52
29

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DIAGNOSTICO DEL HARDWARE DE UNA COMPUTADORA PERSONAL (PC)

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N :

SERVIN ALVAREZ SARA
VILLEGAS DELGADO NORA
MADRIGAL CHAVEZ JUAN CARLOS
SANCHEZ GALLEN DAVID
SORIANO TREJO LUIS ALFONSO

DIRECTOR DE TESIS: ING. RUBEN LIZARDI C.

MEXICO, D. F.

1990

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

I N D I C E

	Página
I. Introducción	1
I.1. Breve historia de las PCs	1
I.2. Las PCs en la UNAM	6
I.3. Papel de las PCs en la sociedad moderna	13
I.4. Justificación del presente trabajo	19
II. Descripción del hardware a diagnosticar	21
II.1. Unidad Central de Procesamiento (CPU)	21
a) Introducción	21
b) Diagrama de terminales	21
c) Descripción de terminales	22
d) Diagrama de bloques interno	27
e) Descripción del diagrama de bloques	28
f) Función general del circuito	31
g) Diagramas de tiempo	33
h) Programación del circuito	34
i) Interconexión del circuito en una PC	53
II.2. Controlador de Acceso Directo a Memoria (DMA)	54
a) Introducción	54
b) Diagrama de terminales	54
c) Descripción de terminales	55
d) Diagrama de bloques interno	58
e) Descripción del diagrama de bloques	58
f) Función general del circuito	59
g) Diagramas de tiempo	66
h) Programación del circuito	69
i) Interconexión del circuito en una PC	77
II.3. Memorias de Acceso Aleatorio y de Sólo Lectura (RAM y ROM)	80
Memorias RAM estáticas y dinámicas	80
Memorias ROM	82
II.4. Puerto serie	85
a) Introducción	85
b) Diagrama de terminales	87
c) Descripción de terminales	87
d) Diagrama de bloques interno	89
e) Descripción del diagrama de bloques	89
f) Función general del circuito	90
g) Programación del circuito	92
h) Interconexión del circuito en una PC	98

II.5. Puerto paralelo 103
a) Introducción 103
b) Diagrama de terminales 103
c) Descripción de terminales 104
d) Diagrama de bloques interno 104
e) Descripción del diagrama de bloques 105
f) Función general del circuito 106
g) Diagramas de tiempo 112
h) Programación del circuito 114
i) Interconexión del circuito en una PC 116
II.6. Teclado 118
a) Introducción 118
b) Funcionamiento del teclado 118
c) Diagrama básico del teclado 119
II.7. Reloj 120
a) Introducción 120
b) Diagrama de terminales 120
c) Descripción de terminales 120
d) Diagrama de bloques interno 122
e) Descripción del diagrama de bloques 122
f) Función general del circuito 123
g) Diagramas de tiempo 124
h) Interconexión del circuito en una PC 124
II.8. Controlador de disco duro 126
a) Introducción 126
b) Diagrama de terminales 126
c) Descripción de terminales 127
d) Diagrama de bloques interno 129
e) Descripción del diagrama de bloques 129
f) Función general del circuito 130
g) Diagramas de tiempo 131
h) Programación del circuito 133
i) Interconexión del circuito en una PC 144
II.9. Controlador de disco flexible 145
a) Introducción 145
b) Diagrama de terminales 145
c) Descripción de terminales 146
d) Diagrama de bloques interno 149
e) Descripción del diagrama de bloques 149
f) Función general del circuito 150
g) Diagramas de tiempo 151
h) Interconexión del circuito en una PC 152
i) Programación del circuito 153

III. Desarrollo de las rutinas para diagnóstico 157
III.1. Módulo del CPU 157
a) Discusión y analisis 157
b) Diseño 157
c) Implantación 160
III.2. Módulo del DMA 161
a) Discusion y análisis 161
b) Diseño 161
c) Implantación 163
III.3. Módulo de las memorias 164
a) Discusión y análisis 164
b) Diseño 165
c) Implantación 167
III.4. Módulo del teclado 168
a) Discusión y análisis 168
b) Diseño 169
c) Implantación 170
III.5. Módulo del disco duro 171
a) Compresión y descompresión de archivos 171
b) Encriptamiento y desencriptamiento de de archivos 174
c) Verificacion de la Tabla de Alojamiento de Archivos 175
III.6. Conjunción de las Rutinas de Diagnóstico 178

IV. Desarrollo del hardware para el diagnóstico 179
IV.1. Discusion del problema 179
a) Necesidad de la tarjeta 179
b) Funciones especificas de la tarjeta 179
c) Listado global de los componentes de la tarjeta 180
d) Comunicación con el exterior (interface tarjeta-PC) 181
e) Control interno 181
IV.2. Diseño de la tarjeta de diagnóstico 182
a) Diseños parciales del hardware
i) Puerto serie 190
ii) Puerto paralelo 196
iii) Reloj 202
iv) Controlador de disco flexible 204
b) Conjunción de los diseños parciales 208
IV.3. Desarrollo del software para el control de la tarjeta 209

V. Implantación y pruebas del sistema de diagnóstico 211
V.1. Diseño de casos de prueba
a) Casos de prueba para la rutina de diagnóstico del CPU 211
b) Casos de prueba para la rutina de diagnóstico del DMA 212
c) Casos de prueba para la rutina de diagnóstico de las memorias 213
d) Casos de prueba para la rutina de diagnóstico del puerto serie 214
e) Casos de prueba para la rutina de diagnóstico del puerto paralelo 215
f) Casos de prueba para la rutina de diagnóstico del teclado 216
g) Casos de prueba para la rutina de diagnóstico del reloj 217
h) Casos de prueba para la rutina de diagnóstico del controlador de disco flexible 218
i) Casos de prueba para la rutina del disco duro 220
V.2. Resultados obtenidos de las pruebas 221

VI. Conclusiones 223
VI.1. Alcances y limitaciones del sistema de diagnóstico 223
a) Conclusiones del diagnóstico del CPU 223
b) Conclusiones del diagnóstico del DMA 224
c) Conclusiones del diagnóstico de las memorias 225
d) Conclusiones del diagnóstico del puerto serie 226
e) Conclusiones del diagnóstico del puerto paralelo 227
f) Conclusiones del diagnóstico del teclado 228
g) Conclusiones del diagnóstico del reloj 229
h) Conclusiones del diagnóstico del controlador de disco flexible 230
i) Conclusiones del disco duro 231
VI.2. Otras alternativas de solución 232
a) Otras alternativas para el diagnóstico del CPU 232

b) Otras alternativas para el diagnóstico del DMA 233
c) Otras alternativas para el diagnóstico de las memorias 234
d) Otras alternativas para el diagnóstico del puerto serie 235
e) Otras alternativas para el diagnóstico del puerto paralelo 236
f) Otras alternativas para el diagnóstico del teclado 237
g) Otras alternativas para el diagnóstico del reloj 238
h) Otras alternativas para el diagnóstico del controlador de disco flexible 239
i) Otras alternativas para las rutinas del disco duro 240

Apéndice A. Descripción del hardware adicional
A.1. Procesador de entrada/salida A-1
a) Introducción A-1
b) Diagrama de terminales A-2
c) Descripción de terminales A-2
d) Diagrama de bloques interno A-5
e) Descripción del diagrama de bloques A-5
f) Función general del circuito A-6
g) Diagramas de tiempo A-14
h) Programación del circuito A-16
i) Interconexión del circuito en una PC A-22
A.2. Controlador de interrupciones A-24
a) Introducción A-24
b) Diagrama de terminales A-24
c) Descripción de terminales A-25
d) Diagrama de bloques interno A-26
e) Descripción del diagrama de bloques A-26
f) Función general del circuito A-28
g) Diagramas de tiempo A-30
h) Programacion del circuito A-31
i) Interconexión del circuito en una PC A-40
A.3. Latch octal A-41
a) Introducción A-41
b) Diagrama de terminales A-41
c) Descripción de terminales A-41
d) Diagrama de bloques interno A-42
e) Función general del circuito A-43

f) Diagramas de tiempo A-43
A.4. Bus transceptor octal A-44
a) Introducción A-44
b) Diagrama de terminales A-44
c) Descripción de terminales A-44
d) Diagrama de bloques interno A-45
e) Función general del circuito A-45
f) Diagramas de tiempo A-46
A.5. Controlador de bus A-47
a) Introducción A-47
b) Diagrama de terminales A-47
c) Descripción de terminales A-47
d) Diagrama de bloques interno A-49
e) Función general del circuito A-50
f) Diagramas de tiempo A-53
g) Interconexión del circuito en una PC A-53
A.6. Timer programable A-55
a) Introducción A-55
b) Diagrama de terminales A-55
c) Descripción de terminales A-55
d) Diagrama de bloques interno A-56
e) Descripción del diagrama de bloques A-57
f) Función general del circuito A-58
g) Diagramas de tiempo A-58
h) Interconexión del circuito en una PC A-63
A.7. Slots e interfaces A-65

Apéndice B. Manual del usuario B-1
B.1. Introducción B-1
B.2. Instalación del sistema B-2
a) Instalación de la tarjeta diagnosticadora B-2
b) Archivos del sistema B-2
B.3. Guía del usuario B-3
a) Salida del sistema al DOS B-3
b) Diagnósticos de los elementos de la unidas central B-3
c) Diagnóstico del teclado B-4
B.3.1. Diagnóstico de los componentes de la tarjeta madre B-4
a) CPU B-4
b) DMAC B-5
c) Memoria RAM B-5
d) Memoria ROM B-6
e) Puerto serie B-6

f) Puerto paralelo B-7
g) Reloj B-7
h) Controlador/manejador de disco flexible B-8
i) Disco duro B-9

Bibliografia

CAPITULO I
INTRODUCCION

I.1. BREVE HISTORIA DE LAS PCs.
I.2. LAS PCs EN LA UNAM.
I.3. PAPEL DE LAS PCs EN LA SOCIEDAD MODERNA.
I.4. JUSTIFICACION DEL PRESENTE TRABAJO.

I. INTRODUCCION.

1.1 BREVE HISTORIA DE LAS PCs.

Para cuando IBM presentó su primera PC al mercado, el 12 de agosto de 1981, el reciente fracaso de la Apple con su Apple II aún estaba fresco en la memoria de los compradores. Fué exacto el momento escogido por IBM.

La nueva máquina, basada en el 8088 de Intel, un microprocesador nuevo de 16 bits, ofrecía a los compradores potenciales una amplia gama de precios, desde 1565 dólares hasta 5000 dólares. El conjunto estaba formado por el CPU, el teclado, monitor a color o monocromático, dos floppys de 5 1/4" y una memoria instalada de 256 KB expandible hasta 512 KB.

De inmediato, la PC atrajo la atención de grandes sectores del mercado. Su microprocesador permitía el manejo de una gran cantidad de memoria, satisfaciendo los requerimientos de los usuarios y permitiendo que el software hecho para ella fuera más sencillo (por ejemplo, ya no se necesitaba realizar overlays en la memoria). La elección del 8088 como procesador de la PC no sólo permitió manejar más memoria, también incrementó la velocidad del proceso. Operando a 4.77 MHz, con un ciclo de acceso de 250 ns, superaba enormemente a sus competidores. Además de estas cualidades, la PC ofrecía un poder de graficación mayor al existente: soportaba hasta 16 colores con una resolución de 320 x 200 pixels, y en alta resolución 2 colores con 640 x 200 pixels.

Acompañando al hardware de la PC, IBM impulsó la creación y el uso de software para su PC. El primer sistema operativo utilizado por la PC fue el SCP DOS, que más tarde derivó en el conocido MS-DOS. La historia de la lucha por ser el sistema operativo (SO) de la PC, ocurrida en aquel tiempo, es conocida, pero vale la pena recordarla.

Cuando IBM planeaba la creación de la PC, puso a la consideración de varias compañías la creación del SO para la nueva máquina. Aparecieron varios candidatos: el CP/M-86 de Digital Research, el UCSD p-System de Softech Microsystems y el IBM PC DOS de Microsoft. El primero y el último fueron los dos candidatos más

viables para IBM, a pesar de la popularidad de los dos primeros. La razón fue el hecho de que entre los usuarios de IBM el SO CP/M gozaba de cierta consideración y tanto el nuevo CP/M-86 como el IBM PC DOS eran muy similares a éste. La elección de IBM por el PC DOS de Microsoft se debió a roces y falta de coordinación entre IBM y Digital Research.

Una vez definido el sistema operativo para la PC, la creación de software para ella fue cuestión de tiempo. En primer lugar, IBM instó a Microsoft a crear un shell (el COMMAND.COM) muy completo, que fuera amable con el usuario y le permitiera explotar toda la capacidad de la PC. Como segundo punto, se creó un intérprete de Basic de Microsoft, que acompañaba a todas las PCs almacenado en memoria ROM. Junto con el Basic, Microsoft aportó también un compilador del lenguaje Pascal. En tercer lugar, IBM impulsó la creación y el uso de programas comerciales tales como la hoja de cálculo Visicalc y el procesador de palabras Easywriter, los que debido al poder de la PC pudieron colocarse rápidamente en la preferencia de los usuarios. El último punto en la estrategia de software seguida por IBM consistió en estimular el desarrollo de software específico para PCs. Se estableció con este fin un departamento que exclusivamente se dedicaba a publicar software creado para PCs, hecho tanto por aficionados como por profesionales.

Hasta ahora sólo se han descrito las características del software y el hardware de la PC de IBM. Para dar una idea más completa del impacto que constituyó la PC en el mercado de computadoras, dividiremos el desarrollo siguiente en dos partes: primero, el desenvolvimiento de la PC en el mercado, posteriormente, describiremos cómo la PC llegó a ser la microcomputadora más popular en el mundo.

Hasta antes de la introducción de la PC, el mercado de computadoras personales era totalmente lidereado por la compañía Apple. Apple, con su microcomputadora Apple, llegó a ser sinónimo de computadora personal. Las grandes compañías de computación (IBM, Burroughs, etc.) no se habían aventurado aún en el mercado de las computadoras personales, permaneciendo en el campo de las macro y mini computadoras. La razón de esto fue el costo de los sistemas de cómputo y la falta de mercado para la computación personal. Gracias a Apple y al desarrollo tecnológico que produjo el microprocesador, fueron salvados aquellos obstáculos.

Cuando IBM entró al mercado de las computadoras personales, Apple seguía liderеando en ventas con su Apple II, a pesar del fracaso con la Apple III. Rápidamente, se notó el efecto de la entrada del gigante azul con su nueva PC. El prestigio que siempre había acompañado a IBM, aunado a una estrategia global de introducción de la PC hizo mella en las ventas de los competidores, y de un 0% que poseía IBM del mercado en 1981, pasó a un 18.8% en 1982 y a un 26% en 1983, colocándose como segundo vendedor atrás de Apple. Para el año de 1984, IBM poseía más del 50% del mercado de computadoras personales, dejando claro quién era el que de ahora en adelante marcaría el rumbo del desarrollo tecnológico para PCs y el establecimiento de estándares en computación personal. (Ver fig. I.1)

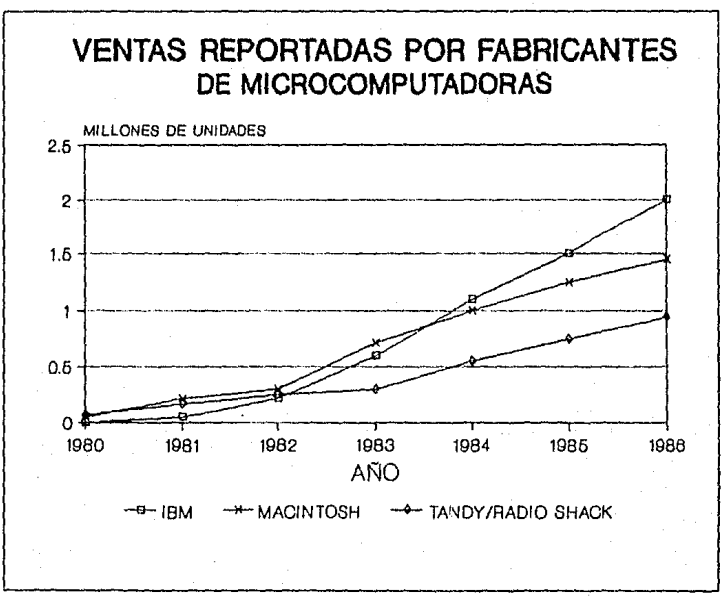


Fig I.1

En agosto de 1983, a dos años de la introducción de la PC, IBM creó un departamento especial, llamado División de Introducción de Sistemas, al que se le asignó el manejo de las PCs. La creación de esta División tuvo como consecuencia el fortalecimiento de la PC en el mercado, y por otra parte, el desarrollo de sistemas similares a la PC. Existe la creencia, en el campo de las computadoras personales, de que la vida de un producto es aproximadamente de 4 a 5 años, por lo que se suponía que la vida de la PC llegaría hasta 1986. Por esta razón, la división mencionada se encargó, a partir de 1983, de desarrollar un producto que tuviera la misma línea de la PC, pero con mayores ventajas. Fue así como, para 1983, nacieron las primeras PC XTs, PC XT/370s y un poco más tarde las PC ATs. El desarrollo de las PCs no sólo ha sido alrededor del CPU. Acompañando a éste, se han dado desarrollos en el terreno de los periféricos (impresoras, discos duros, cajas de Bernoulli, etc) y en comunicaciones (redes locales), lo que ha permitido que en promedio haya tenido IBM ventas por más de un millón de PCs al año.

