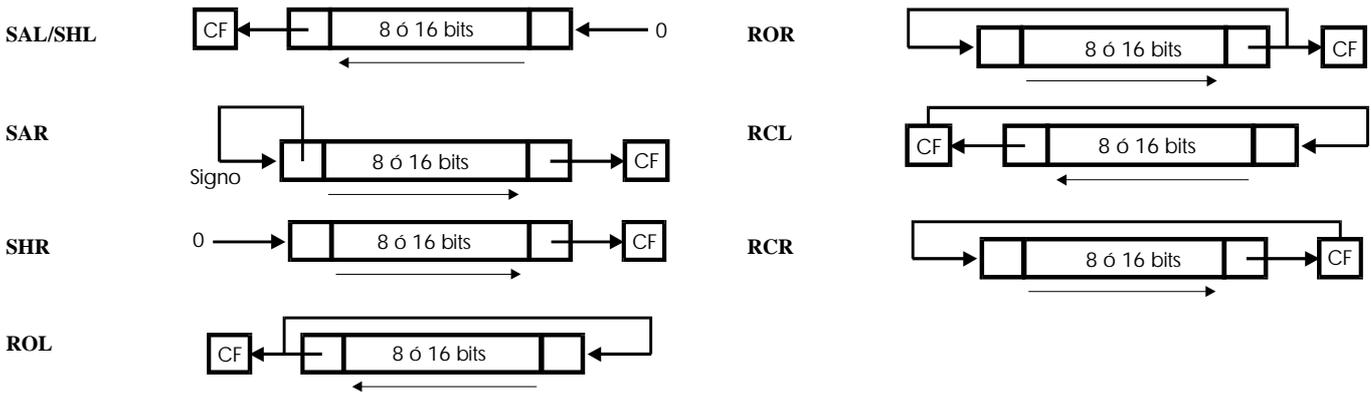


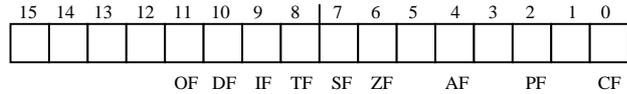
NE-MO-NIC.	OPERANDOS		FLAGS	DESCRIPCIÓN
	DESTINO.	FUENTE		
INSTRUCCIONES DE TRANSFERENCIA				
MOV	REG MEM/REG MEM/REG	MEM/REG REG NUM	No altera flags	COPIA EL OPERANDO FUENTE EN EL OPERANDO DESTINO. Ambos operandos deben ser del mismo tipo (BYTE O PALABRA). RESTRICCIONES: No se pueden mover datos entre dos elementos de memoria. No se puede usar MOV para transferir un valor inmediato a un registro de segmento. Ante cualquiera de estos dos casos, si es necesario, se usará un registro auxiliar. El registro CS no puede ser usado como destino.
PUSH	REG 16 / MEM 16		No altera flags	METE EL OPERANDO EN LA PILA. Decrementa en dos unidades el contenido del Puntero de Pila (SP) y después almacena la palabra especificada por el operando en la posición indicada por el puntero de pila (SS:SP). RESTRICCIONES: Solo se pueden guardar en la pila operandos tipo palabra, y antes hay que inicializar los registros SS y SP.
POP	REG 16 / MEM 16		No altera flags	SACA UNA PALABRA DE LA PILA. Transfiere la palabra de la pila, direccionada por SS:SP al operando destino (tipo palabra), y después incrementa en dos unidades el registro SP. El contenido de la pila no se borra, sino que el puntero es modificado. RESTRICCIONES: El registro CS no puede ser especificado como destino.
XCHG	MEM/REG	REG	No altera flags	INTERCAMBIA EL CONTENIDO DE LOS DOS OPERANDOS. Pueden ser byte o palabra, pero los DOS del mismo tamaño RESTRICCIONES: No se puede intercambiar dos posiciones de memoria. Para ello, se usa un registro auxiliar. No pueden ser operandos registros de segmento
XLAT			No altera flags	TRADUCE. Convierte caracteres de un código en otro. Reemplaza un byte contenido en AL, por un byte de una tabla de conversión (256 bytes de long. max.) creada por el usuario. AL es el índice de la tabla. BX apunta al comienzo de la tabla. AL= DS: BX+AL
IN	AL / AX	PORT / DX	No altera flags	COPIA EL CONTENIDO DE UN PUERTO DE ENTRADA EN EL ACUMULADOR. Tamaño byte o palabra, se almacena en AL o AX respectivamente. Si la dirección del puerto está entre 0 y 255, se especifica directamente en la instrucción. En general, para direcciones entre 0h y FFFFh, se carga en DX la dirección (variable)
OUT	PORT / DX	AL / AX	No altera flags	TRANSFIERE UN BYTE O PALABRA (ubicado en AL o AX respect.) AL PUERTO DE SALIDA ESPECIFICADO. La dirección del puerto puede ser un valor fijo de un byte (de 0 a FFh), o un valor variable que puede ser modificado por el programa, almacenado en DX (de 0 a FFFFh).
LEA	REG 16	MEM	No altera flags	CARGA LA DIRECCIÓN EFECTIVA. Carga el desplazamiento (offset) de la dirección de memoria origen en el registro de 16 bits indicado como destino. RESTRICCIONES: No puede usarse como destino ningún registro de segmento. NOTA: (LEA=dirección, MOV=contenido de la dirección)
LDS	REG 16	MEM	No altera flags	CARGA EL SEGMENTO DE DATOS. Carga el registro de 16 bits especificado como destino con el contenido de la palabra almacenada en la dirección MEM. Las posiciones involucradas son: (byte bajo de REG16)=MEM, (byte alto de REG16)=MEM+1, (byte bajo de DS)=MEM+2, (byte alto de DS)=MEM+3 RESTRICCIONES: No se puede usar como destino un registro de segmento. NOTA: LDS se suele usar para inicializar SI y DS antes de usar instrucc. de cadena
LES	REG 16	MEM	No altera flags	CARGA EL SEGMENTO EXTRA. Carga el registro de 16 bits especificado como destino con el contenido de la palabra almacenada en la dirección MEM. Las posiciones involucradas son: (byte bajo de REG16)=MEM, (byte alto de REG16)=MEM+1, (byte bajo de ES)=MEM+2, (byte alto de ES)=MEM+3 RESTRICCIONES: No se puede usar como destino un registro de segmento. NOTA: LES se suele usar para inicializar DI y ES antes de usar instrucc. de cadena
PUSHF			No altera flags	GUARDA EN LA PILA LA PALABRA DE ESTADO. Decrementa en dos unidades el registro SP, y guarda en la dirección de pila indicada por SS:SP el registro de flags PSW
