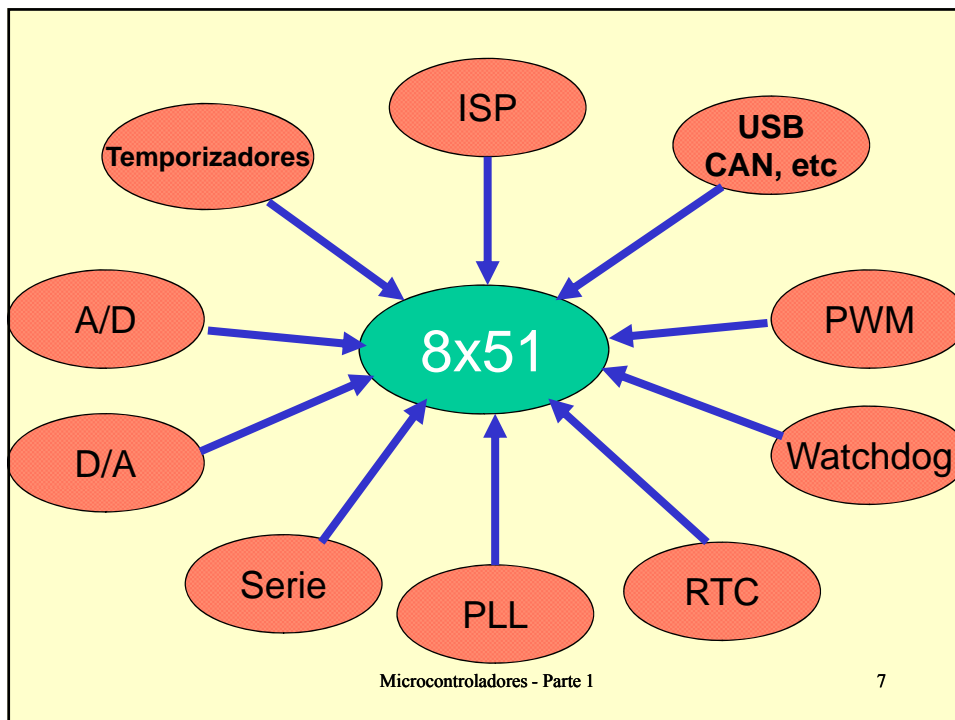
## Filosofía del bit

```

MOV  C,P1.4   ;Se trae al CY el estado del pin 4 de la puerta 1
ANL  C,P1.5   ;AND lógico con el estado de la pata 5 del port 1
ANL  C,P1.6   ;AND lógico del reultado anterior con el estado
              ;de la pata 6 del port 1
CPL  C        ;Se complementa el resultado
MOV  P1.7,C   ;Se escribe el resultado a la pata 7 del port 1
  
```

Microcontroladores - Parte 1

6



## Ejemplos de Familias

High End	AT89C51...					
	RB2	RC2 / IC2	RD2 / ED2 / ID2	15	AC2	AC3
Pins	40/44	40/44	44/64	24/32	44	44
Flash (KB)	16	32	64	16	32	32
Mask ROM	Yes	Yes	Yes	-	-	-
SRAM (B)	1280	1280	2048	512	1280	2304
EEPROM (KB)	-	-	2 (ED2 / ID2)	2	2	2
U(S)ART	1	1	1	1	1	1
SPI	Yes	Yes	Yes	-	-	Yes
TWI	-	IC2	ID2	-	-	-
ADC	-	-	-	10 b	10b	10b
Keyboard	Yes	Yes	Yes	-	-	-
Timers	3	3	3	3	3	3
PCA / PWM	Yes	Yes	Yes	Yes	Yes	Yes
Watchdog	Yes	Yes	Yes	Yes	Yes	Yes
POR / PFD	-	-	Yes	-	-	Yes
ISP	Yes	Yes	Yes	Yes	Yes	Yes
Self Program.	Yes	Yes	Yes	Yes	Yes	Yes
Samples	Now	Now	Now	Now	Now	Now
Production	Now	Now	Now	Now	Now	Now

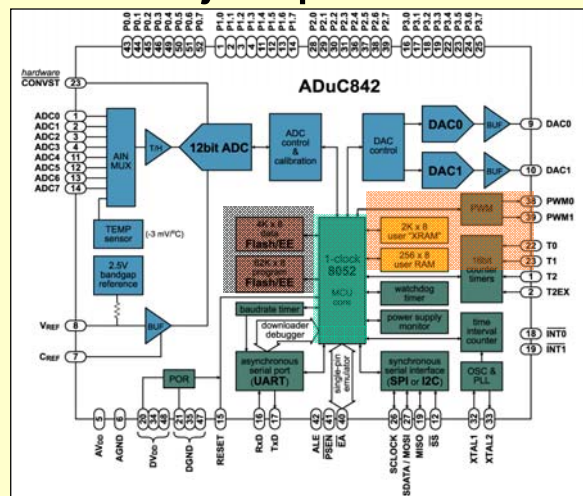
Microcontroladores - Parte 1

8

Part #	ADC	DAC	MCU	Flash/EE Code	Flash/EE Data	RAM	PKGs	Special Features
ADuC812	8-chan 12-bit	Dual 12-bit	12-clock 8052	8K-byte	640-byte	256-byte	52-PQFP 56-CSP	5µs ADC
ADuC814	6-chan 12-bit	Dual 12-bit	12-clock 8052	8K-byte	640-byte	256-byte	28-TSSOP	Small, Low-Cost
ADuC816	Dual 16-bit	Single 12-bit	12-clock 8052	8K-byte	640-byte	256-byte	52-PQFP 56-CSP	Buffered PGA Input
ADuC824	24-bit + 16-bit	Single 12-bit	12-clock 8052	8K-byte	640-byte	256-byte	52-PQFP 56-CSP	Pin-Compatible Upgrade to ADuC816
ADuC831	8-chan 12-bit	Dual 12-bit +Dual PWM	12-clock 8052	62K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Big-Memory" Upgrade to ADuC812
ADuC832	8-chan 12-bit	Dual 12-bit +Dual PWM	12-clock 8052	62K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	Same As ADuC831, But With PLL Clock
ADuC834	24-bit + 16-bit	Sngl 12-bit +Dual PWM	12-clock 8052	62K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Big-Memory" Upgrade to ADuC824
ADuC836	Dual 16-bit	Sngl 12-bit +Dual PWM	12-clock 8052	62K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Big-Memory" Upgrade to ADuC816
ADuC841 *	8-chan 12-bit	Dual 12-bit +Dual PWM	1-clock 8052	8K,32K,62 K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Fast-Core" Upgrade to ADuC831
ADuC842 *	8-chan 12-bit	Dual 12-bit +Dual PWM	1-clock 8052	8K,32K,62 K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Fast-Core" Upgrade to ADuC832
ADuC843 *	8-chan 12-bit	Dual PWM	1-clock 8052	8K,32K,62 K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	stripped-down ADuC842
ADuC844 *	24-bit + 16-bit	Sngl 12-bit +Dual PWM	1-clock 8052	8K,32K,62 K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Fast-Core" Upgrade to ADuC834
ADuC845 *	10-chan 24-bit	Sngl 12-bit +Dual PWM	1-clock 8052	8K,32K,62 K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Fast-Core" ΣΔ with Multi-Channel Input
ADuC846 *	Dual 16-bit	Sngl 12-bit +Dual PWM	1-clock 8052	8K,32K,62 K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	"Fast-Core" Upgrade to ADuC836
ADuC847 *	10-chan 24-bit	(none)	1-clock 8052	8K,32K,62 K-byte	4K-byte	256-byte +2K-byte	52-PQFP 56-CSP	stripped-down ADuC845