Todas las razones del crecimiento en el mercado de las PCs, de la preferencia de los compradores por el producto de IBM distan mucho de ser evidentes. Antes de las PCs los sistemas personales de computación eran aceptables, dado el desarrollo tecnológico; los requerimientos de los usuarios eran bien satisfechos por microcomputadoras como la Apple, ¿por qué entonces, llegó a ser la PC de IBM la computadora personal más popular? La respuesta a esta pregunta es también parte de la historia de las PCs.

Cuando la IBM empezó a planear la entrada de la PC sabía muy bien los problemas que enfrentaría, no bastaban su capacidad financiera y su prestigio, era necesario que la gente creyera en la PC y la necesitara. Algunos errores de compañías como Apple y la explosión que tuvo el mercado de computadoras personales, fueron dos de los factores importantes en el éxito de la PC.

Durante los primeros años de la PC, IBM se dedicó a cambiar las "reglas" del mercado: desde un inicio hizo públicas las especificaciones en detalle de la PC, anexando código fuente del MS-DOS y del BIOS, con esto logró estimular una industria naciente que se dedicaría desde entonces a producir software y hardware compatible con PCs, pero además, este ofrecimiento de especificaciones produjo el nacimiento de "clones" de PC, pequeñas compañías nacieron y se desarrollaron al amparo de la PC y del

impulso dado por IBM al producto. Un segundo aspecto fue la distribución del producto. Debido a las características de éste, IBM tuvo que empezar a distribuirlo a través de pequeños distribuidores, tal y como lo hacía el resto de las compañías. Esta tactica, no esperada por la competencia, hizo llegar a las PCs a un amplio sector que las recibió, además del prestigio de IBM, con muchas opciones de servicio y mantenimiento que otras compañías no ofrecían.

La IBM PC, junto con sus multiples clones, debido a su tecnología avanzada (el microprocesador 8088), su capacidad de expansión (gracias a su arquitectura de "bus abierto", creación de Apple), que permitió satisfacer muchas de las necesidades nuevas de los usuarios (por ejemplo, el uso de tarjetas de comunicación), y la creciente industria de software, que produjo los programas para las PCs, se convirtió, después de dos años de su creación, en el líder en cuanto a microcomputadoras se refiere. La popularidad de la PC trajo como consecuencia la estandarización del microprocesador 8088/8086 y su familia, y al MS-DOS lo colocó como líder dentro de los sistemas operativos para microcomputadoras de 16 bits.

Hasta la fecha, no se ve con claridad qué sistema desplazará a las PCs de su posición privilegiada. Inclusive se duda de la capacidad del nuevo sistema de IBM, el PS/2, para ser el sucesor de la PC.

I.2 LAS COMPUTADORAS PERSONALES EN LA UNAM.

Para poder desarrollar este subtema se necesitó realizar una investigación de campo, es decir se visitaron los centros de cómputo que son considerados los más importantes y representativos de las actividades relacionadas con la computación dentro de la UNAM.

Dichos centros fueron:

- Centro de Cómputo de la Facultad de Ciencias.
- Centro de Informática de la Facultad de Economía.
- Centro de Informática de la Facultad de Contaduría y Admistración.
- Dirección General de Servicios de Cómputo Académico.
- Facultad de Ingeniería:
    - División de Ingeniería Mecánica y Eléctrica (DIME).
    - Centro de Cálculo de la Facultad de Ingeniería (CECAFI).
    - Laboratorio de Computadoras y Programación.

CENTRO DE CÓMPUTO DE LA FACULTAD DE CIENCIAS:

La persona entrevistada fue Raúl Mendoza Castro. La Facultad de Ciencias es considerada una de las instituciones pioneras en materia de computación, dado que la primera computadora llegada a la UNAM fue instalada en dicha Facultad, en el año de 1958.

A partir de esta fecha se ha contado con diferente equipo, que en su mayoría ha sido donación de IBM de México, puesto que existe un convenio con esta institución.

Su equipo abarca lo siguiente:

Primeramente se tuvo en este centro el sistema Eclipse, perforadoras, lectora de tarjetas, máquinas Franklin (64 KB), actualmente se tienen aproximadamente 45 PCs y 8 PS/2.

Las principales aplicaciones de estas máquinas son:

Dar apoyo a las carreras que se imparten en esta Facultad (Actuaría, Matemáticas y Biología) con sus diferentes materias: Programación con Instrumentos de Cálculo, Computación I y

Computacion II, Probabilidad y Estadistica, así como brindar apoyo a diferentes investigaciones, principalmente Biológicas.

Las PCs llegaron a la Facultad de Ciencias en 1987, y son, como ya se mencionó, donaciones principalmente, por lo tanto el cambio de tecnología en este centro está en función del convenio que ya existe.

En este centro se trabaja de las 7:00 a 22:00 horas, de lunes a viernes y de 7:00 a 15:00 horas los sábados.

Con las máquinas que prestan servicio en este centro de cómputo se cubren las necesidades de los usuarios, en general no existe un mantenimiento preventivo y las fallas se presentan en su mayoría en los teclados y en las unidades de disco.

Como planes a futuro se tiene pensado aumentar el número de PS/2 y contar con personal capacitado en DGSCA para dar el debido mantenimiento preventivo y correctivo a sus computadoras.

CENTRO DE INFORMATICA DE LA FACULTAD DE ECONOMIA:

El Centro de Informática de la Facultad de Economía está localizado en el primer piso, fué creado en junio de 1986 -aunque con anterioridad se contaba ya con el equipo, hacía falta personal y espacio-, cuenta con 40 PCs. Trabaja de 9:00 a 20:00 horas de lunes a viernes y de 9:00 a 15:00 el día sábado. El Sr. José Luis Sixtos, uno de los encargados del laboratorio proporciona los datos del mismo.

Este es uno de los pocos laboratorios entrevistados en los cuales anteriormente a las microcomputadoras personales no se contaba con otro equipo. Las marcas de PCs con las que se cuenta son Columbia y Printaform, de las cuales sólo cinco cuentan con disco duro de 20 MB; todas las micros son adquisiciones.

Para propósitos de mantenimiento se tiene un contrato con la compañía TELSA, la cual realiza el mantenimiento preventivo -cada tres meses- y correctivo -este último se atiende por llamada-. En cuestiones de descomposturas del equipo, se presentan fallas más frecuentemente en las fuentes de poder, debido a sobrecargas de voltaje y en las unidades de disco flexible, estas últimas presentan descalibraciones y desalineaciones.

El uso que se da a las PCs en este laboratorio está enfocado a la impartición de cursos de capacitación, usuarios alumnos y

elaboración de bancos de datos. Frecuentemente se menciona a este laboratorio en convocatorias de desarrollo de software, como uno de los patrocinadores.

Las espectativas para este laboratorio son incrementar el número de PCs con equipo nuevo, no se tiene pensado un cambio de tecnología o tipo de equipo, por lo que las PCs seguirán siendo las computadoras a adquirir.

CENTRO DE INFORMATICA DE LA FACULTAD DE CONTADURIA Y ADMINISTRACION:

Este centro fué fundado en el año de 1975, con la adquisición de una computadora CDC con perforadoras UNIVAC, continuando con una computadora Eclipse C130; la siguiente computadora fué una NCR con terminales, y posteriormente una maquina HP-3000 también con terminales. El uso de PCs comienza en 1985, cuando llegan al centro dieciocho PCs de marca IBM y tres marca Printaform. Al respecto, el Lic. Mauro Flores comenta lo siguiente.

Los usuarios de las microcomputadoras son alumnos y personal de la facultad. En este último caso se puede nombrar a la Secretaría General, la que dentro de sus usos propios tiene las labores administrativas, registro de suscriptores, exámenes profesionales, etc. Otro de los usuarios es el personal docente, el cual utiliza las PCs para presentar reportes, listas de asistencias y calificaciones, etc.

El servicio a alumnos se cubre -aunque no se satisface la demanda por microcomputadoras- actualmente con cincuenta PCs, dentro de las cuales se tienen máquinas BPM, IBM y Printaform. En la División de Educación Continua se cuenta con catorce PCs, para trabajos de las áreas básicas, cursos, trabajos, tesis y otros.

En cuanto al mantenimiento a las PCs se carece totalmente de él, aunque está por firmarse un convenio con el Centro de Instrumentos, para que éste último se encargue de dar un mantenimiento preventivo periódico a las PCs. El mismo Centro de Instrumentos realiza las reparaciones del equipo, y la Facultad de Contaduría y Administración cubre el costo del material empleado en la reparación.

Las fallas que se presentan más frecuentemente en el equipo se dan en el teclado, unidades de disco flexible y monitores de video. Se ha observado que las computadoras que presentan más fallas son en orden decreciente Printaform, BPM y finalmente IBM. Se tiene pensado llevar una bitácora de fallas que se vayan presentando en el equipo del Centro de Informática.

Las PCs que dan servicio a alumnos de licenciatura permanecen encendidas doce horas (de 9:00 a 21:00 horas). En dependencias de la Secretaría General las máquinas requieren estar encendidas las veinticuatro horas del día, apagándose los fines de semana, aunque en forma parcial, dado que en estos días se hace un respaldo de la información de la biblioteca.

Se han hecho peticiones por parte de este Centro de Informática para la compra de equipo, las microcomputadoras que se han pedido han sido máquinas PC, por considerar que otro tipo de tecnología -por ejemplo máquinas PS/2- no es la mejor opción por el momento.

En la División de Posgrado de esta facultad se adquirieron recientemente cien PCs con disco duro y veinte con dos unidades de disco flexible. A estas microcomputadoras tienen acceso investigadores de maestría y doctorado, cursos y alumnos de posgrado en general.

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO (DGSCA):

Entrevistado el Ing. Ricardo Martínez, titular del departamento de electrónica de la DGSCA comentó lo siguiente.

No existe la fecha exacta en la cual llega la primera PC a la UNAM, tampoco se sabe el número exacto de estas máquinas actualmente, pero se estima de mil doscientos a mil cuatrocientos el número de ellas.

Por lo que respecta a la DGSCA tampoco se tiene el número exacto de PCs, por lo que nuevamente se estima que existen de ciento diez a ciento veinte microcomputadoras PC, la gran mayoría han sido adquisiciones, y las donaciones son tan pocas que no son representativas en el número total de PCs en la DGSCA.

Aunque comenta que los usuarios son cuidadosos y el número de fallas en los equipos no es exagerado, se reportan fallas en los teclados, unidades de disco, monitores y puertos. No existe un

programa de mantenimiento preventivo periódico, pero el mantenimiento correctivo se realiza a nivel componente, con la sustitución del componente dañado.

En opinión del Ing. Ricardo Martínez un sistema de diagnostico sería de utilidad en México, pues en Estados Unidos de Norteamerica la reparación se concreta a la sustitución general de la tarjeta en donde se sospecha que un componente está defectuoso. Dado que en México no se tiene libre acceso a tarjetas completas, para así realizar la reparación de equipo, ni la disponibilidad monetaria, esto no es posible, pero los componentes individuales se pueden encontrar en forma relativamente fácil.

Las computadoras que se tienen en la DGSCA sirven de apoyo a alumnos en general y a investigadores. Los departamentos que emplean PCs en la DGSCA son Producción, Sistemas Operativos y la División de Cómputo para la Investigación.

DIVISION DE INGENIERIA MECANICA Y ELECTRICA (DIME) DE LA FACULTAD DE INGENIERIA:

Platicando con el Ing. Adolfo Millan N., quien tiene a su cargo los laboratorios de cómputo de esta división, menciono lo siguiente.

Se tienen en dicha división aproximadamente 75 PCs de diferentes marcas; Printaform (en su mayoría), BPM, IBM portátiles, y también 4 PS/2. Casi no existen donaciones, las computadoras han sido compradas directamente por la Facultad de Ingeniería y destinadas a sus diferentes divisiones y departamentos como serían: las divisiones de Ingeniería Civil, de Estudios de Posgrado, Ciencias Básicas, y los departamentos Administrativo, Control, Computación, etc.

Estas máquinas llegaron a la DIME en 1987, y sirven para dar apoyo a las carreras de Ingeniería en Computación e Ingeniería Mecánica y Eléctrica que se cursan en esta facultad.

Hasta la fecha se cubren las necesidades actuales de los usuarios que exclusivamente son alumnos de las carreras antes mencionadas.

Existen 4 laboratorios de Computación en la DIME donde se trabaja 15 horas diarias (de 7:00 a 22:00 horas), existe un mantenimiento periódico preventivo y correctivo por parte de la

Unidad de Mantenimiento y Servicio a Equipo de Cómputo de la División de Ingeniería Mecánica y Eléctrica. Este servicio se brinda aproximadamente cada periodo intersemestral y la Unidad da este servicio a diferentes dependencias de la misma facultad: División de Posgrado, Palacio de Minería, CECAFI, DIME y Laboratorio de Computadoras y Programación.

Donde se reportan la mayoría de máquinas descompuestas es en la DIME y estas radican en los monitores, fuentes de poder, teclados, unidades de disco y los puertos.

Se ha solicitado la compra de un equipo grande como podría ser una MICRO VAX, y no se ha pensado en el cambio a tecnología con PS/2.

Los problemas actuales son debidos (en cuanto a la reparación de las computadoras) a la compra y surtido de refacciones.

CENTRO DE CALCULO DE LA FACULTAD DE INGENIERIA (CECAFI) ANEXO:

Entrevistando al Sr. Salvador Santillán se tuvieron los siguientes datos.

El CECAFI anexo mantenía anteriormente a la llegada de PCs terminales de las computadoras ALTOS I, ALTOS II, VAX, Burroughs. Cuando llegan las PCs a fines de 1987, la necesidad de espacio origina que se coloquen veintidos PCs trabajando en forma normal con una impresora, siete como terminales de ALTOS I con impresora, y siete PCs como terminales de ALTOS II con una impresora, una PC conectada a gráficos con impresora, cuatro PCs como terminales de VAX con impresora y diez PCs como terminales Burroughs con impresora.

Las PCs que funcionan como terminales de otras computadoras tienen la ventaja de poder en un momento funcionar como terminales y al siguiente como PCs independientes, esto se elige dependiendo de las necesidades de servicio que se tengan en cada ocasión en el centro.

Las marcas que se tienen son BPM, IBM, UNIX y Printaform, y presentan pocas fallas. Se tiene un plan de mantenimiento preventivo que se practica aproximadamente cada dos meses. Para reparaciones se ha capacitado personal en COMUNICA, también la Unidad de Mantenimiento y Servicio a Equipo de Cómputo de la División de Ingeniería Mecánica y Eléctrica realiza las

reparaciones mayores.

El CECAFI anexo funciona doce horas diarias, de lunes a viernes, y el sábado de 9:00 a 14:00 horas. Se han cubierto con satisfacción las necesidades de los usuarios, por lo que no se prevee un crecimiento en el número de computadoras.

En la primera semana de haber abierto las inscripciones al CECAFI anexo se inscribieron a trescientos treinta usuarios, con la restricción de ser alumnos de la Facultad de Ingeniería.

LABORATORIO DE COMPUTADORAS Y PROGRAMACION.

La persona encargada, Francisco Monreal V. menciona que este laboratorio existe para dar soporte a la materia de Computadoras y Programación, que se imparte en la División de Ciencias Básicas de la Facultad de Ingeniería.