POPF			OF,DF,IF,TF,SF,ZF,AF,PF,CF.	SACA UNA PALABRA DE LA PILA AL REGISTRO PSW. Saca la palabra de la dirección más baja de la pila (SS:SP) y la copia sobre el registro de estado del procesador (PSW). Posteriormente incrementa en dos unidades el puntero de pila SP
LAHF			No altera flags	ALMACENA EL BYTE BAJO DE PSW (flags SF, ZF, AF, PF y CF) EN AH. Sirve para ejecutar correctamente sobre un 8088/86 programas escritos para un 8080 o un 8085.
SAHF			SF,ZF,AF,PF,CF.	ALMACENA EL REGISTRO AH EN EL BYTE MENOS SIGNIFICATIVO DE PSW. Sirve para ejecutar correctamente sobre un 8088/86 programas escritos para un 8080 o un 8085.
INSTRUCCIONES ARITMETICAS				
ADD	REG MEM/REG MEM/REG	MEM/REG REG NUM	OF,SF,ZF,AF,PF,CF.	SUMAR. Suma el operando fuente al operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. RESTRICCIONES: No se permite la suma de dos posiciones de memoria.
ADC	REG MEM/REG MEM/REG	MEM/REG REG NUM	OF,SF,ZF,AF,PF,CF.	SUMAR CON ACARRERO. Suma el operando fuente al operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. Si el flag de CARRY (CF) está activado, suma uno al resultado. RESTRICCIONES: No se permite la suma de dos posiciones de memoria.
INC	REG / MEM		OF,SF,ZF,AF,PF (no afecta al flag DF)	INCREMENTA EL DESTINO EN UNA UNIDAD. Suma uno a la dirección de memoria o registro especificado. Si dicho operando de destino era FFFFh pasará a valer 0000h después de la ejecución de INC, y no se pondrá a '1' el flag CF
AAA			AF,CF. (Se activan si el n° en AL antes de AAA es > 9)	AJUSTE ASCII PARA LA SUMA. Solo opera sobre el registro AL, corrigiendo el resultado de una suma de dos números BCD desempaquetados (cada n° BCD viene representado por 8 bits) y convirtiéndolo en un número BCD desempaquetado.
DAA			SF,ZF,AF,PF,CF.	AJUSTE DECIMAL PARA LA SUMA. Corrige el resultado almacenado en AL correspondiente a la suma de dos números BCD empaquetados, convirtiéndolo en un número BCD empaquetado. Si el nibble (cuatro bits) de menor peso es superior a 9 o bien la bandera AF se activó durante la suma, la instrucción DAA le suma 6. Si el nuevo valor del nibble de mayor peso es ahora mayor que 9 o bien el flag CF está a '1', se suma 60h al registro AL.
SUB	REG MEM/REG MEM/REG	MEM/REG REG NUM	OF,SF,ZF,AF,PF,CF.	RESTAR. Resta el operando fuente del operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. RESTRICCIONES: No se permite la resta de dos posiciones de memoria.
SBB	REG MEM/REG MEM/REG	MEM/REG REG NUM	OF,SF,ZF,AF,PF,CF.	RESTA CON BORROW. Resta el operando fuente del operando destino, almacenando el resultado en el operando destino. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. Si la bandera de arrastre está activada (CF=1), se resta 1 al resultado. RESTRICCIONES: No se permite la resta de dos posiciones de memoria.
DEC	REG / MEM		OF,SF,ZF,AF,PF (no afecta a CF)	DECREMENTA EL DESTINO EN UNA UNIDAD. Resta uno al operando destino. Dicho operando puede estar almacenado en una posición de memoria o en un registro, y su tamaño puede ser byte o palabra.
NEG	REG / MEM		OF,SF,ZF,AF,PF,CF	FORMAR EL COMPLEMENTO A 2. Invierte el signo del operando destino. El operando puede ser una posición de memoria o un registro.
CMP	REG MEM/REG MEM/REG	MEM/REG REG NUM	OF,DF,SF,ZF,AF,PF,CF.	COMPARAR DOS OPERANDOS. Lo hace mediante la resta del operando fuente del destino. El resultado NO es almacenado, y sólo se actualiza el contenido de los flags. Los operandos pueden ser de tipo byte o palabra, pero AMBOS DEL MISMO TIPO. RESTRICCIONES: No se permite la comparación entre dos posiciones de memoria.
AAS			AF,CF. (Se activan si el n° en AL antes de AAS es > 9)	AJUSTE ASCII PARA LA RESTA. Corrige el resultado en AL de la resta de dos números decimales desempaquetados, convirtiéndolo en un valor decimal desempaquetado.
DAS			SF,ZF,AF,PF,CF	AJUSTE DECIMAL PARA LA RESTA. Corrige el resultado almacenado en AL correspondiente a la resta de dos números BCD empaquetados, convirtiéndolo en un número BCD empaquetado. Si el nibble (4 bits) de menor peso es superior a 9 o bien el flag AF se activó durante la resta, la instrucción DAS le resta 6. Si el nuevo valor del nibble de mayor peso es ahora mayor que 9 o bien el flag CF está a 1, se resta 60h al registro AL.
