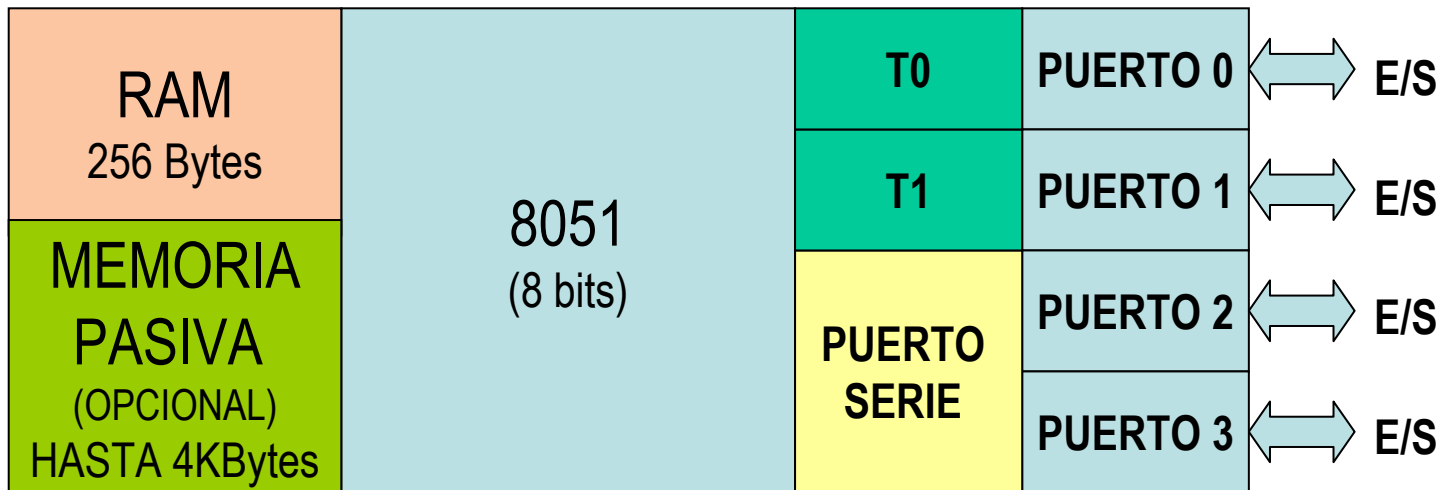
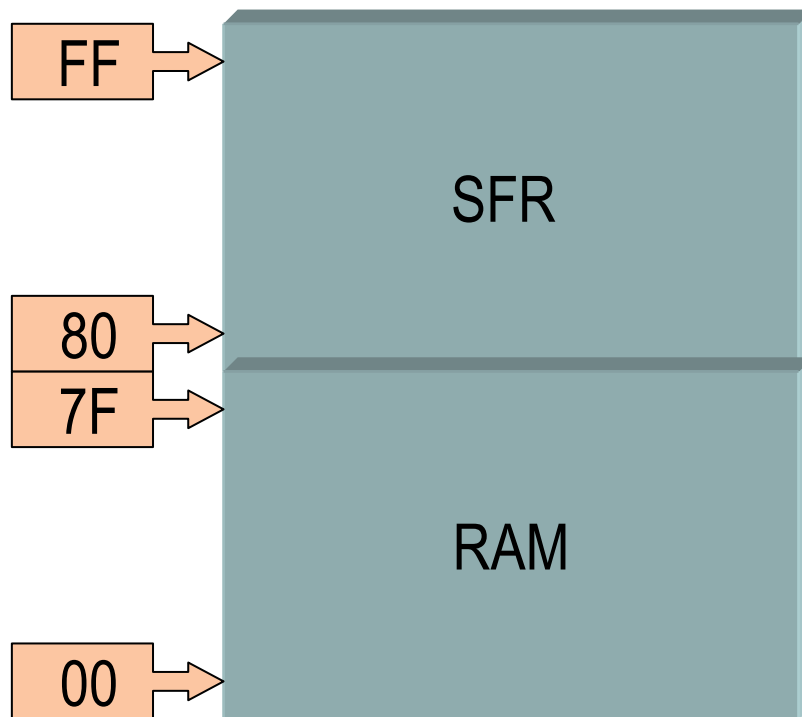


Práctica 1 : INTRODUCCIÓN 8051

- Organización microcontrolador 8051



- IRAM (“Internal RAM”)



- 128 Bytes (parte baja de la memoria):
 - 16 posiciones accesibles bit a bit
 - 4 bancos de 8 registros R0-R7, selección desde PSW (Program Status Word).

BANCO 3	R7	R6	R5	R4	R3	R2	R1	R0	18 _H
BANCO 2	R7	R6	R5	R4	R3	R2	R1	R0	10 _H
BANCO 1	R7	R6	R5	R4	R3	R2	R1	R0	08 _H
BANCO 0	R7	R6	R5	R4	R3	R2	R1	R0	00 _H

127	Scratch Pool Area								7F
48									30H
47	7F	7E	7D	7C	7B	7A	79	78	2FH
46	77	76	75	74	73	72	71	70	2EH
45	6F	6E	6D	6C	6B	6A	69	68	2DH
44	67	66	65	64	63	62	61	60	2CH
43	5F	5E	5D	5C	5B	5A	59	58	2BH
42	57	56	55	54	53	52	51	50	2AH
41	4F	4E	4D	4C	4B	4A	49	48	29H
40	47	46	45	44	43	42	41	40	28H
39	3F	3E	3D	3C	3B	3A	39	38	27H
38	37	36	35	34	33	32	31	30	26H
37	2F	2E	2D	2C	2B	2A	29	28	25H
36	27	26	25	24	23	22	21	20	24H
35	1F	1E	1D	1C	1B	1A	19	18	23H
34	17	16	15	14	13	12	11	10	22H
33	0F	0E	0D	0C	0B	0A	09	08	21H
32	07	06	05	04	03	02	01	00	20H
31	RB 3								1FH
24									RB 2
23	RB 1	17H							
16		RB 0	10H						
15	RB 0		7H						
8		R7	6H						
7	R6	5H							
	R5	4H							
	R4	3H							
	R3	2H							
	R2	1H							
	R1	0H							
0	R0								

- 128 Bytes (parte alta de la memoria):

80	P0	SP	DPL	DPH				PCON	87
88	ICON	IMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
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