Comenzó a funcionar con calculadoras programables, posteriormente tuvieron computadoras Radio Shack y más tarde, a finales de 1985 llegaron las PCs, predominan las máquinas Printaform y todas han sido adquisiciones, existen solamente 8 IBM XT.

Se calcula que el semestre pasado se atendieron el equivalente a 20,000 hrs/hombre.

Existe un mantenimiento correctivo y preventivo por parte de la Unidad de Servicio y Mantenimiento a Equipo de Cómputo de la División de Ingeniería Mecánica y Eléctrica.

En general, se tienen fallas en cuanto a fuentes de poder y monitores. Se han solicitado instalaciones adecuadas, que soporten descargas eléctricas y posean una tierra física. También se ha solicitado la instalación de un equipo adecuado de aire acondicionado y en sus planes a futuro se desea contar con equipo más actualizado como monitores de alta resolución, plotters, lápiz óptico y computadoras con disco duro, para poder dar un panorama general de computación más amplio a los alumnos de esta materia.

I.3. PAPEL DE LAS PCs EN LA SOCIEDAD MODERNA.

Se considera que lo verdaderamente importante no es qué es, ni de qué se compone una microcomputadora, sino para qué sirve y a qué tareas de nuestra vida diaria se aplica.

Es una herramienta, un instrumento de reciente aparición que se suma a la larguísima lista de invenciones humanas. Si se pregunta qué puede hacer una PC se puede contestar de dos maneras: si se enumera todo aquello que realiza en la actualidad, la respuesta es muy extensa, si se expresa todo lo que realizará en un futuro, la respuesta puede ser muy breve: todo.

Lo que implica que las aplicaciones computacionales no están necesariamente limitadas por razones materiales de la computadora, sino por el propio hombre, por lo tanto la única limitación conocida de la computación es la que imponen los límites imaginativos y el ingenio del ser humano. En muchos lugares las aplicaciones computacionales son vistosas y resultan conocidas, en otras, no llaman la atención porque su forma de operar es menos espectacular, pero no menos eficaz.

La computación personal, y como el ejemplo más patente, las PCs, ha tenido el papel de llevar a un amplio sector de la población los beneficios del uso de la computadora, ya no solamente en cálculos complejos o actividades industriales o militares, sino también en la vida diaria.

Las PCs son utilizadas para el procesamiento de datos, en los negocios, en el hogar, en las empresas, en diversas instituciones y en actividades profesionales. Su gran potencial radica en que pueden ser usadas por millones de personas en educación, entretenimiento y otras aplicaciones:

Empresas, hogar, industria, oficina, administración, enseñanza, meteorología, minería, estadística, investigación, medicina, arte, editorial, comercio, biblioteca, navegación, banca, tráfico aéreo, alimentación, agricultura y un amplio etcétera.

Una microcomputadora puede ser utilizada en el hogar para:

- Entretenimiento, con una gama muy amplia de juegos, la mayoría de ellos con impresionantes despliegues gráficos y

sonoros.

- Control de apertura/cierre de puertas y ventanas.
- Alarma contra incendios.
- Compras en el supermercado.
- Control del presupuesto casero y archivo doméstico.
- Transmisión y recepción de documentos.
- Programas educativos.
- Agenda.
- Pagos y transferencia de fondos.
- Obtención de información de bases de datos de uso generalizado como por ejemplo: culturales, técnicas y científicas.

La PC también se encuentra en cualquier materia universitaria: medicina, física, química, matemáticas, ingeniería, biología, economía, administración, ciencias sociales, etc. También está en el mundo empresarial, en la industria, en el deporte, en sistemas de defensa, entretenimiento con juegos por computadora, puede realizar desde un delicadísimo análisis del cerebro humano o asistir a una operación quirúrgica, diseño mecánico, trabajos técnicos y científicos, elaborar nóminas, mantener registros de estudiantes, pacientes o clientes, diseño de circuitos integrados, cartografía, proceso de contabilidad en general, controlar máquinas-herramientas y cualquier otro tipo de producción en la industria, escribir cartas, etiquetar sobres y realizar otros documentos por medio del software procesador de documentos, manipular ventas y proyecciones de costos, a fin de planear las finanzas de una organización, controlar la temperatura, humedad del suelo y otras variables, a fin de informar sobre el estado de los cultivos, permite la rápida expansión en el uso de las técnicas de instrucción por computadora en instituciones educacionales, despliegue visual gráfico para análisis financiero, diseño interactivo, técnicas clínicas, modelación y simulación predictiva, pronóstico del estado del tiempo, oceanografía, astrofísica, socioeconomía, simulación de sistemas biológicos, diseño y automatización en ingeniería, aerodinámica, exploración sismológica, modelado de campos petroleros, ingeniería genética, mecánica cuántica, investigación sobre armas, diseño de plantas, tuberías y análisis de estructuras, diseño de tarjetas impresas, diagramas electrónicos, en definitiva en todas las actividades en las que se puedan conseguir mayores beneficios particulares o sociales.

MEDICINA.

Desde la gestion administrativa de la consulta de un hospital hasta exploraciones radiológicas, estudios angiográficos, tomografía auxiliada por computadora, en el campo de la investigación médica, farmacéutica, biológica, química, etc, formulación de diagnósticos clínicos, bancos de datos para los médicos, en fin, la PC entra en todos los aspectos relacionados con la lucha de los médicos para conseguir un alto nivel de salud de la población.

DISEÑO Y FABRICACION.

Diseño asistido por computadora (CAD) y el de fabricación asistida por computadora (CAM), cuando ambos sistemas actúan simultánea y coordinadamente se multiplican las posibilidades.

Los procesos CAD/CAM no se limitan al campo industrial, sino que se extienden a todas las actividades del diseño.

APLICACIONES INTEGRADAS.

Normalmente se incluyen en paquetes integrados que contienen apartados (menús) con los que se trabaja. Dichas aplicaciones son las siguientes:

- Procesador de textos.

Sirve principalmente para generar textos, cartas, articulos, informes, memorandums e incluso libros. Sus funciones primordiales son: permite editar o escribir y corregir un texto, imprimir dicho texto, y finalmente, una vez que el texto es definitivo, permite almacenarlo en un dispositivo magnético, de donde podrá recuperarse para volver a editarlo o corregirlo.

- Hoja electrónica de cálculo.

Vista en la pantalla de la PC consta de unas filas y columnas que conforman unas cuadrículas, en las cuales es posible capturar datos numéricos, titulos alfanuméricos y fórmulas de cálculo.

Al calcular estas expresiones los resultados aparecen en las cuadrículas que contienen las fórmulas. Si se efectúa un cambio en alguna de las cuadrículas implicadas en la operación, pueden recalcularse automáticamente todas las cuadrículas implicadas en el cambio a través de las fórmulas.

Permite elaborar informes, planificaciones, presupuestos y en general, todo trabajo que requiera cálculos repetitivos.

- Bases de datos.

Las bases de datos son una gran cantidad de datos relacionados entre sí. Estos datos se dividen en varias categorías: registros, archivos, bibliotecas, etc.

No es más que un potente manipulador de las relaciones existentes entre estas jerarquías de información, con él se pueden definir nuevas relaciones o acceder a los datos mediante las ya definidas.

- Gráficos.

Sirve para resumir en un dibujo toda una serie de datos que como tales, y sobre papel, son fríos y poco expresivos.

OFICINA.

Servicios que ofrece una PC en una oficina: procesadores de palabra, correo y mensajería electrónicos, lectores ópticos, videotextos, micrografía, videoconferencias, reconocimiento de voz.

La PC aparece en toda actividad profesional conocida como: Abogado, Contador, Laboratorista, Economista, Médico, Actuario, Arquitecto, Publicista, Administración de Empresas, Educador y/o Pedagogo, Ingeniero, Psicólogo, Agrónomo, Químico, Biólogo, etc.

También está presente en las telecomunicaciones, sin la PC no es posible imaginar resultados óptimos, y por medio de la conjunción de las telecomunicaciones y la computación: Telemática, se puede acceder a información almacenada en una base de datos, a través de las redes públicas de telecomunicación.

La simulación es una de las maneras más importantes en que la PC ayuda al hombre a realizar planificaciones para el futuro,

la computadora en base a un modelo puede generar millones de condiciones diferentes en muy poco tiempo, que pueden tener incidencias, en el modelo, así como también registrar y ordenar todos los resultados obtenidos, las simulaciones ayudan a prever equivocaciones costosas y a identificar nuevas posibilidades: operaciones dentro de un reactor nuclear, el uso de una nueva carretera, simulaciones en los viajes espaciales de una cápsula espacial ficticia, simulador de vuelo.

EDUCACION.

La importancia de las PCs radica en su popularidad y en su flexibilidad como sistema de cómputo. Su introducción ha permitido desarrollos en diversos campos que antes se consideraban propios del ser humano. Uno de estos campos es la educación, en la que el alumno ha pasado a ser un elemento activo en el proceso educativo. La posibilidad de simular escenarios en los que el alumno se sitúa como un sujeto activo con posibilidad de modificar las circunstancias que le rodean, y no sólo como un sujeto pasivo que recibe conocimientos sin intervenir en su educación, ha situado a la computación personal como la piedra angular de una nueva forma de educar en la que el objetivo es hacer pensar al educando. Por supuesto, la simulación por computadora no es privativo de las PCs, pero la sencillez de este sistema y su amplia disponibilidad, además de la enorme cantidad de sotfware y hardware compatible para ellas, las hace la mejor opción.

En una forma de enseñanza activa, al profesor lo libera del trabajo no creativo (listas de calificaciones, material informativo, etc) y dedica su tiempo a la atención de sus alumnos.

NEGOCIOS.

Un campo en el que también las PCs han incidido enormemente, es en el de los negocios. La toma de decisiones, la elaboración de gráficas y estadísticas, así como el control de inventarios, la contabilidad, el almacenamiento de datos en grandes cantidades, etc, se están desarrollando a través de las computadoras personales, utilizando la gran cantidad de software para PCs con que se cuenta, y la relativa facilidad para desarrollar programas

propios de ser necesario. Una vez mas, la enorme difusión de las PCs ha permitido que esto sea realidad, proporcionando un sistema relativamente barato y muy flexible al alcance de la población en general.

EN LA SOCIEDAD EN GENERAL.

Tenemos también a la PC en la astronomía, los bancos, en la cinematografía, juegos como ajedrez, bridge, pócker, videojuegos, redes locales que constituyen uno de los últimos avances tecnológicos, gracias a los progresos experimentados por el hardware y el software, Inteligencia Artificial que consiste en la asimilación de los procesos inductivos y deductivos del cerebro humano, en donde se están haciendo investigaciones en los siguientes campos: robótica, visión artificial, procesamiento del lenguaje natural, modelos del conocimiento y reconocimiento de patrones.

En el sector comercio se almacenan datos y se calculan costos, se llevan inventarios y se maneja a la empresa de una manera más organizada, en los deportes cada día se utilizan más las microcomputadoras, ya que no se puede organizar un acontecimiento deportivo sin utilizarlos, las encontramos cronometrando tiempos, dando datos estadísticos inmediatos, sistemas de resultados y en general controlan con toda exactitud la prueba.

Tomando un ámbito más general, la existencia de la computación personal ha permitido que en países con tecnología atrasada, como es el caso de México, se tenga acceso a recursos de computación, que de lo contrario, serían muy caros y probablemente imposibles de tener o mantener. Las PCs han repercutido favorablemente en el país gracias a su intervención en múltiples áreas, incrementando la productividad, y evitando que otras áreas caigan en el atraso tecnológico por falta de recusos de cómputo.

La lista de las aplicaciones de la PC se puede extender más y más, ya que los usos actuales y los futuros son numerosos.

A pesar de todo no olvidemos que las computadoras, sin el hombre no serían más que máquinas sin forma, ni esencia.

I.4. JUSTIFICACION DEL PRESENTE TRABAJO.

No ha pasado demasiado tiempo, desde el 6 de junio de 1958 en que se inicia la computación en México, con la creación del Centro de Cálculo Electrónico de la Facultad de Ciencias, en donde se instaló un equipo IBM 650 (cuya renta con todo y descuento era de $25,000), con capacidad RAM de 2 KB.

Ocasionalmente estos equipos no eran muy confiables, aún para los grandes precios que alcanzaban. Quizá desde esta época surgió el deseo de que las computadoras fueran 100% confiables.

Rápidamente, el mundo de la computación ha evolucionado, para llegar a la etapa de las PCs. El número de éstas se incrementa día con día, tanto en el mundo entero -se calculaba el 14 de marzo de 1988 en 10 millones el número de PCs, de las cuales eran de 35,000 a 40,000 marca IBM- como en México -la industria de las computadoras personales proporciona más de 500 millones de dólares al PIB-.

Actualmente existen dos corrientes que opinan acerca del futuro de las PCs:

- La primera dice que los días de las PCs están contados, que éstas serán desplazadas por modelos PS/2 o alguna otra máquina con mayor capacidad de proceso.

- La segunda indica que las PCs estarán en primer lugar de ventas durante unos años más, dado que la PS/2 y su micro canal no resuelven ninguna necesidad, y otros tipos de microcomputadoras no se ve por donde puedan llegar.

Tanto una opinión como la otra son sólo suposiciones, lo que sí es cierto es que en estos momentos es importante que las PCs sean lo más confiables y se encuentren en las mejores condiciones posibles, para que cumplan satisfactoriamente con su trabajo.

Entonces, de considerar la primer corriente mencionada como cierta, tenemos que las PCs serán descontinuadas por la gran mayoría de las empresas que se dedican al desarrollo de sistemas de apoyo a las PCs. En estas condiciones, cada día será mas difícil matenerlas en perfectas condiciones de operación. Estamos a tiempo de desarrollar un dispositivo para ayudar a esta tarea. Si tenemos en cuenta que los equipos de microcomputación que se

han tenido en la Facultad de Ingeniería -llámense Cromemco o Radio Shack- presentaron una larga vida útil, un dispositivo que ayude al mantenimiento de las PCs con las cuales cuenta -la Facultad de Ingeniería y la UNAM en general- parece ser de una atracción mayúscula.

Anteriormente, se mencionó que en las visitas realizadas a centros de cómputo de la UNAM se encontraron pocos centros de mantenimiento y detección de fallas de sus equipos PC. El sistema a desarrollar en este trabajo puede llenar ese vacío, sin distraer recursos humanos a los laboratorios, y eliminando la necesidad de instalaciones físicas y equipo a usar en un centro de mantenimiento.

Mas si se atiende a la segunda corriente, los equipos PCs seguirán llegando a la sociedad en general, por consiguiente el número de anomalías en los equipos se incrementarán. Para la atención del mantenimiento a los equipos se hace necesario ser más expedito, dada la revolución informática que surge a diario -mayores procesos, que no pueden ser retrasados-. Por cuestiones de importación no se pueden simplemente sustituir las tarjetas que conforman las PCs -menos aún la llamada mother board-, tenemos que llegar a nivel componente en el diagnóstico para la sustitución de la parte dañada. Es aquí donde surge la necesidad de desarrollar un sistema de diagnóstico para el hardware de una PC.

Observar simplemente el futuro desarrollo de las PCs no es el mejor camino, si se quiere salir de la actual dependencia tecnológica, es hora de desarrollar un sistema visionario para cualquiera que sea la historia futura.

CAPITULO II

DESCRIPCION DEL HARDWARE A DIAGNOSTICAR

II.1. CPU.
II.2. DMA.
II.3. MEMORIAS (RAM Y ROM).
II.4. PUERTO SERIE.
II.5. PUERTO PARALELO.
II.6. TECLADO.
II.7. RELOJ.
II.8. CONTROLADOR DE DISCO DURO.
II.9. CONTROLADOR DE DISCO FLEXIBLE.

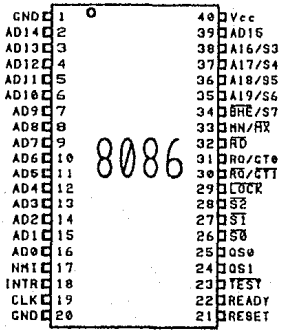
II. DESCRIPCION DEL HARDWARE A DIAGNOSTICAR.

II.1 MICROPROCESADOR DE 16 BITS 8086.

a) Introducción.