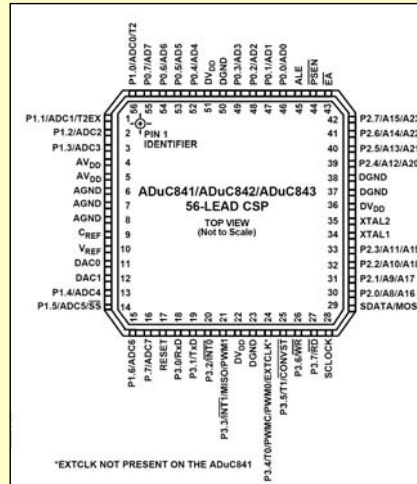
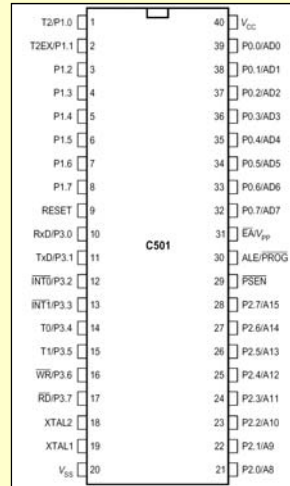
Microcontroladores - Parte 1

# Ejemplos

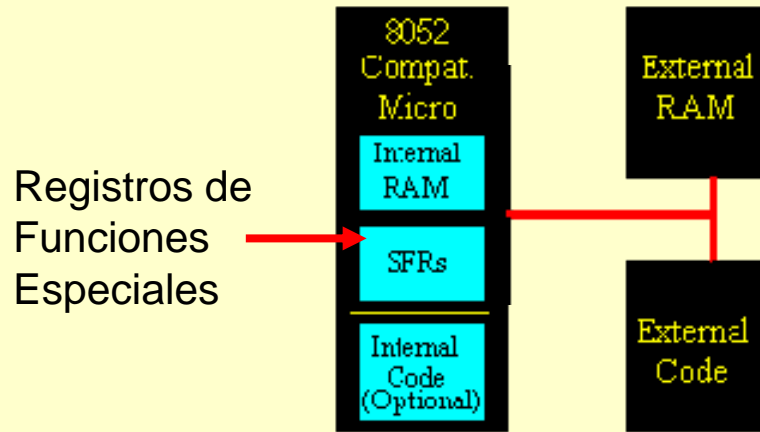


Microcontroladores - Parte 1

## Presentación



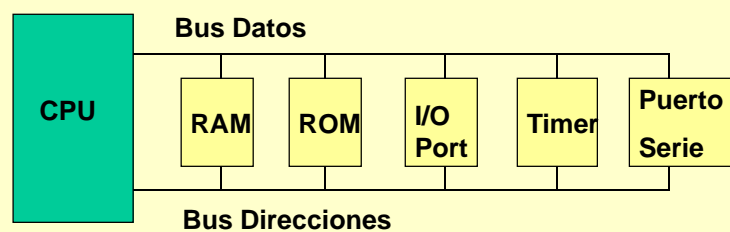
## Microcontrolador



Microcontroladores - Parte 1

13

## Arquitectura



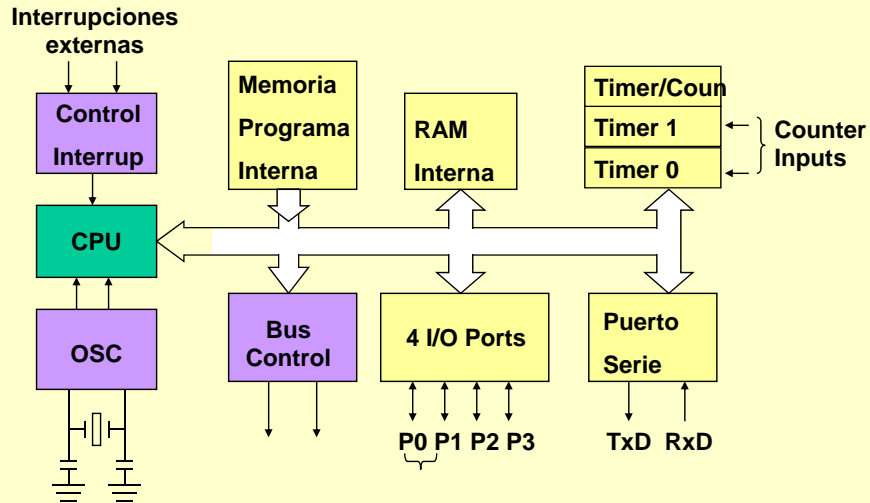
Microcontroladores - Parte 1

Microcontroladores - Parte 1

14

14

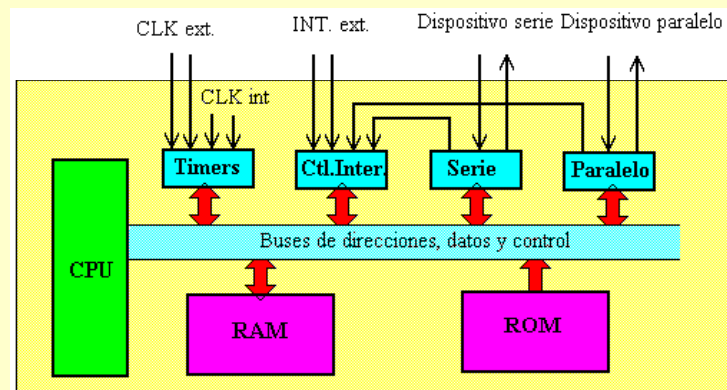
## Diagrama en Bloques



Microcontroladores - Parte 1

15

## Arquitectura

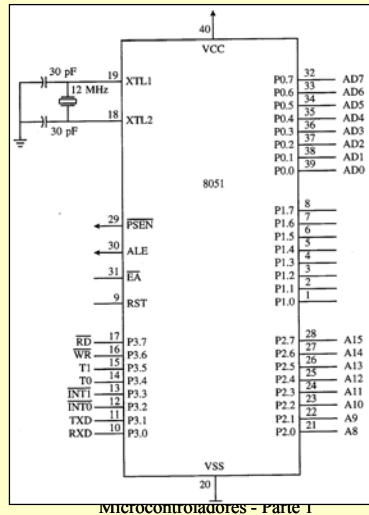


Microcontroladores - Parte 1

16

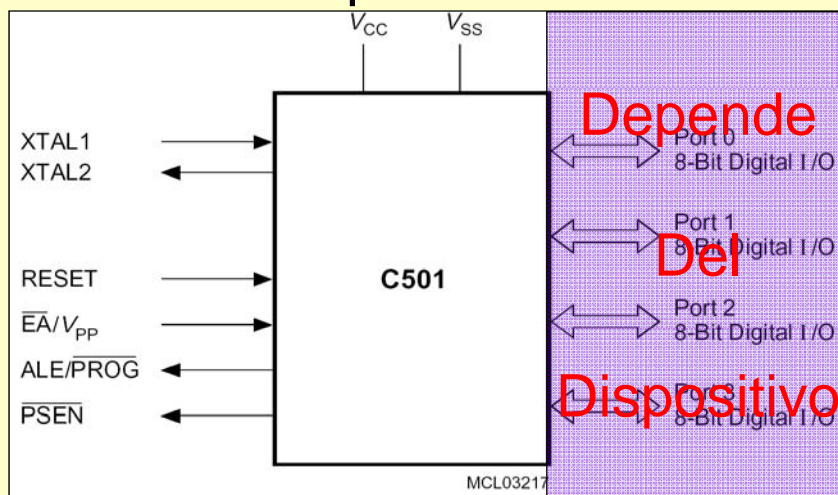


## Conexionado



17