SFR DE PUERTOS DE ENTRADA/SALIDA

SFR DE CONTROL

OTROS SFRs

Direccionables bit a bit los situados en las posiciones divisibles por 8

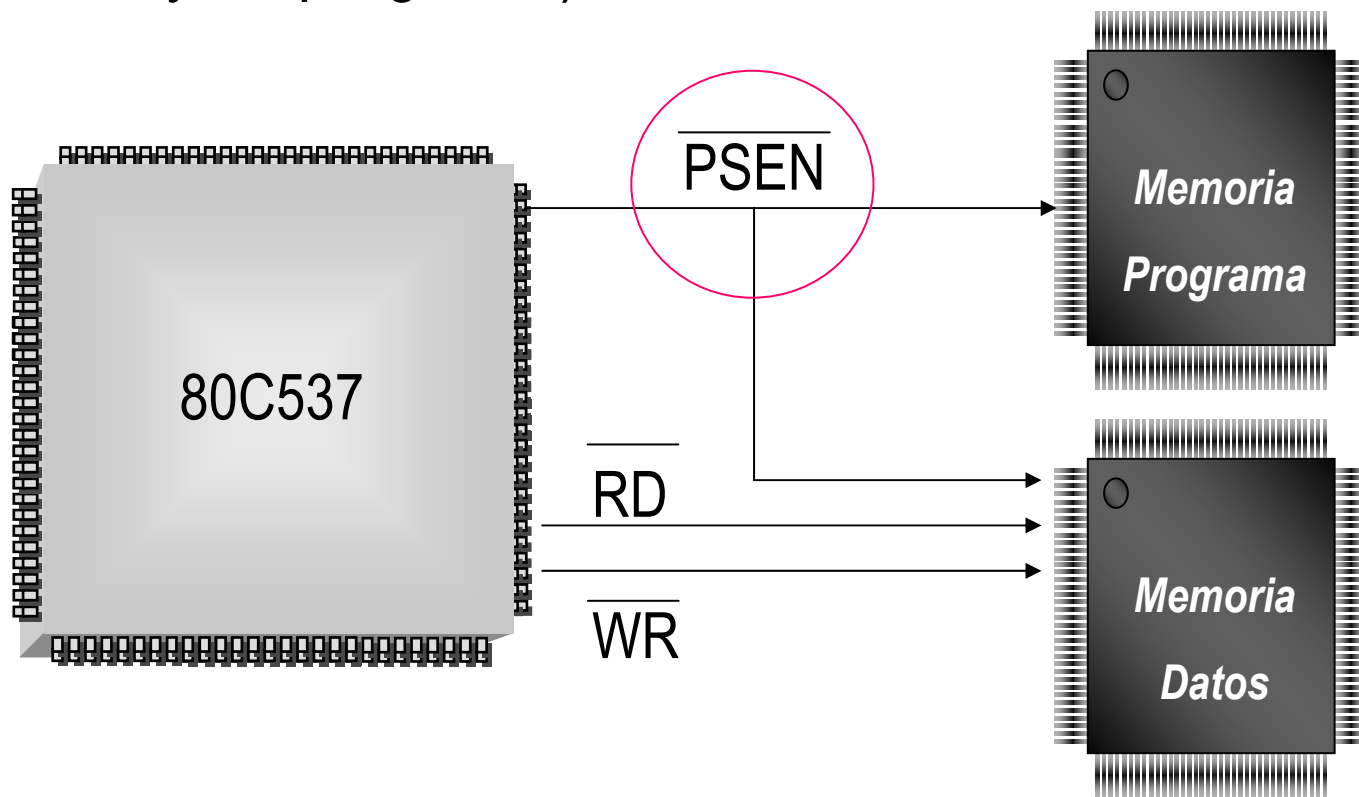
- ACC: acumulador.
- B: registro B, acumulador adicional.
- DPTR: puntero de datos (16 bits: DPH+DPL).
- IE : habilitación de interrupciones.
- IP: prioridad de interrupciones.

- SP (“Stack Pointer”):
 - Tamaño 8 bits. Después de reset apunta a la dirección 07h.
 - Se incrementa antes de PUSH.
 - Se decrementa después de POP.
- PSW (“Program Status Word”):

C	Bit de acarreo		
AC	Bit de acarreo auxiliar para operaciones en BCD		
F0	Flag de propósito general		
RS1 RSO	0	0	Selección del banco de registros 0
	0	1	Selección del banco de registros 1
	1	0	Selección del banco de registros 2
	1	1	Selección del banco de registros 3
OV	Flag de desbordamiento		
F1	Flag de propósito general		
P	Flag de paridad, indica un número par/impar de unos en el acumulador		

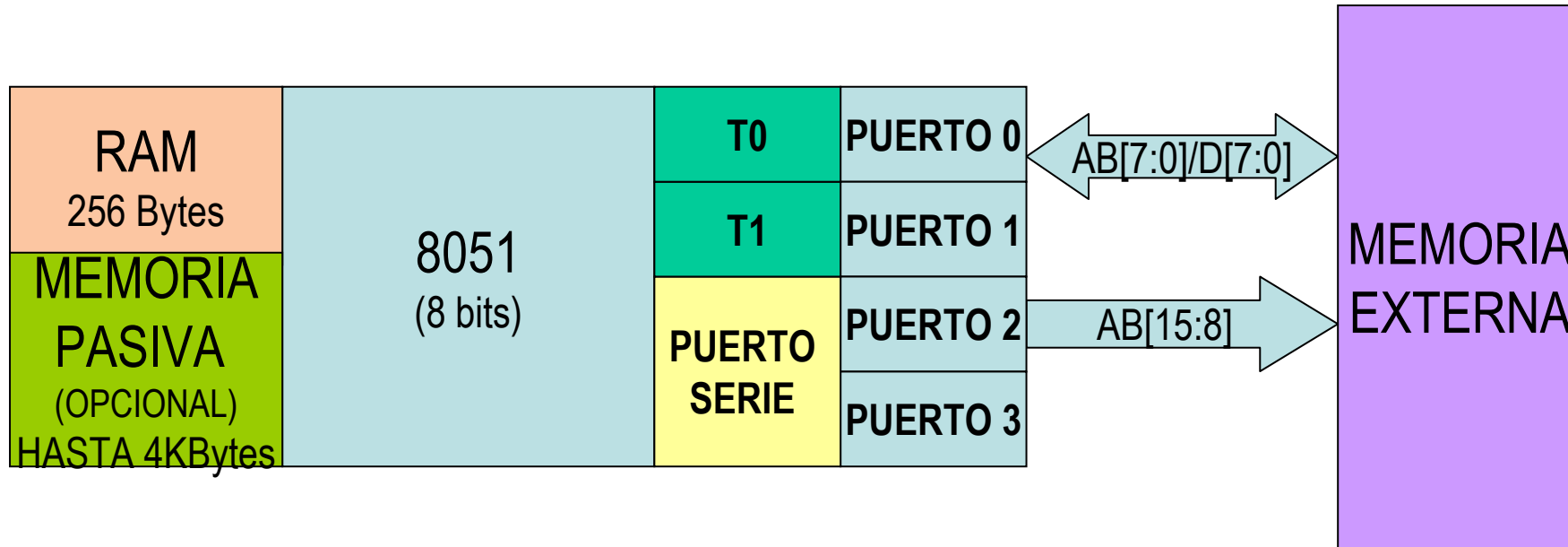
- La memoria pasiva (los datos permanecen al quitar la alimentación) puede ser de tipo ROM, EPROM o FLASH y contiene el programa a ejecutar por el 8051. Puede tener hasta 4 Kbytes de memoria de código interna.
- Es posible utilizar combinaciones de memoria interna y externa. Un ejemplo típico es el que utiliza los 4Kbytes de ROM en el chip y el resto hasta completar como máximo los 64Kbytes en chips de memoria EPROM.