El 8086 es un CPU de 16 bits de alta eficiencia, cuenta con tres tipos de reloj de valor fijo: 5, 8 y 10 MHz. El CPU está implantado en canal-N, sus compuertas de silicio son de tecnología HMOS. EL 8086 opera con dos tipos de configuraciones, como un solo microprocesador o procesador múltiple para alcanzar su nivel alto de ejecución. Su capacidad directa de direccionamiento es de 1MB de memoria, su arquitectura está diseñada para un alto poder en lenguaje ensamblador y lenguajes de alto nivel, tiene 8 y 16 bits signados y no signados, se tiene aritmética en binario o decimal incluyendo la multiplicación y división.

b) Diagrama de terminales.



c) Descripción de Terminales.

Address Data Bus: Estas terminales son de I/O, constituyen el multiplexaje en el tiempo de memoria/direcciones (en T1) y bus de datos (en T2, T3, Tw y T4). AO es análogo a BHE para accesar al bus de datos, correspondiente a las terminales (D7-DO). Están en bajo durante Ti cuando un byte está por ser transferido sobre la parte baja del bus en operaciones de memoria. 8 bits orientados a dispositivos unen a la media baja que normalmente usa AO para función de chip select. Estas líneas se activan alto y no deben estar en tres estados durante la petición de una interrupción y la retención de reconocimiento en el bus local.

Address/Status: Estas terminales son de salida, durante el tiempo T1 estas son las cuatro líneas de dirección más significativas para operaciones en memoria. Durante operaciones I/O estas líneas están en bajo. Durante operaciones en memoria e I/O, información del status está disponible en estas líneas durante los tiempos T2,T3,Tw y T4. El status del bit de la bandera de interrupción (S5) es actualizado en el principio de cada ciclo de reloj (CLK). A17/S4 y A16/S3 son codificadas como sigue:

A17/S4	A16/S3	Característica
0(bajo)	0	alternativa de dato
0	1	stack
1(alto)	0	código o no
1	1	dato
S6 es 0 (bajo)		

Esta información indica que registro de relocalización es usado en el momento para accesamiento de datos. Estas líneas dejan de estar en 3 estados durante la retención de reconocimiento en el bus local.

Bus High Enable/Status: Durante T1 la señal BHE es usada para habilitar datos en la parte media alta del bus de datos (D15-D8). Los ocho bits orientados a dispositivos unen la parte media más alta del bus, normalmente usa BHE como condición de la función chip select. BHE está bajo durante T1 por ciclos de lectura/escritura y el reconocimiento de una interrupción cuando un byte

va a ser transferido en la parte alta del bus. La información del status (S7) es disponible durante los tiempos T2, T3, y T4. La señal se activa bajo, y deja de estar en tres estados durante la retención del bus. Es bajo durante T1 por el primer ciclo de interrupción reconocida .

Read: Esta señal es de salida, la habilitación de Read indica que el procesador está ejecutando en memoria un ciclo de lectura de I/O, dependiendo del estado de la terminal S2. Es usada para leer dispositivos que residen en el bus local del 8086. RD se activa bajo durante T2, T3 y Tw de cualquier ciclo de lectura, y está garantizado que permanece alto en T2 hasta que el bus local del 8086 esté indefinido. La señal deja de estar en tres estados durante la retención de reconocimiento.

READY: Esta señal es de entrada, y es el reconocimiento desde la memoria direccionada o dispositivo I/O, dicha señal completará la transferencia de datos. La señal de READY desde memoria o I/O es sincronizada por el 8284A (generador de reloj) para formar READY. Esta señal se activa alto. La entrada de READY del 8086 no está sincronizada. La operación correcta no está garantizada si la organización y los tiempos de retención no son encontrados.

INTR (Interrup request): Es una entrada de disparo por nivel, que es probada durante un ciclo de reloj de cada intrucción para determinar si el procesador debe aceptar la operación de reconocimiento de una interrupción. Una subrutina es direccionada por la vía del vector de interrupción, localizada en la tabla del sistema de memoria. Puede ser internamente mascarada por software apagando el bit de habilitación de interrupción. INTR está internamente sincronizada. Esta señal se activa alta.

TEST: Esta entrada es examinada por la instrucción "wait". Si la entrada TEST es baja, continua la ejecución, en otro caso el procesador espera en estado muerto. Esta entrada está sincronizada internamente durante cada ciclo de reloj con el flanco del CLK.

NMI (Non maskable interrupt): Esta es una señal de entrada que se genera por un diparo de flanco, que causa dos tipos de interrupciones. Una subrutina es direccionada por medio del vector de interrupción localizado en el sistema de memoria. NMI no es mascarable internamente por software. Una transición de un bajo a un alto inicializa la interrupción al fin de una instrucción actual. Esta entrada es internamente sincronizada

RESET: Esta terminal es una entrada, que causa que el procesador termine inmediatamente su actividad actual. Esta señal se activa alto por cuatro ciclos de reloj bajos. Esto restaura la ejecución, cuando RESET retorna a bajo. RESET está internamente sincronizada.

CLK (Clock): Provee la regulación del tiempo por el procesador y el controlador de bus. Es una señal de entrada.

Vcc: +5 Volts de suministro.

GND: Tierra Electrica.

MN/MX (Minimum/Maximum): Indica el modo de operación del microprocesador.

STATUS (S2, S1, S0): Estas señales son de salida, se activan durante T4, T1, y T2 y retornarán al estado pasivo (1, 1, 1) durante T3 o durante Tw cuando READY es alta. Estos estados son usados por el 8288 (controlador del bus) para generar todas las señales de control de acceso a memoria e I/O. Cualquier cambio por S2, S1 o S0 durante T4 es usado para indicar el principio del ciclo del bus, y el retorno al estado pasivo en T3 o Tw es usado para indicar el final del ciclo del bus. Estas líneas dejan de estar en tres estados con HOLDA. Estas líneas se codifican de la siguiente manera:

S2	S1	S0	Caracteristicas
0 (Low)	0	0	Interrupción Reconocida.
0	0	1	Puerto de lectura I/O.
0	1	0	Puerto de escritura I/O.
0	1	1	Halt
1 (High)	0	0	Código de acceso
1	0	1	Lectura en memoria.
1	1	0	Escritura en memoria.
1	1	1	Estado pasivo

Request/Grant: Esta señal es de I/O, y es usada por otro bus local maestro para forzar al procesador a liberar al bus local al final del ciclo del bus actual. Cada terminal es bidireccional con RQ/GT0 teniendo una alta prioridad en RQ/GT1.

La secuencia de Request/Grant es como sigue:

1.- Un pulso de un CLK desde otro bus local maestro indica el requerimiento del bus local ("hold") al 8086 (primer pulso)

2.- Durante un ciclo de reloj T4 ó T1, un pulso de un CLK desde el 8086 para el requerimiento maestro (segundo pulso), indica que el 8086 ha permitido que el bus local deje de estar en tres estados para aceptar la retención de reconocimiento, estado del siguiente CLK. El bus de interface con el CPU es desconectado lógicamente desde el bus local durante "HOLDA".

3.- Un pulso CLK desde el requerimiento maestro indica al 8086 (tercer pulso) que la retención de requerimiento está casi por terminar y que el 8086 puede utilizar el bus local en el siguiente CLK.

Cada cambio maestro-maestro es una secuencia de tres pulsos, deberá existir un ciclo muerto de reloj en cada cambio de bus. Los pulsos se activan bajo.

Si el requerimiento es pedido mientras el CPU está ejecutando un ciclo en memoria, éste liberará al bus local durante el ciclo T4, cuando las siguientes condiciones se den:

1.- Que el requerimiento actual se dé antes o en T2.
2.- Que en el ciclo actual no esté el byte bajo de una palabra (sobre una dirección anterior).
3.- Que en el ciclo actual no esté el primer reconocimento de una secuencia de interrupción atendida.
4.- Una instrucción no esté actualmente ejecutándose.

Si el bus local está ocioso cuando el requerimiento es hecho los dos eventos posibles son los siguientes:

1.- El bus local será liberado durante el siguiente CLK.
2.- Un ciclo en memoria principiará con tres ciclos de reloj.

LOCK: Esta es una señal de salida, que indica que el bus maestro de otro sistema no está listo para lograr el control del bus del sistema mientras LOCK se active en bajo. La señal de LOCK es activada por la instrucción anterior "LOCK" y permanece activa hasta la terminación de la siguiente instrucción. Esta señal se

activa bajo y deja de estar en tres estados en "HOLDA".

Queue Status: El enclamiento del Status es válido durante el ciclo CLK después que la operación QUEUE es ejecutada.

QS1 y QS0 proveen el status para permitir la localización externa de la instruccion QUEUE interna del 8086.

Estas señales se codifican de la siguiente manera:

QS1	QS0	caracteristicas
0(LOW)	0	no operación
0	1	primer byte de Op code desde la cola
1(HIGH)	0	vacia la cola
1	1	Subsecuente byte desde la cola

Status line: Esta terminal es de salida, y lógicamente es equivalente a la señal S2 en el modo máximo. Esta es usada para distinguir un acceso a memoria desde un acceso I/O. M/$\overline{IO}$ se hace válida en T4, precediendo un ciclo del bus y permanece válida hasta el final del ciclo T4 (M=alto, $\overline{IO}$=bajo). M/$\overline{IO}$ está indifinida para que el bus local no esté en tres estados durante HOLDA.

Write: Esta es una señal de salida, indica que el procesador está ejecutando un ciclo de escritura en I/O o escritura en memoria, dependiendo del estado de la señal M/$\overline{IO}$. $\overline{WR}$ se activa por T2, T3 y Tw de cualquier ciclo de escritura. Esta se activa bajo, y es in definida para quitar al bus local de tres estados durante HOLDA.

$\overline{INTA}$: Esta es una señal de salida y es usada como un habilitador de lectura por ciclos de interrupción reconocida, se activa bajo durante T2, T3 y Tw de cada ciclo de interrupción reconocida.

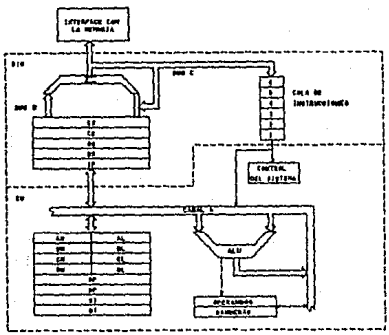
Address Latch Enable: Esta es una señal de salida proporcionada por el procesador para las direcciones del latch a través del 8282/8283. Esta es un pulso activo alto durante T1 en cualquier ciclo del bus. Nota: Esta señal no es indefinida.

Data Transmit/receive: Esta es una señal de salida, necesitada en modo mínimo, se utiliza para el uso del 8286/8287 transmisor del bus de datos. Esta es usada para control de la dirección de datos a través de la salida de transferencia. Lógicamente DT/$\overline{R}$ es equivalente a $\overline{S1}$ en modo máximo. Esta señal se coloca en tres estados para que el bus local no esté en tres estados.

Data Enable: Esta señal es de salida, habilitada por el 8286/8287 en un sistema mínimo que usa la transferencia. DEN se activa bajo durante cada acceso a memoria e I/O y por ciclos INTA. Por cada lectura o ciclo INTA ésta se activa desde la mitad de T2 hasta la mitad de T4, mientras por un ciclo de escritura ésta se activa desde el principio de T2 hasta la mitad de T4. DEN flota para que el bus local no esté en tres estados en HOLDA.

HOLD: Es una señal de I/O, indica que otro maestro está requiriendo el bus local ("hold"). Para ser reconocida HOLD deberá activarse alta. El procesador recibiendo el "hold" requerido enviará HOLDA (alta) como un reconocimiento en la mitad del ciclo de reloj T1. Simultáneamente con la emisión de HOLDA el procesador flotará al bus local y las líneas de control. Después de que HOLD es detectado en bajo, el procesador pondrá, en bajo a HOLDA, y cuando el procesador necesite correr otro ciclo, éste otra vez manejará el bus local y las líneas de control. HOLD no es una entrada asíncrona.

d) Diagrama de Bloques Interno.



e) Descripcion del Diagrama de Bloques.

Organización de Memoria: El procesador proporciona 20 bits de dirección para memoria, que localiza el byte referenciado en ese momento. La memoria está organizada como un arreglo lineal de arriba de un millón de bytes, direccionados como 00000H a FFFFFH.

La memoria está lógicamente dividida en segmentos de código, datos, datos extra y el stack, con 64 Kbytes cada uno, con cada segmento direccionado con 16 bits.

Toda la memoria referenciada está hecha relativa para direcciones contenidas en segmentos de registros de alta velocidad. Los tipos de segmentos serán seleccionados basándose en el direccionamiento necesitado por los programas. El segmento de un registro a ser seleccionado es automáticamente elegido de acuerdo a las reglas de la siguiente tabla.

referencia de memoria necesitada.	Segmento de registros usados.	segmento selección de regla
Instrucciones	CODE (CS)	Automático con toda instrucción de prefetch
Stack	STACK (SS)	En todos los pushes y pops en el stack. Referenciado a la memoria relativa está el registro base BP excepto referencia de datos
Datos Locales	DATA (DS)	Referencia de datos cuando: relativo al stack, destino de la operación de un string ó explícitamente una sobreescritura
Datos Externos (globales)	EXTRA (ES)	Destinado a operaciones de string: Específicamente usando un segmento de sobreescritura.

Toda la información en un mismo tipo de segmento comparte los mismos atributos lógicos (cómo datos y código). Por la estructura

de la memoria en áreas relocalizables de similares características y por la selección automática de segmentos de registros, programas cortos, rápidos y más estructurados.

Operaciones de palabras de (16-bits) pueden ser localizadas en direcciones impares, no manteniendo direcciones siempre pares, como es el caso de muchas computadoras de 16-bits. Para direcciones y operaciones de datos el byte menos significativo de la palabra se encuentra en la posición baja de la dirección valuada y el byte más significativo en la siguiente posición alta de la dirección. El BIU automáticamente ejecuta el número apropiado del acceso a memoria, uno si la palabra operada es el límite de un byte en el nivel, dos si está sobre el límite de un byte viejo. Excepto por la ejecución, este acceso es transparente al software. Esta penalización de ejecución no debe ocurrir por instrucciones en fetch, sólo en operaciones de palabras.

Fisicamente la memoria está organizada como un banco alto (D15-D8) y un banco bajo (D7-D0) de 512K de 8-bits cada byte direccionados en paralelo por las líneas de dirección de los procesadores.

A19-A1. El byte de datos con direcciones actuales es transferido sobre las líneas del bus (D7-D0) mientras el byte de datos viejo direccionado (A0 alto) es transferido sobre las líneas del bus (D15-D8). El procesador proporciona dos señales de habilitación, $\overline{BHE}$ y A0, para selectivamente permitir lectura ó escritura en la posición de un byte viejo, en la posición de un byte actual, o ambos. La intrucción que esté en ese momento es tomada como palabra desde memoria y direccionada por el procesador por el nivel del byte necesariamente.

Operación del Bus: El 8086 tiene comunmente direcciones diferentes y bus de datos referidos al multiplexaje en el tiempo del bus. Esta técnica permite el uso más oficiente de terminales en el procesador a la vez que se permite el uso de un standard. Este bus local puede ser habilitado directamente y usado a través del sistema con direcciones latching proporcionadas sobre memoria y módulos I/O. En adicion el bus puede también ser demultiplexado al procesador con un sólo conjunto de las direcciones sincronizadas si el standar de un bus no multiplexado es deseado por el sistema.

En cada procesador su ciclo de bus es menos de cuatro ciclos CLK. Estos están referidos a T1, T2, T3 y T4. Las direcciones son emitidas desde el procesador durante T1 y la transferencia de datos ocurrida sobre el bus durante T3 y T4. T2 es usada primeramente para cambiar la dirección del bus en operaciones de lectura. En el evento un "NOT READY" indica que es especificado por el direccionado dispositivo, los estados de espera (Tw) son insertados entre T3 y T4. Por cada inserción el estado de espera "wait" (Tw) es de la misma duración que un ciclo CLK. Los períodos pueden ocurrir entre ciclos del bus del 8086. Estos están referidos como a un estado "ocioso" (Ti) o un ciclo inactivo CLK. El procesador utiliza estos ciclos para manejo interno.

Durante T1 en cualquier ciclo del bus la señal ALE (Address Latch Enable) es emitida ya sea por el procesador o el controlador del bus 8288, dependiendo sobre las señales MN/$\overline{MX}$). La salida del flanco de este pulso indica una dirección válida y un estado de información cierto por el ciclo permitido a ser retenido.