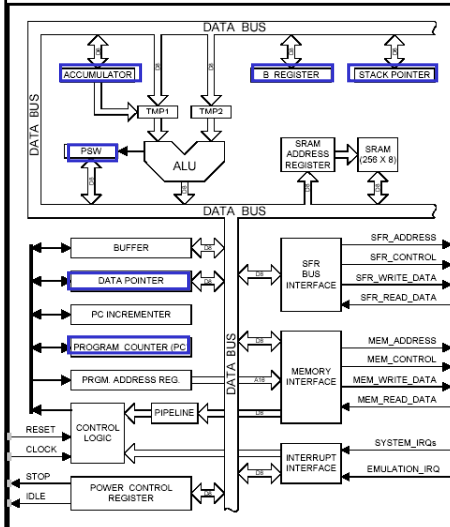
## Arquitectura



Microcontroladores - Parte 1

18

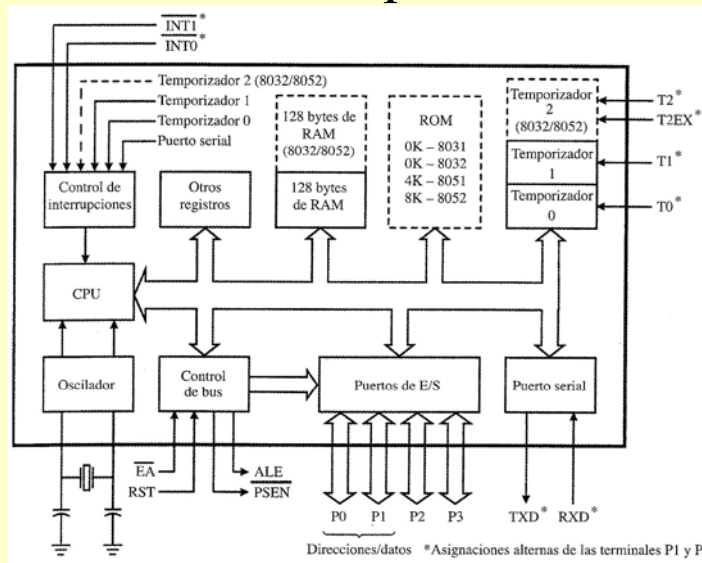
# Arquitectura interna



- A (Acumulador)
- B
- PSW (Program Status Word)
- SP (Stack Pointer)
- PC (Program Counter)
- DPTR (Data Pointer)

Microcontroladores - Parte 1

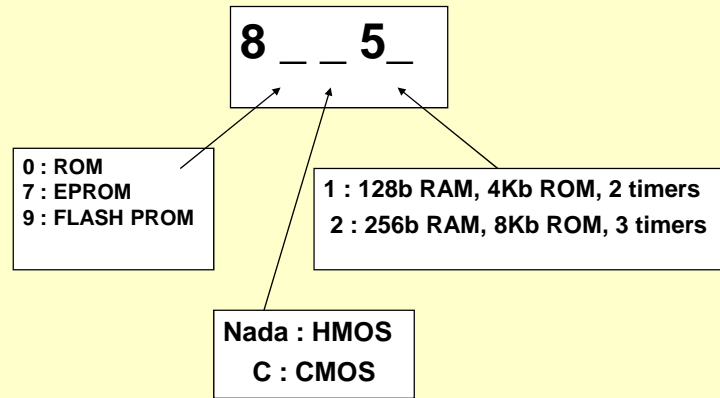
# Bloques



Direcciones/datos \*Asignaciones alternativas de las terminales P1 y P3

Microcontroladores - Parte 1

## Identificación y Nomenclatura



Microcontroladores - Parte 1

21

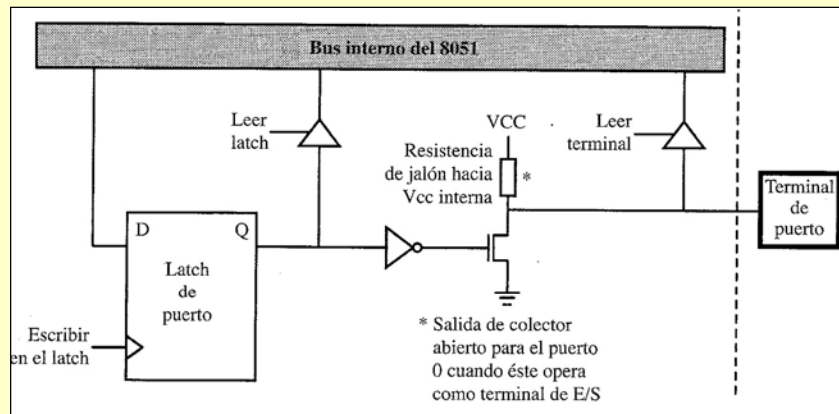
## Contenido del 8051 básico

Puertos de E/S paralelo  
 Generador de Reloj  
 Puerto serie  
 Temporizadores  
 Control de interrupciones