MUL	REG / MEM		OF,CF (se activan cuando la mitad superior del resultado -en DX ó AH- no sea 0)	MULTIPLICAR SIN SIGNO. Multiplica un n° sin signo tamaño byte por un n° sin signo tamaño byte contenido en AL guardando el resultado en AX (AX= AL * operando byte) o multiplica un n° sin signo tamaño word por un n° sin signo tamaño word contenido en AX guardando el resultado en DX (palabra más significativa) y en AX (palabra menos significativa) (DX AX = AX * operando word).
IMUL	REG / MEM		OF,CF (se activan cuando la mitad superior del resultado -en DX ó AH- no sea 0)	MULTIPLICA CON SIGNO. Multiplica un n° con signo tamaño byte por un n° con signo tamaño byte contenido en AL guardando el resultado en AX (AX= AL * operando byte) o multiplica un n° con signo tamaño word por un n° con signo tamaño word contenido en AX guardando el resultado en DX (palabra más significativa) y en AX (palabra menos significativa) (DX AX = AX * operando word).
AAM			SF,ZF,PF	AJUSTE ASCII PARA LA MULTIPLICACIÓN. Corrige el resultado en AX del producto de dos números decimales desempaquetados, convirtiéndolo en un valor desempaquetado. (AH=cociente de AL / 10) y (AL=resto de AL / 10)
DIV	REG / MEM		Todos los flags quedan INDEFINIDOS.	DIVIDIR SIN SIGNO. Divide el contenido del acumulador y su extensión (AH AL si el operando es de tipo byte, o DX AX si el operando es de tipo word) entre el operando fuente. Hay dos posibilidades: Dividir 16 bits entre 8 bits (Dividendo=AX, Divisor=fuente 8bits, Cociente=AL, Resto=AH) o dividir 32 bits entre 16 bits (Dividendo=DX AX, Divisor=operando 16bits, Cociente=AX, Resto=DX). Se genera una interrupción de tipo 0 si el cociente supera Ffh o FFFFh respectivamente.
IDIV	REG / MEM		Todos los flags INDEFINIDOS. Signo resto=signo dividendo	DIVIDIR CON SIGNO. Divide el contenido del acumulador y su extensión (AH AL si el operando es de tipo byte, o DX AX si el operando es de tipo word) entre el operando fuente. Hay dos posibilidades: Dividir 16 bits entre 8 bits (Dividendo=AX, Divisor=fuente 8bits, Cociente=AL, Resto=AH) o dividir 32 bits entre 16 bits (Dividendo=DX AX, Divisor=operando 16bits, Cociente=AX, Resto=DX). Int tipo 0 si 7Fh < cociente < 80h o 7FFFh < cociente < 8000h respectivamente.
AAD			SF,ZF,PF	AJUSTE ASCII PARA LA DIVISION. Convierte dos dígitos en código BCD desempaquetado almacenados en AH y AL en el correspondiente n° binario. El resultado es almacenado en AL.
CBW			No altera flags	CONVIERTE BYTE EN WORD. Copia el bit de signo del byte almacenado en AL sobre todos los bits de AH. A esta operación se denomina <i>extensión del signo de AL</i> . Si es sin signo, se rellena los bits + significativos con 0. Si es con signo, se rellena los bits + significativos con el bit de mayor peso.
CWD			No altera flags	CONVIERTE WORD EN DOUBLEWORD. Copia el bit de signo de la palabra almacenada en AX sobre todos los bits de DX. A esta operación se denomina <i>extensión del signo de AX en DX</i> .
INSTRUCCIONES LÓGICAS				
NOT	REG / MEM		No altera flags	NO LÓGICO. Forma el complemento a uno del operando, esto es, cambia los ceros por unos y los unos por ceros. El operando puede ser un registro o una posición de memoria de tamaño byte o word.
AND	REG MEM/REG MEM/REG	MEM/REG REG NUM	OF=0,SF,ZF,PF,CF=0 (AF indefinida)	Y LÓGICO. Realiza la operación lógica AND entre los operandos fuente y destino, realizada bit a bit y almacenada en el destino. RESTRICCIONES: No se puede realizar la operación entre dos posiciones de memoria.
OR	REG MEM/REG MEM/REG	MEM/REG REG NUM	OF=0,SF,ZF,PF,CF=0 (AF indefinida)	O LÓGICO. Realiza la operación OR inclusiva entre los operandos fuente y destino, realizada bit a bit y almacenada en el destino. RESTRICCIONES: No se puede realizar la operación entre dos posiciones de memoria.