Los bits de status $\overline{S0}$, $\overline{S1}$, y $\overline{S2}$ son usados, en modo máximo por el controlador del bus, para identificar el tipo de transacción del bus de acuerdo a la siguiente tabla:

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Características
0(low)	0	0	Reconocimiento de interrupción
0	0	1	Lectura I/O
0	1	0	Escritura I/O
0	1	1	Halt
1(High)	0	0	Instrucción Fetch
1	0	1	Lectura de datos desde Memoria
1	1	0	Escritura de datos a memoria
1	1	1	Pasivo (Ciclo de no bus)

El bit de status S3 a través de S7 son multiplexados con bits de dirección de alto orden y la señal $\overline{BHE}$, y hasta entonces es válida durante T2 a través de T4. S3 y S4 indican que el segmento de registro fué usado por este ciclo de bus en las direcciones formadas de acuerdo a la siguiente tabla:

S4	S3	Características
0(low)	0	Alternativa de datos
0	1	Stack
1(High)	0	Código ó nada
1	1	Datos

S5 es un reflejo del bit de interrupción. S6=0 y S7 son los bits de status disponibles.

Direccionamiento I/O: En el 8086, las operaciones I/O tienen direcciones para abacar un máximo de 64K de registros I/O de un byte ó 32K de registros I/O de una palabra. Las direcciones I/O aparecen en el mismo formato como las direcciones de memoria sobre las lineas del bus (A0-A15). Las lineas de dirección (A19-A16) son cero en operaciones I/O. Las instrucciones variables I/O que usan el registro DX como un apuntador tienen una completa capacidad de dirección, mientras las direcciones directas I/O no.

Los puertos I/O son direccionados de la misma manera que la posición en memoria. Los bytes pares direccionados son transferidos sobre las lineas del bus (D7-D0), y los bytes impares direccionados sobre (D15-D8). Esto debe ser tomado con cuidado para asegurar que cada registro en un periférico de 8-bits esté localizado en la parte baja del bus para ser direccionado como una dirección par.

f) Función General del Circuito.

La función interna del procesador 8086 está particionada logicamente a través de dos unidades de procesamiento. La primera es la Bus Interface Unit(BIU) y la segunda es la Unidad de Ejecución(EU).

Estas unidades pueden interactuar directamente pero la mayor parte del tiempo trabajan separadas. El BIU proporciona las funciones relacionadas a instrucciones del ciclo de fetch y el encolamiento, los operandos para el fetch y el almacenamiento y relocalización de direcciones. Esta unidad también proporciona el control básico del bus. El traslape de instrucción pre-fetching proporcionada por esta unidad de servicio para incrementar la

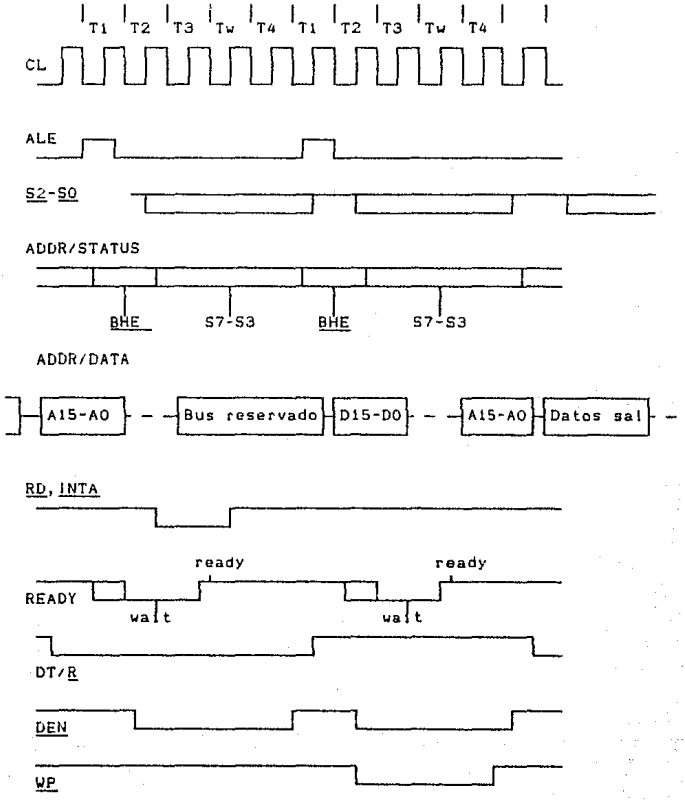
ejecución del procesador a través del incremento del ancho de banda del bus. Más de 6 bytes de instrucción pueden ser encolados mientras que sucede la decodificacion y la ejecución.

La instrucción encolada del mecanismo, posibilita al BIU para mantener la memoria utilizada muy eficientemente. En cualquier momento hay un espacio para dos bytes menos en la cola. El BIU intentará el ciclo fetch de una palabra en memoria. Esto reduce en mucho el tiempo muerto sobre el bus de la memoria. La cola actua como un buffer (FIFO: primero en entrar, primero en salir) desde el cual la unidad de ejecución extrae los bytes de instrucción requeridos. Si la cola está vacía (siguiendo una instrucción de salto por ejemplo) el primer byte introducido en la cola estará inmediatamente disponible para la unidad de ejecución.

La unidad de ejecución recibe instrucciones pre-fecht desde la cola del BIU y proporciona direcciones de operandos relocalizados para el BIU. Los operandos de memoria son pasados a través del BIU para el procesamiento por la unidad de ejecución, que pasa resultados por el BIU para almacenarse.

g) Diagramas de Tiempo:

Sistema Básico de Tiempo.



h) Programacion del circuito

Inicializacion del CPU

La inicialización del CPU se realiza mediante la colocación de un voltaje alto en la terminal RESET del CPU. Cuando se coloca un voltaje alto en este pin, por más de 4 ciclos de reloj, el procesador empezará una secuencia interna para terminar las operaciones que estaba realizando. Posteriormente, cuando reciba el flanco de bajada de la señal RESET, iniciará una secuencia de inicialización interna. Terminada esta secuencia, el CPU pasa a procesar la instrucción que se encuentra en la localidad FFFF0H de la memoria.

Interrupciones.

Para realizar la interface con el exterior, el 8086/88 cuenta con dos tipos de interrupciones, las generadas por software y las generadas por hardware. Las interrupciones por software se realizan internamente al CPU mediante una de las instrucciones del del conjunto de ellas, reportando el número de interrupción requerida. En cambio, las interrupciones por hardware se producen por dispositivos externos al CPU. Estas interrupciones se dividen en dos: mascarables y no mascarables dependiendo de si pueden ser o no ignoradas, respectivamente, por el CPU. Las interrupciones mascarables se generan cuando existe un voltaje alto en la terminal INTR, mientras que las no mascarables ocurren cuando existe un alto en la terminal NMI.

Independientemente al tipo de interrupción generada por hardware, el CPU necesita el número de la interrupción requerida, que puede ser fijo (para la NMI) o variable (para la INTR), en cuyo caso debe ser proporcionado por el dispositivo que interrumpe. Este número es multiplicado por cuatro para posteriormente servir como apuntador a una tabla de vectores de interrupción almacenada en los primeros 256 bytes de la memoria.

La interrupción no mascarable se produce únicamente por la presencia de una transición en la terminal NMI de un voltaje bajo a uno alto. Este voltaje alto debe permanecer así durante al menos dos ciclos de reloj. Cuando se reconoce esta interrupción, se genera una interrupción número 2, la cual será servida hasta que termine de ejecutarse la actual instrucción, o en su defecto, entre movimientos de bloques de una instrucción. En el peor de los

casos, cuando se está ejecutando una instrucción como la multiplicación o la división, puede suceder que la interrupción no mascarable no sea servida a tiempo debido a que haya vuelto a un voltaje bajo. La señal NMI debe estar libre de picos de voltaje que produzcan respuestas indeseables.

Las interrupciones mascarables, cuya petición se reconoce con un voltaje alto en la terminal INTR, pueden ser ignoradas si internamente se ha colocado en cero la bandera de interrupciones. Para responder a estas interrupciones, la señal INTR debe estar en alto durante el último ciclo de la instrucción que está siendo ejecutada, o al final de movimientos de bloques de una instrucción. Durante la respuesta a la interrupción, el CPU deshabilita todas las demás interrupciones colocando la bandera de interrupciones en cero. Posteriormente, el CPU realiza dos ciclos de reconocimiento de interrupción uno enseguida del otro, activando la señal INTA, y recibiendo en el segundo ciclo el número de interrupción solicitada proveniente del sistema de interrupciones externo (por ejemplo, el 8259A). Cuando el CPU sirve a una interrupción la primera acción que realiza es guardar en el stack el registro de estado, para posteriormente, al término de la interrupción, restaurarlo.

Conjunto de Instrucciones

El 8086/88 cuenta con un amplio conjunto de instrucciones que se dividen según su categoría. A continuación se proporciona un listado condensado de estas intrucciones.

Transferencia de Datos

	7 6 5 4 3 2 1 0
MOV (Move): Registro/memoria	1 0 0 0 1 0 d w
	mod reg r/m
Inmediato a registro/memoria	1 1 0 0 0 1 1 w
	mod 0 0 0 r/m
	datos

	7 6 5 4 3 2 1 0
	datos si w = 1
Inmediato a registro	1 0 1 1 w reg
	datos
	datos si w = 1
Memoria a acumulador	1 0 1 0 0 0 0 w
	direc baja
	direc alta
Acumulador a memoria	1 0 1 0 0 0 1 w
	direc baja
	direc alta
reg/mem a registro de seg.	1 0 0 0 1 1 1 0
	mod 0 reg r/m
reg de seg a reg/mem	1 0 0 0 1 1 0 0
	mod 0 reg r/m
PUSH (Push): Registro/memoria	1 1 1 1 1 1 1 1
	mod 1 1 0 r/m
Registro	0 1 0 1 0 reg
Registro de segmento	0 0 0 reg 1 1 0

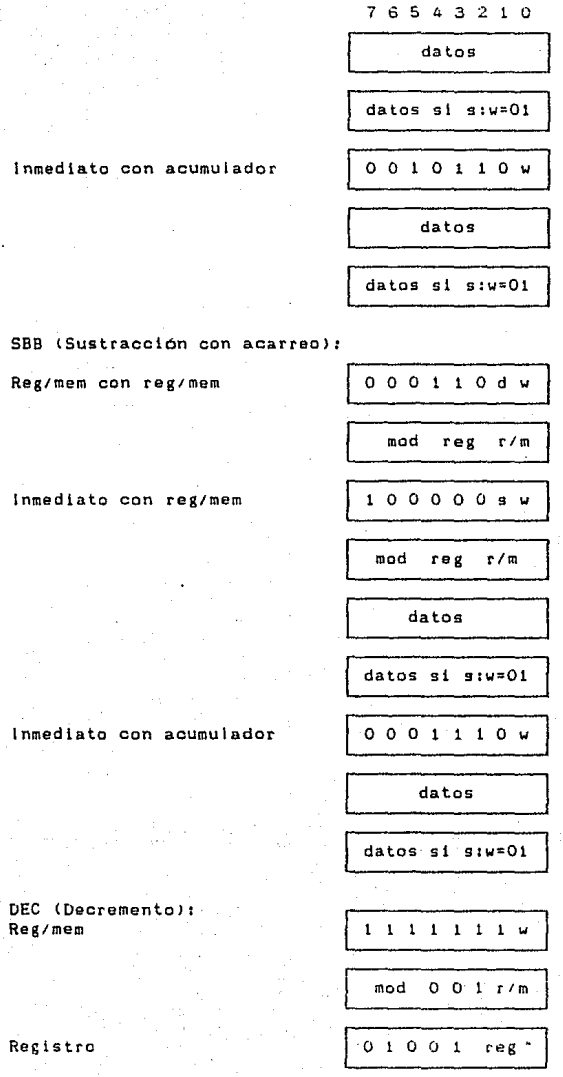
	7 6 5 4 3 2 1 0
POP (Pop): Registro/memoria	1 0 0 0 1 1 1 1
	mod 0 0 0 r/m
Registro	0 1 0 1 1 reg
Registro de segmento	0 0 0 reg 1 1 1
XCHG (Intercambio): Registro/memoria con reg.	1 0 0 0 0 1 1 w
	mod reg r/m
Registro con acumulador	1 0 0 1 0 reg
IN (Lect. puerto): Puerto fijo	1 1 1 0 0 1 0 w
	puerto
Puerto variable	1 1 1 0 1 1 0 w
OUT (Esc. Puerto): Puerto fijo	1 1 1 0 0 1 1 w
	puerto
Puerto variable	1 1 1 0 1 1 1 w
XLAT (Translada byte a AL)	1 1 0 1 0 1 1 1
LEA (Carga EA a registro)	1 0 0 0 1 1 0 1
	mod reg r/m
LDS (Carga apuntador a DS)	1 1 0 0 0 1 0 1
	mod reg r/m

	7 6 5 4 3 2 1 0
LES (Carga apuntador a ES)	1 1 0 0 0 1 0 0
	mod reg r/m
LAHF (Carga AH con banderas)	1 0 0 1 1 1 1 1
SAHF (Carga banderas con AH)	1 0 0 1 1 1 1 0
PUSHF (Push a banderas)	1 0 0 1 1 1 0 0
POPF (Pop a banderas)	1 0 0 1 1 1 0 1

Operaciones Aritméticas

ADD (Suma): Reg/mem con reg/mem	0 0 0 0 0 0 d w
	mod reg r/m
Inmediato a reg/mem	1 0 0 0 0 0 s w
	mod 0 0 0 r/m
	datos
	datos si s:w = 01
Inmediato a acumulador	0 0 0 0 0 1 0 w
	datos
	datos si s:w=01
ADC (Suma con acarreo):	
Reg/mem con reg/mem	0 0 0 1 0 0 d w

	7 6 5 4 3 2 1 0
	mod reg r/m
Inmediato con reg/mem	1 0 0 0 0 0 s w
	mod 0 1 0 r/m
	datos
	datos si s:w=01
Inmediato a acumulador	0 0 0 1 0 1 0 w
	datos
	datos si s:w =01
INC (Incremento): Reg/mem	1 1 1 1 1 1 1 w
	mod 0 0 0 r/m
Registro	0 1 0 0 0 reg
AAA (Ajuste ASCII para suma)	0 0 1 1 0 1 1 1
DAA (Ajuste decimal para suma)	0 0 1 0 0 1 1 1
SUB (Substracción):	
Reg/mem con reg/mem	0 0 1 0 1 0 d w
	mod reg r/m
Inmediato con reg/mem	1 0 0 0 0 0 s w
	mod reg r/m


7 6 5 4 3 2 1 0
datos
datos si s:w=01
Inmediato con acumulador
0 0 1 0 1 1 0 w
datos
datos si s:w=01
SBB (Sustracción con acarreo):
Reg/mem con reg/mem
0 0 0 1 1 0 d w
mod reg r/m
Inmediato con reg/mem
1 0 0 0 0 0 s w
mod reg r/m
datos
datos si s:w=01
Inmediato con acumulador
0 0 0 1 1 1 0 w
datos
datos si s:w=01
DEC (Decremento):
Reg/mem
1 1 1 1 1 1 1 w
mod 0 0 1 r/m
Registro
0 1 0 0 1 reg

	7 6 5 4 3 2 1 0
NEG (Cambio de signo)	1 1 1 1 0 1 1 w
	mod 0 1 1 r/m
CMP (Comparación): Reg/mem con reg.	0 0 1 1 1 0 d w
	mod reg r/m
Inmediato con reg/mem	1 0 0 0 0 0 s w
	mod 1 1 1 r/m
	datos
	datos si s:w=01
Inmediato con acumulador	0 0 1 1 1 1 0 w
	datos
	datos si w = 1
AAS (Ajuste ASCII para sub)	0 0 1 1 1 1 1 1
DAS (Ajuste decimal para sub)	0 0 1 0 1 1 1 1
MUL (Mult. Sin signo)	1 1 1 1 0 1 1 w
	mod 1 0 0 r/m
AAM (Ajuste ASCCI para mult)	1 1 0 1 0 1 0 0
	0 0 0 0 1 0 1 0

	7 6 5 4 3 2 1 0
DIV (División sin signo)	1 1 1 1 0 1 1 w
	mod 1 1 0 r/m
IDIV (División signada)	1 1 1 1 0 1 1 w
	mod 1 1 1 r/m
AAD (Ajuste ASCII para div)	1 1 0 1 0 1 0 1
	0 0 0 0 1 0 1 0
CBW (Convierte byte a word)	1 0 0 1 1 0 0 0
CWD (Convierte word a dword)	1 0 0 1 1 0 0 1