Microcontroladores - Parte 1

22

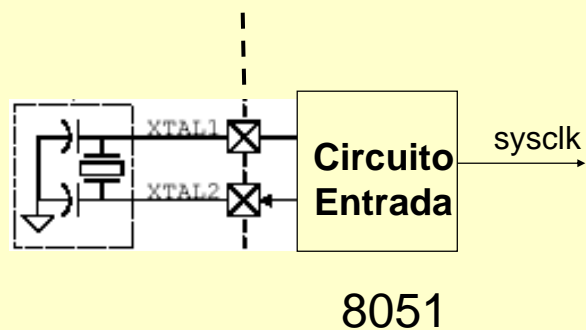
## Esquema de una pata de puerta



Microcontroladores - Parte 1

23

## Circuito de Reloj

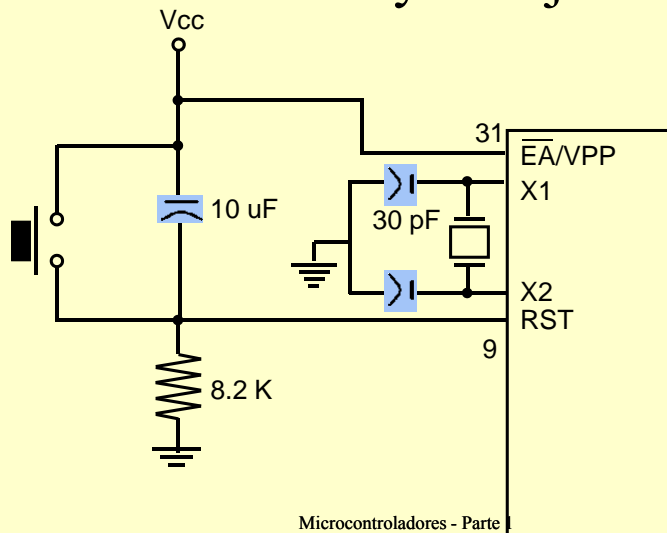


8051 utiliza 12 sysclks por ciclo de máquina

Microcontroladores - Parte 1

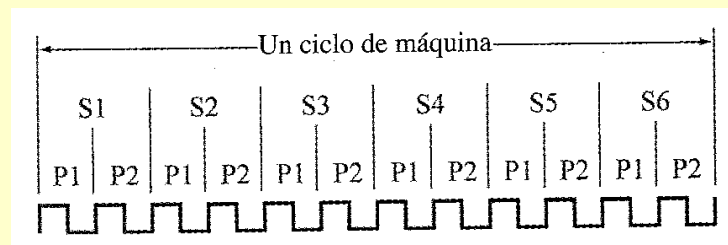
24

## Reset y Reloj



25

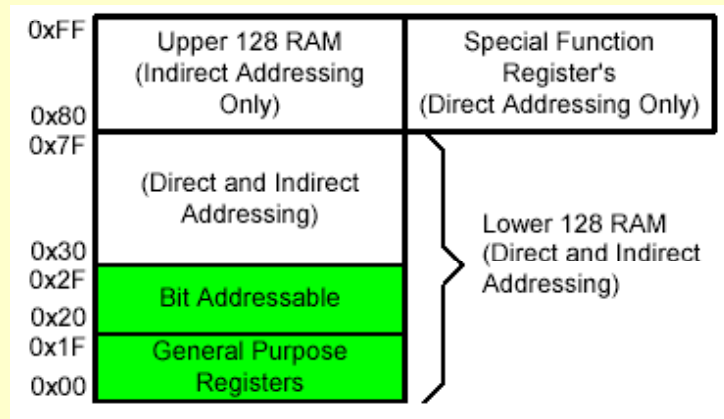
## Ciclo de máquina



Microcontroladores - Parte 1

26

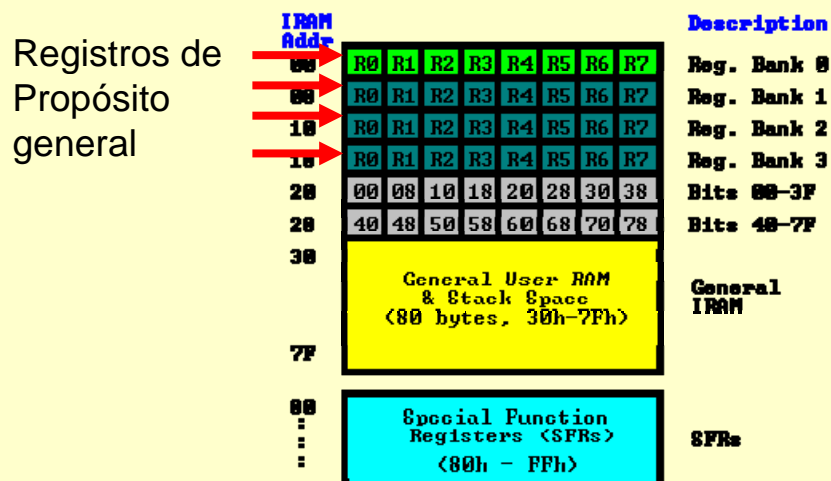
## Memoria De Datos Interna



Microcontroladores - Parte 1

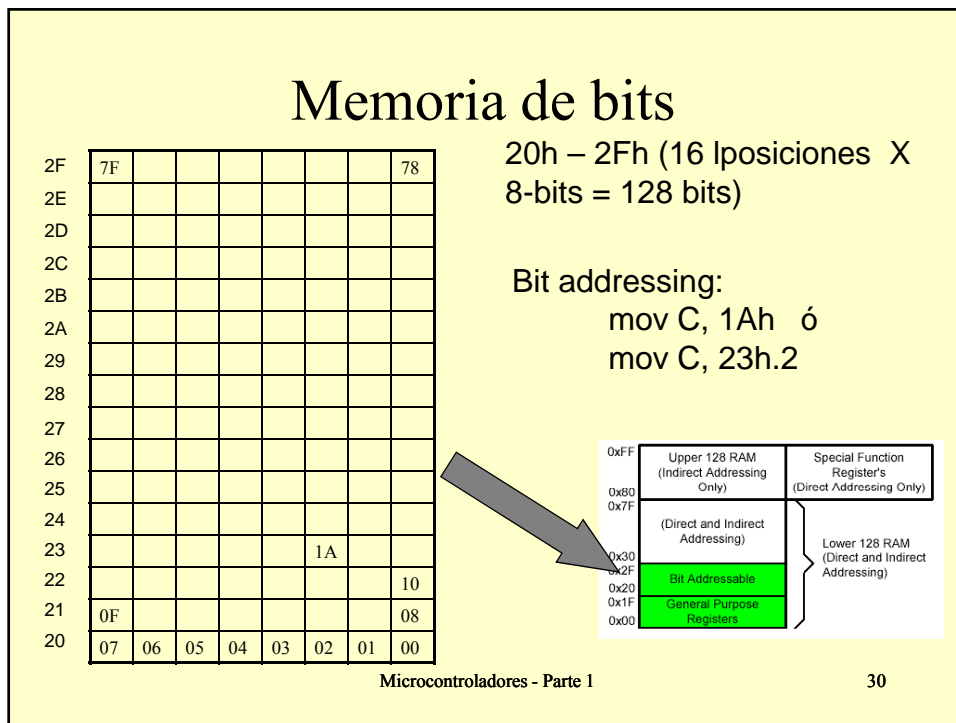
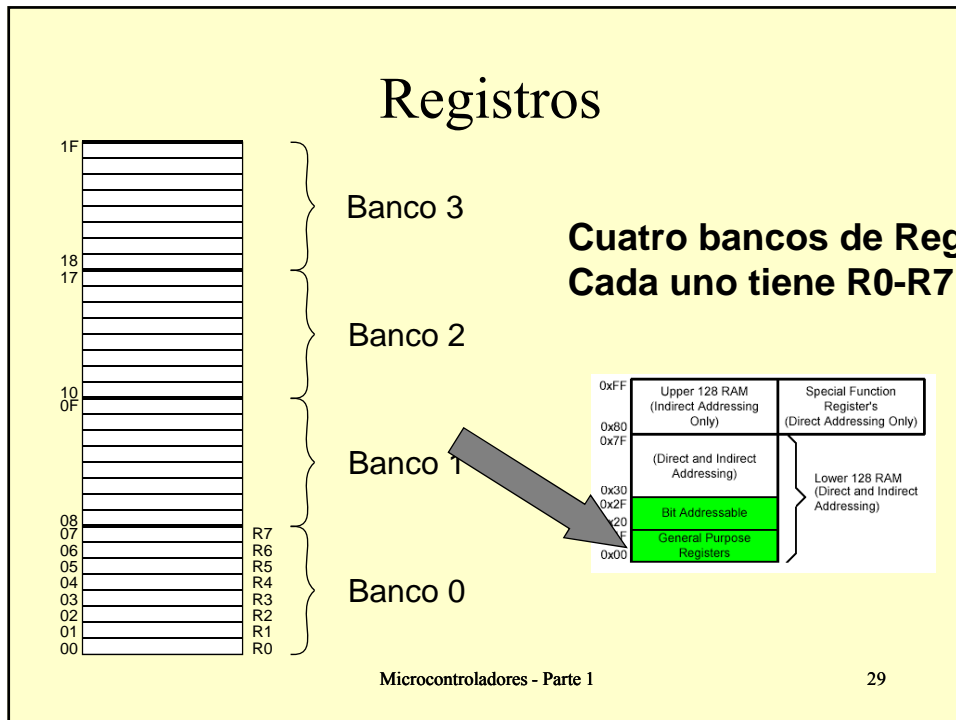
27

