

001.62/25
Os 93
Pt. 2
T-A2.

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

ESPECIALIZACION EN ELECTRONICA Y
TELECOMUNICACIONES

DEPURADOR DE PROGRAMAS PARA EL
MICROPROCESADOR 80486
(PARTE 2)

JORGE BOLIVAR OSORIO HINOJOSA
ALEX JOHNINE TROYA ALDAZ

TESIS PREVIA A LA OBTENCION DEL TITULO DE
INGENIERO EN LA ESPECIALIZACION DE INGENIERIA
ELECTRONICA Y TELECOMUNICACIONES DE LA
ESCUELA POLITECNICA NACIONAL

MAYO 1997

PRO486.EXE Es un programa depurador que abarca las instrucciones hasta el microprocesador 80486, incluyendo las instrucciones del Coprocesador Matemático integrado en el Microprocesador. La sesión de trabajo con el depurador se inicia con una línea de comando como se indica a continuación :

PRO486 *Path/NombreArchivo [PARAMETROS]*

Path indica el camino donde se halla el archivo que se va a depurar, *NombreArchivo* es el nombre del programa que se va a depurar (.EXE o .COM), y *PARAMETROS* son los parámetros, si existen, del mismo. Los tres grupos de parámetros *Path*, *NombreArchivo*, *PARAMETROS* son opcionales y en caso que no se especifiquen el usuario cargará desde el interior del depurador el archivo con el que va a trabajar (mediante la opción Archivo del menú).

El paquete completo incluye los siguientes archivos:

- PRO486.exe.- Archivo principal del paquete
- Child486.exe.- Archivo auxiliar necesario para ejecutar pro486.exe.
- Wincolor.vnt.- Contiene las ventanas del depurador para monitores a color. Debe estar presente.
- Winbn.vnt.- Contiene las ventanas del depurador para monitores B/N. Debe estar presente.
- Pro486.mcr.- Archivo en el que se graban los macros de funciones. No es necesaria su presencia y si no existe, el depurador lo crea automáticamente.

- Edit.com.- Archivo editor de texto, funciona con las opciones <F6> y <F2>, si no está presente no se podrá visualizar al archivo fuente (.asm) ni el archivo de ayuda Pro486.hlp.
- Pro486.hlp.- Contiene la información aquí presentada. Se accesa con la función <F2> desde el menú principal
- Command.com.- Debe estar dentro del directorio en el que se halle pro486.exe o debe estar ubicado en el directorio RAIZ, sirve para la opción Shell del depurador (salir temporalmente al sistema).
- Masm.exe, Link.exe.- Necesarios para usar la opción de menú eXe que enlaza y ensambla un archivo .ASM previamente editado con la opción <F6>. Estos archivos deben estar dentro del mismo directorio en el que se encuentre el depurador, donde además deben estar los archivos ML.EXE, DOSXNT.EXE y ML.ERR.

Este depurador presenta las siguientes facilidades:

- Permite desde el interior del depurador cambiar el programa que se desea depurar (.EXE o .COM). Mediante la opción *Archivo* del menú, para cambiar el archivo que se va a depurar no hace falta que el archivo anterior se haya ejecutado completamente.
- Se puede salir del depurador sin tener que terminar de ejecutar el programa que se está depurando.
- Permite escribir directamente sobre los registros, banderas y sobre localidades de memoria, mediante < F1 >, o haciendo click con el ratón en el área de la ventana respectiva.

- Las variaciones de los datos de memoria producidos por la depuración del programa se actualizan automáticamente.
- Ventana adicional del coprocesador que permite ver el flujo de datos en los registros del coprocesador, mediante < F3 >.
- Permite ver el archivo fuente (.ASM), mediante < F6 >. Se debe ingresar el path y el nombre del archivo fuente que se desea ver.
- Conversión de datos decimal a hexadecimal, mediante < F9 >.
- Tiene la opción de crear macros de funciones (mediante Ctrl-K), un macro puede ser por ejemplo abrir un archivo y depurarlo paso a paso, se termina la creación del macro presionando nuevamente las teclas Ctrl-K, los macros creados se graban automáticamente en el archivo de macros al salir normalmente del programa.
- Con <F4> se intercambia entre pantalla del depurador y pantalla del programa en depuración (pantalla de usuario).
- Ejecutar el programa hasta una dirección dada o hasta que se cumpla una condición ingresada por el usuario, mediante < F7 >.
- Permite insertar puntos de parada (se permiten hasta 256 puntos de parada).
- Permite ingresar o no a depurar paso a paso las subrutinas (call), o lazos, mediante < F8 >.
- Permite Ensamblar y Enlazar un archivo (.ASM) con la opción eXe del menú principal. Importante: antes de usar esta opción previamente debe editarse el archivo con la función <F6>. Siempre se ensambla y enlaza el último archivo ingresado con la función <F6>.
- Permite el uso del Ratón

RECOMENDACIONES PARA EL MEJOR USO DEL DEPURADOR

- El depurador requiere de una computadora con procesador 80386 y posteriores. Esté seguro de usar únicamente las instrucciones que soporta cada procesador, caso contrario el depurador presentará un mensaje de error y no permitirá seguir depurando el archivo, hasta que se cargue uno nuevo.
- Usar el depurador preferentemente en ambiente DOS. Si usted dispone de Windows 95 en "Inicio" elija "Apagar el sistema" seguidamente "Reiniciar el equipo en modo MS-DOS", al terminar retorne a Windows digitando EXIT y ENTER.
- Si está depurando en ambiente Windows se recomienda que el formato de Memoria de los Programas a depurar sea de "Control Simplificado de Segmento" y no de "Control Completo de segmento". Control Simplificado de Segmento es una forma de configurar los segmentos mediante directivos (.model, .stack, .data, .code, .exit, etc.) que cuando son ensamblados configuran automáticamente al programa en sus respectivos segmentos. El control-completo (por ejemplo MYCODE SEGMENTO PARA 'CODE') en cambio requiere que el usuario configure los segmentos a conveniencia.
- Al escribir directamente en localidades de memoria recuerde que puede afectar la información del programa que este depurando u otros programas residentes, lo que puede causar que la computadora no funcione correctamente hasta que la reinicie nuevamente.

OPCIONES DEL MENU

Con las teclas de (← , →) o posicionando el puntero del ratón, elija la opción del menú deseada, con ENTER o con un click del ratón ejecuta la opción remarcada. También se puede seleccionar una opción mediante una letra clave que es la letra en mayúscula del nombre en cada opción de menú. Las opciones de menú terminadas con el símbolo "↓" indica que tiene subopciones.

<.. > Esta opción de los dos puntos en cada nivel del menú permite salir al nivel superior, y si está en el nivel principal permite salir del programa previa confirmación.

<Archivos>.- Permite ingresar el nombre y cargar el archivo a depurar. El archivo a ingresar deberá ser de extensión EXE o COM

<Infor>.- datos de realización del programa

<Depurar↓>.- Contiene las opciones de depuración

<Paso>.- Ejecuta la instrucción remarcada, ingresando o no a las subrutinas o lazos de acuerdo al estado de <F8> que se muestra en la línea 23 (SiSubr, ingreso ; NoSubr, no ingreso).

<Corre>.- Dependiendo del estado de <F7> (Direc, ejecutar hasta una dirección determinada; Cond, ejecutar hasta una condición determinada), ejecuta el programa hasta una predeterminada dirección o Condición.

<eJecuta>.- Ejecuta el programa hasta hallar un punto de parada, o hasta el final si no hay puntos de parada.

<Reinicio>.- Permite reinicializar desde cualquier punto el programa que se esta depurando.

<pArada↓>.- Establecer puntos de parada

<Crea>.- Inserta un punto de parada, primeramente se pide un nombre (máximo de 8 bytes) para el punto de parada, posteriormente se pide ingresar la dirección donde se desea ubicar el punto de parada, por defecto se considera al segmento de código actual, y el usuario debe ingresar el desplazamiento. Con TAB se puede cambiar al campo de segmento para cambiar el valor del segmento por defecto, presionando otra vez TAB se retorna al campo de Desplazamiento. Se acepta los valores ingresados mediante ENTER, y automáticamente el depurador espera por el nombre de otro punto de parada, se termina la creación de puntos de parada con la tecla ESC. Si se ingresa un nombre o dirección ya existente, se borra el punto de parada anterior y se crea el nuevo.

<Act/des>.- Activa o Desactiva un punto de parada creado en forma alterna, el depurador solicita que ingrese el nombre del punto de parada que desea activar o desactivar. Un punto activado tiene la dirección de desplazamiento de color Rojo en la ventana de código, y un punto de ruptura desactivado tiene

las letras de color Verde. Se termina el proceso de Activación/Desactivación mediante la tecla ESC.

<Borra>.- Eliminar un punto de parada, el usuario debe ingresar el nombre del punto de parada que dese borrar.

<Muestra>.- Lista los puntos de parada creados, pueden existir hasta 256 puntos de parada. La ventana solo muestra 8 puntos de parada a la vez, para ver los siguientes ocho se presiona Page Down (Página Abajo), y Page UP (Página arriba) para ver los ocho puntos de parada superiores.

<Ins>.- Ejecuta directamente una instrucción escrita por el usuario, no la inserta en el código de programa que se está depurando. Sólo se han incluido las instrucciones más usadas, en caso que se teclee una instrucción no aceptada el depurador presentará un mensaje de instrucción no válida.

<Memoria↓>.- Permite el manejo de bloques de memoria.

<Llena>.- Llena desde la posición actual de la ventana de memoria un cierto número de localidades con un valor determinado.

<Busca>.- Permite buscar una secuencia de datos en formato hexadecimal o ASCII.

<Copia>.- Copia desde la posición actual un cierto número de localidades de memoria a la dirección especificada.

<Funcion>.- Ver las opciones de teclas de función

<Salir>.- Permite salir del programa previa confirmación.

<Shell>.- Permite salir temporalmente al sistema DOS, se retorna al depurador tecleando EXIT. Para usar esta opción se recomienda correr el depurador desde DOS y no desde el programa NC (Norton Commander), esto para tener memoria disponible.

<eXe>.- Permite ensamblar y enlazar un archivo (.ASM), para esto primero se deberá editar al archivo con la función <F6>, siempre ensambla y enlaza el último archivo editado con esta función. Una vez que ha ensamblado el archivo presenta un mensaje indicando que se ha terminado con el proceso, el usuario debe verificar que no existan mensajes de error. En caso de existir mensajes de error se debe teclear ESC con lo que se termina con el proceso y se retorna al depurador, si no existen mensajes de error se debe presionar ENTER para continuar con el proceso de enlazado. En enlazador solicita al usuario el nombre del archivo .OBJ a enlazar, se puede dar un solo nombre o varios (programación por módulos) usando la opción “ + ” del enlazador. Una vez que se ha terminado con el enlazado se presenta un mensaje de terminación de proceso, el usuario debe verificar que no existan mensajes de error de ser así debe editar el archivo fuente, realizar las correcciones respectivas y volver a repetir el proceso de ensamblado y enlazado.

<Funcion>.- Ver las opciones de teclas de función.

OPCIONES DE LAS TECLAS DE FUNCION

Se activa presionando la tecla de función correspondiente o al hacer click sobre el menú de funciones presentadas en la línea 23.

- <F1>.- Permite acceder a las ventanas del depurador ya sea para alterar datos, registros, banderas, localidades de memoria o simplemente para ver información. También se puede acceder a las ventanas haciendo click sobre la región que la limita.
- <F2>.- Si se está en el nivel principal del menú, muestra este archivo de ayuda (PRO486.HLP), caso contrario muestra ventanas con información rápida en cada subnivel.
- <F3>.- Muestra/Oculta la ventana del coprocesador.
- <F4>.- Intercambia entre pantalla del Depurador y Pantalla del Usuario, para retornar a la pantalla del depurador presione <ESC> o <F4>.
- <F5>.- Amplía la ventana de código, inicialmente solo está visible una parte de la misma. Se puede seguir depurando con ésta ventana ampliada.
- <F6>.- Permite ver cualquier archivo que desee el usuario. Al presionar ésta tecla aparecerá una ventana solicitando el nombre y path del archivo que desee editar. Es importante que en el mismo directorio de PRO486.EXE esté el archivo EDIT.COM. Si no dispone de EDIT.COM, utilice cualquier otro editor de texto renombrando previamente a tal editor como EDIT.COM
- <F7>.- Intercambia entre las opciones de Direc/Cond.
- <F8>.- Si/No ingreso a subrutinas durante la depuración paso a paso.
- <F9>.- Presenta una ventana para hallar el equivalente en hexadecimal de un número decimal (valor máximo 4294967295 = FFFFFFFFH).
- <F10>.- Presenta una ventana con los códigos ASCII.

OPCIONES DE MACROS DE USUARIO

Para acceder a los macros de usuario se debe presionar las teclas Ctrl-K desde el menú principal. Se presenta una lista con los nombres de macros existentes y se pide ingresar el nombre de un macro de usuario (máximo 7 caracteres), si el nombre existe en la lista se presentan tres opciones:

- Probar.- Permite ejecutar paso a paso al macro, cada paso se ejecuta presionando la tecla ENTER y se puede terminar con la prueba en cualquier punto presionando Ctrl-K.
- Correr.- Ejecutar el macro hasta su terminación.
- Borrar.- Borrar el macro.

Si el nombre digitado no es ninguno de los presentados en la lista, se inicia la creación de un macro nuevo. El usuario debe teclear las opciones que se quiere se graben en el macro (no se debe usar el ratón durante la grabación), se termina la grabación del macro presionando nuevamente Ctrl-K. Al salir del depurador este macro será grabado automáticamente en el archivo de macros.

VENTANAS DEL DEPURADOR

Las ventanas que pueden ser accedidas mediante la función <F1> son :
Ventana de Registros, Ventana de Banderas, Ventana de Código del programa, Ventana de la direcciones de memoria, Ventana de Memoria en formato Hexadecimal y la ventana de memoria en formato ASCII.

PANTALLA INICIAL DE PRO486

Línea de Información

Línea de Menú <F2 o *P=Ayuda> « P R O 4 8 6 » J. O. A. T. E P N 1997
 Archivos Infor Depurar Salir Shell Funcion

5263:0100 ADD [SI+BX],AL	5263:0000 CD 20 FF 9F 00 9A F0 FE = f U-
5263:0102 ADD [SI+BX],AL	5263:0008 1D F0 04 8B 4E 1C 6D 01 --+iUm,c
5263:0104 ADD [SI+BX],AL	5263:0010 5C 1A 78 01 5C 1A 3E 1C }-x0->X
5263:0106 ADD [SI+BX],AL	5263:0018 01 01 01 00 02 07 08 FF 000 000
5263:0108 ADD [SI+BX],AL	5263:0020 FF FF FF FF FF FF FF FF
5263:010A ADD [SI+BX],AL	5263:0028 FF FF FF FF 3F 52 9A 01 2R06
5263:0110 ADD [SI+BX],AL	5263:0030 B5 33 14 00 18 00 63 52 ÁST 1 CR
5263:0112 ADD [SI+BX],AL	5263:0038 FF FF FF FF 00 00 00 00
5263:0114 ADD [SI+BX],AL	5263:0040 07 00 00 00 00 00 00 00
5263:0116 ADD [SI+BX],AL	5263:0048 00 00 00 00 00 00 00 00
5263:0118 ADD [SI+BX],AL	5263:0050 CD 21 CB 00 00 00 00 00 =iT
	5263:0058 00 00 00 00 D2 00 D7 18 É I
	5263:0060 65 C4 70 00 16 00 D8 D3 exp - JE

Dirección de la instrucción actual
 Línea Actual
 CS: IP
 S263: 0100
 Código en CS: IP
 00 00

Stack
 IA5C F2F0
 016D 9A00
 1C4E 9FFF
 8804 20CD
 F01D [SP]

Línea de Funciones relativas a archivos
 F2Ayuda F3Coproces F4Pantusr F5Cod F6Asm F7Direc F8Subr F9Dec/Hex F10ASCLII

Ventana de Bandejas (ventana 1)

Bandejas
O D I S Z A P C
0 0 0 0 0 0 0 0

Ventana de Registros (ventana 0)

ES=5263 DS=5263 FS=5263 GS=5263
SS=5263

Memoria

Ventana de Memoria

Información del estado del Depurador

Contenido de la Pila

Línea de Información de la opción de menú actual

Línea de funciones

PANTALLA DEL DEPURADOR CON LA VENTANA DE CODIGO AMPLIADA Y LA VENTANA DEL COPROCESADOR ACTIVA

Opción de menú actual

Nombre del archivo que se está depurando

```

<^O=Menúes> <F2=Ayuda> « P R O 4 8 6 » J. O. A. T. E P N 1997
.. Archivo Infor Depurar, Salir sHell eXe Función
Dirección Instrucción «prucati.exe»
41D8:0010 MOV AX,41F7
41D8:0013 MOV DS,AX
41D8:0015 MOV ES,AX
41D8:0017 CHP [0006],01
41D8:001C JNZ 001E
41D8:001E MOV AH,03
41D8:0020 INT 10
41D8:0022 PUSH CX
41D8:0023 PUSH AX
41D8:0024 PUSH BX
41D8:0025 MOV AX,0000
41D8:0028 MOV [0017],AX
41D8:002B INT 33

```

Banderas

O D I S Z A P C

0 0 0 0 0 0 0 0

FS=0000 GS=0000

Coprocesador Matemático

Tag=FFFF Ipt=00000 Opcode=000 Optr=00000

Stack Exp	Mantisa	Valor
ST(0)		
ST(1)		
ST(2)		
ST(3)		
ST(4)		
ST(5)		
ST(6)		
ST(7)		

PCont 037F IH DH 2H 0H UM PH PC RC IC

1 1 1 1 1 1 3 0 0

PStat 0000 IE DE ZE OE UE PE IR CO CI CD ST CS

0 0 0 0 0 0 0 0 0 0 0

Línea Actual

CS	IP
41D8	: 0010

Código en CS:IP

B8 FC 41

Stack

00AF ST(7)

00AE PCont

00AD 1 1 1 1 1 1 3 0 0

00AB PStat

0007 0 0 0 0 0 0 0 0 0 0 0

```

F1Edit F2Ayuda F3Coproc F4PntUsr F5Cod F6Asm F7Direc F8NoSubr F9Dec/Hex F10ASCII
Permite cargar nuevo programa a depurar (.EXE ó .COM) Normal

```

Ventana de Código :

En la ventana de código no se pueden hacer modificaciones, se ingresa a esta ventana para ver las instrucciones del programa que no se encuentran en la ventana actual. Para recorrer trece instrucciones hacia abajo o arriba se debe presionar PAGE DOWN o PAGE UP, o con el ratón (por cada click se recorre trece instrucciones) sobre la barra de desplazamiento. El desplazamiento hacia arriba solo se puede hacer hasta llegar a la instrucción actualmente remarcada. Otra forma que permite acceder a cualquier dirección es escribiendo directamente en el recuadro respectivo el segmento o el desplazamiento, en este recuadro aparece el mensaje "Ingrese el valor (hexadecimal)", inicialmente esta listo para escribir la dirección de *desplazamiento*, para intercambiar entre escribir la dirección de *segmento* y la dirección de *desplazamiento* se presiona las teclas Flecha Izquierda y Flecha Derecha alternadamente. Se puede escribir una dirección de segmento de código diferente a la actual.