Operaciones Lógicas

NOT (Negación)	1 1 1 1 0 1 1 w
	mod 0 1 0 r/m
SHL/SAL (Corrimientos a izq)	1 1 0 1 0 0 x w
	mod 1 0 0 r/m
SHR (Corrimiento lógico a der)	1 1 0 1 0 0 x w
	mod 1 0 1 r/m
SAR (Corrimiento arit. a der)	1 1 0 1 0 0 x w
	mod 1 1 1 r/m
ROL (Rotación a la izq)	1 1 0 1 0 0 x w

	7 6 5 4 3 2 1 0
	mod 0 0 0 r/m
ROR (Rotación a la der)	1 1 0 1 0 0 x w
	mod 0 0 1 r/m
RCL (Rot. izq. con acarreo)	1 1 0 1 0 0 x w
	mod 0 1 0 r/m
RCR (Rot. der. con acarreo)	1 1 0 1 0 0 x w
	mod 0 1 1 r/m
AND (And lógico): Reg/mem con reg/mem	0 0 1 0 0 0 d w
	mod reg r/m
Inmediato con reg/mem	1 0 0 0 0 0 0 w
	mod 1 0 0 r/m
	datos
	datos si w = 1
Inmediato con acumulador	0 0 1 0 0 1 0 w
	datos
	datos si w = 1
TEST (And sin resultado y con banderas modif): Reg/mem con reg/mem	1 0 0 0 0 1 0 w

	7 6 5 4 3 2 1 0
	mod reg r/m
Inmediato con reg/mem	1 1 1 1 0 1 1 w
	mod 1 0 0 r/m
	datos
	datos si w = 1
Inmediato con acumulador	0 0 1 0 0 1 0 w
	datos
	datos si w = 1
OR (Or lógica): Reg/mem con reg/mem	0 0 0 0 1 0 d w
	mod reg r/m
Inmediato con reg/mem	1 0 0 0 0 0 0 w
	mod 0 0 1 r/m
	datos
	datos si w = 1
Inmediato con acumulador	0 0 0 0 1 1 0 w
	datos
	datos si w = 1

	7 6 5 4 3 2 1 0
XOR (Or exclusiva lógica): Reg/mem con reg/mem	0 0 1 1 0 0 d w
	mod reg r/m
Inmediato con reg/mem	1 0 0 0 0 0 0 w
	mod 1 1 0 r/m
	datos
	datos si w = 1
Inmediato con acumulador	0 0 1 1 0 1 0 w
	datos
	datos si w = 1

Manipulación de cadenas

REP (Repetición)	1 1 1 1 0 0 1 z
MOVS (Mover byte/word)	1 0 1 0 0 1 0 w
CMPS (Compara byte/word)	1 0 1 0 0 1 1 w
SCAS (Busca byte/word)	1 0 1 0 1 1 1 w
LODS (Carga byte/word a AL/AX)	1 0 1 0 1 1 0 w
STOS(Carga byte/word de AL/AX)	1 0 1 0 1 0 1 w

Transferencia de control

CALL (Llamada a rutina)	7 6 5 4 3 2 1 0
Directo en el segmento	1 1 1 0 1 0 0 0
	desp bajo
	desp alto
Indirecto en el segmento	1 1 1 1 1 1 1 1
	mod 0 1 0 r/m
Directo intersegmento	1 0 0 1 1 0 1 0
	offset bajo
	offset alto
	segmento bajo
	segmento alto
Indirecto intersegmento	1 1 1 1 1 1 1 1
	mod 0 1 1 r/m
JMP (Salto incondicional): Directo en el segmento	1 1 1 0 1 0 0 1
	desp bajo
	desp alto
Directo y corto en el seg.	1 1 1 0 1 0 1 1
	desplazamiento

	7 6 5 4 3 2 1 0
Indirecto en el segmento	1 1 1 1 1 1 1 1
	mod 1 0 0 r/m
Directo intersegmento	1 1 1 0 1 0 1 0
	offset bajo
	offset alto
	segmento bajo
	segmento alto
Indirecto intersegmento	1 1 1 1 1 1 1 1
	mod 1 0 1 r/m
RET (Regreso de rutina): En el mismo segmento	1 1 0 0 0 0 1 1
En el seg. y añadiendo el SP	1 1 0 0 0 0 1 0
	datos bajo
	datos alto
Intersegmento	1 1 0 0 1 0 1 1
Intersegmento añadiendo al SP	1 1 0 0 1 0 1 0
	datos bajo
	datos alto

	7 6 5 4 3 2 1 0
JE/JZ (Salto si igual/cero)	0 1 1 1 0 1 0 0
	desplazamiento
JL/JNGE (Salto si menor / no mayor o igual)	0 1 1 1 1 1 0 0
	desplazamiento
JLE/JNG (Salto si menor o igual/no mayor)	0 1 1 1 1 1 1 0
	desplazamiento
JB/JNAE (Salto si menor/ no mayor o igual)	0 1 1 1 0 0 1 0
	desplazamiento
JBE/JNA (Salto si menor o igual/no mayor)	0 1 1 1 0 1 1 0
	desplazamiento
JP/JPE (Salto si paridad/ paridad impar)	0 1 1 1 1 0 1 0
	desplazamiento
JO (Salto si hubo sobreflujo)	0 1 1 1 0 0 0 0
	desplazamiento
JS (Salto si resul. negativo)	0 1 1 1 1 0 0 0
	desplazamiento

	7 6 5 4 3 2 1 0
JNE/JNZ (Salto si no igual/ no cero)	0 1 1 1 0 1 0 1
	desplazamiento
JNL/JGE (Salto si no menor/ mayor o igual)	0 1 1 1 1 1 0 1
	desplazamiento
JNLE/JG (Salto si no menor o igual /mayor)	0 1 1 1 1 1 1 1
	desplazamiento
JNB/JAE (Salto si no menor/ mayor o igual)	0 1 1 1 0 0 1 1
	desplazamiento
JNB/JA (Salto si no menor o igual/mayor)	0 1 1 1 0 1 1 1
	desplazamiento
JNP/JPO (Salto si no par/ paridad impar)	0 1 1 1 0 0 0 1
	desplazamiento
JNO (Salto si no sofreflujo)	0 1 1 1 0 0 0 1
	desplazamiento
JNS (Salto si no signo)	0 1 1 1 1 0 0 1
	desplazamiento

	7 6 5 4 3 2 1 0
LOOP (Ciclo de cx veces)	1 1 1 0 0 0 1 0
	desplazamiento
LOOPZ/LOOPE (Ciclo mientras cero / igual)	1 1 1 0 0 0 0 1
	desplazamiento
LOOPNZ/LOOPNE (Ciclo mientras no cero/no igual)	1 1 1 0 0 0 0 0
	desplazamiento
JCXZ (Salta si cx es cero)	1 1 1 0 0 0 1 1
	desplazamiento
INT (Interrupción): Con número especificado	1 1 0 0 1 1 0 1
	número
Número 3	1 1 0 0 1 1 0 0
INTO (Int. por sobreflujo)	1 1 0 0 1 1 1 0
IRET (Retorno de una int)	1 1 0 0 1 1 1 1

Control del procesador

CLC (Borrar band. de acarreo)	1 1 1 1 1 0 0 0
CMC (Complemento del acarreo)	1 1 1 1 0 1 0 1
STC (Prender band. de acarreo)	1 1 1 1 1 0 0 1

	7 6 5 4 3 2 1 0
CLD (Borrar band. de dir)	1 1 1 1 1 1 0 0
STD (Prender band. de dir)	1 1 1 1 1 1 0 1
CLI (Borrar band. de int)	1 1 1 1 1 0 1 0
STI (Prende band. de int)	1 1 1 1 1 0 1 1
HLT (Paro)	1 1 1 1 0 1 0 0
WAIT (Espera)	1 0 0 1 1 0 1 1
ESC (Escape a disp. externo)	1 1 0 1 1 x x x
	mod x x x r/m
LOCK (Seguro del bus)	1 1 1 1 0 0 0 0

Notas:

AL = parte baja del acumulador (8 bits).

AX = acumulador de 16 bits.

CX = Registro contador.

DS = Apuntador al segmento de datos.

ES = Apuntador a extrasegmento.

Se utiliza la nomenclatura alto y bajo para nombrar los bytes más y menos significativos, respectivamente, de un dato.

Los bytes de una misma instrucción están apilados desde el más significativo hasta el menos significativo.

Los saltos que posean letras B o A se refieren a operaciones no signadas.

Los saltos que posean letras G o L se refieren a operaciones signadas.

Si d=1 entonces se opera hacia registros, si no, se opera desde los registros.

Si w=1 entonces la instrucción es por palabra (word), si no la instrucción es por byte.

Si mod=11 entonces r/m es tratado como un campo REG

Si mod=00 * entonces DESP=0

Si mod=01 entonces DESP=desp bajo con signo extendido a 16 bits.

Si mod=10 entonces DESP=desp alto : disp bajo

Si r/m = 000 entonces EA (Efective Addres) = (BX) + (SI) + DESP

Si r/m = 001 entonces EA = (BX) + (DI) + DESP

Si r/m = 010 entonces EA = (BP) + (SI) + DESP

Si r/m = 011 entonces EA = (BP) + (DI) + DESP

Si r/m = 100 entonces EA = (SI) + DESP

Si r/m = 101 entonces EA = (DI) + DESP

Si r/m = 110 entonces EA = (BP) + DESP

Si r/m = 111 entonces EA = (BX) + DESP

DISP sigue a continuación de los dos primeros bytes de la instrucción (en el caso de que sea requerido).

* Excepto si mod = 00 y r/m = 110 entonces EA = desp alto : desp bajo

Si s:w = 01 entonces los inmediatos 16 bits forman el operando.

Si s:w = 11 entonces el byte inmediato tiene signo extendido para formar el operando de 16 bits.

Si v = 0 entonces el 'contador' = 1, si no, el 'contador' está en (CL).

x = no importa

Prefijo de segmento

0 0 1 reg 1 1 0

Reg tiene el valor según la siguiente tabla:

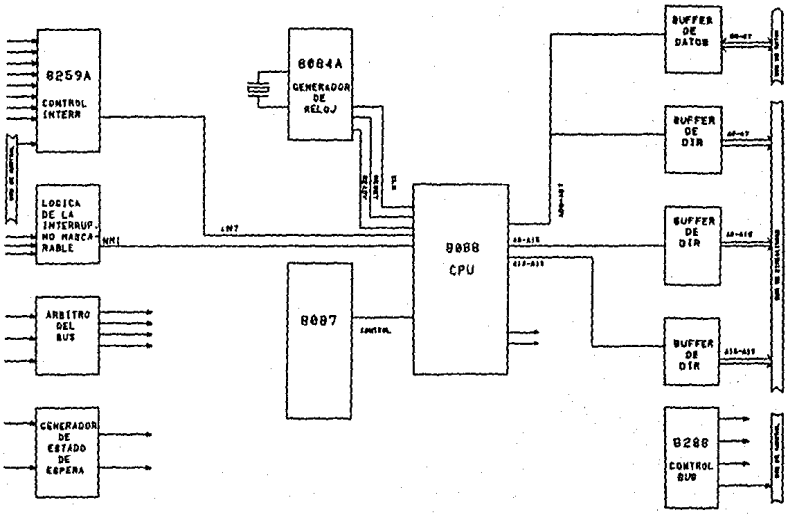
16 bits (w=1)		8 bits (w=0)		Segmentos	
000	AX	000	AL	00	ES
001	CX	001	CL	01	CS
010	DX	010	DL	10	SS
011	BX	011	BL	11	DS
100	SP	100	AH		
101	BP	101	CH		
110	SI	110	DH		
111	DI	111	BH		

Banderas (Flags):

XXXX (OF)(DF)(IF)(SF)(ZF)X(AF)X(PF)X(CF)

i) Interconexión del Circuito en una PC.

El 8086 se encuentra interconectado en una PC de acuerdo al siguiente diagrama:

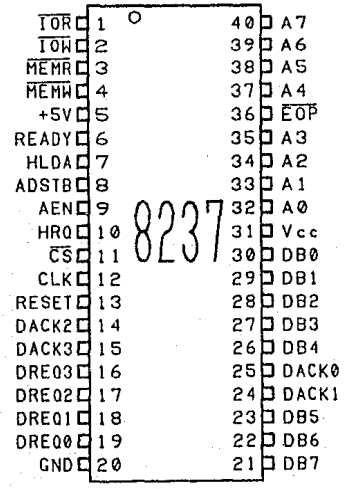


II.2. CONTROLADOR PROGRAMABLE DE ACCESO DIRECTO A MEMORIA 8237.

a) Introducción.

El 8237 es un circuito periférico para el 8086. Este es designado para permitir a dispositivos externos la transferencia directa a memoria de la información. También permite la transferencia de memoria a memoria. El 8237 ofrece una gran variedad de caracteristicas programables para lograr una alta eficiencia en la transferencia de datos y, además se puede reconfigurar el sistema bajo el control de un programa. Cada canal tiene una capacidad de 64KB direccionables y se permite el conteo de palabras. El 8237A-4 y el 8237A-5 trabajan a 4 y 5 Mhz respectivamente, y el estandar 8237A trabaja a 3 Mhz.

b) Diagrama de Terminales.



c) Descripción de Terminales.

Vcc (Power): +5 VCD.
Vss (Ground): Tierra.
CLK (Clock): Esta línea controla las operaciones internas del 8237 y el porcentaje de datos transmitidos.
CS (Chip Select): Es una entrada activa baja usada para seleccionar el 8237 como un dispositivo de I/O durante el ciclo ocioso. Esta señal permite comunicarse con el CPU por el bus de datos.
RESET (Reset): Es una entrada activa alta, la cual pone en ceros a los registros de Estado, Petición y Temporal. Esta también coloca en ceros al primer y al último flip/flop y pone en unos al registro Máscara. Después de un ciclo de reset el dispositivo está en modo ocioso.
READY (Ready): Esta señal de entrada es validada en alto y su función es extender los pulsos de lectura y escritura a la memoria desde el 8237 para ayudar a la memoria o a los dispositivos periféricos de I/O. La señal READY no debe crear transiciones durante el tiempo de especificación setup/hold.
HLDA (Hold Acknowledge): Es una entrada activa alta en la cual el CPU le indica al DMA que le ha cedido el control del bus del sistema.
HRQ (Hold Request): Esta es una señal activa alta de salida. Es usada por el 8237 para solicitar el control del bus del sistema al CPU. Si el correspondiente bit máscara es borrado, la presencia de algún DREQ válido causa que el 8237 solicite la señal HRQ. Después HRQ es activada hasta recibir la señal de HLDA.
IOR (I/O Read): Bidireccional, activa baja y tres estados es esta señal. En el ciclo ocioso, IOR es de entrada usada por el CPU para leer los registros de control. En el ciclo activo, IOR es una señal de salida usada por el 8237 para accesar datos desde un dispositivo periférico durante una transferencia de escritura a DMA.
IOW (I/O Write): Es una señal activa baja, bidireccional y tres estados. En el ciclo ocioso se comporta como una entrada de control usada por el CPU para cargar información al 8237. En el ciclo activo, esta es una señal de control usada por el 8237 para

cargar datos a un periférico durante una transferencia de Lectura a DMA.

MEMR (Memory Read): Es una señal activa baja, tres estados y de salida. Usada para accesar datos desde la localidad de memoria seleccionada durante una lectura a DMA o una transferencia memoria a memoria.

MEMW (Memory Write): Esta línea es activa baja, tres estados y de salida. Usada para escribir datos en la localidad de memoria seleccionada durante una escritura a DMA o una transferencia de memoria a memoria.

AEN (Address Enable): Esta es una señal de salida, activa alta, la cual habilita los 8 bits del latch externo, el cual contiene los 8 bits altos de dirección en el bus de direcciones del sistema. AEN también puede ser usada para deshabilitar otros manejadores de buses del sistema durante la transferencia DMA.

ADSTB (Address Strobe): Es una línea activa alta y de salida la cual es usada para cargar el byte alto de dirección en un latch externo.