## Memoria De datos Interna

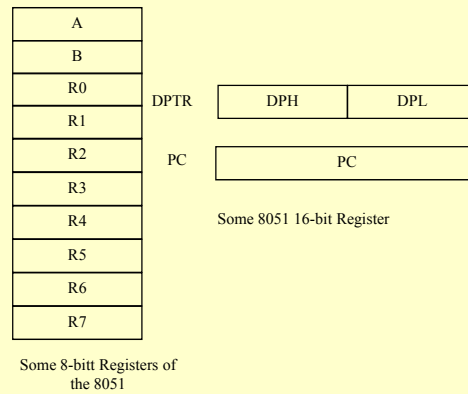


Microcontroladores - Parte 1

28



# Registros

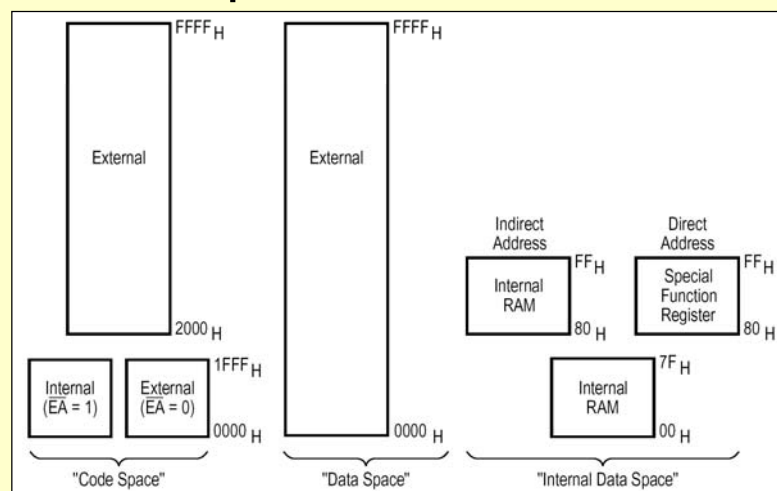


Some 8-bit Registers of the 8051

Microcontroladores - Parte 1

31

# Mapas de memoria



Microcontroladores - Parte 1

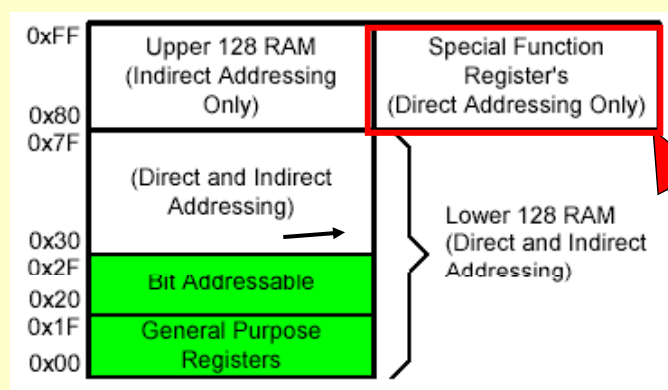
32



## Instrucciones

- **ADD A,R4**
- **ADD A,04h**
- **SETB 24h**
- **MOV 20h,#0FFh** Equivale a **SETB 00h**  
**SETB 01h**  
.....  
**SETB 07h**
- **MOV P0,#01h** Equivale a **SETB 80h**

## SFRs



## SFRs

<u>Dirección</u>	<u>SFR</u>
80H	P0
81H	SP
82H	DPL
83H	DPH
87H	PCON
88H	TCON
89H	TMOD
8AH	TLO
8BH	TL1
8CH	TH0
8DH	TH1

Microcontroladores - Parte 1

35

## SFRs

<u>Dirección</u>	<u>SFR</u>
90H	P1
98H	SCON
99H	SBUF
A0H	P2
A8H	IE
B0H	P3
B8H	IP
D0H	PSW
E0H	ACC

Microcontroladores - Parte 1

36

## SFRs agrupados por función

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	E0 <sub>H</sub> <sup>31</sup>	00 <sub>H</sub>
	B	B-Register	F0 <sub>H</sub> <sup>31</sup>	00 <sub>H</sub>
	DPH	Data Pointer, High Byte	83 <sub>H</sub>	00 <sub>H</sub>
	DPL	Data Pointer, Low Byte	82 <sub>H</sub>	00 <sub>H</sub>
	PSW	Program Status Word Register	D0 <sub>H</sub> <sup>31</sup>	00 <sub>H</sub>
	SP	Stack Pointer	81 <sub>H</sub>	07 <sub>H</sub>
Interrupt System	IE	Interrupt Enable Register	A8 <sub>H</sub> <sup>31</sup>	0X000000B <sup>31</sup>
	IP	Interrupt Priority Register	B8 <sub>H</sub> <sup>31</sup>	0X000000B <sup>31</sup>
Ports	P0	Port 0	80 <sub>H</sub> <sup>31</sup>	FF <sub>H</sub>
	P1	Port 1	90 <sub>H</sub> <sup>31</sup>	FF <sub>H</sub>
	P2	Port 2	A0 <sub>H</sub> <sup>31</sup>	FF <sub>H</sub>
	P3	Port 3	B0 <sub>H</sub> <sup>31</sup>	FF <sub>H</sub>
Serial Channel	PCON <sup>31</sup>	Power Control Register	87 <sub>H</sub>	0XXX0000B <sup>31</sup>
	SBUF	Serial Channel Buffer Register	99 <sub>H</sub>	XX <sub>H</sub> <sup>31</sup>
	SCON	Serial Channel Control Register	98 <sub>H</sub> <sup>31</sup>	00 <sub>H</sub>
Timer 0 / Timer 1	TCON	Timer 0/1 Control Register	88 <sub>H</sub> <sup>31</sup>	00 <sub>H</sub>
	TH0	Timer 0, High Byte	8C <sub>H</sub>	00 <sub>H</sub>
	TH1	Timer 1, High Byte	8D <sub>H</sub>	00 <sub>H</sub>
	TL0	Timer 0, Low Byte	8A <sub>H</sub>	00 <sub>H</sub>
	TL1	Timer 1, Low Byte	8B <sub>H</sub>	00 <sub>H</sub>
	TMOD	Timer Mode Register	89 <sub>H</sub>	00 <sub>H</sub>
Timer 2	T2CON	Timer 2 Control Register	C8 <sub>H</sub> <sup>31</sup>	00 <sub>H</sub>
	T2MOD	Timer 2 Mode Register	C9 <sub>H</sub>	XXXXXXXX0B <sup>31</sup>
	RC2H	Timer 2 Reload/Capture Register, High Byte	CB <sub>H</sub>	00 <sub>H</sub>
	RC2L	Timer 2 Reload/Capture Register, Low Byte	CA <sub>H</sub>	00 <sub>H</sub>
	TH2	Timer 2 High Byte	CD <sub>H</sub>	00 <sub>H</sub>
	TL2	Timer 2 Low Byte	CC <sub>H</sub>	00 <sub>H</sub>
Pow. Sav. Modes	PCON <sup>31</sup>	Power Control Register	87 <sub>H</sub>	0XXX0000B <sup>31</sup>

Microcontroladores - Parte 1

37

## SFRs

80	P0	SP	DPL	DPH				PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1								9F
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
C8									CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	B								F7
F8									FF

Microcontroladores - Parte 1

38

## SFRs

- **P0 (Port 0, Dirección 80h, Bit-Addressable)**
- **SP (Stack Pointer, Dirección 81h)  
Incrementa**
- **DPL/DPH (Data Pointer Low/High,  
Direcciones 82h/83h)**
- **PCON (Power Control, Dirección 87h)**
- **TCON (Timer Control, Dirección 88h, Bit-Addressable)**

## SFRs

- **TMOD (Timer Mode, Dirección 89h)**
- **TL0/TH0 (Timer 0 Low/High, Dirección  
8Ah/8Ch)**
- **TL1/TH1 (Timer 1 Low/High, Dirección  
8Bh/8Dh)**
- **P1 (Port 1, Dirección 90h, Bit-Addressable)**

## SFRs

- **SCON (Serial Control, Dirección 98h, Bit-Addressable)** Se usan SCON, TCON y TMOD
- **SBUF (Serial Control, Dirección 99h)**
- **P2 (Port 2, Dirección A0h, Bit-Addressable)**
- **IE (Interrupt Enable, Dirección A8h)**
- **P3 (Port 3, Dirección B0h, Bit-Addressable)**
- **IP (Interrupt Priority, Direcciones B8h, Bit-Addressable)**

## SFRs

- **PSW (Program Status Word, Direcciones D0h, Bit-Addressable)**
- **ACC (Accumulator, Dirección E0h, Bit-Addressable)**
- **B (B Register, Dirección F0h, Bit-Addressable)**
- **Los propios de cada Microcontrolador**

## SFRs

- **MOV 50h,#01h**
- **MOV 99h,#01h**

## Registros

- Acumulador
- Los Registros R
  - MOV A,R3
  - ADD A,R4
  - MOV R5,A
  - MOV A,R1
  - ADD A,R2
  - SUBB A,R5
- El Registro B
- DPTR
- PC
- SP
- PSW

## PSW

Bit No.	MSB							LSB	
	D7 <sub>H</sub>	D6 <sub>H</sub>	D5 <sub>H</sub>	D4 <sub>H</sub>	D3 <sub>H</sub>	D2 <sub>H</sub>	D1 <sub>H</sub>	D0 <sub>H</sub>	
D0 <sub>H</sub>	CY	AC	F0	RS1	RS0	OV	F1	P	PSW