Cada línea de instrucción de la ventana de código consta de : Dirección de Segmento de código (2 bytes), dirección de desplazamiento (2 bytes), Nombre de la instrucción, Operandos de la instrucción y un carácter que indica el tipo de operandos con los que trabaja la instrucción, de la siguiente forma :

Símbolo	Significado
B	Operando tipo Byte (8 bits)
W	Operando tipo Palabra (16 bits)
D	Operando tipo Doble Palabra (32 bits)
Q	Operando tipo Cuádruple Palabra (64 bits)
T	Operando Temporal (80 bits), usado solamente por las instrucciones del coprocesador matemático.
Espacio en Blanco	Tamaño de operando no definido

Ventana de Registros :

Esta ventana contiene todos los registros de propósito general extendidos y los registros de segmento del procesador, incluidos el apuntador de instrucción EIP y el registro de segmento de código CS. Se accesa a esta ventana presionando <F1> o haciendo click sobre ella, se pueden cambiar los valores de cualquiera de los registros de la ventana. Para moverse entre registros se usa las flechas (flecha arriba, abajo, a la derecha y a la izquierda), o con la barra de desplazamiento usando el ratón. Una vez que se encuentre remarcado el registro al que se desea editar, se escribe el valor en el recuadro que tiene el mensaje "Ingrese el valor en hexadecimal", luego de ingresado el valor se presiona ENTER para aceptar el cambio o ESC para no ejecutar el cambio. Los registros son :

REGISTRO	USO
EAX, EBX, ECX, EDX	Registros de propósito general
EDI, ESI	Registros índice destino y índice fuente respectivamente
EIP	Registro apuntador de instrucciones
ESP	Registro apuntador de pila
EBP	Registro de base
EFL	Registro de banderas (contiene los bits de las banderas mostradas en la ventana de banderas)
DS, ES, GS, FS	Registros de segmento de datos
CS	Registro de segmento de Código
SS	Registro de segmento de Pila

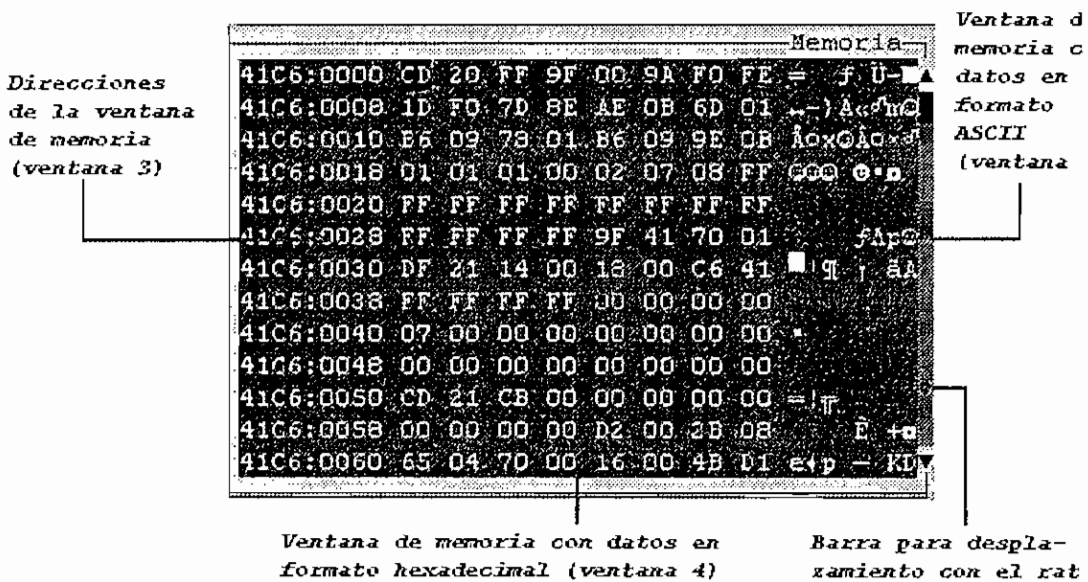
Ventana de Banderas :

Contiene todas las banderas del Procesador, el contenido de las banderas puede ser complementado ubicándose sobre la bandera y luego presionando ENTER. El significado de cada bandera se indica a continuación :

BANDERA	SIGNIFICADO
C	Bandera de acarreo. Indica acarreo en las instrucciones aritméticas, también usada para indicar error después de ejecutar una interrupción (generalmente en las interrupciones DOS)
P	Bandera de paridad (número par de bits)
A	Bandera auxiliar. Indica si hay necesidad de ajuste en las operaciones aritméticas con números BCD
Z	Bandera de resultado cero o de igualdad en una comparación
S	Bandera de resultado o comparación negativa
I	Bandera de interrupción. Habilita o inhabilita la entrada de interrupción externa INTR.
D	Bandera de dirección. Controla la dirección en las operaciones con caracteres incrementando o decrementando automáticamente los registros índices (ESI y EDI)
O	Bandera de desbordamiento (aritmético)

Ventana de Memoria :

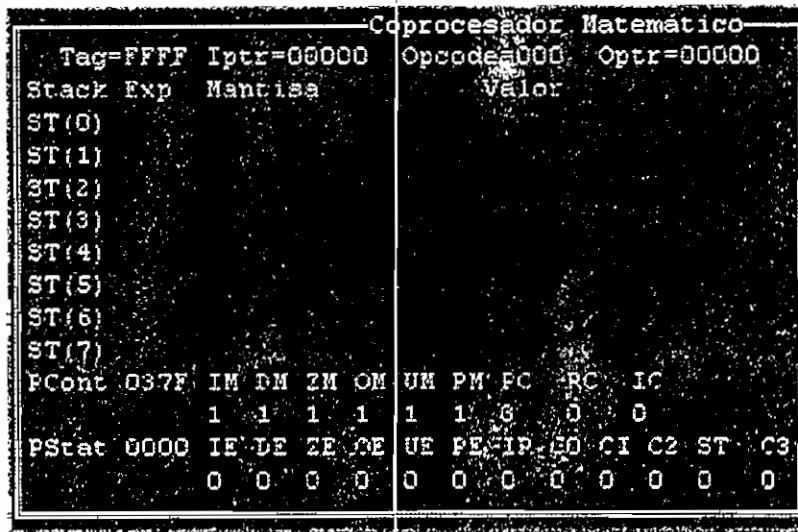
Ventana de Memoria



Consta de tres partes : ventana de dirección de memoria, ventana de memoria en formato hexadecimal y ventana de memoria en formato ASCII. Cada ventana se activa presionando F1 y TAB consecutivamente hasta llegar a la ventana deseada, o haciendo click sobre la ventana. La ventana de dirección permite cambiar el segmento y/o el desplazamiento de la porción de memoria que se muestra en las ventanas de datos de memoria. La ventana de datos en formato Hexadecimal y ASCII permiten variar directamente el contenido de memoria con valores Hexadecimal o ASCII respectivamente. Para moverse dentro de las ventana de datos de memoria se usan las teclas de las flechas. Con el ratón en la barra de desplazamiento se puede mover hacia arriba o hacia abajo las localidades de memoria, en el segmento actual.

Ventana del Coprocesador Matemático :

Ventana del Coprocesador Matemático



La ventana del coprocesador matemático contiene la información de todos los registros del mismo. A continuación se explica la información que contiene cada parte de esta ventana :

- TAG (16 bits) : Dividida en grupos de 2 bits por cada registro de la pila, los 2 bits menos significativos corresponden al registro superior de la pila (referidos a la posición física de los registros), y contiene información acerca del dato contenido en el registro de la siguiente forma:

VALOR SIGNIFICADO

00	Valor válido
01	Valor Cero
10	Valor Infinito, inválido o sin formato
11	Registro Vacío

IPTR (20 bits) : Indica la dirección absoluta de la última instrucción que ejecutó el coprocesador matemático.

- **OPTR (20 bits)** : Indica la dirección absoluta de la localidad de memoria de la computadora con la que el coprocesador matemático intercambio información en la última instrucción que ejecutó.
- **OPCODE (12 bits)** : Indica el código simplificado de la instrucción que ejecutó el coprocesador. Cada instrucción del coprocesador se define completamente con un mínimo de 2 bytes (16 bits), los 5 bits más significativos siempre tienen el valor "11011", por lo que en la ventana solo se muestra el valor de los 11 bits menos significativos. **ST(0) a ST(7)** : Corresponden a los registros de la pila en los que se almacenan los operandos sobre los que el coprocesador realiza las operaciones. **ST(0)** o simplemente **ST** es el tope o parte superior lógica de la pila, que puede ser cualquiera de los 8 registros físicos de la pila. Esta ventana contiene información del exponente (*EXP*), mantisa (*MANTISA*), para el formato real temporal de la IEEE, y se muestra también el valor en formato decimal (*VALOR*).
- **PCONT** : Contiene la información de la palabra de control, primero se muestra el valor de la palabra completa y luego se muestra el valor de ciertos bits importantes en forma independiente. Ver Capítulo 2.3 para mayor infoemación.
- **PSTAT** : Palabra de estado del coprocesador, refleja la condición global del coprocesador. La palabra de estado está dividida en dos campos: Campo bit de indicador de excepción y campo bit de estado. El indicador **ST** determina cual de los registros físicos de la pila es actualmente el que está en el tope (parte superior lógica de la pila). Ver capítulo 2.3.

APENDICE B

**RESUMEN DEL CONJUNTO DE INSTRUCCIONES
DEL
MICROPROCESADOR 80486**

Las instrucciones para los 8086-80286 son en el modo de 16 bits que tiene la configuración que se ilustra en la figura B.1 (a). Estas instrucciones son compatibles con los microprocesadores 80386 y 80486, si están inicializados para utilizar formato de instrucciones de 16 bits (modo real). En el modo protegido, el byte superior del descriptor del 80386 y 80486 contiene el bit D que selecciona el modo de instrucción de 16 o de 32 bits. En esos microprocesadores se utilizan también instrucciones en modo de 32 bits con la configuración que se ilustra en la figura B.1 (b). Esas instrucciones pueden estar en el modo de instrucción de 16 bits, mediante el empleo de prefijos.

Los dos primeros bytes del modo de instrucción de 32 bits, se llaman prefijos de cambio, porque no siempre están presentes. El primer byte modifica el tamaño de la dirección utilizada por la instrucción y el segundo modifica el tamaño del registro. El segundo byte de prefijo cambia el tamaño de registro y del operando de 16 a 32 bits (si está en modo real) o viceversa (si está en modo protegido), para la instrucción con prefijo.

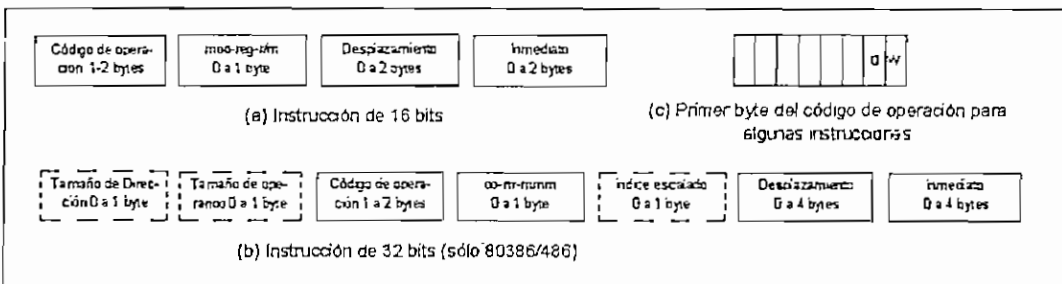


Figura B.1 Los formatos de las instrucciones 8086-80486 (a) forma de 16 bits; (b) forma de 32 bits; (c) Los dos primeros bits del primer byte del código de operación de algunas instrucciones

En muchas instrucciones el primer bit (denominado w) del primer byte del código de operación, indica si w=0 que los operandos son de 8 bits, si w=1 indica que los operandos son de 16 o 32 bits. El segundo bit del mismo byte indica: si d=1

los datos fluyen del campo *mmm* (también representado como *r/m*) al campo de registro *rrr* (también conocido como *reg*). Si el bit *d=0*, los datos fluyen del campo *rrr* al campo *mmm*.

El resumen del conjunto de instrucciones, que continúa después de esta introducción, contiene una lista completa de todas las instrucciones para los microprocesadores 8086, 8088, 80286, 80386 y 80486.

Cada elemento de la instrucción mencionará el código nemotécnico de operación más una breve descripción del propósito de la instrucción. También se menciona los códigos binarios en el lenguaje de máquina para cada instrucción mas cualquier otra información requerida para formar la instrucción tal como el desplazamiento o la información inmediata. Al lado de la versión binaria de la instrucción en el lenguaje de máquina, aparecerán los bits del registro de la bandera así como cualquier cambio que pueda ocurrir para una instrucción específica. En esta lista, un espacio blanco indicará que no hay cambio, un “?” indica un cambio con un resultado no predecible, un “*” indica un cambio predecible un 1 indica que la bandera está establecida y un 0 indica que la bandera fué borrada.

Antes de empezar la lista de instrucciones se requiere algo de información a cerca de la configuración de los bits de las versiones binarias de las instrucciones en el lenguaje de máquina. La tabla B-1 muestra los bits del modificador, codificados como *oo* (conocido también como *mod*) en las listas de las instrucciones para que estas puedan ser formadas con un registro, desplazadas o sin desplazar.

<i>oo</i>	Función
00	Si <i>mmm=110</i> , entonces un desplazamiento sigue al código de operación; de otra manera, ningún desplazamiento es utilizado
01	Un desplazamiento identificado de 8-bits sigue al código de operación
10	Un desplazamiento identificado de 16-bits sigue al código de operación
11	<i>mmm</i> especifica un registro, en vez de a un modo de dirección

TABLA B-1 Los bits del modificador, codificados como *mod* en la lista de instrucciones

La tabla B-2 menciona con el campo de registro/memoria los modos permitidos para direccionar la memoria, codificados como *mmm*. Esta tabla es aplicable a todas las versiones del microprocesador.

<i>mmm</i>	Función
000	DS:[BX + SI]
001	DS:[BX + DI]
010	SS:[BP + SI]
011	SS:[BP + DI]
100	DS:[SI]
101	DS:[DI]
110	SS:[BP]
111	DS:[BX]

TABLA B-2 Descripción del campo de registro/memoria (*mmm*)

<i>rrr</i>	w=0	w=1	reg32
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

TABLA B-3 Opciones del registro para campo (*rrr*)

La tabla B-3 menciona las opciones del registro (*rrr*) cuando está codificado para un registro de 8-bits o 16-bits. Esta tabla también menciona los registros de 32 - bits utilizados con los microprocesadores 80386 y 80486.

La tabla B-4 menciona las asignaciones bits del registro de segmento (*rrr*) para las instrucciones MOV, PUSH y POP, las cuales utilizan estos registros de segmento.

Cuando se utilizan los microprocesadores 80386 y 80486, algunas de las definiciones proporcionadas en las tablas anteriores cambian. Refiérase a las tablas B-5 y B-6 para estos cambios y cómo se aplican los microprocesadores 80386 y 80486.

El resumen del conjunto de instrucciones que sigue menciona todas las instrucciones, con ejemplos, para los microprocesadores 8086, 8088, 80286, 80386 y 80486. Faltan los prefijos para cambiar los segmentos: CS (2EH), SS (36H), DS (3EH), ES (26H), FS (64H) y GS (65H). Estos prefijos son de un byte de largo y colocados en la memoria antes de la instrucción que tiene el prefijo.

En los 8088 y 8086, el valor de la dirección efectiva (*ea* o *a*) debe ser sumado a las instrucciones que trabajan con operandos de memoria. Un desplazamiento es cualquier operando de memoria directa o una constante, o cualquier combinación de las dos. La tabla B.7 muestra el número de ciclos de reloj a sumar para la dirección efectiva. Por ejemplo, la instrucción de `ADD DATA, AL` del 8086 $16+ea$ ciclos de reloj. Debido a que la tabla menciona que las direcciones desplazadas requieren 6 ciclos de reloj, esta instrucción requiere de 22 ciclos de reloj para ejecutar. Observe que todos los tiempos mencionados son los máximos y que en algunos casos el microprocesador puede ejecutar en menos tiempo.

<i>rrr</i>	Registro de segmento
000	ES
001	CS
010	SS
011	DS
100	FS
101	GS

TABLA B-4 Asignaciones del registro para campo (*rrr*) que son usadas para representar los registros de segmentos

ooo	mmm	rrr	Función
00	000	---	DS:[EAX]
00	001	---	DS:[ECX]
00	010	---	DS:[EDX]
00	011	---	DS:[EBX]
00	100	000	DS:[EAX+índice escalonado]
00	100	001	DS:[ECX+índice escalonado]
00	100	010	DS:[EDX+índice escalonado]
00	100	011	DS:[EBX+índice escalonado]
00	100	100	SS:[ESP+índice escalonado]
00	100	101	DS:[disp32+índice escalonado]
00	100	110	DS:[ESI+índice escalonado]
00	100	111	DS:[EDI+índice escalonado]
00	101	---	DS:DISP32
00	110	---	DS:[ESI]
00	111	---	DS:[EDI]
01	000	---	DS:[EAX+disp8]
01	001	---	DS:[ECX+disp8]
01	010	---	DS:[EDX+disp8]
01	011	---	DS:[EBX+disp8]
01	100	000	DS:[EAX+índice escalonado + disp8]
01	100	001	DS:[ECX+índice escalonado + disp8]
01	100	010	DS:[EDX+índice escalonado + disp8]
01	100	011	DS:[EBX+índice escalonado + disp8]
01	100	100	DS:[ESP+índice escalonado + disp8]
01	100	101	DS:[EBP+índice escalonado + disp8]
01	100	110	DS:[ESI+índice escalonado + disp8]
01	100	111	DS:[EDI+índice escalonado + disp8]
01	101	---	SS:[EBP + disp8]
01	110	---	DS:[ESI + disp8]
01	111	---	DS:[EDI + disp8]
10	000	---	DS:[EAX + disp32]
10	001	---	DS:[ECX + disp32]
10	010	---	DS:[EDX + disp32]
10	011	---	DS:[EBX + disp32]
10	100	000	DS:[EAX+índice escalonado+disp32]
10	100	001	DS:[ECX+índice escalonado+disp32]
10	100	010	DS:[EDX+índice escalonado+disp32]
10	100	011	DS:[EBX+índice escalonado+disp32]
10	100	100	SS:[ESP+índice escalonado+disp32]
10	100	101	SS:[EBP+índice escalonado+disp32]
10	100	110	DS:[ESI+índice escalonado+disp32]
10	100	111	DS:[EDI+índice escalonado+disp32]
01	101	---	SS:[EBP + disp32]
01	110	---	DS:[ESI + disp32]
01	111	---	DS:[EDI + disp32]

TABLA B-5 Las combinaciones posibles de *ooo*, *mmm* y *rrr* para el conjunto de instrucciones del 80386 y 80486 utilizando el modo de direccionamiento de 32 bits

rrr	Registro para Índice
000	EAX
001	ECX
010	EDX
011	EBX
100	Sin índice
101	EBP
110	ESI
111	EDI

TABLA B-6 Los registros para índice están especificados con rrr en el microprocesador 80386 y 80486

El bit-D, en el descriptor del segmento del código, indica el tamaño predeterminado del operando y la dirección para los microprocesadores 80386 y 80486. Si D=1, entonces todas las direcciones y operandos son de 32 bits y si D=0, todas las direcciones y operandos son de 16 bits. En el modo real, el bit D será establecido en cero por los microprocesadores 80386 y 80486, así que los operandos y las direcciones son de 16 bits.

El prefijo del tamaño de la dirección (67H) se debe colocar antes de las instrucciones en el 80386 y 80486 para cambiar el tamaño predeterminado como fué seleccionado por el bit D.

TIPO	CICLOS DE RELOJ	EJEMPLO
Base o índice	5	MOV CL,[DI]
Desplazamiento	6	MOV AL,DATA
Base más índice [BP + DI] o [BX + SI]	7	MOV BL,[BP+DI]
Base más índice [BP + SI] o [BX + DI]	8	MOV CL,[BP+SI]
Desplazamiento más base o índice	9	MOV DH,[DI+20H]
Base más índice más desplazamiento [BP+DI+disp] o [BX+SI+disp]	11	MOV CX,DATA[BX+SI]
Base más índice más desplazamiento [BP+SI+disp] o [BX+DI+disp]	12	MOV CX,[BX+DI+2]
Prefijo para evitar el segmento	ea+2	MOV AL,ES:DATA

TABLA B-7 Cálculos de dirección efectivos para los microprocesadores 8086 y 8088

Por ejemplo, la instrucción `MOV AX, [ECX]` debe tener el prefijo del tamaño de la dirección colocado antes en el código de la máquina si el tamaño predeterminado es de 16 bits. Si el tamaño predeterminado es de 32 bits, el prefijo de la dirección no es necesario con esta instrucción. El prefijo para modificar operando (66H) funciona de la misma manera que el prefijo para el tamaño de la dirección. En el ejemplo anterior, el tamaño del operando es de 16-bits. Si el bit-D selecciona operandos y direcciones de 32-bits, esta instrucción requiere al prefijo para el tamaño del operando.

Algunas instrucciones repetitivas, tienen un número de ciclos de reloj base más un número de ciclos de reloj por cada iteración. El símbolo "n" que se presenta en estas instrucciones repetitivas, en la fórmula para el cálculo de los ciclos de reloj, representa el número de iteraciones. Por ejemplo la instrucción `REPE SCAS` para el 80486 tiene la fórmula $7 + 5n$ para calcular los ciclos de reloj, donde 7 es el número base, 5 es el número de ciclos de reloj por cada iteración y n representa el número de iteraciones. El tiempo en nanosegundos de cada ciclo de reloj se obtiene dividiendo un microsegundo para el número de Mhz a la que está funcionando el microprocesador.