NE- MÓ- NIC.	OPERANDOS		FLAGS	DESCRIPCIÓN
	DESTINO,	FUENTE		
JG JNLE	DESPLAZAMIENTO		No altera flags	SALTO SI MAYOR. Salta si se cumple ZF=0' y SF=0F. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante. El número 00000011 es mayor que 11111000 NOTA: Usaremos MAYOR Y MENOR con números con signo, y SUPERIOR e INFERIOR para números sin signo.
JGE JNL	DESPLAZAMIENTO		No altera flags	SALTO SI MAYOR O IGUAL. Salta si se cumple SF=0F. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante. El número 00000011 es mayor que 11111000 NOTA: Usaremos MAYOR Y MENOR con números con signo, y SUPERIOR e INFERIOR para números sin signo.
JL JNGE	DESPLAZAMIENTO		No altera flags	SALTO SI MENOR. Salta si se cumple SF distinto de 0F. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante. El número 00000011 es mayor que 11111000 NOTA: Usaremos MAYOR Y MENOR con números con signo, y SUPERIOR e INFERIOR para números sin signo.
JLE JNG	DESPLAZAMIENTO		No altera flags	SALTO SI MENOR O IGUAL. Salta si se cumple ZF=1' o SF distinto de 0F. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante. El número 00000011 es mayor que 11111000 NOTA: Usaremos MAYOR Y MENOR con números con signo, y SUPERIOR e INFERIOR para números sin signo.
JNE JNZ	DESPLAZAMIENTO		No altera flags	SALTO SI NO IGUAL. Salta si se cumple ZF=0'. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante. NOTA: Usaremos MAYOR Y MENOR con números con signo, y SUPERIOR e INFERIOR para números sin signo.
JNO	DESPLAZAMIENTO		No altera flags	SALTO SI NO HAY DESBORDAMIENTO. Salta si se cumple OF=0'. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante.
JNP JPO	DESPLAZAMIENTO		No altera flags	SALTO SI NO HAY PARIDAD, O SI ES PARIDAD IMPAR. Salta si se cumple PF=0'. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante.
JNS	DESPLAZAMIENTO		No altera flags	SALTO SI POSITIVO. Salta si se cumple SF=0'. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante.
JO	DESPLAZAMIENTO		No altera flags	SALTO SI HAY DESBORDAMIENTO. Salta si se cumple OF=1'. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante.
JP JPE	DESPLAZAMIENTO		No altera flags	SALTO SI HAY PARIDAD. Salta si se cumple PF=1'. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante.
JS	DESPLAZAMIENTO		No altera flags	SALTO SI SIGNO. Salta si se cumple SF=1'. El salto está comprendido entre 128 posiciones hacia atrás y 127 hacia delante.
LOOP	DIRECCIÓN		No altera flags	BUCLE. Provoca la repetición de una serie de instrucciones un número determinado de veces, especificado por el registro contador CX. Si CX es distinto de 0', salta a la dirección. Esa dirección debe estar entre 128 posiciones hacia atrás y 127 hacia delante.
LOOPE LOOPZ	DIRECCIÓN		No altera flags	BUCLE SI IGUAL O CERO. Provoca la repetición de una serie de instrucciones un número determinado de veces, especificado por el registro contador CX. Si CX es distinto de 0' y ZF=1', salta a la dirección. Esa dirección debe estar entre 128 posiciones hacia atrás y 127 hacia delante.