EOP (End of Process): Activa baja y bidireccional es esta línea, la cual proporciona información concerniente a la terminación del servicio de DMA. El 8237 permite una señal externa para terminar un ciclo activo DMA. El 8237 también genera un pulso cuando el terminal count (TC) es rechazado por algún canal. Este genera una señal EOP de salida. La recepción de EOP interna o externa causan que el 8237 termine el servicio, acepte la petición y si la autoinicialización es habilitada, se escriben los registros bases y los registros presentes de cada canal. El bit máscara y el bit TC en la palabra de estado serán puestos en 1 desde el presente canal activo por la señal EOP, a menos que el canal sea programado para Autoinicialización. En este caso el bit máscara permanece igual. Durante la transferencia memoria a memoria, EOP será salida cuando el TC desde el canal 1 ocurre. EOP es puesta en alta con un resistor pull-up si este no es usado para prevenir fines erróneos de procesos de entrada.

DREQ0-DREQ3 (DMA Request): Son señales de entrada y cada una representa a un canal individual de recepción asíncrona usadas por los circuitos periféricos para obtener servicio DMA. DREQ0 tiene la más alta prioridad mientras que DREQ3 tiene la mas baja prioridad. Una petición es generada activando la línea DREQ de

algún canal. Luego DACK reconocerá la señal DREQ. La polaridad de DREQ es programable. La señal RESET inicializa estas líneas en activas altas. DREQ deberá mantenerse activa hasta que la correspondiente DACK es activada.

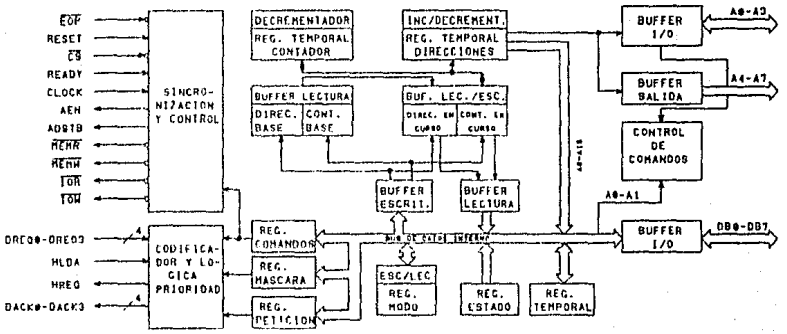
DACK0-DACK3 (DMA Acknowledge): Estas son cuatro líneas de salida, las cuales son usadas para notificar a los periféricos individuales cuando han sido atendidos por un ciclo DMA. El sensado de estas líneas es programable. RESET inicializa estas líneas en activas bajas.

A0-A3 (Address): Son 4 líneas bidireccionales, activas altas y tres estados. En el ciclo ocioso ellas son entradas, y son usadas por el CPU para direccionar los registros a ser cargados o leídos. En el ciclo activo estas líneas son salidas y proveen los 4 bits bajos de las direcciones de salida.

A4-A7 (Address): Estas 4 señales tres estados, de salida y activas altas son las que proveen los 4 bits de dirección. Estas líneas son habilitadas sólo durante el servicio DMA.

DB0-DB7 (Data Bus): Las líneas tres estados del bus de datos son bidireccionales, conectadas al bus de datos del sistema. Las salidas son habilitadas por la condición de un programa durante la lectura I/O para obtener el contenido de los registros de direcciones, estado, temporal o el registro contador de palabra hacia el CPU. Las salidas son deshabilitadas y las entradas son leídas durante un ciclo de escritura I/O cuando el CPU está programando los registros de control del 8237. Durante un ciclo DMA los 8 bits más significativos de la dirección son colocados en el bus de datos para ser cargados en un latch externo por la señal ADSTB. En la transferencia de memoria a memoria, los datos desde la memoria pueden entrar al 8237 por el bus de datos durante el ciclo de lectura desde memoria, luego utilizando el bus de datos como salida se ejecuta el ciclo de escritura a memoria en la nueva localidad de memoria.

d) Diagrama de Bloques Interno.



e) Descripcion del Diagrama de Bloques Interno.

El Diagrama de Bloques del 8237 incluye la lógica mayor y todos los registros internos (los cuales se describirán en la parte de programacion del circuito). La interconexión de las rutas de datos también es mostrada. Lo que no se muestra son todas las señales de control entre los bloques. El 8237 contiene 344 bits de memoria interna en forma de registros. A continuación se proporciona una lista de todos los registros:

NOMBRE	TAMAÑO	No. DE REGISTROS
Dirección Base	16 bits	4
Palabra Contadora Base	16 bits	4
Dirección en Curso	16 bits	4
Palabra Contadora en Curso	16 bits	4
Dirección Temporal	16 bits	1
Palabra Contadora Temporal	16 bits	1
Estado	8 bits	1
Comando	8 bits	1
Temporal	8 bits	1
Modo	6 bits	4
Máscara	4 bits	1
Petición	4 bits	1

El 8237 contiene 3 bloques básicos de control lógico. El bloque de control de tiempo genera señales de control internas y externas para el 8237. El bloque Control de Comandos decide los varios comandos que llevan al 8237 a dar un servicio DMA a una petición. Este también decodifica la palabra Control de Modo usada para seleccionar el tipo de servicio de DMA. El bloque Codificador de Prioridad resuelve las prioridades que se generan entre los canales que piden servicio al DMA simultaneamente.

El bloque Control de Tiempo deriva tiempos internos desde el reloj de entrada. El 8237 tiene como entrada el reloj O2TTL desde el 8284 o el CLK desde el 8085AH o 8084A. Para el 8085AH-2 se maneja una frecuencia de alrededor de 3.9 Mhz, y el 8985 CLK(OUT) no satisface al reloj del 8237A-5. En este caso, un reloj externo se necesitará para manejar el 8237A-5.

f) Función General del Circuito.

El 8237 es designado para operar en dos ciclos mayores. Estos son llamados el ciclo ocioso y el ciclo activo.

Cada ciclo de dispositivo está compuesto de un número dado de estados. El 8237 puede asumir 7 estados separados, cada uno compuesto de un ciclo de reloj. "SI" es el estado inactivo. Este es transmitido cuando el 8237 no ha validado las peticiones pendientes de DMA. Mientras está en estado "SI", el DMA es inactivo, pero quizá en la condición del programa sea programado

por el CPU. El estado "S0" es el primer estado de un servicio DMA. El 8237 ha requerido esperar pero el procesador aún no ha enviado un reconocimiento. El 8237 puede ser programado hasta recibir la señal HLDA del CPU. Un reconocimiento desde el CPU puede empezar .

S1,S2,S3, y S4 son estados de trabajo del servicio de DMA. Si más tiempo es necesitado para completar una transferencia que el disponible con el tiempo normal, el estado de espera "SW" puede ser insertado entre S2 o S3 y S4 por el uso de la señal READY del 8237. Note que los datos transferidos directamente desde un dispositivo I/O a memoria o viceversa, con IOR y MEMW (o MEMR e IOW) empezarán a activarse al mismo tiempo.

Los datos no son leídos dentro o manejados fuera del 8237 en transferencias DMA de I/O a Memoria o Memoria a I/O.

Las transferencias Memoria a Memoria requieren un ciclo "lectura desde" y un ciclo "escritura a memoria" para completar cada transferencia. Los estados, que se asemejan a los estados normales de trabajo, usan dos dígitos para identificarse. Ocho estados son requeridos para una simple transferencia. Los primeros cuatro estados (S11, S12,S13, S14) son usados para el ciclo "lectura desde memoria", y los últimos cuatro estados (S21, S22, S23, S24) para la "escritura a memoria" de la transferencia.

i) Ciclo Ocioso.

Cuando ningún canal requiere el servicio, el 8237 entrará en el ciclo ocioso y ejecutará estados "SI". En este ciclo el 8237 probará todos los ciclos de reloj de las líneas DREQ para determinar si algún canal está requiriendo un servicio DMA. El dispositivo también probará CS, para saber si existe una tentativa por el CPU para escribir o leer los registros del 8237. Cuando CS es baja y HLDA es baja, el 8237 entra en Condición de Programa. El CPU puede ahora establecer cambios o inspeccionar la definición interna de las partes para leerse o escribirse en los registros internos.

Las líneas de dirección A0-A3 son entradas hacia los dispositivos y seleccionan cuales registros serán leídos o escritos. Las líneas IOR y IOW son usadas para seleccionar, leer o escribir. Debido al número o tamaño de los registros internos, un flip-flop es usado para generar y adicionar un bit de dirección.

Este bit es usado para determinar el byte alto y bajo de la dirección de 16 bits y los registros de Palabra Contadora. El flip-flop es inicializado por un RESET. Aparte un comando de Software también puede colocar en cero este flip-flop.

Los comandos de software especial pueden ser ejecutados por el 8237 en la Condición de Programa. Estos comandos son decodificados y colocados en las direcciones correspondientes con el CS y el IOW. Los comandos no pueden hacer uso del bus de datos. Las instrucciones incluyen borrado del primer y último flip-flop y el Borrado Maestro.

ii) Ciclo Activo.

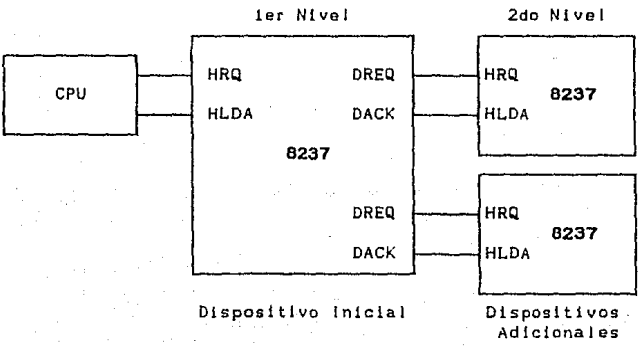
Cuando el 8237 está en el ciclo ocioso y un canal no mascarado requiere servicio DMA, el dispositivo habilita la señal HRQ hacia el CPU y entrará en el ciclo activo. En este ciclo el servicio DMA tomará posesión en uno de los 4 modos:

Modo de Transferencia Simple.- El dispositivo es programado para hacer sólo una transferencia. La Palabra Contadora será decrementada y las direcciones se decrementarán o incrementarán siguiendo cada transferencia. Cuando la Palabra Contadora termina una vuelta desde 0000H hasta FFFFH, la señal TC causará una autoinicialización si los canales han sido programados. La señal DREQ necesitará mantenerse activa hasta que la señal DACK se activa para ser reconocida. Si la señal DREQ se mantiene activa desde el principio hasta el fin de la "transferencia simple", esta se activará de nuevo y recibirá una nueva señal HLDA, entonces, otra "transferencia simple" será ejecutada. En sistemas 8080A, 8085AH, 8086/88 se asegura un ciclo de máquina completo para la ejecución entre transferencia DMA. Detalles de tiempos entre el 8237 y otros protocolos para el control del bus dependerán sobre manera de las características del CPU utilizado.

Modo de Transferencia en Bloque.- El 8237 es activado por DREQ para continuar realizando transferencias durante el servicio hasta un TC, causado porque la "Palabra Contadora" llegó a FFFFH, o una señal externa EOP es encontrada . La señal DREQ necesitará mantenerse activa hasta que DACK empiece a activarse. De nuevo, una Autoinicialización ocurrirá hasta el fin del servicio si el canal ha sido programado por éste.

Modo de Transferencia por Demanda.- El 8237 es programado para continuar creando transferencias hasta que TC, una señal externa EOP es encontrada, o hasta que DREQ se coloca inactiva. Estas transferencias podrán continuar hasta que el dispositivo de I/O haya acabado su capacidad de datos. Después que el dispositivo I/O termine su transferencia, el servicio DMA es reestablecido por la señal DREQ. Durante el tiempo entre servicios, cuando el CPU está disponible para operar, los valores de dirección y palabra contadora son guardados en el 8237 en los registros "Dirección en Curso" y "Palabra Contadora en Curso". Sólo EOP puede causar la autoinicialización hasta el fin del servicio. EOP es generada por TC o por una señal externa.

Modo Cascada.- Este modo es usado para poner en cascada varios 8237, y así expander el sistema. Las señales HRQ y HLDA desde el 8237 adicional son conectadas a las señales DREQ y DACK de un canal del 8237 inicial. Este permite al DMA hacer peticiones de los dispositivos adicionales para propagarse a través de la red de prioridad del dispositivo predecesor. La cadena de prioridad es preservada y los nuevos dispositivos necesitarán esperar para que sea reconocida su petición. Después, el canal cascada del 8237 inicial es usado sólo para la priorización del dispositivo adicional, éste no emitirá alguna dirección o señales de control desde si mismo y podrá entrar en conflicto con las salidas del canal activo con el dispositivo adicionado. El 8237 responderá con el DREQ y DACK pero todas las otras salidas excepto HRQ serán deshabilitadas. La entrada READY es ignorada.



iii) Tipos de Transferencia.

Cada uno de los 3 tipos activos de modos de transferencia pueden llevar tres tipos diferentes de transferencia. Estos son Leer, Escribir y Verificar. La transferencia de Lectura mueve los datos de E/S a un mecanismo de la memoria activando a MEMW e IOR. La transferencia de Escritura mueve los datos de la memoria a un mecanismo de E/S, activando MEMR e IOW. Las transferencias de Verificación son pseudotransferencias. El 8237 opera como una transferencia de Lectura o Escritura generando destinatarios y respondiendo a EOP, etc. Sin embargo, la memoria y todas las lineas de control de E/S permanecen inactivas. La entrada READY es ignorada en el modo de verificación.

Transferencia "Memoria a Memoria".- Para llevar a cabo movimientos de bloques de datos desde un espacio en una dirección de memoria a otra, con un mínimo de esfuerzo y tiempo de programa, el 8237 incluye una transferencia de Memoria a Memoria. Programando un bit en el registro de Comandos se seleccionan los canales 0 y 1 para operar como canales de transferencia de Memoria a Memoria. La transferencia es iniciada por la activación de la señal DREQ desde el canal 0. El 8237 requiere un servicio de DMA de manera normal al CPU. Después de que HLDA es verdadera, el 8237 usa cuatro estados de transferencia en modos de transferencia de Bloque, leyendo datos desde la memoria. El registro de Dirección en Curso del canal 0 es la fuente para las direcciones usadas, y es decrementado o incrementado de una manera normal. La lectura de un byte de datos desde la memoria es guardado en el registro Temporal del 8237. El canal 1 ejecutará con cuatro estados una transferencia de los datos desde el registro Temporal hasta la memoria usando las direcciones en el registro de Dirección en Curso e incrementando o decrementando éste de una manera normal. La Palabra Contadora en Curso del canal 1 es decrementada. Cuando el conteo del canal 1 llega a FFFFH TC es generada, causando una salida EOP que termina el servicio.

El canal 0 puede ser programado para retener las mismas direcciones de todas las transferencias. Estas permitiran una simple palabra para ser escrita a un bloque de memoria.

El 8237 responderá con señales externas EOP durante transferencias Memoria a Memoria. La comparación de datos en un bloque buscado podrá usar esta entrada para terminar el servicio cuando una marca es encontrada. Las operaciones de memoria a memoria pueden ser detectadas como una señal activa AEN sin ninguna salida DACK.

iv) Autoinicialización.

Por medio de la programación de un bit en el registro Modo, un canal puede ser puesto como canal de autoinicialización. Durante el inicio de la autoinicialización, los valores originales de la Dirección en Curso y la Palabra Contadora en curso son automáticamente restaurados desde la Dirección Base y la Palabra Base Contadora de este canal, siguiendo un EOP. Los registros base son cargados simultáneamente con los registros en curso por el microprocesador y quedando sin cambio a través del servicio DMA. El bit máscara no es alterado cuando el canal está en autoinicialización. Después de la autoinicialización el canal está listo para ejecutar otro servicio DMA, sin intervención del CPU, tan pronto es detectada DREQ como válida. En una orden para la autoinicialización en ambos canales en una transferencia memoria a memoria ambas palabras contadoras deberán ser programadas idénticamente. Si se interrumpe externamente, los pulsos de EOP deberán ser ampliados en ambos ciclos.

v) Prioridad.

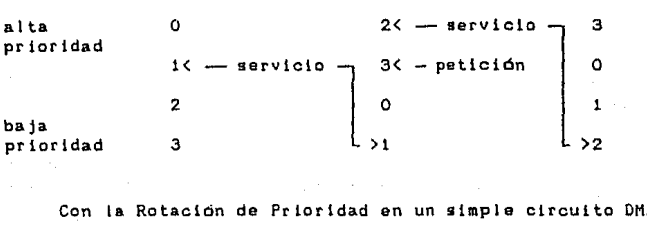
El 8237 tiene dos tipos de prioridad disponible así como Software para seleccionar las opciones. La primera es prioridad fija, la cual coloca los canales en orden de prioridad basado en el orden descendente del número de canal. El canal con la más baja prioridad es el tres, seguido por el dos, uno y el cero, que es el de más alta prioridad.