AAA		Ajuste de ASCII después de sumar	
00110111		O D I T S Z A P C	
Ejemplo		? ? ? * ? *	Relojes
AAA	8086	8	
	8088	8	
	80286	3	
	80386	4	
	80486	3	
AAD		Ajuste de ASCII antes de dividir	
11010101 00001010		O D I T S Z A P C	
Ejemplo		? * * ? * ?	Relojes

AAD		8086	60
		8088	60
		80286	14
		80386	19
		80486	14
AAM Ajuste de ASCII después de multiplicar			
11010100 00001010		O D I T S Z A P C ? ? ? ? ? ? ? ?	
Ejemplo		Relojes	
AAM		8086	83
		8088	83
		80286	16
		80386	17
		80486	15
AAS Ajuste de ASCII después de restar			
00111111		O D I T S Z A P C ? ? ? ? ? ? ? ?	
Ejemplo		Relojes	
AAS		8086	8
		8088	8
		80286	3
		80386	4
		80486	3
ADC Sumar con restante			
000100dw oorrmmmm disp		O D I T S Z A P C * * * * * * * * * *	
Formato	Ejemplos	Relojes	
ADC reg,reg.	ADC AX,BX	8086	3
	ADC AL,BL	8088	3
	ADC EAX,EBX	80286	2
	ADC CX,SI	80386	2
	ADC ESI,EDI	80486	1

ADC mem,reg	ADC DATA,AL ADC LIST,S] ADC DATA [DI],CL ADC [EAX],BL ADC [EBX÷2*ECX],EDX	8086	16+ea
		8088	24+ea
		80286	7
		80386	7
		80486	3
ADC reg,mem	ADC BL,DATA ADC SI,LIST ADC CL,DATA [DI] ADC CL,[EAX] ADC EDX,[EBX+100H]	8086	9+ea
		8088	13+ea
		80286	7
		80386	6
		80486	2
100000sw 00010mmm disp información			
Formato	Ejemplos	Relojes	
ADC reg,imm	ADC CX,3 ADC DI,1AH ADC DL,34H ADC EAX,12345 ADC CX,1234H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
ADC mem,imm	ADC DATA,33 ADC LIST,'A' ADC DATA [DI],2 ADC BYTE PTR [EAX],3 ADC WORD PTR [DI],669H	8086	17+ea
		8088	23+ea
		80286	7
		80386	7
		80486	3
0001010w información			
Formato	Ejemplos	Relojes	
ADC acc,imm	ADC AX,3 ADC AL,1AH ADC AH,34 ADC EAX,3 ADC AL,'Z'	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

ADD Sumar		O D I T S Z A P C * * * * *	
Formato	Ejemplos	Relojes	
00000dw oorrmmm disp	ADD AX,BX ADD AL,BL ADD EAX,EBX ADD CX,SI ADD ESI,EDI	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
ADD mem,reg	ADD DATA,AL ADD LIST,SI ADD DATA [DI],CL ADD [EAX],CL ADD [EBX+4*EDX],EBX	8086	16+ea
		8088	24+ea
		80286	7
		80386	7
		80486	3
ADD reg,mem	ADD BL,DATA ADD SI,LIST ADD CL,DATA [DI] ADD CL,[EAX] ADD EDX,[EBX+200H]	80286	9+ea
		8088	13+ea
		80286	7
		80386	6
		80486	2
10000sw oo000mmm disp información			
Formato	Ejemplos	Relojes	
ADD reg,imm	ADD CX,3 ADD DI,1AH ADD DL,34H ADD EAX,123456 ADD CX,18AFH	80286	4
		8088	4
		80286	3
		80386	2
		80486	1
ADD mem,imm	ADD DATA,33 ADD LIST,A ADD DATA [DI],2 ADD BYTE PTR [EAX],3 ADD WORD PTR [DI],6A8H	8086	17+ea
		8088	23+ea
		80286	7
		80386	7
		80486	3

0000010w información			
Formato	Ejemplos	Relojes	
ADD acc,imm	ADD AX,3 ADD AL,1AH ADD AH,56 ADD EAX,3 ADD AL,'D'	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

AND AND lógico

001000aw oorrmmm disp		O D I T S Z A P C 0 ? 0	
Formato	Ejemplos	Relojes	
AND reg,reg	AND CX,BX AND DL,BL AND ECX,EBX AND BP,SI AND EDX,EDI	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
AND mem,reg	AND BIT,CH AND LIST,DI AND DATA [BX],CL AND [ECX],AL AND [EDX+8*ECX],EDI	8086	16+ea
		8088	24+ea
		80286	7
		80386	7
		80486	3
AND reg,mem	AND BL,DATA AND SI,LIST AND CL,DATA [DI] AND CL,[EAX] AND EDX,[EBX+34AH]	8086	9+ea
		8088	13+ea
		80286	7
		80386	6
		80486	2

100000sw oo100mmm disp información			
Formato	Ejemplos	Relojes	
AND reg,imm	AND BP,1 AND DI,10H AND DL,34H AND EBP,12345 AND SP,1234H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

AND mem,imm	AND DATA,33 AND LIST,4 AND DATA [SI],2 AND BYTE PTR [EAX],3 AND DWORD PTR [DI],32	8086	17+ea
		8088	23+ea
		80286	7
		80386	7
		80486	3

0010010w información

Formato	Ejemplos	Relojes	
AND acc,imm	AND AX,15 AND AL,1FH AND AH,34 AND EAX,3 AND AL,'R'	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

ARPL Ajustar el nivel de privilegio requerido

01100011 oorrmmm disp O D I T S Z A P C

Formato	Ejemplos	Relojes	
ARPL reg,reg	ARPL AX,BX ARPL BX,SI ARPL CX,DX ARPL BX,AX ARPL DI,SI	8086	—
		8088	—
		80286	10
		80386	20
		80486	9
ARPL mem,reg	ARPL NUMB,AX ARPL LIST,DI ARPL DATA [BX],CX ARPL [ECX],AX ARPL [EDX+4*ECX],DI	8086	—
		8088	—
		80286	11
		80386	21
		80486	9

BOUND Revisar límites de tabla

01100010 oorrmmm disp O D I T S Z A P C

Formato	Ejemplos	Relojes	
BOUND	BOUND AX,BETS	8086	—

reg,mem	BOUND BX,SAID BOUND CX,DATA BOUND BX,[DI] BOUND DI,[BX÷2]	8088	—
		80286	13
		80386	10
		80486	7
BSF Verificación delantera de bit			
00001111 10111100 oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
BSF reg,reg	BSF AX,BX BSF BX,SI BSF ECX,EBX BSF EBX,EAX BSF DI,SI	8086	—
		8088	—
		80286	—
		80386	10+3n
		80486	6+2
BSF reg,mem	BSF AX,DATA BSF BP,LISTG BSF ECX,MEMORY BSF EAX,DATA6 BSF DI,[ECX]	8086	—
		8088	—
		80286	—
		80386	10+3n
		80486	7+43
BSR Verificación inversa de bit			
00001111 10111101 oorrmmm disp:		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
BSR reg,reg	BSR AX,BX BSR BX,SI BSR ECX,EBX BSR EBX,EAX BSR DI,SP	8086	—
		8088	—
		80286	—
		80386	10+3n
		80486	6+103
BSR reg,mem	BSR AX,DATA BSR BP,LISTG BSR ECX,MEMORY BSR EAX,DATA6 BSR DI,[EBX]	8086	—
		8088	—
		80286	—
		80386	10+3n
		80486	7+104

BSWAP Intercambio de bytes

00001111 11001rrr		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
BSWAP reg	BSWAP EAX BSWAP EBX BSWAP ECX BSWAP EDX BSWAP EDI	8086	---
		8088	---
		80286	---
		80386	---
		80486	1

BT Prueba de bits

00001111 10111010 oo100mmm disp información		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
BT reg,imm8	BT AX,2 BT CX,4 BT BP,10H BT CX,8 BT BX,2	8086	---
		8088	---
		80286	---
		80386	3
		80486	3
BT mem,imm8	BT DATA,2 BT UST,2 BT DATA [DI],2 BT [BX],1 BT FROG,3	8086	---
		8088	---
		80286	---
		80386	6
		80486	3
00001111 10100011 disp.			
Formato	Ejemplos	Relojes	
BT reg,reg	BT AX,CX BT CX,DX BT BP,AX BT SI,CX BT CX,BP	8086	---
		8088	---
		80286	---
		80386	3
		80486	3

BT mem,reg	BT DATA1,AX BT LIST,DX BT DATA3,CX BT DATA9,BX BT DATA [DI],AX	8086	—
		8088	—
		80286	—
		80386	12
		80486	8

BTC Prueba de bits y complemento

00001111 10111010 00111mmm disp información		O D I T S Z A P C *	
Formato	Ejemplos	Relojes	
BTC reg,imm8	BTC AX,2 BTC CX,4 BTC BP,10H BTC CX,8 BTC BX,2	8086	—
		8088	—
		80286	—
		80386	5
		80486	5
BTC mem,imm8	DATA1,2 BTC LIST,2 BTC DATA [DI],3 BTC [BX],1 BTC TOAD,5	8086	—
		8088	—
		80286	—
		80386	8
		80486	8

00001111 10111011 disp		Relojes	
Formato	Ejemplos	Relojes	
BTC reg,reg	BTC AX,CX BTC CX,DX BTC BP,AX BTC SI,CX BTC CX,BX	8086	—
		8088	—
		80286	—
		80386	6
		80486	6
BTC mem,reg	BTC DATA1,AX BTC LIST,DX BTC DATA3,CX BTC DATA9,BX BTC DATA [DI],AX	8086	—
		8088	—
		80286	—
		80386	13
		80486	13

BTR Prueba de bits y arranque nuevo

00001111 10111010 00110mmm disp información		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
BTR reg,imm8	BTR AX,2 BTR CX,4 BTR BP,10H BTR CX,8 BTR BX,2	8086	---
		8088	---
		80286	---
		80386	6
		80486	6
BTR mem,imm8	BTR DATA1,2 BTR LIST,2 BTR DATA [DI],4 BTR [BX],1 BTR SLED,6	8086	---
		8088	---
		80286	---
		80386	8
		80486	8
00001111 10110011 diso			
Formato	Ejemplos	Relojes	
BTR reg,reg	BTR AX,CX BTR CX,DX BTR BP,AX BTR SI,CX BTR BP,CX	8086	---
		8088	---
		80286	---
		80386	6
		80486	6
BTR mem,reg	BTR DATA1,AX BTR LIST,DX BTR DATA3,CX BTR DATA9,BX BTR DATA [BX],AX	8086	---
		8088	---
		80286	---
		80386	13
		80486	13
BTS Prueba de bits y establecer			
00001111 10111010 00101mmm disp información		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
BTS reg,imm8	BTS AX,2 BTS CX,4 BTS BP,10H BTS CX,8 BTS BX,3	8086	---
		8088	---
		80286	---
		80386	6
		80486	6

BTS mem,imm8	BTS DATA1,2	8086	—
	BTS LIST,2	8088	—
	BTS DATA [BP],7	80286	—
	BTS [BX],1	80386	8
	BTS FROG,3	80486	8

00001111 10101011 disp

Formato	Ejemplos	Relojes	
BTS reg,reg	BTS AX,CX	8086	—
	BTS CX,DX	8088	—
	BTS BP,AX	80286	—
	BTS SI,CX	80386	6
	BTS CX,BP	80486	6
BTS mem,reg	BTS DATA1,AX	8086	—
	BTS LIST,DX	8088	—
	BTS DATA3,CX	80286	—
	BTS DATA9,BX	80386	13
	BTS DATA [BP],AX	80486	13

CALL Procedimiento de llamar otro programa (subrutina)

11101000 disp O D I T S Z A P C

Formato	Ejemplos	Relojes	
CALL label (near)	CALL FOR FUN	8086	19
	CALL HOME	8088	23
	CALL ET	80286	7
	CALL WAITING	80386	3
	CALL SOMEONE	80486	3

10011010 disp

Formato	Ejemplos	Relojes	
CALL label (far)	CALL FAR PTR DATES	8086	28
	CALL WHAT	8088	36
	CALL WHERE	80286	13
	CALL FARCE	80386	17
	CALL WHOM	80486	18

11111111 oo010mmm			
Formato	Ejemplos	Relojes	
CALL reg (near)	CALL AX CALL BX CALL CX CALL DI CALL SI	8086	16
		8088	20
		80286	7
		80386	7
		80486	5
CALL mem (near)	CALL ADDRESS CALL NEAR PTR [DI] CALL DATA CALL FROG CALL HERO	8086	21 ÷ ea
		8088	29 + ea
		80286	11
		80386	10
		80486	5

11111111 oo011mmm			
Formato	Ejemplos	Relojes	
CALL mem (far)	CALL FAR LIST [SI] CALL FROM HERE CALL TO THERE CALL SIXX CALL OCT	8086	37 ÷ ea
		8088	53 + ea
		80286	16
		80386	22
		80486	17

CBW Convertir byte a palabra

10011000		O D I T S Z A P C	
Ejemplo		Relojes	
CBW		8086	2
		8088	2
		80286	2
		80386	3
		80486	3

CDQ Convertir doble palabra a cuádruple palabra

10011001	O D I T S Z A P C	
Ejemplo	Relojes	
CDQ	8086	—
	8088	—
	80286	—
	80386	2
	80486	2

CLC Borrar bandera de restante

11111000	O D I T S Z A P C 0	
Ejemplo	Relojes	
CLC	8086	2
	8088	2
	80286	2
	80386	2
	80486	2

CLD Borrar bandera de dirección

11111100	O D I T S Z A P C 0	
Ejemplo	Relojes	
CLD	8086	2
	8088	2
	80286	2
	80386	2
	80486	2

CLI Borrar bandera de interrupción

11111010	O D I T S Z A P C 0	
Ejemplo	Relojes	
CLI	8086	2
	8088	2
	80286	3
	80386	3
	80486	5

CLTS Borrar bandera de conmutador de tareas		
00001111 00000110	O D I T S Z A P C	
Ejemplo	Relojes	
CLTS	8086	—
	8088	—
	80286	2
	80386	5
	80486	7

CMC Complementar bandera de restante		
10011000	O D I T S Z A P C	
Ejemplo	Relojes	
CMC	8086	2
	8088	2
	80286	2
	80386	2
	80486	2

CMP Comparar operandos			
001110dw oorrmmmm disp	O D I T S Z A P C		
Formato	Ejemplos	Relojes	
CMP reg,reg	CMP AX,BX CMP AL,BL CMP EAX,EBX CMP CX,SI CMP ESI,EDI	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
CMP mem,reg	CMP DATA,AL CMP LIST,SI CMP DATA [BX],CL CMP [EAX],AL CMP [EBX+2*ECX],EBX	8086	9+ea
		8088	13+ea
		80286	7
		80386	5
		80486	2

CMP reg,mem	CMP ESI,DATA CMP SI,LIST CMP CL,DATA [DI] CMP CL,[EAX] CMP EDX,[EBX+200H]	8086	9+ea
		8088	13+ea
		80286	5
		80386	6
		80486	2

10000sw 00111mmm disp información

Formato	Ejemplos	Relojes	
CMP reg,imm	CMP CX,3 CMP DI,1AH CMP DL,34H CMP EBX,12345 CMP CX,123AH	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
CMP mem,imm	CMP DATA,33 CMP LIST,'A' CMP DATA [DI],87H CMP BYTE PTR [EAX],3 CMP WORD PTR [SI],7	8086	10+ea
		8088	14+ea
		80286	6
		80386	5
		80486	2

0011110w información

Formato	Ejemplos	Relojes	
CMP acc,imm	CMP AX,3 CMP AL,1AH CMP AH,34 CMP EAX,3 CMP AL,'I'	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

CMPS Comparar cadenas

1010011w		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
CMPSB CMPSW CMPSD	CMPSB CMPSW CMPSD CMPS DATA1 REPE CMPSB REPNE CMPSW	8086	22
		8088	30
		80286	8
		80386	10
		80486	8

CMPXCHG Comparar e intercambiar			
00001111 1011000w 11rrrrr		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
CMPXCHG reg,reg	CMPXCHG EAX,EBX CMPXCHG ECX,EDX	8086	—
		8088	—
		80286	—
		80386	—
		80486	6
00001111 1011000w 00rrrrmm			
Formato	Ejemplos	Relojes	
CMPXCHG mem,reg	CMPXCHG DATA,EAX CMPXCHG DATA2,EBX	8086	—
		8088	—
		80286	—
		80386	—
		80486	7
CWD Convertir palabra a doble palabra			
10011001		O D I T S Z A P C	
Ejemplo		Relojes	
CWD		8086	5
		8088	5
		80286	2
		80386	2
		80486	3
CWDE Convertir palabra a doble palabra extendida			
10011000		O D I T S Z A P C	
Ejemplo		Relojes	
CWDE		8086	—
		8088	—
		80286	—
		80386	3
		80486	3

DAA		Ajuste de decimales después de sumar	
00100111		O D I T S Z A P C	
Ejemplo		? * * * * *	Relojes
DAA		8086	4
		8088	4
		80286	3
		80386	4
		80486	2
DAS		Ajuste de decimales después de restar	
00101111		O D I T S Z A P C	
Ejemplo		? * * * * *	Relojes
DAS		8086	4
		8088	4
		80286	3
		80386	4
		80486	2
DEC		Decremento	
1111111w oo001mmm disp		O D I T S Z A P C	
Formato	Ejemplos	* * * * *	Relojes
DEC reg8	DEC BL DEC BH DEC CL DEC DH DEC AH	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
DEC mem	DEC DATA DEC LIST DEC DATA [SI] DEC BYTE PTR [EAX] DEC WORD PTR [DI]	8086	15+ea
		8088	23+ea
		80286	7
		80386	6
		80486	3

01001rrr			
Formato	Ejemplos	Relojes	
DEC reg16 DEC reg32	DEC AX DEC EAX DEC CX DEC EBX DEC DI	8086	3
		8088	3
		80286	2
		80386	2
		80486	1

DIV División sin identificación

1111011w oo110mmm disp		O	D	I	T	S	Z	A	P	C
Formato	Ejemplos	?					?	?	?	?
		Relojes								
DIV reg	DIV BL DIV BH DIV ECX DIV SH DIV CH	8086	162							
		8088	162							
		80286	22							
		80386	38							
		80486	40							
DIV mem	DIV DATA DIV LIST DIV DATA [DI] DIV BYTE PTR [EAX] DIV WORD PTR [DI]	8086	166							
		8088	176							
		80286	25							
		80386	41							
		80486	40							

ENTER Crear un marco de pila

11001000 información		O	D	I	T	S	Z	A	P	C
Formato	Ejemplos									
		Relojes								
ENTER imm,0	ENTER 4,0 ENTER 8,0 ENTER 100,0 ENTER 200,0 ENTER 1024,0	8086	—							
		8088	—							
		80286	11							
		80386	10							
		80486	14							

ENTER imm,1	ENTER 4,1 ENTER 10,1	8086	--
		8088	--
		80286	15
		80386	12
		80486	17
ENTER imm,imm	ENTER 3,6 ENTER 100,3	8086	--
		8088	--
		80286	12
		80386	15
		80486	17

ESC Escape

11011nnn oonnnmmm O D I T S Z A P C

nnnnnn = Código de operación para coprocesador
Formato Ejemplos

Relojes

ESC imm,reg	ESC 5,AL ESC 5,5H ESC 6,CH FADD ST,ST(3)	8086	2
		8088	2
		80286	20
		80386	var
		80486	var
ESC imm,mem	ESC 2,DATA ESC 3,FROG FADD DATA FMUL FROG	8086	6÷ea
		8088	12÷ea
		80286	20
		80386	var
		80486	var

HLT Alto

11110100 O D I T S Z A P C

Ejemplo Relojes

HLT	8086	2
	8088	2
	80286	2
	80386	5
	80486	4

IDIV División con identificación			
1111011w 00111mmm disp		O D I T S Z A P C ? ? ? ? ?	
Formato	Ejemplos	Relojes	
IDIV reg	IDIV BL	8086	184
	IDIV BH	8088	184
	IDIV ECX	80286	25
	IDIV BH	80386	43
	IDIV CX	80486	43
IDIV mem	IDIV DATA	8086	190
	IDIV LIST	8088	194
	IDIV DATA [DI]	80286	28
	IDIV BYTE PTR [EAX]	80386	46
	IDIV WORD PTR [DI]	80486	44
IMUL Multiplicación con identificación			
1111011w 00101mmm disp		O D I T S Z A P C * ? ? ? ? *	
Formato	Ejemplos	Relojes	
IMUL reg	IMUL BL	8086	154
	IMUL CL	8088	154
	IMUL CX	80286	21
	IMUL ECX	80386	38
	IMUL EBX	80486	42
IMUL mem	IMUL DATA	8086	160
	IMUL LIST	8088	164
	IMUL DATA [SI]	80286	24
	IMUL BYTE PTR [EAX]	80386	41
	IMUL WORD PTR [DI]	80486	42

011010sl oorrmmm disp información			
Formato	Ejemplos	Relojes	
IMUL reg,imm	IMUL CX,16 IMUL DX,100 IMUL EAX,20	8086	—
		8088	—
		80286	21
		80386	38
		80486	42
IMUL reg,reg,imm	IMUL DX,AX,2 IMUL CX,DX,3 IMUL BX,AX,33	8086	—
		8088	—
		80286	21
		80386	38
		80486	42
IMUL reg,mem,imm	IMUL CX,DATA,4	8086	—
		8088	—
		80286	24
		80386	38
		80486	42
00001111 10101111 oorrmmm disp			
Formato	Ejemplos	Relojes	
IMUL reg,reg	IMUL CX,DX IMUL DX,BX IMUL EAX,ECX	8086	—
		8088	—
		80286	—
		80386	38
		80486	42
IMUL reg,mem	IMUL DX,DATA IMUL CX,FROG IMUL BX,LISTS	8086	—
		8088	—
		80286	—
		80386	41
		80486	42
IN Entrada de información del puerto			
1110010w número del puerto		O D I T S Z A P C	
Formato	Ejemplos	Relojes	

IN acc,pt	IN AL,12H IN AX,12H IN AL,OFFH IN AX,OFFH IN EAX,10H	8086	10
		8088	14
		80286	5
		80386	12
		80486	14

1110110w			
Formato	Ejemplos	Relojes	
IN acc,DX	IN AL,DX IN AX,DX IN EAX,DX	8086	8
		8088	12
		80286	5
		80386	13
		80486	14

INC Incremento

1111111w oo000mmm disp		O D I T S Z A P C			
Formato	Ejemplos	Relojes			

INC reg8	INC BL INC BH INC CL INC DH INC AH	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
INC mem	INC DATA INC LIST INC DATA [BX] INC BYTE PTR [EAX] INC WORD PTR [BX] INC DWORD PTR [ECX]	8086	15+ea
		8088	23+ea
		80286	7
		80386	6
		80486	3

01000rrr			
Formato	Ejemplos	Relojes	

INC reg16 INC reg32	INC AX INC EAX INC CX INC EBX INC DI	8086	3
		8088	3
		80286	2
		80386	2
		80486	1

INS		Cadena de entrada de información del puerto	
0110110w		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
INSB INSW INSD	INSB INSW INSD INS DATA REP INSB	8086	—
		8088	—
		80286	5
		80386	15
		80486	17
INT		Interrupción	
11001101 tipo		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
INT type	INT 10H INT 255 INT 21H INT 20H INT 15H	8086	51
		8088	71
		80286	23
		80386	37
		80486	30
11001100			
Ejemplo		Relojes	
INT 3		8086	52
		8088	72
		80286	23
		80386	33
		80486	26
INTO		Interrupción en desbordamiento	
11001110		O D I T S Z A P C	
Ejemplo		Relojes	
INTO		8086	53
		8088	73
		80286	24
		80386	35
		80486	28