LOOPNE LOOPNZ	DIRECCIÓN		No altera flags	BUCLE SI NO IGUAL O NO CERO. Provoca la repetición de una serie de instrucciones un número determinado de veces, especificado por el registro contador CX. Si CX es distinto de 0' y ZF=0', salta a la dirección. Esa dirección debe estar entre 128 posiciones hacia atrás y 127 hacia delante.
INT	TIPO DE INTERRUPCIÓN	IF=0, TF=0		INTERRUPCIÓN. Realiza una interrupción por software. Abandona el curso normal del programa para ejecutar la rutina de atención de la interrupción, salvando antes el contenido de IF e IP en la pila. El vector de interrupción está ubicado en la posición de mem (0000:4* int) a la (0000:4* int+3)
INTO		IF=0, TF=0		INTERRUPCIÓN SI OVERFLOW. Genera la interrupción interna tipo 4 en el caso de que la bandera de overflow OF=1'
IRET		TODAS LAS BANDERAS SON RESTAURADAS		RETORNO DE UNA INTERRUPCIÓN. Devuelve el control del programa a la instrucción siguiente a donde se había interrumpido, sacando su dirección de la pila. Se restauran los valores de CS, IP e IF. Todas las banderas son restauradas con los valores que tenían antes de la interrupción.
INSTRUCCIONES DE MANIPULACIÓN DE FLAGS				
CLC		CF=0		PONE A CERO EL FLAG DE CARRY
CLD		DF=0		PONE A CERO EL FLAG DE DIRECCIÓN
CLI		IF		BORRA EL FLAG DE INTERRUPCIÓN. Desactiva el permiso de interrupción para las interrupciones enmascarables. Se seguirán tratando las interrupciones no enmascarables NMI y las software.
CMC		CF		COMPLEMENTAR EL FLAG DE ACARREO. Si CF=0' lo pone a 1', si CF=1' lo pone a 0'
STC		CF=1		PONER FLAG DE CARRY. Pone a 1' el flag de carry
STD		DF=1		PONER FLAG DE DIRECCIÓN. Pone a 1' el flag de dirección.
STI		IF		PONER FLAG DE INTERRUPCIÓN. Pone a uno el flag de interrupción, permitiendo las interrupciones enmascarables. No tiene efecto hasta que no se haya ejecutado la instrucción que sigue a la STI
INSTRUCCIONES DE CONTROL DEL MICROPROCESADOR				
ESC	COD. OP.	REG / MEM		ESCAPE. Se usa para pasar instrucciones a un coprocesador. El primer operando es un código de operación. El segundo operando indica dónde se encuentra el dato que se le va a pasar
HLT				PARADA DEL PROCESADOR. Cesa la actividad de búsqueda. Sólo saldrá de este estado mediante una petición de interrupción externa enmascarable, no enmascarable, o una señal RESET. Útil para diagnosticar equipos.
LOCK				CIERRE DEL BUS. Usado en sistemas multiprocesador para compartir recursos, LOCK se usará delante de una instrucción crítica que deba tener prioridad absoluta en el control del bus.
NOP				NO OPERACIÓN. Sólo consume tres ciclos de reloj. Usado para ajustar retardos.
WAIT				ESPERA. El microprocesador entra en estado de letargo sin hacer nada. Se puede salir mediante la activación a nivel bajo de la entrada TEST, o con una petición de interrupción. Se usa para sincronizar.