- Arquitectura Harvard (distinción entre memoria externa de datos y de programa):



PSEN ("Program Store Enable")

- Bus de datos y direcciones



- Direcccionamiento directo:
 - La dirección del operando se especificada en el código de instrucción.
 - Tamaño del campo de dirección: 8 bits → Espacio de memoria direccionable: 256 bytes.
 - *Ejemplo: ADD A,30h*
- Direcccionamiento inmediato:
 - EL operando se indica mediante una constante
 - *Ejemplo: ADD A,#10*
- Direcccionamiento por registro:
 - El operando es el contenido de un registro que se especifica en el código de instrucción.
 - *Ejemplo: ADD A,R0*

- Direccionamiento indirecto:
 - El código de instrucción especifica un registro que contiene la dirección del operando:
 - R0,R1 (del banco seleccionado): 8 bits
 - DPTR: 16 bits.
 - *Ejemplo: ADD A,@R0*
- Direccionamiento indexado:
 - El código de instrucción se obtiene de la suma de un registro base (DPTR/PC) y un registro índice (ACC).
 - Facilita el acceso a tablas de datos.
 - *Ejemplo: MOVC A,@ A + DPTR*

- Instrucciones y modos de direccionamiento orientadas al bit que permiten realizar controladores lógicos.
- Instrucciones aritméticas en BCD.
- Acceso a memoria externa a través de punteros para facilitar manejo de tablas y reducir el tamaño de programa.

Modos direccionamiento					
Mnemónico	Resultado de la instrucción	DIR	IND	REG	INM
MOV A, src	(A) = (src)	X	X	X	X
MOV des, A	(des) = (A)	X	X	X	
MOV Rn, scr	(Rn) = (src)	X			X
MOV direc,src	(direc) = (src)	X	X	X	X
MOV @Ri,src	((Ri)) = (src)	X			X
MOV DPTR, #dato16	(DPTR) = constante 16 bits.				X
PUSH src	INC SP; MOV @SP, src	X			
POP des	MOV des,@SP; DEC SP	X			
XCH A,des	intercambia contenido de A y des	X	X	X	
XCHD A, @Ri	intercambia los 4 bits de menor peso del contenido de A y el apuntado por Ri		X		

() indica contenido

Ri=R0 o R1 del banco de registros seleccionado.

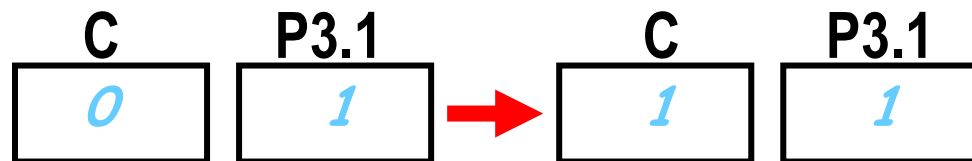
DIR: direc. Directo; IND: direc. Indirecto; REG: direc. Registro; INM: direc. Inmediato

- Ejemplos:

```
MOV A,R0  
MOV @R1,A  
MOV A,#3  
MOV 25H,26  
MOV DPTR,#2F7AH
```



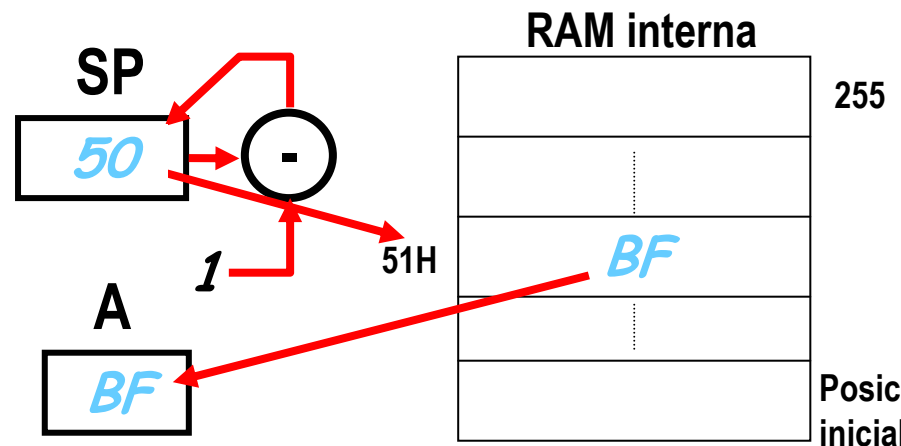
```
MOV C,P3.1
```



- Ejemplo:

PUSH ACC

POP ACC

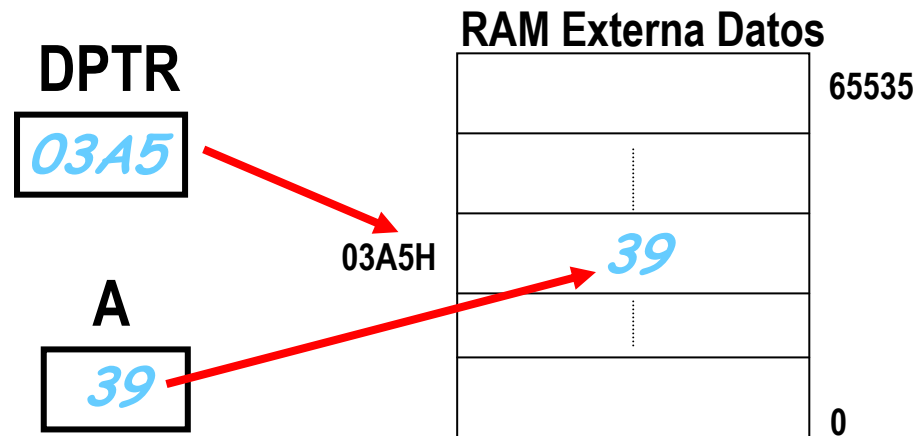


Mnemónico	Resultado	Espacio direcciones
MOVX A, @Ri	$(A) = ((Ri))$	8 bits
MOVX @Ri, A	$((Ri)) = (A)$	8 bits
MOVX A, @DPTR	$(A) = ((DPTR))$	16 bits
MOVX @DPTR, A	$((DPTR)) = (A)$	16 bits
Ri=R0 o R1 del banco de registros seleccionado.		

Ejemplo:

MOVX A,@R0

MOVX @DPTR,A

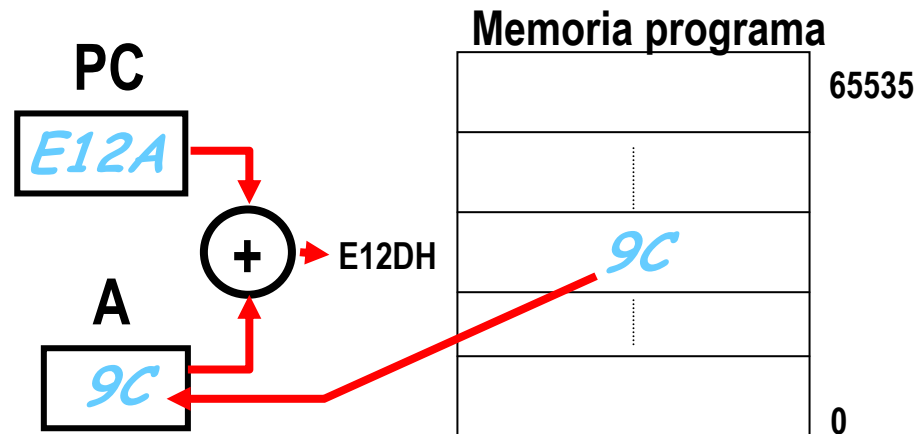


Mnemónico:	Resultado instrucción:
MOVC A, @A + DPTR	$(A) = ((A) + (DPTR))$
MOVC A, @A + PC	$(A) = ((A) + (PC))$
PC: contador de programa	

Ejemplo:

MOVC A,@A+DPTR

MOVC A,@A+PC



Mnemónico:	Resultado de la instrucción:
CLR A	pone a cero el acumulador
CPL A	complementa el acumulador (cambia 0 por 1 y viceversa)
RL A	rota 1 bit a la izquierda el acumulador de forma circular (el bit 7 pasa a ser el bit 0)
RLC A	rota 1 bit a la izquierda el acumulador de forma circular a través del bit acarreo (el bit 7 pasa a ser el acarreo y el acarreo pasa a ser el bit 0)
RR A	rota 1 bit a la derecha el acumulador de forma circular (el bit 0 pasa a ser el bit 7)
RRC A	rota 1 bit a la derecha el acumulador de forma circular a través del bit acarreo (el bit 0 pasa a ser el acarreo y el acarreo pasa a ser el bit 7).
SWAP A	intercambia los “nibbles” (4 bits) bajo y alto del acumulador. Es igual a una rotación de 4 bits (RL o RR)

Modos de direccionamiento:					
Mnemónico:	Resultado :	DIR	IND	REG	INM
ANL A, op	(A)=(A) AND (op)	x	x	x	x
ORL A, op	(A)=(A) OR (op)	x	x	x	x
XRL A, op	(A)= (A) $\oplus$ (op)	x	x	x	x
DIR: direc. Directo; IND: direc. Indirecto; REG: direc. Registro; INM: direc. Inmediato					

Mnemónico:	Resultado de la instrucción:	Operando:
CLR bit	pone a cero el bit	C o cualquier bit
SETB bit	pone a uno el bit	C o cualquier bit
CPL bit	complementa el bit	C o cualquier bit
C: bit de acarreo		

Mnemónico:	Resultado:	bit
ANL C, bit	$(C) = (C) \text{ AND } (\text{bit})$	cualquier bit
ORL C, bit	$(C) = (C) \text{ OR } (\text{bit})$	cualquier bit
MOV C, bit	$(C) = (\text{bit})$	cualquier bit
MOV bit, C	$(\text{bit}) = (C)$	cualquier bit

Mnemónico:	Condición:	Resultado
JB bit , rel	si bit = 1	salta a la dirección(PC) + rel
JBC bit , rel	si bit = 1	salta a la dirección(PC) + rel y desactiva el bit
JNB bit , rel	si bit = 0	salta a la dirección(PC) + rel
JC rel	si C = 1	salta a la dirección(PC) + rel
JNC rel	si C = 0	salta a la dirección(PC) + rel
"rel" es un entero de 8 bits con signo (-128 a 127)		

Mnemónico	Resultado	A	DIR	IND	REG	INM
ADD A, byte	suma aritmética de byte con el acumulador guardando el resultado en este último. Los bits C y OV se ven afectados. Si se suman enteros sin signo el desbordamiento se indica con C=1 mientras que si se suman enteros con signo se indica mediante OV=1		X	X	X	X
ADDC A, byte	realiza la suma aritmética del operando2 con el acumulador y con el acarreo (C) guardando el resultado en el acumulador. C y OV se ven afectadas de igual forma que en ADD		X	X	X	X
DA A	corrige el resultado de una suma almacenado en el acumulador para ponerlo en BCD, añadiendo 0110 si el resultado del dígito BCD es mayor que 9					
DEC byte	decrementa en 1 el byte indicado. Si decrementamos 00H obtendremos FFH	X	X	X	X	X
DIV AB	divide el acumulador (A) entre el registro B (ambos enteros sin signo). La parte entera del cociente se guarda en A y el resto entero se guarda en B. Una división por cero pone OV=1					
INC byte	incrementa en 1 el byte indicado. Si incrementamos FFH obtendremos 00H	X	X	X	X	X
INC DPTR	incrementa en 1 el puntero de datos (16 bits)					
MUL AB	multiplica el acumulador y el registro B (ambos enteros sin signo). El byte bajo del resultado se guarda en A y el byte alto en B					
SUBB A, byte	resta el byte y el acarreo (C) del acumulador y guarda el resultado en este último. El acarreo de la resta (C) se pone a 1 cuando el resultado es negativo. Esto permite hacer restas de mayor precisión (16 bits, etc.). El desbordamiento se indica mediante OV=1			X	X	X