## PSW

Bit	Function															
CY	Carry Flag Used by arithmetic instruction.															
AC	Auxiliary Carry Flag Used by instructions which execute BCD operations.															
F0	General Purpose Flag															
RS1 RS0	Register Bank select control bits These bits are used to select one of the four register banks.															
	<table border="1"> <thead> <tr> <th>RS1</th> <th>RS0</th> <th>Function</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Bank 0 selected, data address 00<sub>H</sub>-07<sub>H</sub></td> </tr> <tr> <td>0</td> <td>1</td> <td>Bank 1 selected, data address 08<sub>H</sub>-0F<sub>H</sub></td> </tr> <tr> <td>1</td> <td>0</td> <td>Bank 2 selected, data address 10<sub>H</sub>-17<sub>H</sub></td> </tr> <tr> <td>1</td> <td>1</td> <td>Bank 3 selected, data address 18<sub>H</sub>-1F<sub>H</sub></td> </tr> </tbody> </table>	RS1	RS0	Function	0	0	Bank 0 selected, data address 00 <sub>H</sub> -07 <sub>H</sub>	0	1	Bank 1 selected, data address 08 <sub>H</sub> -0F <sub>H</sub>	1	0	Bank 2 selected, data address 10 <sub>H</sub> -17 <sub>H</sub>	1	1	Bank 3 selected, data address 18 <sub>H</sub> -1F <sub>H</sub>
RS1	RS0	Function														
0	0	Bank 0 selected, data address 00 <sub>H</sub> -07 <sub>H</sub>														
0	1	Bank 1 selected, data address 08 <sub>H</sub> -0F <sub>H</sub>														
1	0	Bank 2 selected, data address 10 <sub>H</sub> -17 <sub>H</sub>														
1	1	Bank 3 selected, data address 18 <sub>H</sub> -1F <sub>H</sub>														
OV	Overflow Flag Used by arithmetic instruction.															
F1	General Purpose Flag															
P	Parity Flag Set/cleared by hardware after each instruction to indicate an odd/even number of "one" bits in the accumulator, i.e. even parity.															

## Modos de Direccionamiento

- **Direccionamiento Inmediato:** MOV A,#20h
- **Direccionamiento Directo:** MOV A,30h (SFRs)
- **Direccionamiento Indirecto:** MOV A,@R0
- **Externo Directo:** MOVX A,@DPTR
- **Código Indirecto:** MOVC A,@A+DPTR

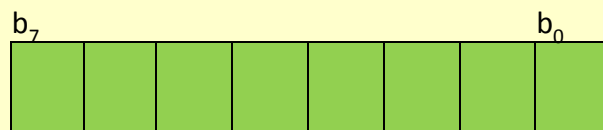
## Modos de direccionamiento

- Implícito
- Inmediato
- Registro
- Directo
- Registro Indirecto
- Indexado



## Implícito

- No requiere operandos adicionales



Código Operación

## Direccionamiento Directo

- Se debe dar la dirección del segundo operando



Código Operación

Dirección Operando

## Inmediato

```
MOV  A,#65H
MOV  A,#'A'
MOV  R6,#65H
MOV  DPTR,#2343H
MOV  P1,#65H
```

Ejemplo :

```
Num      EQU      30
...
MOV      R0,Num
MOV      DPTR,#data1
...
ORG      100H
data1:   db       "pepe"
```

Microcontroladores - Parte 1

51

## Registro

```
MOV      Rn, A      ;n=0,..,7
```

```
ADD      A, Rn
```

```
MOV      DPL, R6
```

```
MOV      DPTR, A
```

```
MOV      Rm, Rn
```

Microcontroladores - Parte 1

52

## Direccionamiento Directo

Puede utilizarse para toda la RAM interna baja (0 – 7F) y los SFRs

```
MOV R0, 40H
MOV 56H, A
MOV A, 4 ; ≡ MOV A, R4
MOV 6, 2 ; copia R2 a R6
; MOV R6,R2 ¡ es inválido !
```

### SFRs

```
MOV 0E0H, #66H ; ≡ MOV A,#66H
MOV 0F0H, R2 ; ≡ MOV B, R2
MOV 80H,A ; ≡ MOV P1,A
```

☒ Bit Addressable  
Page 359,360

## Registro indirecto

- Un registro se usa como puntero a datos.

```
MOV A,@Ri
MOV @R1,B
```

Ejemplo:

Escriba un programa que copie un bloque de 10 bytes de RAM comenzando en la 37H a la memoria que comienza en 59h.

```
MOV R0,37h ; puntero fuente
MOV R1,59h ; puntero destino
MOV R2,10 ; cantidad
L1: MOV A,@R0
MOV @R1,A
INC R0
INC R1
DJNZ R2,L1
```

## Acceso a la memoria de código interna por direccionamiento indexado

- Este modo se emplea para acceder a tablas de búsqueda (look-up tables) en el espacio de programa

```
MOVC    A,@A+DPTR
```

A ← contenido de la dirección en ROM A +DPTR

### Nota:

Como se busca acceder a la memoria de código, se emplea MOVC en lugar de MOV.

- Ejemplo:  
Suponer que se tiene un mensaje de texto "Hola" a partir de la dirección de ROM 250h  
Escriba un programa que lo transfiera a la posición de RAM 40h.

Solucion:

```

ORG      0
MOV      DPTR,#MYDATA
MOV      R0,#40H
L1:      CLR      A
          MOVC    A,@A+DPTR
          JZ      L2
          MOV     @R0,A
          INC     DPTR
          INC     R0
          SJMP   L1
L2:      SJMP   L2
;-----
ORG      250H
MYDATA:  DB      "Hola",0 ; El 0 es fin de mensaje

END
```

- Ejemplo:  
Escriba un programa que lea una variable x de la puerta P1 y escriba  $x^2$  a P2, continuamente .

Solucion:

```

ORG          0
MOV  DPTR, #TAB1
MOV  A,#0FFH
MOV  P1,A

L01:
MOV  A,P1
MOVC A,@A+DPTR
MOV  P2,A
SJMP L01

;-----
ORG  300H
TAB1: DB  0,1,4,9,16,25,36,49,64,81

      END

```

Microcontroladores - Parte 1

57

## Instrucciones que afectan los flags

Instructions	CY	OV	AC
ADD	X	X	X
ADDC	X	X	X
SUBB	X	X	X
MUL	0	X	
DIV	0	X	
DA	X		
RRC	X		
RLC	X		
SETB C	1		
CLR C	0		
ANL C,bit	X		
ANL C,/bit	X		
ORL C,bit	X		
MOV C,bit	X		
CJNE	X		

Note: X can be 0 or 1

Microcontroladores - Parte 1

58

## Distintos tipos de instrucciones

- Movimiento de datos
- Aritméticas
- Lógicas
- Booleanas
- De bifurcación

## Movimiento de datos

**MOV dest,fuente** ; dest = fuente

```

MOV    A,#72H      ;A=72H
MOV    A, #'r'     ;A='r' ó 72H
MOV    R4,#62H     ;R4=62H
MOV    B,0F9H      ;B=contenido de la RAM 0xf9

MOV    DPTR,#7634H
MOV    DPL,#34H
MOV    DPH,#76H

MOV    P1,A        ;mover A al port 1

```

## Movimiento de datos

### Nota 1:

MOV A,#72H ≠ MOV A,72H  
Justificar diferencias

### Nota 2:

MOV A,R3 ≡ MOV A,3

## Aritméticas

**ADDA, fuente** ;A=A+fuentes

ADD A,#6 ;A=A+6

ADD A,R6 ;A=A+R6

ADD A,6 ;A=A+[6] ó A=A+R6

ADD A,0F3H ;A=A+[0F3H]