ESQUEMA DE DESPLAZAMIENTOS:



PALABRA DE ESTADO PSW:



CODIFICACIÓN DE INSTRUCCIONES :

COD_OP	D	W	MOD	REG	R / M	DESPLAZA.	VALOR
--------	---	---	-----	-----	-------	-----------	-------

D: 0 REG = FUENTE
 1 REG = DESTINO

W: 0 TAMAÑO BYTE
 1 TAMAÑO WORD

DIRECCIÓN DE REGISTRO	REGISTROS W=1	REGISTROS W=0
000	AX	AL
001	CX	CL
010	DX	DL
011	BX	BL
100	SP	AH
101	BP	CH
110	SI	DH
111	DI	BH

Dirección de Registro	Registro de Segmento
00	ES
01	CS
10	SS
11	DS

R / M	MOD	00	01	10	W=0	W=1
000		(BX)+(SI)	(BX)+(SI)+D8	(BX)+(SI)+D16	AL	AX
REG. DE SEGMENTO		DS	DS	DS		
001		(BX)+(DI)	(BX)+(DI)+D8	(BX)+(DI)+D16	CL	CX
REG. DE SEGMENTO		DS	DS	DS		
010		(BP)+(SI)	(BP)+(SI)+D8	(BP)+(SI)+D16	DL	DX
REG. DE SEGMENTO		SS	SS	SS		
011		(BP)+(DI)	(BP)+(DI)+D8	(BP)+(DI)+D16	BL	BX
REG. DE SEGMENTO		SS	SS	SS		
100		(SI)	(SI)+D8	(SI)+D16	AH	SP
REG. DE SEGMENTO		DS	DS	DS		
101		(DI)	(DI)+D8	(DI)+D16	CH	BP
REG. DE SEGMENTO		DS	DS	DS		
110		D16	(BP)+D8	(BP)+D16	DH	SI
REG. DE SEGMENTO		DS	DS	DS		
111		(BX)	(BX)+D8	(BX)+D16	BH	DI
REG. DE SEGMENTO		DS	DS	DS		

USO DE LOS REGISTROS DE SEGMENTO:

TIPO DE OPERACIÓN	SEGMENTO POR DEFECTO	SEGMENTO OPCIONAL	OFFSET
Busqueda de instrucción	CS	NINGUNO	IP
Operación con la pila	SS	NINGUNO	SP
Operación con cadena fuente	DS	CS, ES, SS	SI
Operación con cadena destino	ES	NINGUNO	DI
Lectura / Escritura de datos	DS	CS, ES, SS	DIRECCIÓN EFECTIVA
BP Usado como registro base	SS	CS, DS, ES	DIRECCIÓN EFECTIVA

AH	Función
AH = 06h	Desplazamiento (scroll) hacia arriba AL es el número de líneas. Si AL=0 se borra la ventana. CH fila esquina superior izquierda. DH fila esquina inferior derecha. BH relleno. CL columna esquina superior izquierda. DL columna esquina inferior derecha.
AH = 07h	Desplazamiento (scroll) hacia abajo AL es el número de líneas. Si AL=0 se borra la ventana. CH fila esquina superior izquierda. DH fila esquina inferior derecha. BH relleno. CL columna esquina superior izquierda. DL columna esquina inferior derecha.
AH = 08h	Leer carácter y atributo de la posición actual BH es el número de página. Devuelve: AL el carácter leído AH atributo del carácter leído.
AH = 09h	Escribir el carácter y el atributo en la posición actual del cursor BH es el número de página. CX es el número de caracteres a escribir. BL es el atributo del carácter o el color. AL es el carácter a escribir.
AH = 0Eh	Escribir el carácter en la pantalla y avanzar el cursor AL es el carácter a escribir. BL es el color del carácter o su atributo. BH es el número de la página.
AH = 0Fh	Leer el estado actual de la pantalla Devuelve: AL el modo. AH número de columnas de la pantalla. BH número de página activa.

Funciones de la interrupción 16h.

AH	Función
AH = 00h	Leer pulsación del teclado Salida: AH = Código BIOS de rastreo; AL = Código ASCII de la tecla. Sólo reconoce 84 teclas, elimina del buffer las del teclado extendido. Espera tecla. Elimina tecla del buffer al leerla.
AH = 01h	Comprobar pulsación del teclado Salida: si no hay tecla en el buffer ⇒ Flag de cero activado. si hay tecla ⇒ Flag de cero desactivado; AH = Código BIOS de rastreo. AL = Código ASCII de la tecla. No espera tecla. No reconoce teclado extendido. No elimina tecla del buffer.
AH = 02h	Obtener estado del teclado Salida: AL = Estado del teclado.
AH = 10h	Leer pulsación del teclado extendido Salida: AH = Código BIOS de rastreo. AL = Código ASCII de la tecla. Espera tecla. Elimina tecla del buffer.
AH = 11h	Comprobar pulsación del teclado extendido Salida: Si no hay tecla en el buffer ⇒ Flag de cero activado Si hay tecla ⇒ Flag de cero desactivado AH = Código BIOS de rastreo AL = Código ASCII de la tecla No elimina tecla del buffer.
AH = 12h	Obtener estado del teclado extendido Salida: AH = Estado del teclado extendido. AL = Estado del teclado.