DIR: direc. Directo; IND: direc. Indirecto; REG: direc. Registro; INM: direc. Inmediato

Mnemónico	Dirección de salto	Operaciones adicionales
SJMP rel	$(PC) = (PC) + rel$	
LJMP addr16	$(PC) = addr16$	
AJMP addr11	$(PC) = addr11$	
JMP @A + DPTR	$(PC) = (A) + (DPTR)$	
LCALL addr16	$(PC) = addr16$	Guarda PC en la pila (*)
ACALL addr11	$(PC) = addr11$	Guarda PC en la pila (*)
RET	$(PC) = ((SP))$	Repone el PC de la pila
RETI	$(PC) = ((SP))$	Repone el PC de la pila

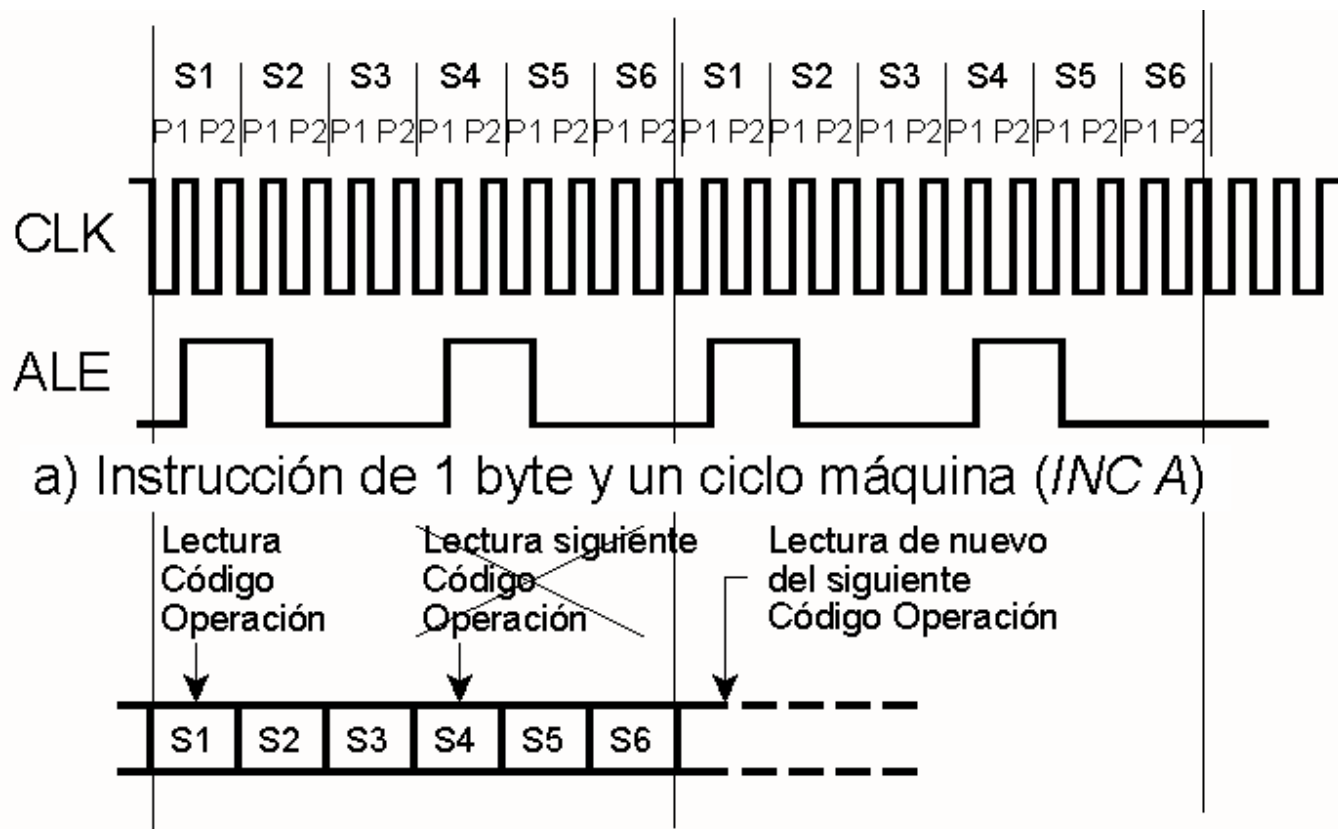
(*) $-128 \leq rel \leq +127$

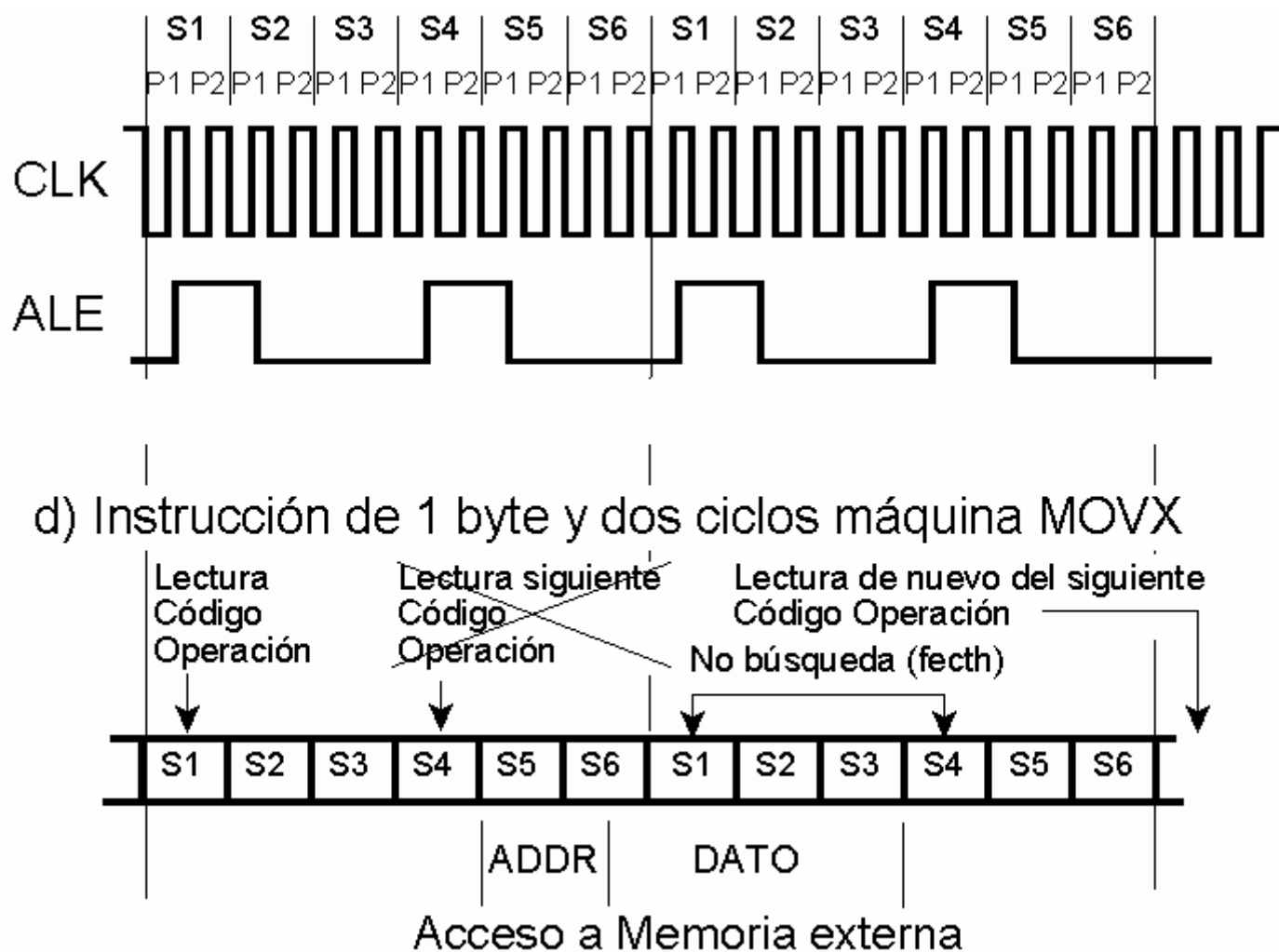
(**) Primero se almacena el byte de menor peso y después el de mayor peso

Mnemónico	Condición	Dirección de salto	DIR	IND	REG	INM
JZ rel	Salto si (A) = 0	(PC) = (PC) + rel				
JNZ rel	Salto si (A) $\neq$ 0	(PC) = (PC) + rel				
DJNZ byte,rel	Decrementa (byte) y salta si (byte) $\neq$ 0	(PC) = (PC) + rel	X		X	
CJNE A,byte,rel	Salto si (A) $\neq$ (byte)	(PC) = (PC) + rel	X			X
CJNE byte,#dato,rel	Salto si (byte) $\neq$ dato	(PC) = (PC) + rel		X	X	

DIR: direc. Directo; IND: direc. Indirecto; REG: direc. Registro; INM: direc. Inmediato
 $-128 \leq \text{rel} \leq +127$

- Ciclo máquina: 12 ciclos del oscilador (12MHz $\rightarrow$ 1 μ s)
- Las instrucciones se ejecutan generalmente en 1 o 2 ciclos máquina en función del número de bytes del código de la instrucción (1/2/3) y del tiempo de ejecución.