INVD		Invalidar al cache para información	
00001111 00001000		O D I T S Z A P C	
Ejemplo		Relojes	
INVD		8086	---
		8088	---
		80286	---
		80386	---
		80486	4
INVLPG		Invalidar elemento TLB	
00001111 00000001 00111mmm		O D I T S Z A P C	
Formato Ejemplos		Relojes	
INVLPG mem	INVLPG DATA INVLPG LIST	8086	---
		8088	---
		80286	---
		80386	---
		80486	12
IRET		Retorno de interruptor	
11001101 información		O D I T S Z A P C	
Formato Ejemplos		Relojes	
IRET IRETD	IRET IRETD IRET 10H	8086	32
		8088	44
		80286	17
		80386	22
		80486	15

Jconditional Salto condicional			
0111cccc disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
Jcc label (8-bit disp)	JA BELOW JB ABOVE JG GREATER JE EQUAL JZ ZERO	8086	16/4
		8088	16/4
		80286	7/3
		80386	7/3
		80486	3/1
00001111 1000cccc disp			
Formato	Ejemplos	Relojes	
Jcc label (16-bit disp)	JNE NOT_MORE JLE LESS_THAN	8086	—
		8088	—
		80286	—
		80386	7/3
		80486	3/1
Condición			
Códigos	Mnemonic	Bandera	Descripción
0000	JO	O = 1	Saltar si existe desbordamiento
0001	JNO	O = 0	Saltar si no existe desbordamiento
0010	JB/JNAE	C = 1	Saltar si está abajo
0011	JAE/JNB	C = 0	Saltar si está arriba o igual
0100	JE/JZ	Z = 1	Saltar si es igual/cero
0101	JNE/JNZ	Z = 0	Saltar si no es igual/no es cero
0110	JBE/JNA	C = 1 + Z = 1	Saltar si está abajo o igual
0111	JA/JNBE	C = 0 + Z = 0	Saltar si está arriba
1000	JS	S = 1	Saltar si existe signo
1001	JNS	S = 0	Saltar si no existe signo
1010	JP/JPE	P = 1	Saltar si la paridad es pares
1011	JNP/JPO	P = 0	Saltar si la paridad es nones
1100	JL/JNGE	S · O	Saltar si es menor que
1101	JGE/JNL	S = O	Saltar si es mayor o igual
1110	JLE/JNG	Z = 1 + S · O	Saltar si es menor o igual
1111	JG/JNLE	Z = 0 + S = O	Saltar si es mayor

JCXZ/JECXZ Salto, si CX (ECX) es igual a cero			
11100011		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
JCXZ label JECXZ label	JCXZ LOTS JCXZ OVER JECXZ UPPER JECXZ UNDER JCXZ NEXT	8086	18/6
		8088	18/6
		80286	8/4
		80386	9/5
		80486	8/5
JMP Salto incondicional			
11101011 disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
JMP label (corto)	JMP SHORT UP JMP SHORT DOWN JMP SHORT OVER JMP SHORT CIRCUIT JMP SHORT ARM	8086	15
		8088	15
		80286	7
		80386	7
		80486	3
11101001 disp		Relojes	
Formato	Ejemplos	Relojes	
JMP label (cerca)	JMP VER JMP FROG JMP UNDER JMP NEAR PTR OVER	8086	15
		8088	15
		80286	7
		80386	7
		80486	3
11101010 disp		Relojes	
Formato	Ejemplos	Relojes	
JMP label (lejos)	JMP VER JMP FROG JMP UNDER JMP FAR PTR THERE	8086	15
		8088	15
		80286	11
		80386	12
		80486	17

11111111 00100mmm			
Formato	Ejemplos	Relojes	
JMP reg (cerca)	JMP AX JMP EAX JMP CX JMP DX	8086	11
		8088	11
		80286	7
		80386	7
		80486	3
JMP mem (cerca)	JMP DATA JMP LIST JMP DATA [DI+2]	8086	18+ea
		8088	18+ea
		80286	11
		80386	10
		80486	5
11111111 00101mmm			
Formato	Ejemplos	Relojes	
JMP mem (lejos)	JMP WAY OFF JMP TABLE JMP UP	8086	24+ea
		8088	24+ea
		80286	15
		80386	12
		80486	13
LAHF Cargar AH de las banderas			
10011111		O D I T S Z A P C	
Ejemplo		Relojes	
LAHF		8086	4
		8088	4
		80286	2
		80386	2
		80486	3

LAR Cargar los derechos de acceso			
00001111 00000010 oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LAR reg,reg	LAR AX,BX LAR CX,DX LAR EAX,ECX	8086	---
		8088	---
		80286	14
		80386	15
		80486	11
LAR reg,mem	LAR CX,DATA LAR AX,LIST LAR ECX,FROG	8086	---
		8088	---
		80286	16
		80386	16
		80486	11
LDS Cargar apuntador lejano			
11000101 oorrmmm		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LDS reg,mem	LDS DI,DATA LDS SI,LIST LDS BX,ARRAY LDS CX,PNTR	8086	16+ea
		8088	24+ea
		80286	7
		80386	7
		80486	6
LES Cargar apuntador lejano			
11000100 oorrmmm		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LES reg,mem	LES DI,DATA LES SI,LIST LES BX,ARRAY LES CX,PNTR	8086	16+ea
		8088	24+ea
		80286	7
		80386	7
		80486	6

LFS Cargar apuntador lejano			
00001111 10110100 oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LFS reg,mem	LFS DI,DATA LFS SI,LIST LFS BX,ARRAY LFS CX,PNTR	8086	---
		8088	---
		80286	---
		80386	7
		80486	6
LGS Cargar apuntador lejano			
00001111 10110101 oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LGS reg,mem	LGS DI,DATA LGS SI,LIST LGS BX,ARRAY LGS CX,PNTR	8086	---
		8088	---
		80286	---
		80386	7
		80486	6
LSS Cargar apuntador lejano			
00001111 10110010 oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LSS reg,mem	LSS DI,DATA LSS SI,LIST LSS BX,ARRAY LSS CX,PNTR	8086	---
		8088	---
		80286	---
		80386	7
		80486	6
LEA Cargar la dirección efectiva			

10001101 oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LEA reg,mem	LEA DI,DATA LEA SI,LIST LEA BX,ARRAY LEA CX,PNTR LEA BP,ADDR	8086	2+ea
		8088	2+ea
		80286	3
		80386	2
		80486	2

LEAVE Salir del procedimiento de nivel alto

11001001		O D I T S Z A P C	
Ejemplo		Relojes	
LEAVE		8086	—
		8088	—
		80286	5
		80386	4
		80486	5

LGDT Cargar tabla del descriptor global

00001111 00000001 00010mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LGDT mem64	LGDT DESCRIP LGDT TABLE	8086	—
		8088	—
		80286	11
		80386	11
		80486	11

LIDT Cargar tabla del interruptor para descriptor

00001111 00000001 00011mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LIDT mem64	LIDT DATA LIDT DESCRIP	8086	—
		8088	—
		80286	12
		80386	11
		80486	11

LLDT Cargar tabla local del descriptor			
00001 11 00000000 00010mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LLDT reg	LLDT AX LLDT CX	8086	—
		8088	—
		80286	17
		80386	20
		80486	11
LLDT mem	LLDT DATA LLDT LIST	8086	—
		8088	—
		80286	19
		80386	24
		80486	11
LMSW Cargar palabra del estado de la máquina			
00001111 00000001 00110mmm disp		O D I T S Z A P C	
Solo se debe utilizar con el 80286			
Formato	Ejemplos	Relojes	
LMSW reg	LMSW AX LMSW CX	8086	—
		8088	—
		80286	3
		80386	10
		80486	2
LMSW mem	LMSW DATA LMSW LIST	8086	—
		8088	—
		80286	6
		80386	13
		80486	3
LOCK Cerrar el canal			

11110000		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LOCK inst	LOCK:XCHG AX,BX LOCK:MOV AL,AH	8086	2
		8088	2
		80286	0
		80386	0
		80486	1

LODS Cargar el operando de la cadena

1010110w		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LODSB LODSW LODSD	LODSB LODSW LODSD LODS DATA LODS ES:DATA	8086	12
		8088	16
		80286	5
		80386	5
		80486	5

LOOP Repetir hasta que CX = 0

11100010 disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LOOP label	LOOP DATA LOOP BACK	8086	17/5
		8088	17/5
		80286	8/4
		80386	11
		80486	7/6

LOOPE Repetir mientras sea igual

11100001 disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LOOPE label LOOPZ label	LOOPE NEXT LOOPE AGAIN LOOPZ REPEAT	8086	18/6
		8088	18/6
		80286	8/4
		80386	11
		80486	9/6

LOOPNE Repetir mientras que no sea igual

11100000 disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LOOPNE label LOOPNZ label	LOOPNE AGAIN LOOPNE BACK LOOPNZ REPL	8086	19/5
		8088	19/5
		80286	8/4
		80386	11
		80486	9/6

LSL Cargar el límite del segmento

00001111 00000011 corrrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LSL reg,reg	LSL AX,BX LSL CX,BX LSL DX,AX	8086	—
		8088	—
		80286	14
		80386	25
		80486	10
LSL reg,mem	LSL AX,LIMIT LSL EAX,NUMB	8086	—
		8088	—
		80286	16
		80386	26
		80486	10

LTR Cargar registro para tareas

00001111 00000000 oo001mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
LTR reg	LTR AX LTR CX LTR DX	8086	—
		8088	—
		80286	17
		80386	23
		80486	20

LTR mem	LTR TASK LTR EDGE	8085	—
		8088	—
		80286	19
		80386	27
		80486	20

MOV Mover información

100010dw oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
MOV reg,reg	MOV CL,CH MOV BH,CL MOV CX,DX MOV EAX,ECX MOV EBP,ESI	8086	2
		8088	2
		80286	2
		80386	2
		80486	1
MOV mem,reg	MOV DATA,DL MOV NUMB,CX MOV TEMP,EBX MOV TEMP1,CH MOV DATA2,CL	8086	9+ea
		8088	13+ea
		80286	3
		80386	2
		80486	1
MOV reg,mem	MOV DL,DATA MOV DX,NUMB MOV EBX,TEMP MOV CH,TEMP1 MOV CL,DATA2	8086	10+ea
		8088	12+ea
		80286	5
		80386	4
		80486	1
1100011w oo000mmm disp información			
Formato	Ejemplos	Relojes	
MOV mem,imm	MOV DATA,23H MOV LIST,12H MOV BYTE PTR [DI],2 MOV NUMB,234H MOV DWORD PTR [SI],100	8086	10+ea
		8088	14+ea
		80286	3
		80386	2
		80486	1

1011wrrr información			
Formato	Ejemplos	Relojes	
MOV reg,imm	MOV BX,23H MOV CX,12H MOV CL,2 MOV ECX,123423H MOV DI,100	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
101000dw disp			
Formato	Ejemplos	Relojes	
MOV mem,acc	MOV DATA,AL MOV NUMB,AX MOV NUMB1,EAX	8086	10
		8088	14
		80286	3
		80386	2
		80486	1
MOV acc,mem	MOV AL,DATA MOV AX,NUMB MOV EAX,TEMP	8086	10
		8088	14
		80286	5
		80386	4
		80486	1
100011d0 oosssmmm disp			
Formato	Ejemplos	Relojes	
MOV seg,reg	MOV SS,AX MOV DS,DX MOV ES,CX	8086	2
		8088	2
		80286	2
		80386	2
		80486	1
MOV seg,mem	MOV SS,DATA MOV DS,NUMB MOV ES,TEMP1	8086	8+ea
		8088	12+ea
		80286	2
		80386	2
		80486	1

MOV reg,seg	MOV AX,DS MOV DX,ES MOV CX,CS	8086	2
		8088	2
		80286	2
		80386	2
		80486	1
MOV mem,seg	MOV DATA,SS MOV NUMB,ES MOV TEMP1,DS	8086	9÷ea
		8088	13÷ea
		80286	3
		80386	2
		80486	1
00001111 001000d0 11rrmmm			
Formato	Ejemplos	Relojes	
MOV reg,cr	MOV EAX,CR0 MOV EBX,CR2 MOV ECX,CR3	8086	—
		8088	—
		80286	—
		80386	6
		80486	4
MOV cr,reg	MOV CR0,EAX MOV CR2,EBX MOV CR3,ECX	8086	—
		8088	—
		80286	—
		80386	10
		80486	4
00001111 001000d1 11rrmmm			
Formato	Ejemplos	Relojes	
MOV reg,dr	MOV EBX,DR6 MOV EAX,DR6 MOV EDX,DR1	8086	—
		8088	—
		80286	—
		80386	22
		80486	10

MOV dr,reg	MOV DR1,ECX MOV DR2,ESI MOV DR6,EBP	8086	—
		8088	—
		80286	—
		80386	22
		80486	11

00001111 001001d0 11rrmmm

Formato	Ejemplos	Relojes	
MOV reg,tr	MOV EAX,TR6 MOV EDX,TR7	8086	—
		8088	—
		80286	—
		80386	12
		80486	4
MOV tr,seg	MOV TR6,EDX MOV TR7,ESI	8086	—
		8088	—
		80286	—
		80386	12
		80486	6

MOVS Mover información de la cadena

1010010w		O D I T S Z · A P C	
Formato	Ejemplos	Relojes	
MOVSB MOVSW MOVSD	MOVSB MOVSW MOVSD MOVS DAT1,DAT2 REP MOVSB	8086	18
		8088	26
		80286	5
		80386	7
		80486	7

MOVSX Moverse con signo extendido

00001111 1011111w oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
MOVSX reg,reg	MOVSX BX,AL MOVSX EAX,DX	8086	—
		8088	—
		80286	—
		80386	3
		80486	3
MOVSX reg,mem	MOVSX AX,DATA MOVSX EAX,NUMB	8086	—
		8088	—
		80286	—
		80386	6
		80486	3

MOVZX Moverse con cero extendido

00001111 1011011w oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
MOVZX reg,reg	MOVZX BX,AL MOVZX EAX,DX	8086	—
		8088	—
		80286	—
		80386	3
		80486	3
MOVZX reg,mem	MOVZX AX,DATA MOVZX EAX,NUMB	8086	—
		8088	—
		80286	—
		80386	6
		80486	3

MUL Multiplicación sin identificar			
1111011w oo100mmm disp		O D I T S Z P C * ? ? ? ? *	
Formato	Ejemplos	Relojes	
MUL reg	MUL BL MUL CX MUL ECX	8086	118
		8088	143
		80286	21
		80386	38
		80486	42
MUL mem	MUL DATA MUL BYTE PTR [SI] MUL WORD PTR [SI] MUL DWORD PTR [ECX]	8086	139
		8088	143
		80286	24
		80386	41
		80486	42
NEG Negación			
1111011w oo011mmm disp		O D I T S Z A P C * * * * * * * *	
Formato	Ejemplos	Relojes	
NEG reg	NEG AX NEG CX NEG EDX	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
NEG mem	NEG DATA NEG NUMB NEG WORD PTR [DI]	8086	16+ea
		8088	24+ea
		80286	7
		80386	6
		80486	3

NOP Sin operación			
10010000		O D I T S Z A P C	
Ejemplo		Relojes	
NOP	8086	3	
	8088	3	
	80286	3	
	80386	3	
	80486	3	
NOT El complemento de uno			
1111011w 00010mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
NOT reg	NOT AX NOT CX NOT EDX	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
NOT mem	NOT DATA NOT NUMB NOT WORD PTR [DI]	8086	16+ea
		8088	24+ea
		80286	7
		80386	6
		80486	3
OR OR-inclusivo			
000010dw 00rrmmm disp		O D I T S Z A P C 0 * * ? * 0	
Formato	Ejemplos	Relojes	
OR reg,reg	OR CL,BL OR CX,DX OR ECX,EBX	8086	3
		8088	3
		80286	2
		80386	2
		80486	1

OR mem,reg	OR DATA,CL OR NUMB,CX OR [DI],CX	8086	16+ea
		8088	24+ea
		80286	7
		80386	7
		80486	3
OR reg,mem	OR CL,DATA OR CX,NUMB OR CX,[SI]	8086	9+ea
		8088	13+ea
		80286	7
		80386	6
		80486	2

10000sw 00001mmm disp data

Formato	Ejemplos	Relojes	
OR reg,imm	OR CL,3 OR DX,1000H OR EBX,100000H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
OR mem,imm	OR DATA,33 OR NUMB,4AH OR NUMS,123498H OR BYTE PTR [ECX],2	8086	17+ea
		8088	25+ea
		80286	7
		80386	7
		80486	3

0000110w información

Formato	Ejemplos	Relojes	
OR acc,imm	OR AL,3 OR AX,1000H OR EAX,100000H	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

OUT Salida de información al puerto			
1110011w número de puerto		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
OUT pt,acc	OUT 12H,AL OUT 12H,AX OUT 0FFH,AL OUT 0FEH,AX OUT 10H,EAX	8086	10
		8088	14
		80286	3
		80386	10
		80486	10
1110111w			
Formato	Ejemplos	Relojes	
OUT DX,acc	OUT DX,AL OUT DX,AX OUT DX,EAX	8086	8
		8088	12
		80286	3
		80386	11
		80486	10

OUTS Salida de información de la cadena al puerto			
1110011w número de puerto		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
OUTSB OUTSW OUTSD	OUTSB OUTSW OUTSD OUTS DATA REP OUTSB	8086	—
		8088	—
		80286	5
		80386	14
		80486	10

POP Sacar datos de la pila			
01011rrr		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
POP reg	POP CX POP AX POP EBX	8086	8
		8088	12
		80286	5
		80386	4
		80486	1

10001111 oo000mmm disp			
Formato	Ejemplos	Relojes	
POP mem	POP DATA POP LISTS POP NUMBS	8086	17+ea
		8088	25+ea
		80286	5
		80386	5
		80486	4
00sss111			
Formato	Ejemplos	Relojes	
POP seg	POP DS POP ES POP SS	8086	8
		8088	12
		80286	5
		80386	7
		80486	3
00001111 10sss001			
Formato	Ejemplos	Relojes	
POP seg	POP FS POP GS	8086	—
		8088	—
		80286	—
		80386	7
		80486	3

POPA/POPAD Sacar todos los registros de la pila

01100001		O D I T S Z A P C	
Ejemplo	Relojes		
POPA POPAD	8086	—	
	8088	—	
	80286	19	
	80386	24	
	80486	9	

POPF/POPFD Sacar las banderas de la pila

10011101		O D I T S Z A P C						
Ejemplo		Relojes						
POPF POPFD		8086	—					
		8088	—					
		80286	5					
		80386	5					
		80486	6					

PUSH Empujar información a la pila

01010rrr		O D I T S Z A P C						
Formato	Ejemplos	Relojes						
PUSH reg	PUSH CX PUSH AX PUSH ECX	8086	11					
		8088	15					
		80286	3					
		80386	2					
		80486	1					

11111111 00110mmm disp

Formato	Ejemplos	Relojes						
PUSH mem	PUSH DATA PUSH LIST PUSH NUMB PUSH DWORD PTR [ECX]	8086	16÷ea					
		8088	24÷ea					
		80286	5					
		80386	5					
		80486	4					

00sss110

Formato	Ejemplos	Relojes						
PUSH seg	PUSH DS PUSH CS PUSH ES	8086	10					
		8088	14					
		80286	3					
		80386	2					
		80486	3					

APENDICE B

00001111 10sss000			
Formato	Ejemplos	Relojes	
PUSH seg	PUSH FS PUSH GS	8086	—
		8088	—
		80286	—
		80386	2
		80486	3
011010s0 información			
Formato	Ejemplos	Relojes	
PUSH imm	PUSH 2000H PUSH 5322H PUSHW 10H PUSHD 100000H	8086	—
		8088	—
		80286	3
		80386	2
		80486	1
PUSHA/PUSHAD Empujar a todos los registros			
01100000 O D I T S Z A P C			
Ejemplo	Relojes		
PUSHA PUSHAD	8086	—	
	8088	—	
	80286	17	
	80386	18	
	80486	11	
PUSHF/PUSHFD Empujar las banderas a la pila			
10011100 O D I T S Z A P C			
Ejemplo	Relojes		
PUSHF PUSHFD	8086	10	
	8088	14	
	80286	3	
	80386	4	
	80486	3	

RCL/RCR/ROL/ROR Rotar			
1101000w ooTTTmmm disp		O D I T S Z A P C	
TTT = 000 = ROL TTT = 001 = ROR TTT = 010 = RCL TTT = 011 = RCR			
Formato	Ejemplos	Relojes	
ROL reg,1 ROR reg,1	ROL CL,1 ROL DX,1 ROR CH,1 ROL SI,1	8086	2
		8088	2
		80286	2
		80386	3
		80486	3
RCL reg,1 RCR reg,1	RCL CL,1 RCL SI,1 RCR AH,1 RCR EBX,1	8086	2
		8088	2
		80286	2
		80386	9
		80486	3
ROL mem,1 ROR mem,1	ROL DATA,1 ROL BYTE PTR [DI],1 ROR NUMB,1 ROR DWORD PTR [ECX],1	8086	15+ea
		8088	23+ea
		80286	7
		80386	7
		80486	4
RCL mem,1 RCR mem,1	RCL DATA,1 RCL BYTE PTR [DI],1 RCR NUMB,1 RCR WORD PTR [ECX],1	8086	15+ea
		8088	23+ea
		80286	7
		80386	10
		80486	4
1101001w ooTTTmmim disp			
Formato	Ejemplos	Relojes	
ROL reg,CL ROR reg,CL	ROL CH,CL ROL DX,CL ROR CH,CL ROL SI,CL	8086	8+4n
		8088	8+4n
		80286	5+n
		80386	3
		80486	3