## Aritméticas

### SUBB A,fuente ;A=A-fuente-CY

SETB C ;CY=1  
SUBB A,R5 ;A=A-R5-1

### ADC A,fuente ;A=A+fuente+CY

SETB C ;CY=1  
ADC A,R5 ;A=A+R5+1

Microcontroladores - Parte 1

63

## Aritméticas

**DEC byte ;byte=byte-1**  
**INC byte ;byte=byte+1**

INC R7  
DEC A  
DEC 40H ; [40]=[40]-1

**CPL A ;Complemento a 1**

Ejemplo:

```
MOV A,#55H ;A=01010101 B
CPL A
MOV P1,A
ACALL DELAY
SJMP L01
```

**NOP & RET & RETI**

Microcontroladores - Parte 1

64



## Lógicas

### ANL - ORL - XRL

Ejemplo:

```
MOV  R5,#89H
ANL  R5,#08H
```

### RR - RL - RRC - RLC A

Ejemplo:

```
RR   A
```

Microcontroladores - Parte 1

65

## Booleanas - Manipulación de bits

```
SETB   bit           ; bit=1
CLR    bit           ; bit=0
```

```
SETB  C           ; CY=1
SETB  P0.0        ; bit 0 del port 0 =1
SETB  P3.7        ; bit 7 del port 3 =1
SETB  ACC.2       ; bit 2 del ACCUMULATOR =1
SETB  05          ; pone en 1 el bit 5 de la RAM 20h
```

### Nota:

CLR opera idéntica a SETB

p.e.:

```
CLR   C           ;CY=0
```

Pero la siguiente instrucción es solo para CLR:

```
CLR   A           ;A=0
```

Microcontroladores - Parte 1

66

## Manipulación de bits

- **Para conmutar cada bit del Puerto 1 existen 3 modos:**
- **Modo 1: Enviar los datos a P1 a través del ACC**

```

– BACK:  MOV  A,#55H ;A=01010101B
          MOV  P1,A
          ACALL DELAY
          MOV  A,#0AAH ;A=10101010B
          MOV  P1,A
          ACALL DELAY
          SJMP BACK
  
```

Microcontroladores - Parte 1

67

## Manipulación de bits

- **Modo 2: Acceder al Puerto 1 directamente**
- ```

BACK:  MOV      P1,#55H ;P1=01010101B
        ACALL   DELAY
        MOV      P1,#0AAH ;P1=10101010B
        ACALL   DELAY
        SJMP    BACK
  
```
- **Modo 3: Read-modsiy-write :**
- ```

          MOV      P1,#55H ;P1=01010101B
AGAIN:  XRL     P1,#0FFH
        ACALL   DELAY
        SJMP    AGAIN
  
```

Microcontroladores - Parte 1

68

## Manipulación de bits

- (a) Observar el estado de P1.2 hasta que pase a alto.
- (b) Cuando P1.2 pase a alto, escriba 45H al puerto 0 y
- (c) Genere un pulso alto a bajo en P2.3.

```
SETB P1.2      ;programo P1.2 como entrada
MOV  A,#45H    ;A=45H
```

AGAIN:

```
JNB  P1.2,AGAIN ;salir cuando P.2=1
MOV  P0,A       ; sacar A a P0
SETB P2.3      ; P2.3 alto
CLR  P2.3      ; Transición H-L en P2.3
```

## Bifurcación

- Escriba un programa que limpie el acumulador y luego sume 10 veces 3 al acumulador
- Solucion1:

```
MOV  A,#0;
MOV  R2,#10
repita : ADD  A,#03
        DJNZ R2,repita ;repetir hasta R2=0 (10 veces)
MOV  R5,A
```

## Otros saltos condicionales (relativos al PC)

JZ	Saltar si A=0
JNZ	Saltar si A/=0
DJNZ	Decrement y Saltar si A/=0
CJNE A,byte	Saltar si A/=byte
CJNE reg,#data	Saltar si byte/=#data
JC	Saltar si CY=1
JNC	Saltar si CY=0
JB	Saltar si bit=1
JNB	Saltar si bit=0
JBC	Saltar si bit=1 and clear bit

Microcontroladores - Parte 1

71

## **SJMP y LJMP:**

### LJMP(Salto largo)

LJMP es un salto incondicionales. Es una instrucción de 3 bytes en la cual el primer byte es el código de operación y el segundo y tercer byte representan la dirección del offset de 16 bits, o sea que el rango del salto es de 0x0 a 0xFFFF.

### SJMP(Salto corto)

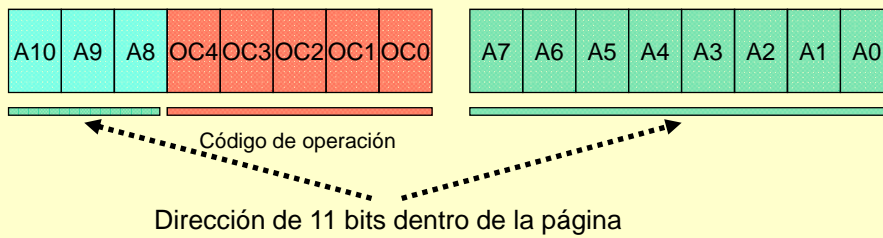
Es una instrucción de 2 bytes. El primer byte es el código de operación y el segundo byte es la dirección relativa al valor actual del PC en la que se halla el destino del salto. Este valor se expresa en complemento a 2, de forma que el destino se deberá encontrar en el entorno de -128 a +127 bytes de la próxima instrucción (a la que apunta el PC).

Microcontroladores - Parte 1

72

## AJMP

- Es un salto dentro de la página de 2 kbytes (11 bits) en la que se halla la próxima instrucción al AJMP



## Salto

- Condicionales

```
JB 45h,PEPE
```

```
NOP
```

```
PEPE:.....
```

- Directos

```
LJMP Por_ahi
```

```
.....
```

```
Por_ahi:
```

- **SJMP** (+127/-128)

- **AJMP** (2K)

## CJNE , JNC

### Ejercicio:

Escriba un programa que compare R0,R1.  
si  $R0 > R1$  escriba 1 al puerto 2,  
si  $R0 < R1$  escriba 0FFh al puerto 2,  
y si no, escriba 0 al puerto 2.

## CALL

Similar a los JMPs anteriores, salvo que se guarda en la pila la dirección de retorno.  
No existe el CALL relativo.

## Subrutinas

- LCALL
- RET

## Tablas de salto

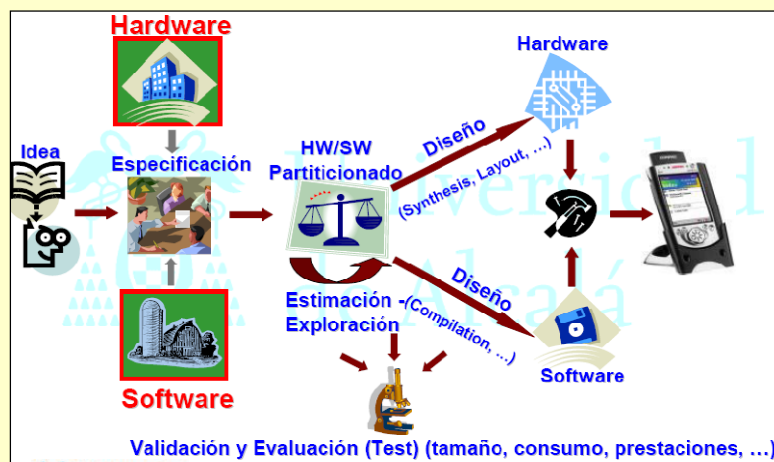
- Se emplean en sistemas secuenciales en los cuales el pasaje de un estado al otro se realiza por medio de una rutina de acción.
- Esas rutinas se asemejan a un case().

## Tablas de salto

MOV DPTR,#TABLA\_DE\_SALTO  
 MOV A,#INDICE  
 RL A ; Cada salto (AJMP) es de 2 bytes  
 JMP @A + DPTR

TABLA\_DE\_SALTO:  
 AJMP SALTO1  
 AJMP SALTO2  
 AJMP SALTO3  
 .....