Nemónico	Descripción	Código	Bytes	Ciclos
ADD A,Rn	Suma registro al acumulador	0010 1rrr	1	1
ADD A,directo	Suma un byte al acumulador	0010 0101	2	1
ADD A,@Ri	Suma indirecto al acumulador	0010 011i	1	1
ADD A,#dato	Suma dato inmediato al acumulador	0010 0100	2	1
ADDC A,Rn	Suma registro y acarreo al acumulador	0011 1rrr	1	1
ADDC A,directo	Suma byte y acarreo al acumulador	0011 0101	2	1
ADDC A,@Ri	Suma indirecto y acarreo al acumulador	0011 011i	1	1
ADDC A,#dato	Suma inmediato y acarreo al acumulador	0011 0100	2	1
SUBB A,Rn	Sustraer registro y acarreo del acumulador	1001 1rrr	1	1
SUBB A,directo	Sustraer byte y acarreo del acumulador	1001 0101	2	1
SUBB A,@Ri	Sustraer indirecto y acarreo del acumulador	1001 011i	1	1
SUBB A,#dato	Sustraer inmediato y acarreo del acumulador	1001 0100	2	1
INC A	Incrementa acumulador	0000 0100	1	1
INC Rn	Incrementa registro	0000 1rrr	1	1
INC directo	Incrementa una posición de RAM	0000 0101	2	1
INC @Ri	Incrementa una posición indirecta	0000 011i	1	1
DEC A	Decrementa acumulador	0001 0100	1	1
DEC Rn	Decrementa registro	0001 1rrr	1	1
DEC directo	Decrementa una posición de RAM	0001 0101	2	1
DEC @Ri	Decrementa una posición indirecta	0001 011i	1	1
INC DPTR	Incrementa el puntero de datos	1010 0011	1	2
MUL AB	Multiplica A por B	1010 0100	1	4
DIV AB	Divide A por B	1000 0100	1	4
DA A	Ajuste decimal del acumulador	1101 0100	1	1