RCL reg,CL ROR reg,CL	RCL DL,CL RCL SI,CL ROR AH,CL ROR BX,CL	8086	8+4n
		8088	8+4n
		80286	5+n
		80386	9
		80486	8
ROL mem,CL ROR mem,CL	ROL DATA,CL ROL BYTE PTR [DI],CL ROR NUMB,CL ROR WORD PTR [ECX],CL	8086	20+a+4n
		8088	28+a+4n
		80286	8+n
		80386	7
		80486	4
RCL mem,CL ROR mem,CL	RCL DATA,CL RCL BYTE PTR [DI],CL ROR NUMB,CL ROR WORD PTR [ECX],CL	8086	20+a+4n
		8088	28+a+4n
		80286	8+n
		80386	10
		80486	9
1100000w ooTTTTmmm disp información			
Formato	Ejemplos	Relojes	
ROL reg,imm ROR reg,imm	ROL CL,4 ROL DX,5 ROR CH,12 ROL SI,9	8086	—
		8088	—
		80286	5+n
		80386	3
		80486	2
RCL reg,imm ROR reg,imm	RCL CL,2 RCL SI,3 ROR AH,5 ROR BX,13 . .	8086	—
		8088	—
		80286	5+n
		80386	9
		80486	8
ROL mem,imm ROR mem,imm	ROL DATA,4 ROL BYTE PTR [DI],2 ROR NUMB,2 ROR WORD PTR [ECX],3	8086	—
		8088	—
		80286	8+n
		80386	7
		80486	4

RCL mem,imm RCR mem,imm	RCL DATA,6 RCL BYTE PTR [DI],7 RCR NUMB,6 RCR WORD PTR [ECX],5	8086	---
		8088	---
		80286	8+n
		80386	10
		80486	9

REP Repetir prefijo

11110010 1010010w		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
REP MOVS	REP MOVSB REP MOVSW REP MOVSD REP MOVS DATA1,DATA2	8086	9+17n
		8088	9+25n
		80286	5+4n
		80386	8+4n
		80486	12+3n
11110010 1010101w			
Formato	Ejemplos	Relojes	
REP STOS	REP STOSB REP STOSW REP STOSD REP STOS DATA3	8086	9+10n
		8088	9+14n
		80286	4+3n
		80386	5+5n
		80486	7+4n
11110010 0110110w			
Formato	Ejemplos	Relojes	
REP INS	REP INSB REP INSW REP INSD REP INS DATA4	8086	—
		8088	—
		80286	5+4n
		80386	13+6n
		80486	16+8n

11110010 0110111w			
Formato	Ejemplos	Relojes	
REP OUTS	REP OUTSB REP OUTSW REP OUTSD REP OUTS DATA5	8086	---
		8088	---
		80286	5+4n
		80386	12+5n
		80486	17+5n

REPE/REPNE Repetir condicional

11110011 1010011w			
			O D I T S Z A P C
Formato	Ejemplos	Relojes	
REPE CMPS	REPE CMPSB REPE CMPSW REPE CMPSD REPE CMPS DATA6,DATA7	8086	9+22n
		8088	9+30n
		80286	5+9n
		80386	5+9n
		80486	7+7n

11110011 1010111w			
Formato	Ejemplos	Relojes	
REPE SCAS	REPE SCASB REPE SCASW REPE SCASD REPE SCAS DATA8	8086	9+15n
		8088	9+19n
		80286	5+8n
		80386	5+8n
		80486	7+5n

11110010 1010011w			
Formato	Ejemplos	Relojes	
REPNE CMPS	REPNE CMPSB REPNE CMPSW REPNE CMPSD REPNE CMPS DATA9,DATA10	8086	9+22n
		8088	9+30n
		80286	5+9n
		80386	5+9n
		80486	7+7n

11110010 1010111w			
Formato	Ejemplos	Relojes	
REPNE SCAS	REPNE SCASB REPNE SCASW REPNE SCASD REPNE SCAS DATA11	8086	9+15n
		8088	9+19n
		80286	5+8n
		80386	5+8n
		80486	7+5n
RET Regresar del procedimiento			
11000011 O D I T S Z A P C			
Ejemplo	Relojes		
RET (cerca)	8086	16	
	8088	20	
	80286	11	
	80386	10	
	80486	5	
11000010 información			
Formato	Ejemplos	Relojes	
RET imm (cerca)	RET 4 RET 100H	8086	20
		8088	24
		80286	11
		80386	10
		80486	5
11001011			
Ejemplos	Relojes		
RET (lejos)	8086	26	
	8088	34	
	80286	15	
	80386	18	
	80486	13	

11001010 información			
Formato	Ejemplos	Relojes	
RET imm (lejos)	RET 4 RET 100H	8086	25
		8088	33
		80286	11
		80386	10
		80486	5
SAHF Archivar AH en las banderas			
10011110		O D I T S Z A P C * * * * * * * * *	
Ejemplo		Relojes	
SAHF		8086	4
		8088	4
		80286	2
		80386	3
		80486	2
SAL/SAR/SHL/SHR Camoiar de posición			
1101000w ooTTTmmm disp		O D I T S Z A P C * * * * * ? * * *	
TTT = 100 = SHL/SAL TTT = 101 = SHR TTT = 111 = SAR			
Formato	Ejemplos	Relojes	
SAL reg,1 SHL reg,1 SHR reg,1 SAR reg,1	SAL CL,1 SHL DX,1 SHR CH,1 SAR SI,1	8086	2
		8088	2
		80286	2
		80386	3
		80486	3
SAL mem,1 SHL mem,1 SHR mem,1 SAR mem,1	SAL DATA,1 SHL BYTE PTR [DI],1 SHR NUMB,1 SAR WORD PTR [ECX],1	8086	15+ea
		8088	23+ea
		80286	7
		80386	7
		80486	4

1101001w ooTTTmmm disp			
Formato	Ejemplos	Relojes	
SAL reg,CL SHL reg,CL SHR reg,CL SAR reg,CL	SAL CH,CL SHL DX,CL SHR CH,CL SAR SI,CL	8086	8÷4n
		8088	8÷4n
		80286	5+n
		80386	3
		80486	3
SAL mem,CL SHL mem,CL SHR mem,CL SAR mem,CL	SAL DATA,CL SHL BYTE PTR [DI],CL SHR NUMB,CL SAR WORD PTR [ECX],CL	8086	20+a-4n
		8088	28+a+4n
		80286	6÷n
		80386	7
		80486	4

1100000w ooTTTmmm disp información			
Formato	Ejemplos	Relojes	
SAL reg,imm SHL reg,imm SHR reg,imm SAR reg,imm	SAL CL,4 SHL DX,5 SHR CH,12 SAR SI,9	8086	—
		8088	—
		80286	5+n
		80386	3
		80486	2
SAL mem,imm SHL mem,imm SHR mem,imm SAR mem,imm	SAL DATA,6 SHL BYTE PTR [DI],7 SHR NUMB,6 SAR WORD PTR [ECX],5	8086	—
		8088	—
		80286	8+n
		80386	7
		80486	4

SBB Restar con pedir prestado

000110dw oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
SBB reg,reg	SBB CL,DL SBB AX,DX SBB CH,CL SBB EAX,EBX	8086	3
		8088	3
		80286	2
		80386	2
		80486	1

SBB mem,reg	SBB DATA,CL SBB BYTES,CX SBB NUMBS,ECX SBB [EAX],CX	8086	16÷ea
		8088	24÷ea
		80286	7
		80386	6
		80486	3
SBB reg,mem	SBB CLDATA SBB CX,BYTES SBB ECX,NUMBS SBB CX,[EDX]	8086	9÷ea
		8088	13÷ea
		80286	7
		80386	7
		80486	2
100000sw 0011mmm disp información			
Formato	Ejemplos	Relojes	
SBB reg,imm	SBB CL,4 SBB DX,5 SBB CH,12 SBB SI,9	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
SBB mem,imm	SBB DATA,6 SBB BYTE PTR [DI],7 SBB NUMB,6 SBB WORD PTR [ECX],5	8086	17÷ea
		8088	25÷ea
		80286	7
		80386	7
		80486	3
0001110w información			
Formato	Ejemplos	Relojes	
SBB acc,imm	SBB AL,4 SBB AX,5 SBB AH,12 SBB AX,9	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

SCAS		Examinar cadena	
1010111w		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
SCASB SCASW SCASD	SCASB	8086	15
	SCASW	8088	19
	SCASD	80286	7
	SCAS DATA	80386	7
	REP SCASB	80486	6
SET		Establecer con condición	
00001111 1001cccc 0000mmm		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
SETcd reg8	SETA BL	8086	---
	SETB CH	8088	---
	SETG DL	80286	---
	SETI BH	80386	4
	SETZ AL	80486	3
SETcd mem8	SETI DATA	8086	---
	SETLE BYTES	8088	---
		80286	---
		80386	5
		80486	3
Condición	Códigos Mnemonic	Bandera	Descripción
0000	SETO	O = 1	Establecer si existe desbordamiento
0001	SETNO	O = 0	Establecer si no existe desbordamiento
0010	SETB/SETNAE	C = 1	Establecer si está abajo
0011	SETAE/SETNB	C = 0	Establecer si está arriba o igual
0100	SETI/SETZ	Z = 1	Establecer si es igual/cero
0101	SETNE/SETNZ	Z = 0	Establecer si no es igual/no es cero
0110	SETBE/SETNA	C = 1 + Z = 1	Establecer si está abajo o igual
0111	SETA/SETNBE	C = 0 + Z = 0	Establecer si está arriba
1000	SETS	S = 1	Establecer si existe signo
1001	SETNS	S = 0	Establecer si no existe signo
1010	SETP/SETPE	P = 1	Establecer si la paridad es pares
1011	SETNP/SETPO	P = 0	Establecer si la paridad es nones
1100	SETL/SETNGE	S = 0	Establecer si es menor que
1101	SETGE/SETNL	S = 0	Establecer si es mayor o igual
1110	SETLE/SETNG	Z = 1 + S = 0	Establecer si es menor que o igual
1111	SETG/SETNLE	Z = 0 + S = 0	Establecer si es mayor

SGDT/SIDT/SLDT Almacenar la tabla del descriptor			
00001111 00000001 0000mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
SGDT mem	SGDT MEMORY SGDT GLOBAL	8086	—
		8088	—
		80286	11
		80386	9
		80486	10
00001111 00000001 00001mmm disp		Relojes	
Formato	Ejemplos	Relojes	
SIDT mem	SIDT DATAS SIDT INTERRUPT	8086	—
		8088	—
		80286	12
		80386	9
		80486	10
00001111 00000000 0000mmm disp		Relojes	
Formato	Ejemplos	Relojes	
SLDT reg	SLDT CX SLDT DX	8086	—
		8088	—
		80286	2
		80386	2
		80486	2
SLDT mem	SLDT NUMBS SLDT LOCALS	8086	—
		8088	—
		80286	3
		80386	2
		80486	3

SHLD/SHRD Cambio de posición con doble precisión			
00001111 10100100 oorrmmm disp información		O	D
Formato	Ejemplos	?	T
		S	Z
		A	P
		C	Relojes
SHLD reg,reg,imm	SHLD AX,CX,10 SHLD DX,BX,8 SHLD CX,DX,2	8086	—
		8088	—
		80286	—
		80386	3
		80486	2
SHLD mem,reg,imm	SHLD DATA,CX,8	8086	—
		8088	—
		80286	—
		80386	7
		80486	3
00001111 10101100 oorrmmm disp información		S	Z
Formato	Ejemplos	A	P
		C	Relojes
SHRD reg,reg,imm	SHRD CX,DX,2	8086	—
		8088	—
		80286	—
		80386	3
		80486	2
SHRD mem,reg,imm	SHRD DATA,CX,3	8086	—
		8088	—
		80286	—
		80386	7
		80486	3
00001111 10100101 oorrmmm disp		S	Z
Formato	Ejemplos	A	P
		C	Relojes
SHLD reg,reg,CL	SHLD DX,BX,CL	8086	—
		8088	—
		80286	—
		80386	3
		80486	3

SHLD mem,reg,CL	SHLD DATA,AX,CL	8086	—
		8088	—
		80286	—
		80386	7
		80486	3

00001111 10100101 oorrmmm disp

Formato	Ejemplos	Relojes	
SHRD reg,reg,CL	SHRD DX,BX,CL	8086	—
		8088	—
		80286	—
		80386	3
		80486	3
SHRD mem,reg,CL	SHRD DATA,AX,CL	8086	—
		8088	—
		80286	—
		80386	7
		80486	3

SMSW Almacenar palabra del estado de la máquina

00001111 00000001 o0100mmm disp O D I T S Z A P C

(sólo se debe usar por el 80286)

Formato	Ejemplos	Relojes	
SMSW reg	SMSW AX SMSW DX SMSW CX	8086	—
		8088	—
		80286	2
		80386	10
		80486	2
SMSW mem	SMSW DATA	8086	—
		8088	—
		80286	3
		80386	3
		80486	3

STC Establecer la bandera de cargar

11111001		O D I T S Z A P C	
Ejemplo		1 Relojes	
STC	8086	2	
	8088	2	
	80286	2	
	80386	2	
	80486	2	
STD Establecer bandera de dirección			
11111101		O D I T S Z A P C	
Ejemplo		1 Relojes	
STD	8086	2	
	8088	2	
	80286	2	
	80386	2	
	80486	2	
STI Establecer interruptor para bandera			
11111011		O D I T S Z A P C	
Ejemplo		1 Relojes	
STI	8086	2	
	8088	2	
	80286	2	
	80386	3	
	80486	5	
STOS Almacenar información de la cadena			
1010101w		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
STOSB STOSW STOSD	STOSB	8086	11
	STOSW	8088	15
	STOSD	80286	3
	STOS DATA	80386	4
	REP STOSB	80486	5

STR		Almacenar registro para tareas	
00001111 · 00000000 oo001mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
STR reg	STR DX STR CX STR AX	8086	—
		8088	—
		80286	2
		80386	2
		80486	2
STR mem	STR DATA	8086	—
		8088	—
		80286	3
		80386	2
		80486	3

SUB		Restar	
001010dw oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
SUB reg,reg	SUB CL,DL SUB AX,DX SUB CH,CL SUB EAX,EBX	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
SUB mem,reg	SUB DATA,CL SUB BYTES,CX SUB NUMBS,ECX SUB [EAX],CX	8086	16+ea
		8088	24+ea
		80286	7
		80386	6
		80486	3
SUB reg,mem	SUB CL,DATA SUB CX,BYTES SUB ECX,NUMBS SUB CX,[EDX]	8086	9+ea
		8088	13+ea
		80286	7
		80386	7
		80486	2

10000sw 00101mmm disp información			
Formato	Ejemplos	Relojes	
SUB reg,imm	SUB CL,4 SUB DX,5 SUB CH,12 SUB SI,9	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
SUB mem,imm	SUB DATA,6 SUB BYTE PTR [DI],7 SUB NUMB,6 SUB WORD PTR [ECX],5	8086	17+ea
		8088	25+ea
		80286	7
		80386	7
		80486	3

0010110w información			
Formato	Ejemplos	Relojes	
SUB acc,imm	SUB AL,4 SUB AX,5 SUB AH,12 SUB AX,9	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

TEST Operandos de prueba (comparación lógica)

1000011w 00rrrrmmm disp			
Formato	Ejemplos	Relojes	
TEST reg,reg	TEST CL,DL TEST CX,DX TEST CL,CH TEST ECX,EBX	8086	5
		8088	5
		80286	2
		80386	2
		80486	1
TEST reg,mem mem,reg	TEST DATA,CL TEST CL,DATA	8086	9+ea
		8088	13+ea
		80286	6
		80386	5
		80486	2

O D I T S Z A P C
0 * * ? * 0

1111011w oo000mmm disp información			
Formato	Ejemplos	Relojes	
TEST reg,imm	TEST CL,4 TEST DX,5 TEST CH,12H TEST SI,256	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
TEST mem,imm	TEST DATA,6	8086	11+ea
		8088	11÷ea
		80286	6
		80386	5
		80486	2

1010100w infrmación			
Formato	Ejemplos	Relojes	
TEST acc,imm	TEST AL,4 TEST AX,5 TEST AH,12 TEST AX,9 TEST EAX,2	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

VERR/VERW Verificar leer o escribir

00001111 00000000 oo100mmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
VERR reg	VERR BX VERR CX VERR DX	8086	---
		8088	---
		80286	14
		80386	10
		80486	11
VERR mem	VERR DATA	8086	---
		8088	---
		80286	16
		80386	11
		80486	11

00001111 00000000 00101mmm disp			
Formato	Ejemplos	Relojes	
VERW reg	VERW AX VERW CX VERW DX	8086	—
		8088	—
		80286	14
		80386	15
		80486	11
VERW mem	VERW DATA	8086	—
		8088	—
		80286	16
		80386	16
		80486	11

WAIT Esperar para el coprocesador

10011011 O D I T S Z A P C			
Ejemplos	Relojes		
WAIT FWAIT	8086	4	
	8088	4	
	80286	3	
	80386	6	
	80486	6	

WBINVD Escribir e invalidar el cache para información

00001111 00001001 O D I T S Z A P C			
Ejemplo	Relojes		
WBINVD	8086	—	
	8088	—	
	80286	—	
	80386	—	
	80486	5	

XADD Intercambiar y sumar

0000111p ?!100000w 11rrrrr		O D I T S Z A P C			
Formato	Ejemplos	Relojes			
XADD reg,reg	XADD EBX,ECX XADD EDX,EAX XADD EDI,EBP	8086	—		
		8088	—		
		80286	—		
		80386	—		
		80486	3		
00001111 1100000w oorrmmm disp		Relojes			
Formato	Ejemplos	Relojes			
XADD mem,reg	XADD DATA,EAX XADD [DI],EAX XADD [ECX],EDX	8086	—		
		8088	—		
		80286	—		
		80386	—		
		80486	4		
XCHG Intercambiar					
1000011w 1oorrrmmm		O D I T S Z A P C			
Formato	Ejemplos	Relojes			
XCHG reg,reg	XCHG BL,CL XCHG AX,DX XCHG EDI,EBP	8086	4		
		8088	4		
		80286	3		
		80386	3		
		80486	3		
XCHG reg,mem mem,reg	XCHG CL,DATA XCHG DATA,CL XCHG DX,[DI] XCHG ECX,[EBP]	8086	17+ea		
		8088	25+ea		
		80286	5		
		80386	5		
		80486	5		

10010reg			
Formato	Ejemplos	Relojes	
XCHG acc,reg XCHG reg,acc	XCHG DATA,AL XCHG AX,FRIED	8086	3
		8088	3
		80286	3
		80386	3
		80486	3

XLAT Traducir

11010111		O D I T S Z A P C	
Ejemplo		Relojes	
XLAT		8086	11
		8088	11
		80286	5
		80386	3
		80486	4

XOR OR-exclusivo

001100dw oorrmmm disp		O D I T S Z A P C	
Formato	Ejemplos	Relojes	
XOR reg,reg	XOR BL,CL XOR CX,DX XOR CH,CL XOR EAX,EBX	8086	3
		8088	3
		80286	2
		80386	2
		80486	1
XOR mem,reg	XOR DATA,CL XOR BYTES,CX XOR NUMBS,ECX XOR [EAX],CX	8086	16+ea
		8088	24+ea
		80286	7
		80386	6
		80486	3

XOR reg,mem	XOR CL,DATA XOR CX,BYTES XOR ECX,NUMBS XOR CX,[EDX]	8086	9+ea
		8088	13+ea
		80286	7
		80386	7
		80486	2
100000sw oo110mmm disp información			
Formato	Ejemplos	Relojes	
XOR reg,imm	XOR BL,33 XOR CX,234H XOR CH,'A' XOR EAX,123445	8086	4
		8088	4
		80286	3
		80386	2
		80486	1
XOR mem,imm	XOR DATA,34 XOR BYTES,1234 XOR NUMBS,123 XOR [EAX],11	8086	17+ea
		8088	25+ea
		80286	7
		80386	7
		80486	3
0011010w información			
Formato	Ejemplos	Relojes	
XOR acc,imm	XOR AL,33 XOR AX,234H XOR AL,'A' XOR EAX,123445	8086	4
		8088	4
		80286	3
		80386	2
		80486	1

APENDICE C

RESUMEN DEL CONJUNTO DE INSTRUCCIONES DEL COPROCESADOR MATEMATICO

Las instrucciones para los coprocesadores 80387 compañero del 80386, del 80486DX y su coprocesador integrado, el 80486SX y su compañero coprocesador 80487SX se las presenta a continuación siendo estas básicamente las mismas que fueron proporcionadas por las versiones anteriores, con unas cuantas instrucciones adicionales.

El 80387, 80486 y 80486SX contienen las siguientes instrucciones adicionales: FCOST (coseno), FPREM1 (residuo parcial), FSDN (seno), FSINCOS (seno y coseno), y FUCOM/FUCOMP/FUCOMPP (una comparación desordenada). Las instrucciones de seno y coseno son los elementos más importante que se han agregado al conjunto de instrucciones. En las versiones anteriores del coprocesador era necesario calcular seno y coseno de la tangente.

A continuación se indican los conjuntos de instrucciones para todas las versiones del coprocesador. También indica el número de períodos del reloj necesarios para ejecutar cada instrucción.