INTERFAZ PARALELA 8255A

Tabla 1. Operación de lectura en la PPI

A1	A0	RD#	WR#	CS#	Operación de entrada (Lectura)
0	0	0	1	0	Puerto A → Bus de Datos
0	1	0	1	0	Puerto B → Bus de Datos
1	0	0	1	0	Puerto C → Bus de Datos
1	1	0	1	0	Palabra de control → Bus de datos

Tabla 2. Operaciones de escritura en la PPI

A1	A0	RD#	WR#	CS#	Operación de salida (Escritura)
0	0	1	0	0	Bus de Datos → Puerto A
0	1	1	0	0	Bus de Datos → Puerto B
1	0	1	0	0	Bus de Datos → Puerto C
1	1	1	0	0	Bus de Datos → Palabra de control

Tabla 3. Selección de la PPI

A1	A0	RD#	WR#	CS#	Deshabilitación
x	x	x	x	1	Bus de datos → Three-State
x	x	1	1	0	Bus de datos → Three-State

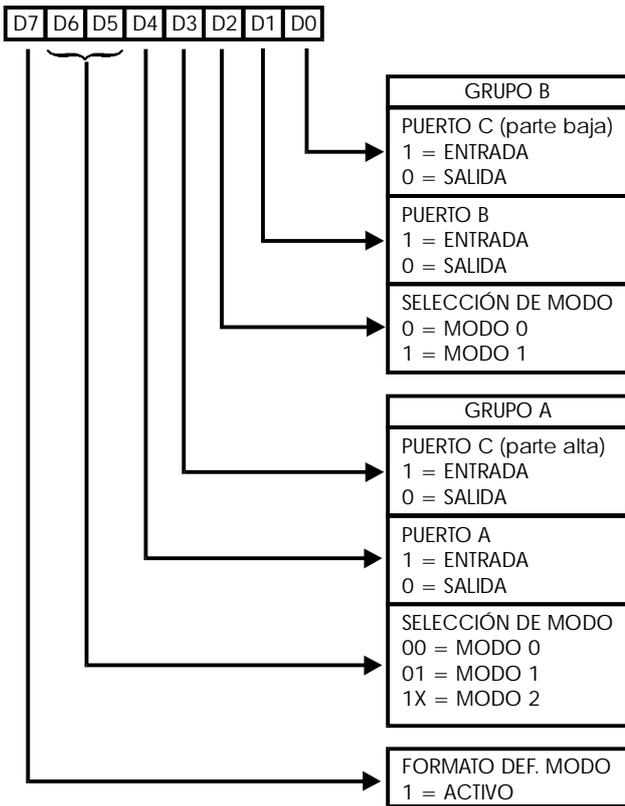


Figura 1. Registro de control

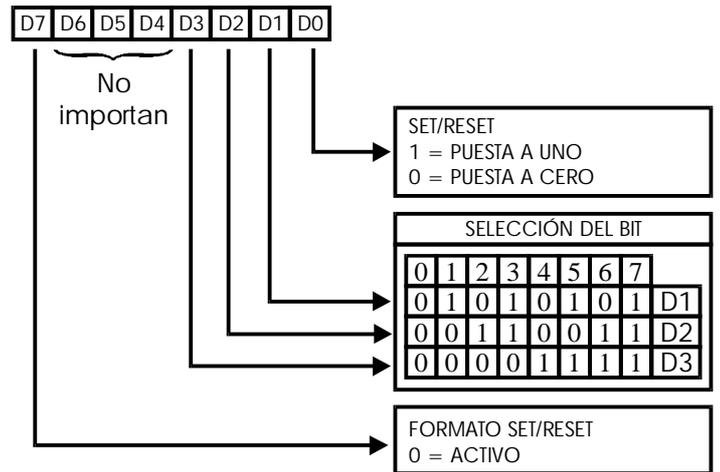


Figura 2. Palabra de control. Set/reset de los bits del puerto C