Operaciones Lógicas

Mnemónico	Descripción	Código	Bytes	Ciclos
ANL A,Rn	AND de un registro y el acumulador	0101 1r r r	1	1
ANL A,directo	AND de un byte y el acumulador	0101 0101	2	1
ANL A,@Ri	AND de un indirecto y el acumulador	0101 011i	1	1
ANL A,#dato	AND de un inmediato y el acumulador	0101 0100	2	1
ANL directo,A	AND del acumulador y un byte	0101 0010	2	1
ANL directo,#dato	AND de un valor inmediato y un byte	0101 0011	3	2
ORL A, Rn	OR de un registro y el acumulador	0100 1r r r	1	1
ORL A,directo	OR de un byte y el acumulador	0100 0101	2	1
ORL A,@Ri	OR de un indirecto y el acumulador	0100 011i	1	1
ORL A,#dato	OR de un inmediato y el acumulador	0100 0100	2	1
ORL directo,A	OR del acumulador y un byte	0100 0010	2	1
ORL directo,#dato	OR de un valor inmediato y un byte	0100 0011	3	2
XRL A, Rn	EOR de un registro y el acumulador	0110 1r r r	1	1
XRL A,directo	EOR de un byte y el acumulador	0110 0101	2	1
XRL A,@Ri	EOR de un indirecto y el acumulador	0110 011i	1	1
XRL A,#dato	EOR de un inmediato y el acumulador	0110 0100	2	1
XRL directo,A	EOR del acumulador y un byte	0110 0010	2	1
XRL directo,#dato	EOR de un valor inmediato y un byte	0110 0011	3	2
CLR A	Limpia el acumulador	1110 0100	1	1
CLR bit	Limpia el bit direccionado	1100 0010	2	1
CLR C	Limpia el bit de acarreo	1100 0011	1	1
CPL A	Complementa el acumulador	1111 0100	1	1
RL A	Rota acumulador a la izquierda	0010 0011	1	1
RLC A	Rota acumulador a la izquierda con acarreo	0011 0011	1	1
RR A	Rota acumulador a la derecha	0000 0011	1	1
RRC A	Rota acumulador a la derecha con acarreo	0001 0011	1	1
SWAP A	Intercambia nibbles en el acumulador	1100 0100	1	1

Transferencia de Datos

Mnemónico	Descripción	Código	Bytes	Ciclos
MOV A,Rn	Mueve registro al acumulador	1110 1r r r	1	1
MOV A,directo	Mueve directo al acumulador	1110 0101	2	1
MOV A,@Ri	Mueve indirecto al acumulador	1110 011i	1	1
MOV A,#dato	Mueve inmediato al acumulador	0111 0100	2	1
MOV Rn,A	Mueve acumulador al registro	1111 1r r r	1	1
MOV Rn,directo	Mueve directo al registro	1010 1r r r	2	2
MOV Rn,#dato	Mueve inmediato al registro	0111 1r r r	2	1
MOV directo,A	Mueve el acumulador a directo	1111 0101	2	1
MOV directo,Rn	Mueve registro a directo	1000 1r r r	2	2
MOV directo,directo	Mueve directo a directo	1000 0101	3	2
MOV directo,@Ri	Mueve indirecto a directo	1000 011i	2	2
MOV directo,#dato	Mueve dato a directo	0111 0101	3	2
MOV @Ri,A	Mueve acumulador a indirecto	1111 011i	1	1
MOV @Ri,directo	Mueve directo a indirecto	1010 011i	2	2
MOV @Ri,#dato	Mueve inmediato a indirecto	0111 011i	2	1
MOV DPTR,#dato16	Mueve dato16 al puntero de datos	1001 0000	3	2
MOVC A,@A+DPTR	Mueve código relativo a DPTR y A	1001 0011	1	2
MOVC A,@A+PC	Mueve código relativo a PC y A	1000 0011	1	2
MOVX A,@Ri	Mueve RAM externa a A	1110 001i	1	2
MOVX A,@DPTR	Mueve RAM externa a A	1110 0000	1	2
MOVX @Ri,A	Mueve A a RAM externa	1111 001i	1	2
MOVX @DPTR,A	Mueve A a RAM externa	1111 0000	1	2
PUSH directo	Guarda un byte en la pila	1100 0000	2	2
POP directo	Lee un byte de la pila	1101 0000	2	2
XCH A,Rn	Intercambia registro con el acumulador	1100 1r r r	1	1
XCH A,directo	Intercambia directo con el acumulador	1100 0101	2	1
XCH A,@Ri	Intercambia indirecto con el acumulador	1100 011i	1	1
XCHD A,@Ri	Intercambia nibbles inferiores	1101 011i	1	1

Manipulación de variables booleanas

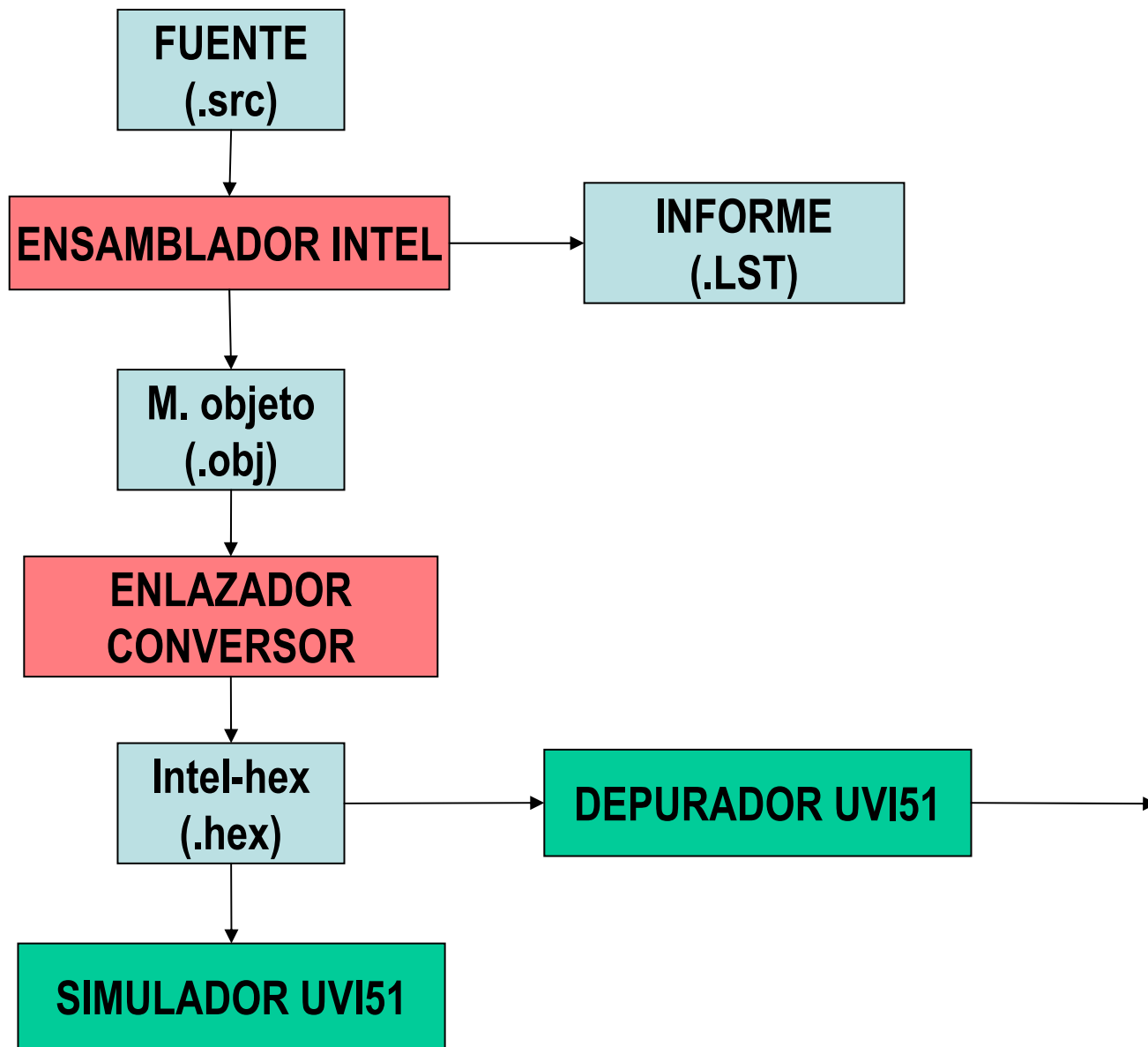
Mnemónico	Descripción	Código	Bytes	Ciclos
CLR C	Desactiva el bit de acarreo	1100 0011	1	1
CLR bit	Desactiva un bit directo	1100 0010	2	1
SETB C	Activa el bit de acarreo	1101 0011	1	1
SETB bit	Activa un bit directo	1101 0010	2	1
CPL C	Complementa el bit de acarreo	1011 0011	1	1
CPL bit	Complementa un bit directo	1011 0010	2	1
ANL C,bit	AND del bit directo y C	1000 0010	2	2
ANL C,/bit	AND del complemento del bit directo y C	1011 0000	2	2
ORL C,bit	OR del bit directo y C	0111 0010	2	2
ORL C,/bit	OR del complementario del bit directo y C	1010 0000	2	2
MOV C,bit	Mueve bit directo a C	1010 0010	2	1
MOV bit,C	Mueve C a bit directo	1001 0010	2	2