F2XM1 2^{ST-1}			
11011001 11110000			
Ejemplo		Ciclos de reloj	
F2XM1	8087	310—630	
	80287	310—630	
	80387	211—476	
	80486/7	140—279	
FABS Valor absoluto de ST			
11011001 11100001			
Ejemplo		Ciclos de reloj	
FABS	8087	10—17	
	80287	10—17	
	80387	22	
	80486/7	3	
FADD/FADDP/FIADD Suma			
11011000	oo000mmm disp	Memoria de 32 bits (FADD)	
11011100	oo000mmm disp	Memoria de 64 bits (FADD)	
11011000	11000rrr	FADD ST,ST(rrr)	
11011110	11000rrr	FADDP ST,ST(rrr)	
11011110	oo000mmm disp	Memoria de 16 bits (FIADD)	
11011010	oo000mmm disp	Memoria de 32 bits (FIADD)	
Formato	Ejemplos	Ciclos de reloj	
FADD FADDP FIADD	DATOS FADD	8087	70—143
	FADD ST,ST(1)	80287	70—143
	FADDP		
	NUMERO FIADD	80387	23—72
	FADD ST,ST(3)	80486/7	8—20
FADDP ST,ST(2)			
FADD ST(2),ST			
FCLEX/FNCLEX Borrar errores			
11011011 11100010			
Ejemplo		Ciclos de reloj	
FCLEX FNCLEX	8087	2—8	
	80287	2—8	
	80387	11	
	80486/7	7	

FCOM/FCOMP/FCOMPP/FICOM/FICOMP		Comparar	
11011000	oo010mmm disp	Memoria de 32 bits (FCOM)	
11011100	oo010mmm disp	Memoria de 64 bits (FCOM)	
11011000	11010rrr	FCOM ST(rrr)	
11011000	oo011mmm disp	Memoria de 32 bits (FCOMP)	
11011100	oo011mmm disp	Memoria de 64 bits (FCOMP)	
11011000	11011rrr	FCOMP ST(rrr)	
11011110	11011001	FCOMPP	
11011110	oo010mmm disp	Memoria de 16 bits (FICOM)	
11011010	oo010mmm disp	Memoria de 32 bits (FICOM)	
11011110	oo011mmm disp	Memoria de 16 bits (FICOMP)	
11011010	oo011mmm disp	Memoria de 32 bits (FICOMP)	
Formato	Ejemplos	Ciclos de reloj	
FCOM	FCOM ST(2)	8087	40—93
FCOMP	FCOMP DATO		
FCOMPP	FCOMPP	80287	40—93
FICOM	FICOM NUMERO		
FICOMP	FICOMP DATO3	80387	24—63
		80486/7	15—20
FCOS		Coseno de ST	
11011001	11111111		
Ejemplo	Ciclos de reloj		
FCOS	8087	—	
	80287	—	
	80387	123—772	
	80486/7	193—279	
FDECSTP		Decrementa el apuntador de pila	
11011001	11110110		
Ejemplo	Ciclos de reloj		
FDECSTP	8087	6—12	
	80287	6—12	
	80387	22	
	80486/7	3	
FDISI/FNDISI		Deshabilita interrupciones	
11011011	11100001		
(se ignoran en el 80287, 80387 y 80486/7)			

Ejemplo	Ciclos de reloj	
FDISI	8087	2—8
FNDISI	80287	—
	80387	—
	80486/7	—

FDIV/FDIVP/FIDIV División

11011000	oo110mmm disp	Memoria de 32 bits (FDIV)
11011100	oo100mmm disp	Memoria de 64 bits (FDIV)
11011d00	11111rrr	FDIV ST,ST(rrr)
11011110	11111rrr	FDIVP ST,ST(rrr)
11011110	oo110mmm disp	Memoria de 16 bits (FDIV)
11011010	oo110mmm disp	Memoria de 32 bits (FDIV)

Formato	Ejemplos	Ciclos de reloj	
FDIV	FDIV DATO	8087	191—243
FDIVP	FDIV ST,ST(3)	80287	191—243
FIDIV	FDIVP		
	FDIV NUMERO	80387	88—140
	FDIV ST,ST(5)	80486/7	8—89
	FDIVP ST,ST(2)		
	FDIV ST(2),ST		

FDIVR/FDIVRP/FIDIVR División inversa

11011000	oo111mmm disp	Memoria de 32 bits (FDIVR)
11011100	oo111mmm disp	Memoria de 64 bits (FDIVR)
11011d00	11110rrr	FDIVR ST,ST(rrr)
11011110	11110rrr	FDIVRP ST,ST(rrr)
11011110	oo111mmm disp	Memoria de 16 bits (FDIVR)
11011010	oo111mmm disp	Memoria de 32 bits (FDIVR)

Formato	Ejemplos	Ciclos de reloj	
FDIVR	FDIVR DATO	8087	191—243
FDIVRP	FDIVR ST,ST(3)	80287	191—243
FIDIVR	FDIVRP		
	FDIVR NUMERO	80387	88—140
	FDIVR ST,ST(5)	80486/7	8—89
	FDIVRP ST,ST(2)		
	FDIVR ST(2),ST		

FENI/FNENI Deshabilita interrupciones

11011011 11100000
 (se ignoran en el 80287, 80387 y 80486/7)

Ejemplo	Ciclos de reloj	
FENI	8087	2—8
FNENI	80287	—
	80387	—
	80486/7	—

FFREE Registro libre			
11011101 11000rrr			
Formato	Ejemplos	Ciclos de reloj	
FFREE	FFREE FFREE ST(1) FFREE ST(2)	8087	9—16
		80287	9—16
		80387	18
		80486/7	3
FINCSTP Incrementa el apuntador de pila			
11011001 11110111			
Ejemplo	Ciclos de reloj		
FINCSTP	8087	6—12	
	80287	6—12	
	80387	21	
	80486/7	3	
FINIT/FNINIT Inicializar coprocesador			
11011001 11110110			
Ejemplo	Ciclos de reloj		
FINIT FNINIT	8087	2—8	
	80287	2—8	
	80387	33	
	80486/7	17	
FLD/FILD/FBLD Cargar datos a ST(0)			
11011001 0000mmm disp Memoria de 32 bits (FLD) 11011101 0000mmm disp Memoria de 64 bits (FLD) 11011011 00101mmm disp Memoria de 80 bits (FLD) 11011111 00000mmm disp Memoria de 16 bits (FILD) 11011011 00000mmm disp Memoria de 32 bits (FILD) 11011111 00101mmm disp Memoria de 64 bits (FILD) 11011111 00100mmm disp Memoria de 80 bits (FBLD)			
Formato	Ejemplos	Ciclos de reloj	
FLD FILD FBLD	FLD DATO FILD DATO1 FBLD DEC_DATO	8087	17—310
		80287	17—310
		80387	14—275
		80486/7	3—103

FLD1 Cargar $\div 1.0$ a ST(0)		
11011001 11101000		
Ejemplo	Ciclos de reloj	
FLD1	8087	15--21
	80287	15--21
	80387	24
	80486/7	4
FLDZ Cargar $\div 0.0$ a ST(0)		
11011001 11101110		
Ejemplo	Ciclos de reloj	
FLDZ	8087	11--17
	80287	11--17
	80387	20
	80486/7	4
FLDPI Cargar π a ST(0)		
11011001 11101011		
Ejemplo	Ciclos de reloj	
FLDPI	8087	16--22
	80287	16--22
	80387	40
	80486/7	8
FLDL2E Cargar $\log_2 e$ a ST(0)		
11011001 11101010		
Ejemplo	Ciclos de reloj	
FLDL2E	8087	15--21
	80287	15--21
	80387	40
	80486/7	8

FLDL2T Cargar log ₂ 10 a ST(0)			
11011001 11101001			
Ejemplo		Ciclos de reloj	
FLDL2T		8087	16—22
		80287	16—22
		80387	40
		80486/7	8
FLDLG2 Cargar log ₂ 2 a ST(0)			
11011001 11101000			
Ejemplo		Ciclos de reloj	
FLDLG2		8087	16—24
		80287	16—24
		80387	41
		80486/7	8
FLDLN2 Cargar log ₂ 2 a ST(0)			
11011001 11101101			
Ejemplo		Ciclos de reloj	
FLDLN2		8087	17—23
		80287	17—23
		80387	41
		80486/7	8
FLDCW Cargar registro de control			
11011001 00101mmm disp			
Formato	Ejemplos	Ciclos de reloj	
FLDCW	FLDCW DATO FLDCW ESTADO	8087	7—14
		80287	7—14
		80387	19
		80486/7	4

FLDENV Cargar ambiente			
11011001 oo100mmm disp			
Formato	Ejemplos	Ciclos de reloj	
FLDENV	FLDENV AMBIENTE FLDENV DATO	8087	35—45
		80287	25—45
		80387	71
		80486/7	34—44

FMUL/FMULP/FIMUL Multiplicación			
11011000	oo001mmm disp	Memoria de 32 bits (FMUL)	
11011100	oo001mmm disp	Memoria de 64 bits (FMUL)	
11011000	11001rr	FMUL ST,ST(rr)	
11011110	11001rr	FMULP ST,ST(rr)	
11011110	oo001mmm disp	Memoria de 16 bits (FMUL)	
11011010	oo001mmm disp	Memoria de 32 bits (FMUL)	
Formato	Ejemplos	Ciclos de reloj	
FMUL FMULP FIMUL	FMUL DATO FMUL ST,ST(2) FMULP ST,ST FIMUL DATOS	8087	110—168
		80287	110—168
		80387	23—32
		80486/7	11—27

FNOP No operación			
11011001 11010000			
Ejemplo		Ciclos de reloj	
FNOP		8087	10—16
		80287	10—16
		80387	12
		80486/7	3

FPATAN Arco tangente parcial de ST(0)			
11011001 11110011			
Ejemplo		Ciclos de reloj	
FPATAN..		8087	250—800
		80287	250—800
		80387	314—487
		80486/7	218—303

FPREM Residuo parcial

11011001 11111000

Ejemplo	Ciclos de reloj	
FPREM	8087	15—190
	80287	15—190
	80387	74—155
	80486/7	70—138

FPREM1 Residuo parcial (IEEE)

11011001 11110101

Ejemplo	Ciclos de reloj	
FPREM1	8087	—
	80287	—
	80387	95—185
	80485/7	72—167

FPTAN Tangente parcial de ST(0)

11011001 11110010

Ejemplo	Ciclos de reloj	
FPTAN	8087	30—450
	80287	30—450
	80387	191—497
	80486/7	200—273

FRNDINT Redondear ST(0) a número entero

11011001 11111100

Ejemplo	Ciclos de reloj	
FRNDINT	8087	16—50
	80287	16—50
	80387	66—80
	80486/7	21—30

FRSTOR Recuperar estado			
11011101 oo110mmm disp			
Formato	Ejemplos	Ciclos de reloj	
FRSTOR	FRSTOR DATO FRSTOR ESTADO FRSTOR MAQUINA	8087	197—207
		80287	197—207
		80387	308
		80486/7	120—131
FSAVE/FNSAVE Salvar estado de la máquina			
11011101 oo110mmm disp			
Formato	Ejemplos	Ciclos de reloj	
FSAVE FNSAVE	FSAVE ESTADO FNSAVE ESTADO FSAVE MAQUINA	8087	197—207
		80287	197—207
		80387	375
		80486/7	143—154
FSCALE Escalar ST(0) por ST(1)			
11011001 11111101			
Ejemplo	Ciclos de reloj		
FSCALE	8087 32—38		
	80287 32—38		
	80387 67—86		
	80486/7 30—32		
FSETPM Inicializar modo protegido			
11011011 11100100			
Ejemplo	Ciclos de reloj		
FSETPM	8087 —		
	80287 2—18		
	80387 12		
	80486/7 —		

FSIN Seno de ST(0)		
11011001 11111110		
Ejemplo	Ciclos de reloj	
FSIN	8087	—
	80287	—
	80387	122—771
	80486/7	193—279
FSINCOS Encontrar seno y coseno de ST(0)		
11011001 11111011		
Ejemplo	Ciclos de reloj	
FSINCOS	8087	—
	80287	—
	80387	194—809
	80486/7	243—329
FSQRT Raíz cuadrada de ST(0)		
11011001 11111010		
Ejemplo	Ciclos de reloj	
FSQRT	8087	180—186
	80287	180—186
	80387	122—129
	80486/7	83—87
FST/FSTP/FIST/FISTP/FBSTP Almacenar		
11011001	00010mmm disp	Memoria de 32 bits (FST)
11011101	00010mmm disp	Memoria de 64 bits (FST)
11011101	11010rrr	FST ST(rrr)
11011001	00011mmm disp	Memoria de 32 bits (FSTP)
11011101	00011mmm disp	Memoria de 64 bits (FSTP)
11011011	00111mmm disp	Memoria de 80 bits (FSTP)
11011101	11001rrr	FSTP ST(rrr)
11011111	00010mmm disp	Memoria de 16 bits (FIST)
11011011	00010mmm disp	Memoria de 32 bits (FIST)
11011111	00011mmm disp	Memoria de 16 bits (FISTP)
11011011	00011mmm disp	Memoria de 32 bits (FISTP)
11011111	00111mmm disp	Memoria de 64 bits (FISTP)
11011111	00110mmm disp	Memoria de 80 bits (FBSTP)

Formato	Ejemplos	Ciclos de reloj	
FST FSTP FIST FISTP FBSTP	FST DATO FST ST(3) FST FSTP FIST DATO2 FBSTP DATO6 FISTP DATO9	8087	15—540
		80287	15—540
		80387	11—534
		80486/7	3—176

FSTCW/FNSTCW Almacenar registro de control

11011001 oo111mmm disp

Formato	Ejemplos	Ciclos de reloj	
FSTCW FNSTCW	FSTCW CONTROL FNSTCW ESTADO FSTCW MAQUINA	8087	12—18
		80287	12—18
		80387	15
		80486/7	3

FSTENV/FNSTENV Almacenar ambiente

11011001 oo110mmm disp

Formato	Ejemplos	Ciclos de reloj	
FSTENV FNSTENV	FSTENV CONTROL FNSTENV ESTADO FSTENV MAQUINA	8087	40—50
		80287	40—50
		80387	103—104
		80486/7	58—67

FSTSW/FNSTSW Almacenar registro de estado

11011101 oo111mmm disp

Formato	Ejemplos	Ciclos de reloj	
FSTSW FNSTSW	FSTSW CONTROL FNSTSW ESTADO FSTSW MAQUINA	8087	12—18
		80287	12—18
		80387	15
		80486/7	3

FSUB/FSUBP/FISUB Restar

11011000 oo100mmm disp	Memoria de 32 bits (FSUB)
11011100 oo100mmm disp	Memoria de 64 bits (FSUB)
11011d00 11101rrr	FSUB ST,ST(rrr)
11011110 11101rrr	FSUBP ST,ST(rrr)
11011110 oo100mmm disp	Memoria de 16 bits (FISUB)
11011010 oo100mmm disp	Memoria de 32 bits (FISUB)

Formato	Ejemplos	Ciclos de reloj	
FSUB FSUBP FISUB	FSUB DATO	8087	70—143
	FSUB ST,ST(2)	80287	70—143
	FSUB ST(2),ST		
	FSUBP	80387	29—82
FISUB DATO3	80486/7	6—35	

FSUBR/FSUBRP/FISUBR Resta inversa

11011000	oo101mmm disp	Memoria de 32 bits (FSUBR)
11011100	oo101mmm disp	Memoria de 64 bits (FSUBR)
11011d00	11100rrr	FSUBR ST,ST(rrr)
11011110	11100rrr	FSUBRP ST,ST(rrr)
11011110	oo101mmm disp	Memoria de 16 bits (FISUBR)
11011010	oo101mmm disp	Memoria de 32 bits (FISUBR)

Formato	Ejemplos	Ciclos de reloj	
FSUBR FSUBRP FISUBR	FSUBR DATO	8087	70—143
	FSUBR ST,ST(2)	80287	70—143
	FSUBR ST(2),ST		
	FSUBRP	80387	29—82
FISUBR DATO3	80486/7	6—35	

FTST Comparar ST(0) con + 0.0

11011001 11100100

Ejemplo	Ciclos de reloj	
FTST	8087	36—48
	80287	36—48
	80387	28
	80486/7	4

FUCOM/FUCOMP/FUCOMPP Comparación desordenada

11011101	11100rrr	FUCOM ST,ST(rrr)
11011101	11101rrr	FUCOMP ST,ST(rrr)
11011101	11101001	FUCOMPP

Formato	Ejemplos	Ciclos de reloj	
FUCOM FUCOMP FUCOMPP	FUCOM ST,ST(2)	8087	—
	FUCOM	80287	—
	FUCOMP ST,ST(3)		
	FUCOMP	80387	24—26
FUCOMPP	80486/7	4—5	

FWAIT Esperar

10011011

Ejemplo	Ciclos de reloj	
FWAIT	8087	4
	80287	3
	80387	6
	80486/7	i-3

FXAM Examinar ST(0)

11011001 11100101

Ejemplo	Ciclos de reloj	
FXAM	8087	12-23
	80287	12-23
	80387	30-38
	80486/7	8

FXCH Intercambiar ST(0) con otro registro

11011001 11001rrr FXCH ST,ST(rrr)

Formato	Ejemplos	Ciclos de reloj	
FXCH	FXCH,ST,ST(1) FXCH FXCH ST,ST(4)	8087	10-15
		80287	10-15
		80387	18
		80486/7	4

FXTRACT Extraer componentes de ST(0)

11011001 11110100

Ejemplo	Ciclos de reloj	
FXTRACT	8087	27-55
	80287	27-55
	80387	70-76
	80486/7	16-20

FYL2X $ST(1) \times \log_2 ST(0)$

11011001 11110001		
Ejemplo	Ciclos de reloj	
FYL2X	8087	900—1100
	80287	900—1100
	80387	120—538
	80486/7	196—329
FXL2XP1 $ST(1) \times \log_2 [ST(0) + .1.0]$		
11011001 11111001		
Ejemplo	Ciclos de reloj	
FXL2XP1	8087	700—1000
	80287	700—1000
	80387	257—547
	80486/7	171—326

Nota: d = dirección, donde d = 0 para ST como el destino y d = 1 para ST como fuente, rrr = número de registro de punto decimal flotante, oo = modo, mmm = campo r/m, y disp = desplazamiento.

APENDICE D

DIRECTIVOS COMUNES DEL ENSAMBLADOR MASM611

DIRECTIVOS COMUNES DEL ENSAMBLADOR MASM611

DIRECTIVO	FUNCION
.286	Selecciona el conjunto de instrucciones para el 80286
.286P	Selecciona el conjunto de instrucciones en modo protegido para el 80286
.386	Selecciona el conjunto de instrucciones para el 80386
.386P	Selecciona el conjunto de instrucciones en modo protegido para el 80386
.486	Selecciona el conjunto de instrucciones para el 80486
.486P	Selecciona el conjunto de instrucciones en modo protegido para el 80486
.287	Selecciona el coprocesador numérico 80287
.387	Selecciona el coprocesador numérico 80387
ALING2	Inicia los datos en un segmento con límites de palabra o doble palabra
ASSUME	Indica los nombres de cada segmento al ensamblador; no carga los registros de segmento
AT	Indica la dirección física de cada segmento que se emplea con el enunciado SEGMENT
BYTE	Indica un operando de tamaño de byte como en BYTE PTR o THIS BYTE
DB	Define uno o más bytes (8 bits)
DD	Define palabra o dobles palabras (32 bits)
DQ	Define palabra o cuádruples palabras (64 bits)
DT	Define diez bytes (80 bits)
DUP	Genera duplicación de caracteres o números
DW	Define palabra o palabras (16 bits)
DWORD	Indica un operando de tamaño de doble palabra como en THIS DWORD
END	Indica el final del programa
ENDM	Indica el final de una secuencia de macro
ENDP	Indica el final de un procedimiento
ENDS	Indica el final de un segmento
EQU	Iguala los datos con los de una etiqueta
FAR	Identifica una dirección lejana como en JMP FAR PTR LISTAS
MACRO	Define el nombre, parámetros e inicio de un macro
NEAR	Especifica una dirección cercana como en JMP NEAR PTR AYUDA
OFFSET	Especifica una dirección de desplazamiento
ORG	Inicializa el origen dentro de un segmento
PROC	Define el inicio de un procedimiento
PTR	Indica un apuntador a la memoria
SEGMENT	Define el comienzo de un segmento de memoria
STACK	Indica que un segmento es segmento de pila
STRUC	Define el comienzo de una estructura de datos
THIS	Se emplea con EQU para establecer una etiqueta de un Byte, palabra o doble palabra
USES	Un directivo de la versión 6.0 de MASM que salva en forma automática los registros utilizados en un procedimiento
USE16	Ordena al ensamblador que utilice el modo de instrucción y tamaños de datos de 16 bits para 80386 y 80486
USE32	Ordena al ensamblador que utilice el modo de instrucción y tamaños de datos de 32 bits para 80386 y 80486
WORD	Actúa como operando palabra como en WORD PTR o THIS WORD

APENDICE E

FUNCIONES DE LA INTERRUPCION 10H DEL BIOS

00H	SELECCIONAR EL MODO DE VIDEO
Entrada	AH = 00H AL = número de modo
Salida	Modo cambiado y la pantalla borrada
01H	SELECCIONAR TIPO DE CURSOR
Entrada	AH = 01H CH = número de la línea para iniciar CL = número de la última línea
Salida	Tamaño del cursor modificado
02H	SELECCIONAR POSICION DEL CURSOR
Entrada	AH = 02H BH = número de página (normalmente 0) DH = número de hilera (comenzando con 0) DL = número de columna (comenzando con 0)
Salida	Cambia el cursor a una nueva posición
03H	LEER POSICION DEL CURSOR
Entrada	AH = 03H BH = número de página
Salida	CH = línea inicial (tamaño del cursor) CL = línea final (tamaño del cursor) DH = hilera actual DL = columna actual
04H	LEER LAPIZ DE LUZ
Entrada	AH = 04H (no apoyado en VGA)