● **FORMATO PUERTO C EN MODO1 ENTRADA (arriba) /SALIDA (abajo):**

I/O	I/O	IBFa	INTEa	INTRa	INTEb	IBFb	INTRb
D7	D6	D5	D4	D3	D2	D1	D0
OBFa	INTEa	I/O	I/O	INTRa	INTEb	OBFb	INTRb

● **FORMATO DE LA PALABRA DE ESTADO O PUERTO C EN MODO 2:**

OBFa	INTE1	IBFa	INTE2	INTRa	X	X	X
D7	D6	D5	D4	D3	D2	D1	D0

INTERFAZ SERIE USART 8251A

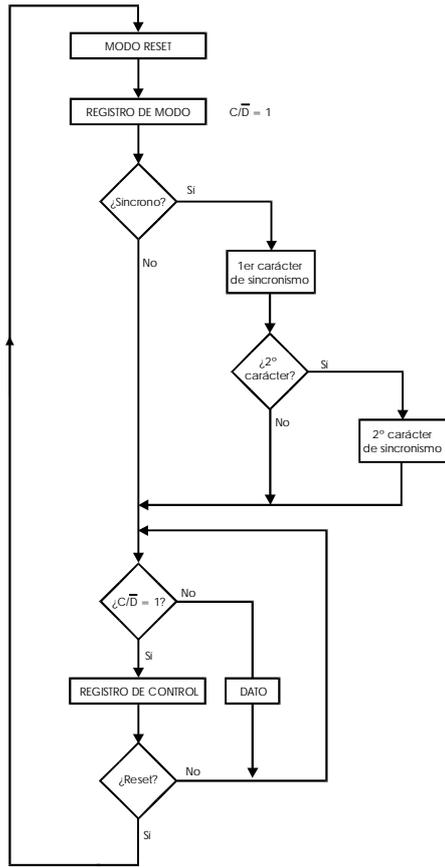


Figura 3. Secuencia de programación

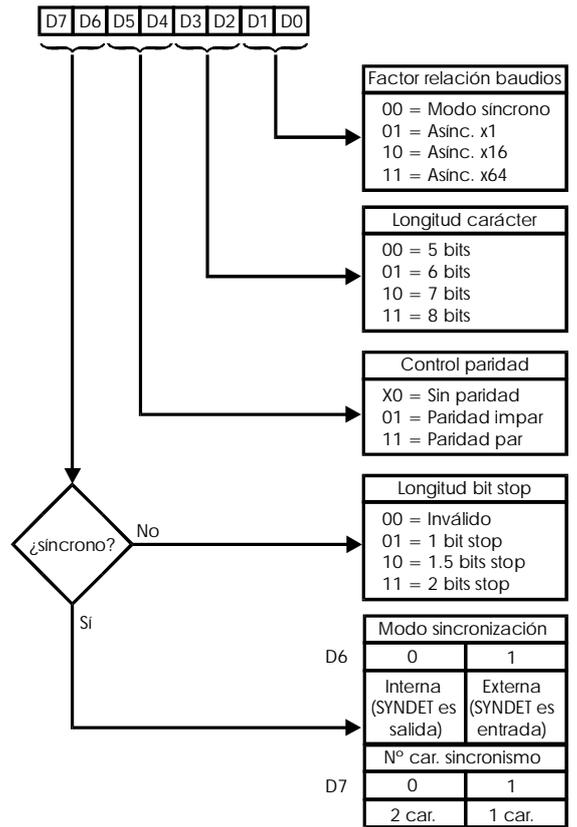


Figura 4. Formato del registro de modo

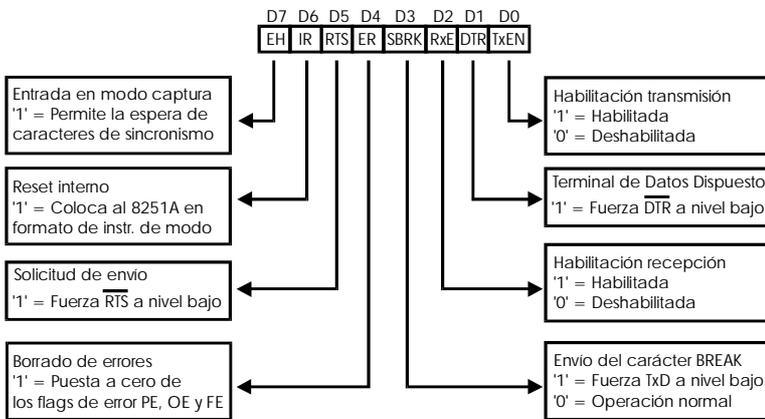


Figura 5. Palabra de control

Registro	C/D	Tipo
Modo	1	Escritura
Control	1	Escritura
Estado	1	Lectura
1º carácter sincronismo	1	Escritura
2º carácter sincronismo	1	Escritura
Registro de transmisión	0	Escritura
Registro de recepción	0	Lectura

Registros internos de la USART

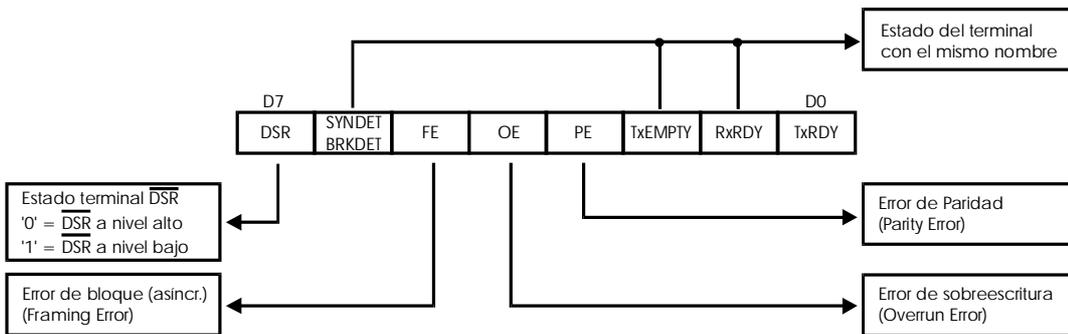


Figura 6. Registro de estado