## Etapas del diseño de un SS.EE.

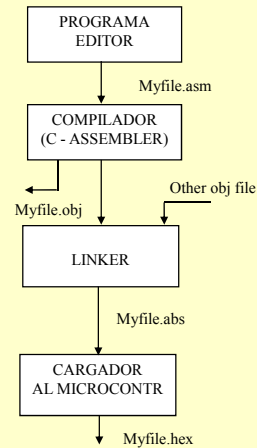




## Como desarrollar un programa

```

ORG      0H
MOV      R5,#25H
MOV      R7,#34H
MOV      A,#0
ADD      A,R5
ADD      A,#12H
ACA:    SJMP ACA
END
  
```



Microcontroladores - Parte 1

81

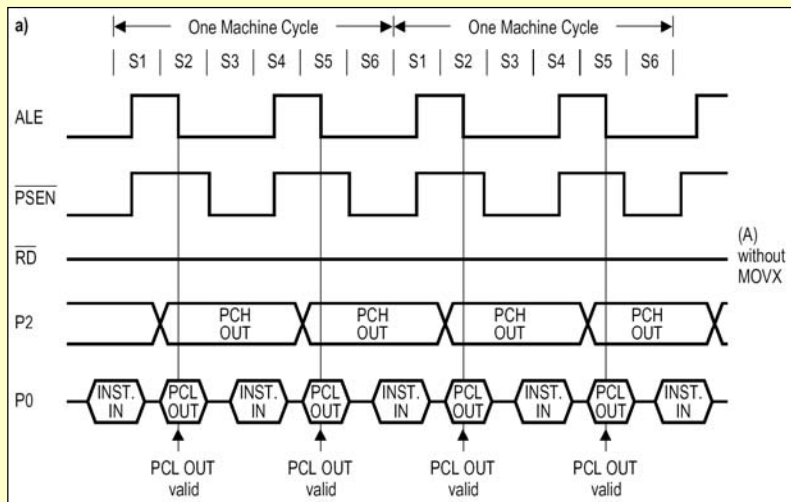
## Herramientas

- **Compilador / Simulador / Depurador**
  - Keil, IAR, Tasking, Raisonance, Crossware, Hi-Tech, SDCC...
- **Emuladores**
  - Signum, Hitex, Ceibo, Phytion, Metalink, Raisonance, Nohau, Brendes, Isystem...
- **Programadores**
  - Advantech, Advin, BP, Ceibo, Data I/O, Dataman, Eetools, Elnec, HiLo, ICE technology, Leap, MQP, Needhams, Phytion, Stag, System General, Xeltek...

Microcontroladores - Parte 1

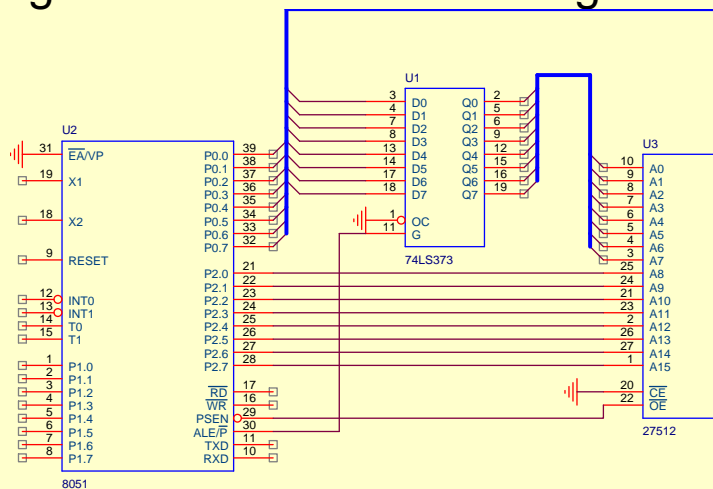
82

### Diagrama Circuital-Mem. Prog. Externa



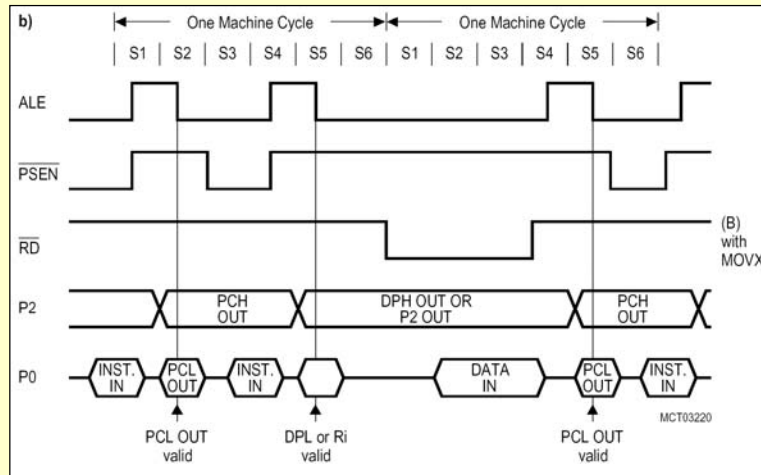
Microcontroladores - Parte 1

### Diagrama Circuital-Mem. Prog. Externa



Microcontroladores - Parte 1

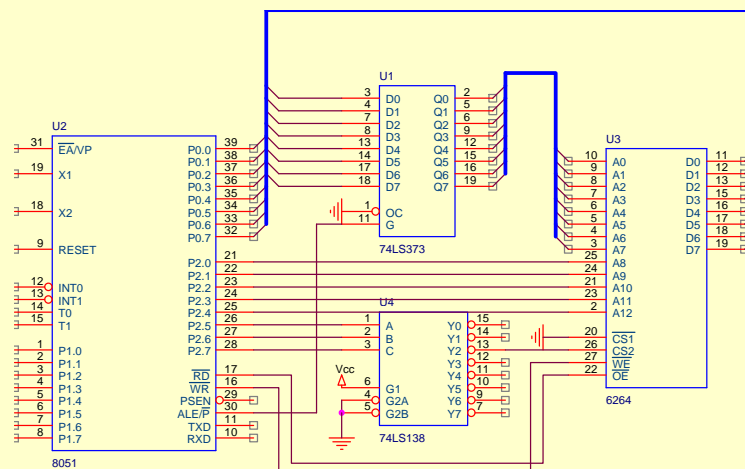
## Diagrama Circuital-Mem. Datos Externa



Microcontroladores - Parte 1

85

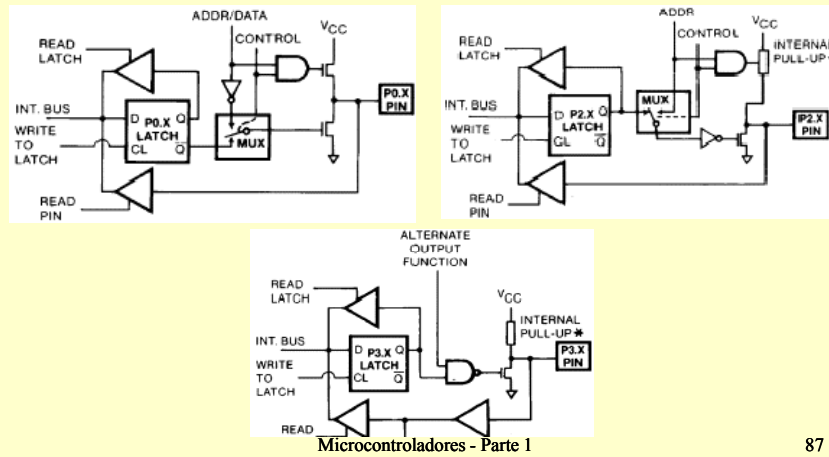
## Diagrama Circuital-Mem. Datos Externa



Microcontroladores - Parte 1

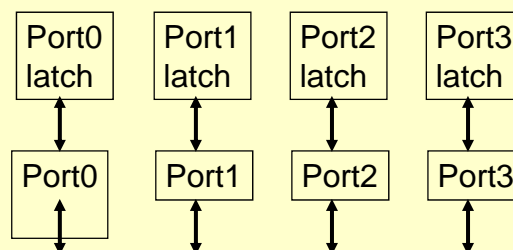
86

## Diagrama de las patas



87

## Puertos de E/S paralelo



Microcontroladores - Parte 1

88

## Puerto Serie