Salida	AH = 0, activa el lápiz de luz BX = columna de pixels CX = hilera de pixels DH = hilera de caracteres DL = columna de caracteres
05H	SELECCIONAR PAGINA DE LA PANTALLA
Entrada	AH = 05H AL = número de página
Salida	Número de página seleccionado. A continuación están los números de página válidos. Modo 0 y 1 apoyan las páginas 0-7 Modo 2 y 3 apoyan las páginas 0-7 Modo 4, 5 y 6 apoyan la página 0 Modo 7 y D apoyan las páginas 0-7 Modo E apoya las páginas 0-3 Modo F y 10 apoyan las páginas 0-1 Modo 11, 12 y 13 apoyan la página 0
06H	BUSCAR CON AVANCE PAGINA
Entrada	AH = 06H AL = número de líneas para buscar (0 borra la ventana) BH = atributo de caracteres para las nuevas líneas CH = hilera superior para buscar en la ventana CL = columna izquierda de la ventana para buscar DH = última hilera de la ventana para buscar DL = columna derecha de la ventana para buscar
Salida	Desplaza la ventana de abajo hasta arriba de la pantalla. Líneas en blanco llenan la parte inferior utilizando el atributo de caracter en BH.
07H	BUSCAR CON RETROCESO PAGINA
Entrada	AH = 07H AL = número de líneas para buscar (0 borra a ventana) BH = atributo del carácter para las líneas nuevas CH = hilera superior para de la ventana para buscar CL = columna izquierda para la ventana para buscar DH = última hilera de la ventana para buscar DL = columna derecha de la ventana para buscar
Salida	Desplaza la ventana de arriba a abajo de la pantalla. Líneas en blanco llenan desde arriba utilizando el atributo del carácter en BH.
08H	LEER ATRIBUTO/CARACTER EN LA POSICION ACTUAL DEL CURSOR
Entrada	AH = 08H BH = número de la página

Salida	AL = código del carácter ASCII AH = atributo del carácter Nota: Esta función no avanza al cursor.
09H	ESCRIBIR ATRIBUTO/CARACTER EN LA POSICION ACTUAL DEL CURSOR
Entrada	AH = 09H AL = código del carácter ASCII BH = número de la página BL = atributo del carácter CX = número de caracteres para escribir
Salida	Nota: Esta función no avanza el cursor.
0AH	ESCRIBIR CARACTER EN LA POSICION ACTUAL DEL CURSOR
Entrada	AH = 0AH AL = código del carácter ASCII BH = número de la página CX = número de caracteres para escribir
Salida	Nota: Esta función no avanza el cursor.
0FH	LEER EL MODO DE VIDEO
Entrada	AH = 0FH
Salida	AL = modo de video actual AH = número de columnas de caracteres BH = número de la página
10H	ESTABLECER EL REGISTRO DE LA PALETA VGA
Entrada	AH = 10H AL = 10H BX = número de color (0-255) CH = verde (0-63) CL = azul (0-63) DH = rojo (0-63)
Salida	El color del registro en paleta será cambiado. Nota: los primeros 16 colores (0-15) son utilizados en el modo de texto VGA de 16 colores y otros modos.
10H	LEER EL REGISTROS DE LA PALETA VGA
Entrada	AH = 10H AL = 15H BX = número de color (0-255)

Salida	CH = verde CL = azul DH = rojo
11H	OBTENER EL CONJUNTO DE CARACTERES ROM
Entrada	AH = 11H AL = 30H BH = 2 = conjunto de caracteres 8 X 14 de ROM BH = 3 = conjunto de caracteres 8 X 8 de ROM BH = 4 = conjunto de caracteres extendido 8 X 8 de ROM BH = 5 = conjunto de caracteres 9 X 14 de ROM BH = 6 = conjunto de caracteres 8 X 16 de ROM BH = 7 = conjunto de caracteres 9 X 16 de ROM
Salida	CX = bytes por carácter DL = líneas por carácter ES:SP = dirección del conjunto de caracteres

TABLA A-5 Funciones extendidas de VGA

BX	Función
100H	640 X 400 con 256 colores
101H	640 X 480 con 256 colores
102H	800 X 600 con 16 colores
103H	800 X 600 con 256 colores
104H	1.024 X 768 con 16 colores
105H	1.024 X 768 con 256 colores
106H	1.280 X 1.024 con 16 colores
107H	1.280 X 1.024 con 256 colores
108H	80 X 60 en el modo texto
109H	132 X 25 en el modo texto
10AH	132 X 43 en el modo texto
10BH	132 X 50 en el modo texto
10CH	132 X 60 en el modo texto

APENDICE F

FUNCIONES DE LA INTERRUPCION 21H DEL DOS

00H	TERMINAR UN PROGRAMA
Entrada	AH = 00H CS = dirección del prefijo para el segmento del programa
Salida	Se tecldea DOS
01H	LEER EL TECLADO
Entrada	AH = 01H
Salida	AL = caracter ASCII
Notas	Cuando la solicitud de función AL = 00H se puede reutilizar para leer un caracter ASCII extendido. Refiérase al Capítulo 6, Tabla 6-1, para un listado de los códigos del teclado extendido ASCII. Esta solicitud de una función automáticamente hace eco cuando esta escrita en la pantalla de video.
02H	ESCRIBIR AL EQUIPO ESTANDAR DE SALIDA DE INFORMACION
Entrada	AH = 02H AL = caracter ASCII que se debe mostrar
Notas	Esta solicitud de la función normalmente muestra información en la pantalla de video.
03H	LEER CARACTER DE COM1
Entrada	AH = 03H
Salida	AL = caracter ASCII leído del puerto de comunicaciones
Notas	Esta solicitud de función lee información del puerto serial de comunicaciones.
04H	ESCRIBIR A COM1
Entrada	AH = 04H DL = caracter que se debe enviar a COM1
Notas	Esta función transmite información por medio del puerto de comunicaciones serial.

05H	ESCRIBIR A LPT1
Entrada	AH = 05H DL = carácter ASCII que se debe imprimir
Notas	Imprime DL en la impresora de línea conectada a LPT1.
06H	DIRIGIR LEER/ESCRIBIR DE LA CONSOLA
Entrada	AH = 06H DL = OFFH o DL = carácter ASCII
Salida	AL = carácter ASCII
Notas	<p>Cuando DL = OFFH se usa, entonces esta función lee la consola. Cuando DL = carácter ASCII, entonces esta función muestra al carácter ASCII en la pantalla de video de la consola.</p> <p>Cuando un caracter se lee del teclado de la consola, la bandera cero (ZF) indica cuándo un caracter fue teclado. Una condición cero indica que ninguna tecla está escrita y una condición no cero indica que AL contiene el código ASCII de la tecla o un 00H. Cuando AL = 00H, la función debe ser reusada para leer un carácter ASCII extendido del teclado. Observe que la tecla no manda eco a la pantalla de video.</p>
07H	DIRIGIR LA ENTRADA DE INFORMACION A LA CONSOLA SIN ECO
Entrada	AH = 07H
Salida	AL = carácter ASCII
Notas	Funcionan exactamente como el número de función 06H con DL = OFFH, pero no regresan de la función hasta que la tecla sea oprimida.
08H	LEER LA ENTRADA DE INFORMACION ESTANDAR SIN ECO
Entrada	AH = 08H
Salida	AL = carácter ASCII
Notas	Funciona como la función 07H, excepto que lee al equipo de entrada de información estándar. El aparato de entrada de información estándar puede ser asignado ya sea como el teclado o el puerto COM. Esta función también responde a un Ctrl-break, en donde las funciones 06H y 07H no lo hacen. Un Ctrl-break ocasiona que se ejecute INT 23H.
09H	MOSTRAR UNA CADENA DE CARACTERES
Entrada	AH = 09H DS:DX = dirección de la cadena de caracteres
Notas	La cadena de caracteres debe terminar con un ASCII \$ (24H). La cadena de caracteres puede ser de cualquier tamaño y puede contener caracteres de control tal como el regreso de línea (0DH) y la alimentación de línea (0AH).

0AH	LA ENTRADA DE INFORMACION DEL TECLADO CON BUFER
Entrada	AH = 0AH DS:DX = dirección del búfer para la entrada de información del teclado
Notas	El primer byte del búfer contiene el tamaño del búfer (hasta 255). El segundo byte está ocupado con el número de caracteres escritos al regresar. Desde el tercer byte hasta el final del búfer contiene la cadena de caracteres escrita seguida por un retorno (0DH). Esta función continúa leyendo el teclado (mostrando información como fue teclada) hasta que el número especificado de caracteres sea escrito o hasta que la tecla de retorno (enter) sea teclada.
0BH	ESTADO DE LA PRUEBA DEL EQUIPO ESTANDAR PARA LA ENTRADA DE INFORMACION
Entrada	AH = 0BH
Salida	AL = estado del equipo para la entrada de información
Notas	Esta función prueba al equipo estándar de entrada de información para determinar si hay información disponible. Cuando AL = 00, no hay información disponible. Cuando AL = 0FFH, entonces cuando hay información disponible que debe ser aceptada utilizando el número de la función 0BH.
0CH	BORRAR EL BUFER DEL TECLADO E INVOCAR LA FUNCION DEL TECLADO
Entrada	AH=0CH AL = 01H, 06H, 07H o 0AH
Salida	véase la salida para las funciones 01H, 06H, 07H o 0AH
Notas	El búfer del teclado retiene las teclas mientras que los programas ejecutan otras tareas. Esta función vacía o borra al búfer y después invoca a la función del teclado ubicado en el registro AL.
0DH	ELIMINAR A LOS BUFERES DEL DISCO
Entrada	AH = 0DH
Notas	Borra todos los nombres de archivos almacenados en los búferes del disco. Esta función no cierra los archivos especificados por los búferes del disco, así que se debe tener cuidado con su uso.
0EH	SELECCIONAR UNIDAD PREDETERMINADA DE DISCO
Entrada	AH = 0EH DL = el número deseado de la unidad predeterminada de disco
Salida	AL = el número total de lectores de disco existentes en el sistema
Notas	Lector de disco A = 00H, lector de disco B = 01H, lector de disco C = 02H, etc.

0FH		@ABRIR ARCHIVO CON FCB
Entrada	AH = 0FH DS:DX = dirección del bloque para control de un archivo no abierto (FCB)	
Salida	AL = 00H cuando el archivo es encontrado AL = 0FFH cuando el archivo no es encontrado	
Notas	El bloque de control de archivo (FCB) es usado sólo con el software del DOS antiguo y nunca se debe utilizar con programas nuevos. Los bloques para control de archivo no permiten nombres de rutas como lo hacen los códigos de función de archivo más nuevos presentados más adelante. La Figura A-2 muestra la estructura del FCB. Para abrir un archivo el archivo debe estar presente en el disco o ser creado con una solicitud de la función 16H.	
10H		@CERRAR EL ARCHIVO CON FCB
Entrada	AH = 10H DS:DX = dirección del bloque para control del archivo abierto (FCB)	
Salida	AL = 00H cuando el archivo está cerrado AL = 0FFH cuando es encontrado un error	
Notas	Los errores que ocurren normalmente indican que el disco está lleno o que el medio está mal.	
11H		@BUSCAR LA PRIMERA COMBINACION (FCB)
Entrada	AH = 11H DS:DX = dirección del bloque para control de archivo que se debe buscar	
Salida	AL = 00H cuando el archivo fue encontrado AL = 0FFH cuando el archivo no fue encontrado	

FIGURA A-2 Contenido del bloque para control de archivos (FCB).

Desplazamiento	Contenido
00H	Lector de disco
01H	Nombre de archivo de 8-caracteres
09H	Extensión para archivo de 3-caracteres
0CH	Número de bloque actual
0EH	Tamaño del registro
10H	Tamaño del archivo
14H	Fecha de creación
16H	Espacio reservado
20H	Número de registro actual
21H	Número de registro relativo

Notas	Los caracteres de comodines (? o *) se pueden utilizar para buscar el nombre de un archivo. El caracter de comodín ? iguala a cualquier carácter y el * iguala a cualquier nombre o extensión.
12H	@BUSCAR LA PROXIMA COMBINACION (FCB)
Entrada	AH = 12H DS:DX = dirección del bloque de control para archivo que tiene que ser buscado
Salida	AL = 00H cuando el archivo fue encontrado AL = 0FFH cuando el archivo no fue encontrado
Notas	Esta función es usada después de que la función 11H encuentra el primer nombre de archivo que combine.
13H	@BORRAR EL ARCHIVO USANDO FCB
Entrada	AH = 13H DS:DX = dirección del bloque para control de archivo que se tiene que borrar
Salida	AL = 00H cuando el archivo fue borrado AL = 0FFH cuando ocurrió un error
Notas	Los errores que ocurren mas frecuentemente son errores del medio defectuoso.
14H	@LEER EN FORMA SECUENCIAL (FCB)
Entrada	AH = 14H DS:DX = dirección del bloque para control de archivo que tiene que ser leído
Salida	AL = 00H cuando es leído exitosamente AL = 01H cuando se alcanza el final del archivo AL = 02H cuando DTA tiene un ajuste de segmento AL = 03H cuando menos de 128 bytes fueron leídos
15H	@ESCRIBIR EN FORMA SECUENCIAL (FCB)
Entrada	AH = 15H DS:DX = dirección del bloque para control de archivo que tiene que escribirse
Salida	AL = 00H cuando se escribió con éxito AL = 01H cuando el disco está lleno AL = 02H cuando DTA tuvo un ajuste del segmento
16H	@CREAR UN ARCHIVO (FCB)
Entrada	AH = 16H DS:DX = dirección de un bloque para control de un archivo no abierto
Salida	AL = 00H cuando el archivo fue creado AL = 01H cuando el disco está lleno

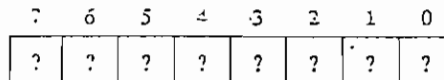
17H	@RENOMBRAR UN ARCHIVO (FCB)
Entrada	AH = 17H DS:DX = dirección de un bloque para control de un archivo modificado
Salida	AL = 00H cuando el archivo es renombrado AL = 01H cuando ocurrió un error
Notas	Refiérase a la Figura A-3 para el FCB modificado utilizado para renombrar un archivo.
18H	NO ASIGNADO
19H	REGRESAR EL LECTOR DE DISCO ACTUAL
Entrada	AH = 19H
Salida	AL = lector de disco actual
Notas	AL = 00H para el lector de disco A, 01H para el lector de disco B, etc.
1AH	ESTABLECER EL AREA DE TRANSFERENCIA DEL DISCO
Entrada	AH = 1AH DS:DX = dirección para el DTA nuevo
Notas	El área de transferencia del disco está normalmente ubicada dentro del prefijo del segmento del programa en la dirección desolazada 80H. El DTA se utiliza por DOS para todas las transferencias de información del disco utilizando bloques de control del archivo.
1BH	CONSEGUIR LA TABLA DE ASIGNACIONES (FAT) PARA LA UNIDAD PREDETERMINADA
Entrada	AH = 1BH

FIGURA A-3 Contenido del bloque para control de archivo modificado (FCB).

Desplazamiento	Contenido
00H	Lector de disco
01H	Nombre de archivo de 8-caracteres
09H	Extensión de 3-caracteres
0CH	Número de bloque actual
0EH	Tamaño del registro
10H	Tamaño del archivo
14H	Fecha de creación
16H	Segundo nombre de archivo

Salida	AL = número de sectores por unidades DS:BX = dirección del descriptor del medio CX = tamaño de un sector en bytes DX = número de unidades en un lector de disco
Notas	Refiérase a la Figura A-4 para el formato del byte del descriptor del medio. El registro DS será cambiado por esta función así que asegúrese de grabarlo antes de usar esta función.
1CH	OBTENER UNA TABLA DE ASIGNACION (FAT) PARA CUALQUIER UNIDAD
Entrada	AH = 1CH DL = número de lector de disco
Salida	AL = número de sectores por unidad DS:BX = dirección del descriptor del medio CX = tamaño de un sector en bytes DX = número de unidades en el lector de disco
1DH	NO ASIGNADO
1EH	NO ASIGNADO
1FH	NO ASIGNADO
20H	NO ASIGNADO
21H	@LECTURA ALEATORIA UTILIZANDO FCB
Entrada	AH = 21H DS:DX = dirección del FCB abierto

FIGURA A-4 Contenido del byte descriptivo del medio.



Bit 0 = 0 cuando no es de dos lados
= 1 cuando es de dos lados

Bit 1 = 0 cuando no hay ocho sectores por pista
= 1 cuando hay ocho sectores por pista

Bit 2 = 0 cuando no es removible
= 1 cuando es removible

Salida	AL = 00H cuando es leído exitosamente AL = 01H cuando se alcanza el final del archivo AL = 02H cuando el segmento se ajusta AL = 03H cuando menos de 128 bytes son leídos
22H	@ESCRIBIR DE MANERA ALEATORIA USANDO FCB
Entrada	AH = 22H DS:DX = dirección del FCB abierto
Salida	AL = 00H cuando fue escrito exitosamente AL = 01H cuando el disco está lleno AL = 02H cuando el segmento se ajusta
23H	@INDICAR EL NUMERO DE REGISTROS (FCB)
Entrada	AH = 23H DS:DX = dirección de FCB
Salida	AL = 00H número de registros AL = 0FFH cuando el archivo no se encuentra
24H	@ESTABLECER EL TAMAÑO RELATIVO DEL REGISTRO (FCB)
Entrada	AH = 24H DS:DX = dirección de FCB
Notas	Establece el campo del registro al valor contenido en el FCB.
25H	ESTABLECER EL INTERRUPTOR PARA VECTOR
Entrada	AH = 25H AL = número del interruptor para el vector DS:DX = dirección del nuevo procedimiento para el interruptor
Notas	Antes de cambiar el interruptor para vector, se sugiere que el interruptor para el vector actual sea primero grabado utilizando la función 35H de DOS. Esto permite un vínculo para que el vector original pueda ser posteriormente restaurado.
26H	CREAR EL PREFIJO DE SEGMENTO DEL NUEVO PROGRAMA
Entrada	AH = 26H DX = dirección del segmento del PSP nuevo
Notas	La Figura A-5 muestra la estructura del prefijo del segmento del programa.

FIGURA A-5 Contenido del prefijo del segmento del programa (PSP).

Desplazamiento	Contenido
00H	INT 20H
02H	Parte superior de la memoria
04H	Reservado
05H	Código de operación
06H	Número de bytes en el segmento
0AH	Dirección para terminar (desplazamiento)
0CH	Dirección para terminar (segmento)
0EH	Dirección para Ctrl-break (desplazamiento)
10H	Dirección para Ctrl-break (segmento)
12H	Dirección para error crítico (segmento)
14H	Dirección para error crítico (desplazamiento)
16H	Reservado
2CH	Dirección para el ambiente (segmento)
2EH	Reservado
50H	Solicitud al DOS
52H	Reservado
5CH	Bloque 1 para control de archivo
6CH	Bloque 2 para control de archivo
80H	Distancia de la línea de comando
81H	Línea de comando

27H	@LECTURA ALEATORIA DE BLOQUES DE ARCHIVOS (FCB)
Entrada	AH = 27H CX = el número de registros DS:DX = dirección del FCB abierto
Salida	AL = 00H cuando fue leído exitosamente AL = 01H cuando se alcanza el final del archivo AL = 02H cuando el segmento se ajusta AL = 03H cuando menos de 128 bytes fueron leídos CX = el número de registros leídos

28H	@ESCRIBIR EN FORMA ALEATORIA BLOQUES DE ARCHIVO (FCB)
Entrada	AH = 28H CX = el número de registros DS:DX = dirección del FCB abierto
Salida	AL = 00H cuando fue escrito exitosamente AL = 01H cuando el disco está lleno AL = 02H cuando el segmento se ajusta CX = el número de registros escritos
29H	@LINEA DE COMANDO PARSE (FCB)
Entrada	AH = 29H AL = esconder parse DS:SI = dirección de FCB DS:DI = dirección de línea de comando
Salida	AL = 00H cuando no son encontrados ningunos caracteres del nombre del archivo AL = 01H cuando son encontrados los caracteres del nombre del archivo AL = 0FFH cuando el especificador del lector del disco fue incorrecto DS:SI = dirección del carácter después del nombre DS:DI = dirección del primer byte de FCB
2AH	LEER LA FECHA DEL SISTEMA
Entrada	AH = 2AH
Salida	AL = día de la semana CX = año (1980-2099) DH = mes DL = día del mes
Notas	El día de la semana es codificado como Domingo = 00H a Sábado = 06H. El año es un número binario igual a 1980 hasta 2099.
2BH	ESTABLECER LA FECHA DEL SISTEMA
Entrada	AH = 2BH CX = el año (1980-2099) DH = mes DL = día del mes
2CH	LEER EL TIEMPO DEL SISTEMA
Entrada	AH = 2CH
Salida	CH = horas (0 - 23) CL = minutos DH = segundos DL = cientos de segundos

2DH	ESTABLECER EL TIEMPO DEL SISTEMA
Entrada	AH = 2DH CH = horas CL = minutos DH = segundos DL = cientos de segundos
2EH	DISCO VERIFICAR ESCRITO
Entrada	AH = 2EH AL = 00H para desactivar verificar en escrito AL = 01H para activar verificar en escrito
2FH	LEER AREA DE TRANSFERENCIA DEL DISCO
Entrada	AH = 2FH
Salida	ES:BX = contiene la dirección de DTA
30H	LEER EL NUMERO DE LA VERSION DE DOS
Entrada	AH = 30H
Salida	AH = número fraccionado de la versión AL = número de versión del número entero
Notas	Por ejemplo, el número 3.2 de la versión DOS se compone como 3 en AL y un 14H en AH.
31H	TERMINAR Y PERMANECER COMO RESIDENTE (TSR)
Entrada	AH = 31H AL = el código de retorno de DOS DX = número de párrafos para reservar
Notas	Un párrafo es de 16 bytes y el código de salida de DOS será leído al nivel del archivo de lote con ERRORCODE.
32H	NO ASIGNADO
33H	VERIFICAR CTRL-BREAK