Instrucciones de salto

Mnemónico	Descripción	Código	Bytes	Ciclos
ACALL addr11	Salto absoluto a un subprograma	a a a1 0001	2	2
LCALL addr16	Salto absoluto largo a un subprograma	0001 0010	3	2
RET	Retorna de una subrutina	0010 0010	1	2
RETI	Retorna de una interrupción	0011 0010	1	2
AJMP addr11	Salto absoluto	a a a0 0001	2	2
LJMP addr16	Salto absoluto largo	0000 0010	3	2
SJMO rel	Salto corto relativo	1000 0000	2	2
JMP @A+DPTR	Salto relativo a DPTR	0111 0011	1	2
JZ rel	Salta si acumulador es cero	0110 0000	2	2
JNZ rel	Salta si acumulador no es cero	0111 0000	2	2
JC rel	Salta si acarreo activo	0100 0000	2	2
JNC rel	Salta si acarreo a cero	0101 0000	2	2
JB bit,rel	Salta si bit está activo	0010 0000	3	2
JNB bit,rel	Salta si el bit está inactivo	0011 0000	3	2
JBC bit,rel	JB bit,rel + CLR bit	0001 0000	3	2
CJNE A,directo,rel	Compara directo con A y salta si distinto	1011 0101	3	2
CJNE A,#dato,rel	Compara inmediato con A y salta si distinto	1011 0100	3	2
CJNE Rn,#dato,rel	Compara inmediato con Rn y salta si distinto	1011 1r r r	3	2
CJNE @Ri,#dato,rel	Compara inmediato con @Ri y salta si distinto	1011 011i	3	2
DJNZ Rn,rel	Decrementa registro y salta si no es cero	1101 1r r r	2	2
DJNZ directo,rel	Decrementa directo y salta si no es cero	1101 0101	3	2
NOP	No Operación	0000 0000	1	1

- Son direccionables de dos formas:
 - **Completa (byte)**: cada puerto tiene asignada una dirección dentro del conjunto de registros especiales (SFRs). Se accede a ellos mediante una instrucción de transferencia de datos.
 - **Bit a bit**: se puede trabajar con cada bit de los puertos por separado mediante las instrucciones de manejo de bits. Para direccionar un bit la sintaxis es **Px.y** donde "x" es el número del puerto e "y" el bit deseado (0 a 7).

- Durante la inicialización del microcontrolador, el contenido de los puertos de E/S es FFh. Esta situación ha de tenerse en cuenta siempre y cuando se utilice uno de los puertos como salida ya que la salida estará en estado alto al menos todo el tiempo de la fase "**RESET**" del microcontrolador y no pasará al estado deseado por el usuario hasta que se ejecute una instrucción de escritura en el registro correspondiente al puerto del que forma parte esa salida.
- Por el contrario, cuando la línea deba ser utilizada como entrada, este estado inicial es indispensable para una buena recepción de información exterior. Si por el contrario se encuentra en estado bajo, impedirá los cambios de estado de la señal externa. Por tanto, para utilizar una línea de un puerto de entrada, es imprescindible asegurarse de que el bit que le corresponde en el registro del puerto esté en estado 1, tal como habrá sido puesto por la inicialización de controlador.



- ORG valor : actualiza contador de programa.
- Símbolo EQU valor: Asocia símbolo con un valor numérico u otro símbolo ya definido.
- Símbolo BIT valor : Dirección de bit.
- Símbolo DATA valor: Dirección dato en RAM interna con direccionamiento directo.
- Símbolo XDATA valor: Dirección dato en RAM externa.
- Símbolo CODE valor: Dirección de código.

- DB valor : genera el código del valor indicado.
- DB 'c': genera el ASCII del carácter entre comillas.
- DB "cadena": genera los códigos correspondientes a la cadena indicada.
- DB 'c', valor, "cadena": genera los códigos correspondientes a los parámetros separados mediante comas.
- END : final del archivo.

tabla XDATA 0	; Definición de etiquetas
valor EQU 55h	
dirección DATA 40h	
ORG 0	; Posición inicial después RESET
AJMP inicio	; Espacio para vectores interrup.
ORG 100h	; Comienzo de instrucciones
Inicio:	; Programa cíclico.
CLR A	
MOV DPTR,#tabla	
MOVBX A,@DPTR	
sjmp inicio	
END	; Fin de programa