Entrada	AH = 33H AL = 00H para solicitar el ctrl-break actual AL = 01H para cambiar ctrl-break DL = 00H para desactivar ctrl-break DL = 01H para activar ctrl-break
Salida	DL = Estado actual de ctrl-break
34H	OBTENER DIRECCION DE LA BANDERA InDOS
Entrada	AH = 34H
Salida	ES:BX dirección de la bandera InDOS
Notas	La bandera InDOS está disponible en las versiones 3.2 o más actuales de DOS e indica la actividad de DOS. Cuando InDOS = 00H, DOS está inactivo o 0FFH cuando DOS está activo.
35H	LEER EL INTERRUPTOR PARA VECTOR
Entrada	AH = 35H AL = número del interruptor para vector
Salida	ES:BX = dirección archivada en el vector
Notas	Esta función de DOS es usada con la función 25H para instalar/retirar los identificadores de interruptores.
36H	DETERMINAR EL ESPACIO LIBRE EN EL DISCO
Entrada	AH = 36H DL = número del lector de disco
Salida	AX = FFFFH cuando el lector de disco es inválido AX = número de sectores por unidad BX = número de unidades disponibles CX = bytes por sector DX = número de unidades en un lector de disco
Notas	El lector de disco predeterminado es DL = 00H, lector de disco A = 01H, lector de disco B = 02H, etc.
37H	NO ASIGNADO
38H	REGRESAR AL CODIGO DEL PAIS
Entrada	AH = 38H AL = 00H para el código actual del país BX = código del país de 16-bits DS:DX = dirección del búfer de información

Salida	AX = código de error cuando el restante fue habilitado BX = código para contador DS:DX = dirección del búfer de información
39H	CREAR SUBDIRECTORIO
Entrada	AH = 39H DS:DX = dirección para el nombre del subdirectorio de la cadena ASCII
Salida	AX = código de error cuando el restante está activado
Notas	Cadena ASCII es el nombre del subdirectorio en el código ASCII terminado con 00H en vez de un regreso de carro/alimentación de línea.
3AH	BORRAR SUBDIRECTORIO
Entrada	AH = 3AH DS:DX = dirección del nombre del subdirectorio de la cadena ASCII
Salida	AX = código de error cuando el restante fue activado
3BH	CAMBIAR SUBDIRECTORIO
Entrada	AH = 3BH DS:DX = dirección del nombre nuevo del subdirectorio de la cadena ASCII-Z
Salida	AX = código de error cuando el restante está activado
3CH	CREAR UN ARCHIVO NUEVO
Entrada	AH = 3CH CX = palabra de atributo DS:DX = dirección del nombre de archivo de la cadena ASCII-Z
Salida	AX = código de error cuando el restante está activado AX = identificador de archivo cuando el restante fue eliminado
Notas	La palabra de atributo puede contener cualquiera de los siguientes (sumados): 01H acceso sólo de lectura, 02H = archivo o directorio escondido, 04H archivo del sistema, 08H = nivel del volumen, 10H = subdirectorio, y 20H = bit de archivo. En la mayoría de los casos un archivo es creado con 0000H.
3DH	ABRIR UN ARCHIVO
Entrada	AH = 3DH AL = código de acceso DS:DX = dirección del nombre de archivo de la cadena ASCII-Z
Salida	AX = código de error cuando el restante está activado AX = identificador de archivo cuando el restante fue eliminado

Notas	El código de acceso en AL = 00H para acceso de sólo lectura, AL = 01H para acceso de sólo escribir, y AL = 02H para acceso de leer/escribir. Para los archivos compartidos en un ambiente de red, bit 4 de AL = 1 negará el acceso de leer/escribir, bit 5 de AL = 1 negará un acceso a escribir, bits 4 y 5 de AL = 1 negarán el acceso a leer, bit 6 de AL = 1 no niega ninguno, bit 7 de AL = 0 ocasiona que el archivo sea heredado por el hijo y bit 7 de AL = 1 está limitado al proceso actual.
3EH	CERRAR UN ARCHIVO
Entrada	AH = 3EH BX = identificador de archivo
Salida	AX = código de error cuando el restante está activado
3FH	LEER UN ARCHIVO
Entrada	AH = 3FH BX = identificador de archivos CX = número de bytes que deben leerse DS:DX = dirección del búfer del archivo para mantener los datos al leer
Salida	AX = código de error cuando el restante está activado AX = número de bytes leídos cuando el restante está eliminado
40H	ESCRIBIR UN ARCHIVO
Entrada	AH = 40H BX = identificador de archivos CX = número de bytes para escribirse DS:DX = dirección del búfer para archivo que sostiene los datos escritos
Salida	AX = código de error cuando el restante está activado AX = número de bytes escritos cuando el restante está eliminado
41H	ELIMINAR UN ARCHIVO
Entrada	AH = 41H DS:DX = dirección del nombre de archivo de la cadena ASCII
Salida	AX = código de error cuando el restante está activado
42H	MOVER EL APUNTADOR DEL ARCHIVO
Entrada	AH = 42H AL = técnica de movimiento BX = identificador de archivos CX:DX = número de bytes que el apuntador movió

Salida	AX = código de error cuando el restante está activado AX:DX = apuntador para bytes movido
Notas	La técnica de movimiento ocasiona que el apuntador se mueva del principio del archivo cuando AL = 00H, de la localidad actual si AL = 01H y del final del archivo si AL = 02H. La cuenta será guardada para que DX contenga los 16-bits menos significantes, CX o AX contienen los 16 bits más importantes.
43H ATRIBUTOS DE LEER/ESCRIBIR DEL ARCHIVO	
Entrada	AH = 43H AL = 00H para leer atributos AL = 01H para escribir atributos CX = palabra de atributo (véase la función 3CH) DS:DX = dirección del nombre del archivo o la cadena ASCII-Z
Salida	AX = código de error cuando el restante está activado CX = palabra de atributo del restante eliminado
44H CONTROL DEL PERIFERICO I/O (IOTCL)	
Entrada	AH = 44H AL = código (vease notas) AL = 01H para escribir atributos BX = identificador de archivos o número de periférico CX = número de bytes DS:DX = datos o dirección
Salida	X = código de error cuando el restante está activado AX y DX = parámetros
Notas	Los códigos encontrados en AL son los siguientes: 00H = leer el estado del periférico (DX = estado) 01H = escribir el estado del periférico (DX = estado escrito) 02H = leer información del periférico (DS:DX = dirección del búfer) 03H = escribir información al periférico (DS:DX = dirección del búfer) 04H = leer información del lector del disco 05H = escribir información al lector del disco 06H = leer el estado de la entrada de información (AL = 00H listo o 0FH no listo) 07H = leer el estado de la salida de información (AL = 00H listo o 0FH no listo) 08H = ¿medio removible? (AL = 00H removible, 01H fijo) 09H = ¿periférico local o remoto? (bit 12 del juego DX para remoto) 0AH = ¿identificador local o remoto? (bit 15 del juego DX para remoto) 0BH = cambiar la cuenta de la entrada 0CH = control del I/O genérico para los periféricos de caracteres 0DH = control del I/O genérico para los periféricos de bloque 0EH = número de salida de los periféricos lógicos (AL = número) 0FH = cambiar el número de los periféricos lógicos
45H DUPLICAR EL IDENTIFICADOR DE ARCHIVOS	
Entrada	AH = 45H BX = identificador de archivos actual

Salida	AX = código de error cuando el restante está activado AX = duplicar identificador de archivos
46H	FORZAR IDENTIFICADORES DUPLICADOS DE ARCHIVOS
Entrada	AH = 46H BX = identificador de archivos actual CX = identificador de archivos nuevo
Salida	AX = código de error cuando el restante está activado
Notas	Esta función trabaja como la función 45H excepto que la función 45H te permite a DOS seleccionar al identificador nuevo mientras que esta función le permite al usuario seleccionar el identificador nuevo.
47H	LEER DIRECTORIO ACTUAL
Entrada	AH = 47H DL = número de lector de disco DS:SI = dirección os un búfer de 64 bytes para el nombre del directorio
Salida	DS:SI direcciona el nombre del directorio actual cuando el restante fue eliminado
48H	ASIGNAR EL BLOQUE DE MEMORIA
Entrada	AH = 48H BX = número de párrafos para asignar CX = identificador de archivos nuevo
Salida	BX = el bloque más grande disponible si el restante fue eliminado
49H	LIBERAR EL BLOQUE DE MEMORIA ASIGNADO
Entrada	AH = 49H ES = segmento de la dirección del bloque que tiene que liberarse CX = identificador de archivos nuevo
Salida	El restante indica un error cuando sea establecido
4AH	MODIFICAR EL BLOQUE DE MEMORIA ASIGNADO
Entrar	AH = 4AH BX = nuevo tamaño del bloque en párrafos ES = dirección del segmento del bloque para ser modificado
Salida	BX = bloque más grande disponible cuando el restante fue eliminado.

4BH	CARGAR O EJECUTAR UN PROGRAMA
Entrada	AH = 4BH AL = código de función ES:BX = dirección del bloque para parámetro DS:DX = dirección del comando de la cadena ASCII-Z
Salida	El restante indica un error cuando fue activado
Notas	Los códigos de las funciones son AL = 00H para cargar y ejecutar un programa y AL = 03H para cargar un programa pero no ejecutarlo. La Figura A-6 muestra el bloque de parámetros usado con esta función.
4CH	ELIMINAR UN PROCESO
Entrada	AH = 4CH AL = código de error
Salida	Regresa el control a DOS
Notas	Esta función regresa el control a DOS con el código de error grabado para que se pueda obtener utilizando el sistema de procesamiento de lote de DOS ERROR LEVEL. Normalmente usamos esta función con un código de error de 00H para regresar a DOS.
4DH	LEER EL CODIGO DE SALIDA
Entrada	AH = 4DH

FIGURA A-6 Los bloques para los parámetros utilizados con la función 4BH (EXEC). (a) Para el código de función 00H, (b) Para el código de función 03H.

(a)	
Desplazamiento	Contenido
00H	Dirección para el ambiente (segmento)
02H	Dirección para la línea de comando (desplazamiento)
04H	Dirección para la línea de comando (segmento)
06H	Dirección para el bloque 1 para el control del archivo (desplazamiento)
08H	Dirección para el bloque 1 para control del archivo (segmento)
0AH	Dirección para el bloque 2 para control del archivo (segmento)
0CH	Dirección para el bloque 2 para control del archivo (desplazamiento)

(b)	
Desplazamiento	Contenido
00H	Dirección para el segmento destino sobrepuesto
02H	Factor de reubicación

Salida	AX = indica el código de error
Notas	Esta función es usada para obtener un código de estado de salida creado ejecutando un programa con la función 4BH de DOS. Los códigos de salida son: AX = 0000H para una terminación normal—sin errores, AX = 0001H para una terminación de ctrl-break, AX = 0002H para un error crítico de periférico y AX = 0003H para una terminación por un INT 31H.
4EH	ENCONTRAR EL PRIMER ARCHIVO QUE COMBINE
Entrada	AH = 4EH CX = atributos del archivo DS:DX = dirección del nombre del archivo de la cadena ASCII-Z
Salida	El restante está habilitado para un archivo no encontrado
Notas	Esta función buscará en el directorio actual u otro para el primer archivo que combine. A la salida, el DTA contiene la información del archivo. Vea la Figura A-7 para el área de transferencia del disco (DTA).
4FH	ENCONTRAR EL PROXIMO ARCHIVO QUE COMBINE
Entrada	AH = 4FH
Salida	El restante está activado para un archivo no encontrado
Notas	Esta función es usada después de que el primer archivo fue encontrado con la función 4EH.
50H	ESTABLECER LA DIRECCION DEL PREFIJO DEL SEGMENTO DEL PROGRAMA (PSP)
Entrada	AH = 50H BX = dirección desplazada del PSP nuevo
Notas	Un cuidado excesivo se debe usar con esta función porque no es posible ninguna recuperación de error.

FIGURA A-7 Área para transferencia de información (DTA) usada para encontrar un archivo.

Desplazamiento	Contenido
15H	Atributos
16H	Tiempo para creación
18H	Fecha de creación
1AH	Tamaño del archivo de palabra inferior
1CH	Tamaño del archivo de palabra superior
1EH	Buscar nombre del archivo

51H	OBTENER LA DIRECCION DE PSP
Entrada	AH = 51H
Salida	BX = dirección actual del segmento PSP
52H	NO ASIGNADO
53H	NO ASIGNADO
54H	LEER EL ESTADO DE VERIFICACION DEL DISCO
Entrada	AH = 54H
Salida	AL = 00H cuando verificar no está activada AL = 01H cuando verificar está activada
55H	NO ASIGNADO
56H	RENOMBRAR ARCHIVO
Entrada	AH = 56H ES:DI = dirección de la cadena ASCII-Z conteniendo el nuevo nombre del archivo DS:DX = dirección de la cadena ASCII-Z que contiene el archivo que se tiene que renombrar
Salida	El restante está activado para la condición de error
57H	LEER LA MARCA DE FECHA Y TIEMPO DEL ARCHIVO
Entrada	AH = 57H AL = código de función BX = identificador de archivos CX = nuevo tiempo DX = nueva fecha
Salida	El restante está activado para la condición de error CX = hora cuando el restante fue eliminado DX = fecha cuando el restante fue eliminado
Notas	AL = 00H para leer la fecha y el tiempo o 01H para escribir la fecha y tiempo.
58H	NO ASIGNADO

59H	OBTENER LA INFORMACION DE ERROR EXTENDIDA
Entrada	AH = 59H BX = 0000H para la versión 3.X de DOS
Salida	AX = código de error extendido BH = clase de error BL = acción recomendada CH = locus
Notas	<p>A continuación están los códigos de error encontrados en AX:</p> <ul style="list-style-type: none"> 0001H = número de función inválido 0002H = archivo no encontrado 0003H = ruta de acceso no encontrada 0004H = ningunos identificadores de archivos están disponibles 0005H = acceso negado 0006H = identificador de archivos inválido 0007H = falla del bloque de control de la memoria 0008H = memoria insuficiente 0009H = dirección inválida del bloque de memoria 000AH = falla en el ambiente 000BH = formato inválido 000CH = código de acceso inválido 000DH = información inválida 000EH = unidad desconocida 000FH = lector de disco inválido 0010H = intento para quitar el directorio actual 0011H = no el mismo periférico 0012H = no más archivos 0013H = disco con protección contra escribir 0014H = unidad desconocida 0015H = lector de disco no listo 0016H = comando desconocido 0017H = error de información (CRC revisar error) 0018H = largo de la estructura de una petición mala 0019H = buscar error 001AH = tipo de medio desconocido 001BH = sector no encontrado 001CH = impresora fuera de papel 001DH = escribir falla 001EH = leer falla 001FH = falla general 0020H = compartir violación 0021H = cerrar violación 0022H = cambio de disco inválido 0023H = FCB no disponible 0024H = exceso del búfer compartido 0025H = incompatibilidad de la página del código 0026H = manejar la operación del final del archivo no terminada 0027H = disco lleno 0028H—0031H reservado 0032H = requerimiento de red no apoyado 0033H = máquina remota no mencionada 0034H = nombre duplicado en la red 0035H = nombre de la red no encontrado 0036H = red ocupada 0037H = periférico ya no existe en la red 0038H = límite del comando netBIOS excedido

0039H = error en el hardware para el adaptador de la red
 003AH = respuesta incorrecta de la red
 003BH = error inesoeraco en la red
 003CH = adaptador remoto es incompatible
 003DH = cola de espera para imprimir está llena
 003EH = no hay suficiente espacio para imprimir el archivo
 003FH = archivo para impresión fue borrado
 0040H = nombre de la red borrado
 0041H = acceso a . red negado
 0042H = tipo incorrecto del periférico de la red
 0043H = nombre de la red no encontrado
 0044H = nombre de la red excedió el límite
 0045H = límite de la sesión netBIOS excedido
 0046H = pausa temporal
 0047H = requerimiento de la red no aceptado
 0048H = pausa de impresión o de redireccionamiento del disco
 0049H — 004FH reservado
 0050H = archivo ya existe
 0051H = duplicar FCB
 0052H = no se puede hacer el directorio
 0053H = falla en INT 24H (error crítico)
 0054H = demasiados redireccionamientos
 0055H = duplicar redireccionamiento
 0056H = clave inválida
 0057H = parámetro inválido
 0058H = falla al escribir en la red
 0059H = función no apoyada por la red
 005AH = componente del sistema requerido no instalado
 0065H = periférico no instalado

A continuación están los códigos de clase error como se encuentran en BH:

01H = no hay recursos disponibles
 02H = error transitorio
 03H = error de autorización
 04H = error del software interno
 05H = error del hardware
 06H = falla en el sistema
 07H = error de aplicación del software
 08H = elemento no encontrado
 09H = formato inválido
 0AH = elemento bloqueado
 0BH = error del medio
 0CH = elemento ya existe
 0DH = error no conocido

A continuación está la acción recomendada como se encuentra en BL:

01H = reintente la operación
 02H = oemorar y reintente la operación
 03H = reintento del usuario
 04H = abortar el procesamiento
 05H = salida inmediata
 06H = ignorar error
 07H = reintentar con la intervención del usuario

A continuación está una lista de locus en CH:

01H = fuente desconocida
 02H = bloquear el error del periférico

	<p>03H = área de la red 04H = error del periférico serial 05H = error de la memoria</p>
5AH	CREAR UN NOMBRE UNICO PARA ARCHIVO
Entrada	<p>AH = 5AH CX = código de atributo DS:DX = dirección de la ruta de acceso del directorio de la cadena ASCII-Z</p>
Salida	<p>El restante está activado para una condición de error AX = identificador de archivo cuando el restante fue eliminado DS:DX = dirección del nombre de directorio agregada</p>
Notas	<p>La ruta de acceso del directorio del archivo ASCII-Z debe terminar con una diagonal invertida (\). A la salida, el nombre del directorio será agregado como un nombre unico de archivo.</p>
5BH	CREAR UN ARCHIVO DOS
Entrada	<p>AH = 5BH CX = código de atributo DS:DX = dirección de la cadena ASCII-Z contiene el nombre del archivo</p>
Salida	<p>El restante está activado para la condición de error AX = identificador de archivo cuando el restante fue eliminado</p>
Notas	<p>La función sólo trabaja en la versión 3.X o más de DOS.</p>
5CH	CERRAR/ABRIR EL CONTENIDO DEL ARCHIVO
Entrada	<p>AH = 5CH BX = identificador de archivos CX:DX = dirección desplazada del área cerrada/abierta SI:DI = número de bytes para cerrar o abrir empezando en el desplazamiento</p>
Salida	<p>El restante está activado para la condición de error.</p>
5DH	ESTABLECER INFORMACION DE ERROR EXTENDIDO
Entrada	<p>AH = 5DH AL = 0AH DS:DX = dirección de la estructura de información del error extendido</p>
Notas	<p>Esta función es usada por la versión 3.1 o más de DOS para archivar la información de error extendido.</p>
5EH	RED/IMPRESORA
Entrada	<p>AH = 5EH AL = 00H (obtener el nombre de la red) DS:DX = dirección de la cadena ASCII-Z que contiene el nombre de la red</p>

Salida	El restante está activado para la condición de error CL = número netBIOS cuando el restante está eliminado
Entrada	AH = 5EH AL = 02H (definir la impresora de la red) BX = lista de redireccionamiento CX = distancia de la cadena para la instalación DS:DX = dirección del búfer para establecer la impresora
Salida	El restante está activado para una condición de error
Entrada	AH = 5EH AL = 03H (leer la cadena de establecimiento de la impresora de la red) BX = lista de redireccionamiento DS:DX = dirección del búfer para establecer la impresora
Salida	El restante está activado para la condición de error CX = distancia de la cadena de instalación cuando el restante fue eliminado ES:DI = dirección del búfer para establecer la impresora
62H	OBTENER LA DIRECCION PSP
Entrada	AH = 62H
Salida	BX = dirección del segmento del programa actual
Notas	La función sólo trabaja en la versión 3.0 o más de DOS
65H	OBTENER INFORMACION DEL PAIS EXTENDIDA
Entrada	AH = 65H AL = código de función ES:DI = dirección del búfer que va a recibir la información
Salida	El restante está activado para condición de error CX = distancia de la información del país
Notas	La función sólo trabaja con la versión 3.3 o más de DOS.
66H	OBTENER/ESTABLECER PAGINA DE CODIGO
Entrada	AH = 66H AL = código de función BX = número de la página del código
Salida	El restante está activado para una condición de error BX = número de la página de código activo DX = número de la página de código predeterminado
Notas	Un código de función en AL de 01H obtiene el número de la página del código y un código de 02H establece el número de la página del código.

67H	ESTABLECER LA CUENTA DEL IDENTIFICADOR
Entrada	AH = 67H BX = número de identificadores deseados
Salida	El restante está activado para la condición de error
Notas	Esta función está disponible para la versión 3.3 o más de DOS.
68H	ARCHIVO COMPROMETIDO
Entrada	AH = 68H BX = número de identificador
Salida	El restante está activado para la condición de error. De otra manera, la marca de la fecha y la hora será escrita al directorio
Notas	Esta función está disponible para la versión 3.3 o más de DOS.
6CH	ARCHIVO ABIERTO EXTENDIDO
Entrada	AH = 6CH AL = 00H BX = modo abierto CX = atributos DX = abrir bandera DS:SI = dirección del nombre del archivo de la cadena ASCII-2
Salida	AX = código de error cuando el restante está establecido AX = identificador cuando el restante fue eliminado CX = archivo 0001H existía y fue abierto CX = archivo 0002H no existía y fue creado
Notas	Esta función está disponible en la versión 4.0 o más de DOS.