Después del reconocimiento de alguno de los canales para servicio, los otros canales son impedidos para intervenir con el servicio hasta que éste es completado.

La lectura de un byte de datos desde la memoria es guardado en el registro temporal del 8237.

El segundo es la Rotación de Prioridad. El siguiente canal a ser servido pasará a tener la más baja prioridad rotando a los

demás adecuadamente:



Con la Rotación de Prioridad en un simple circuito DMA, algún dispositivo que requiere servicio se garantiza que será atendido; después, no más de tres servicios de altas prioridades podrán ocurrir. Esto previene que algún canal monopolice el sistema.

vi) Compresión de tiempo.

Pensando lograr igualar gran desempeño donde las características del sistema lo permitan, el 8237 puede comprimir el tiempo de transferencia a dos ciclos de reloj.

El estado S3 es usado para extender el tiempo de acceso de los pulsos de reloj. Removiendo el estado S3, los pulsos de lectura serán igualados a los pulsos de escritura y una transferencia consistirá sólo de un estado S2 para cambiar las direcciones y el estado S4 para ejecutar la lectura escritura. El estado S1 ocurrirá cuando A8-A15 necesite actualizarse (Ver Generación de direcciones).

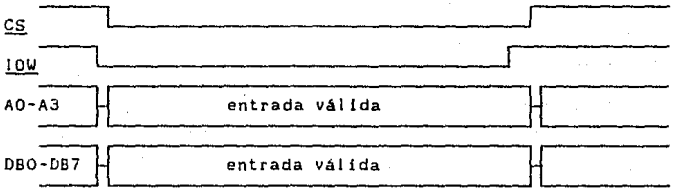
vii) Generación de Direcciones.

Para reducir el conteo, el 8237 multiplexa los ocho bits altos de direcciones con las líneas de datos. El estado S1 es usado para sacar los bits de dirección altos a un latch externo, desde el cual ellos podrán colocarse en el bus de direcciones. La señal ADSTB es usada para cargar estos bits desde las líneas de datos hasta el latch. La señal AEN es usada para habilitar los bits dentro del bus de direcciones a través de la habilitación tres estados. Los bits bajos de dirección son dados por el 8237 directamente. Las líneas A0-A7 serán conectadas al bus de direcciones.

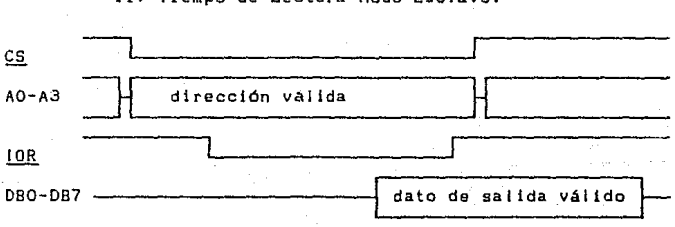
Durante los modos de transferencia de Bloque y Demanda, los cuales incluyen múltiples transferencias, las direcciones generadas serán secuenciales. Para muchas transferencias los datos se mantendrán en el latch de direcciones externo. Estos datos sólo necesitarán cargarse cuando un acarreo o un "tomar prestado" desde A7 a A8 toman lugar en la secuencia normal de direcciones. Para salvar tiempo y rápidas transferencias, el 8237 ejecutará estados S1 sólo cuando se actualice A8-A15 en el latch. En este largo servicio, el estado S1 y ADSTB podrán ocurrir sólo una vez para todas las 256 transferencias y 255 ciclos de reloj para cada 256 transferencias.

g) Diagramas de Tiempos.

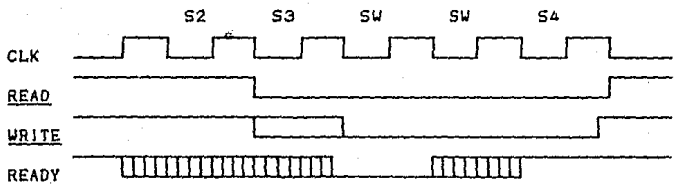
i) Tiempo de Escritura Modo Esclavo:



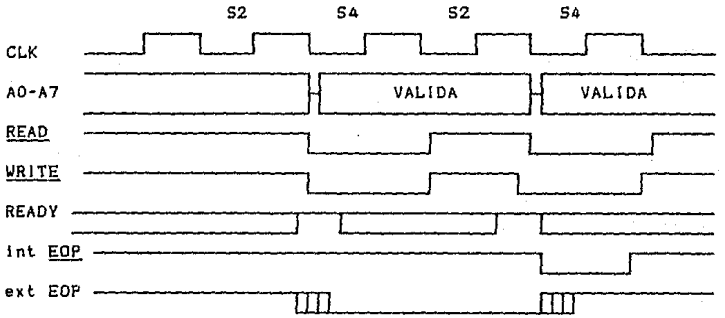
ii) Tiempo de Lectura Modo Esclavo:



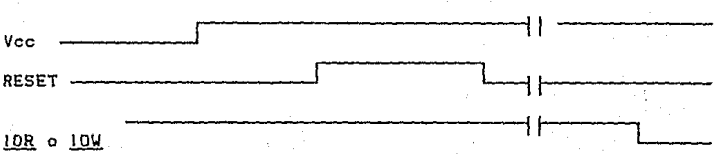
iii) Tiempo Ready:



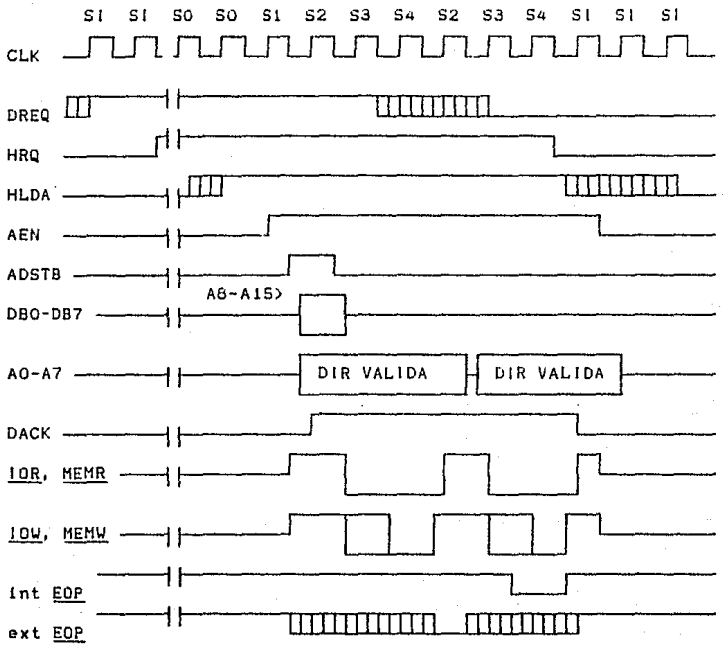
iv) Tiempo de Compresión de Transferencia:



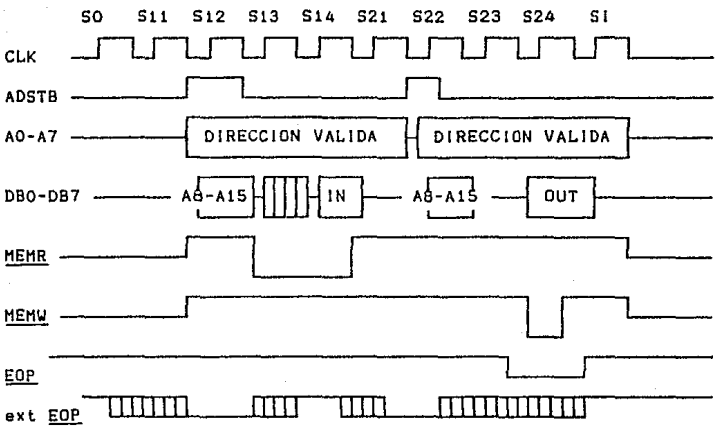
v) Tiempo de RESET:



vi) Tiempo de Transferencia DMA:



vii) Tiempos de Transferencia de Memoria a Memoria:



h) Programación del Circuito.

El 8237 acepta programación desde el microprocesador en cualquier tiempo que HLDA sea inactiva; esta es verdadera si HRQ es activa. La responsabilidad del microprocesador es asegurar que la programación y la señal HLDA sean mutuamente exclusivas. Note que este problema puede suceder si la petición de DMA ocurre, en un canal no mascarado mientras el 8237 es empezado a programar. Por tanto, el microprocesador puede estar reprogramando los dos bytes del registro de dirección del canal uno cuando el canal uno recibe una petición DMA. Si el 8237 es habilitado (bit 2 en el registro de comandos=0) y el canal uno es no mascarado, un servicio DMA ocurrirá, sólo un byte del registro de Direcciones será reprogramado. Esto se puede resolver deshabilitando el 8237 (activando el bit 2 en el registro de Comandos) o mascarando el canal antes de programar alguno de los registros. Una vez que la programación es completada, el 8237 podrá habilitar o deshabilitar las máscaras.

Después de activar el suministro de voltaje es recomendable que todas las localidades internas, especialmente los registros Modo, sean cargados con algún valor válido. Esto permitirá ver si algunos canales no son usados.

1) Descripción de Registros.

Registro de Dirección en Curso.- Cada canal tiene 16 bits en este registro. Este registro retiene el valor de la dirección usado durante una transferencia DMA. La dirección es automáticamente incrementada o decrementada después de cada transferencia y los valores intermedios de las direcciones son guardados es este registro durante la transferencia. Este registro es escrito o leído por el microprocesador en dos bytes sucesivos. Este registro podrá ser reinicializado por una Autoinicialización. La Autoinicialización toma lugar sólo después de ser activada la señal EOP.

Registro Palabra Contadora en Curso.- Cada canal tiene 16 bits en este registro. Este determina el número de transferencias a ser ejecutadas. El actual número de transferencias será uno más que el programado en este registro. Este registro es decrementado después de cada trasferencia. El valor intermedio del conteo de palabras es guardado en este registro durante la transferencia. Cuando el valor en registro llega desde cero hasta FFFFH, un TC será generado. Este registro es cargado o leído por el microprocesador en dos bytes sucesivos en la condición de un programa. Después del fin de servicio DMA, éste también podrá reinicializarse por una Autoinicialización para regresar al valor original. La Autoinicialización ocurre cuando una señal EOP ocurre. Si éste no tiene Autoinicialización, el registro tendrá un FFFFH después de la señal TC.

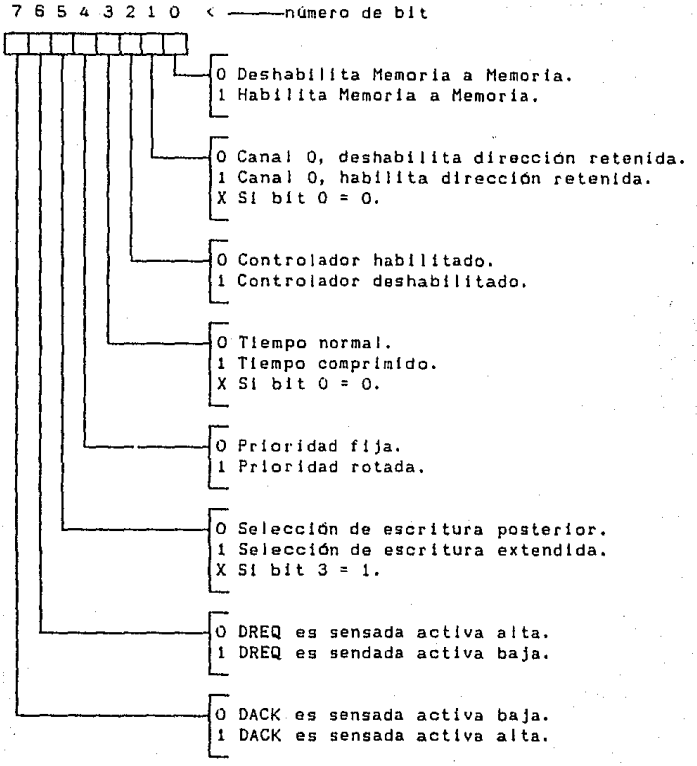
Registros Base de Palabra Contadora y Dirección Base.- Cada canal tiene un par de estos registros. Estos registros de 16 bits guardan el valor original asociado al registro en curso. Durante la Autoinicialización, estos valores son usados para restaurar los registros en curso a sus valores originales. Estos registros son escritos con sus correspondientes registros en curso, byte por

byte en la condición de un programa. Estos registros no pueden ser leídos por el microprocesador.

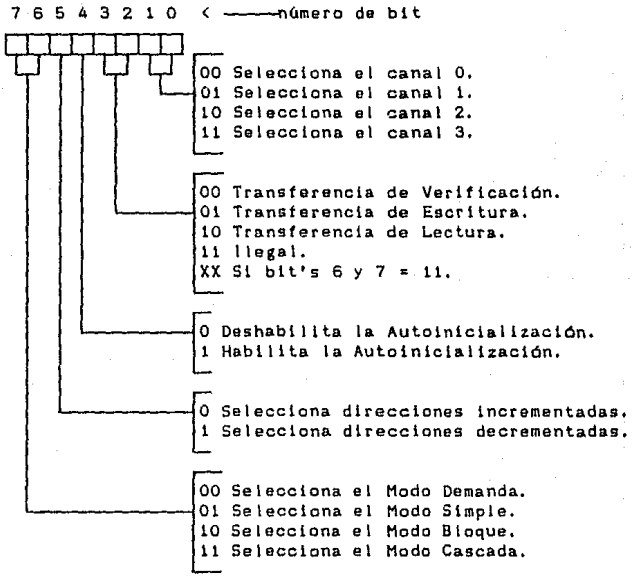
Código de Comandos para Registros Palabra Contadora y Dirección

(CANAL)	REGISTRO	(W-escritura;R-lectura)	_CS	IOR	IOW	A3	A2	A1	A0	F/F	BUS DE DATOS DB0-DB7
0	Dirección Base y en Curso	W	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Dirección en Curso	R	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Palabra Contadora Base y en Curso	W	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Palabra Contadora en Curso	R	0	0	1	0	0	0	1	0	W0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Dirección Base y en Curso	W	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Dirección en Curso	R	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Palabra Contadora Base y en Curso	W	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Palabra Contadora en Curso	R	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Dirección Base y en Curso	W	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Dirección en Curso	R	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Palabra Contadora Base y en Curso	W	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Palabra Contadora en Curso	R	0	0	1	0	1	0	1	0	W0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Dirección Base y en Curso	W	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Dirección en Curso	R	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Palabra Contadora Base y en Curso	W	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Palabra Contadora en Curso	R	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

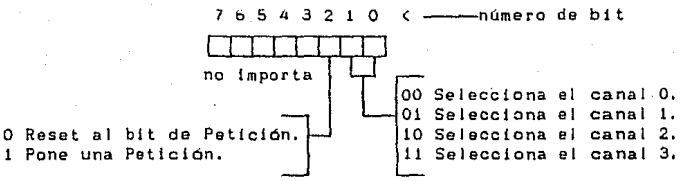
Registro Comando.- Este registro de 8 bits controla la operación del 8237. Este es programado por el microprocesador en la condición de un programa y es borrado por un RESET o una instrucción del Master Clear. La siguiente tabla lista las funciones de los bits de comandos:



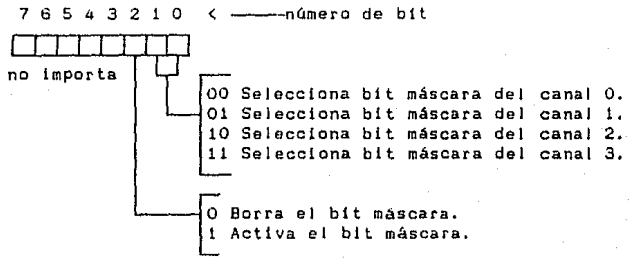
Registro Modo.- Cada canal tiene 16 bits en su registro Modo correspondiente. Cuando el registro es escrito por el microprocesador en la condición de programa, los bits 0 y 1 determinan cuál registro Modo de qué canal es escrito:



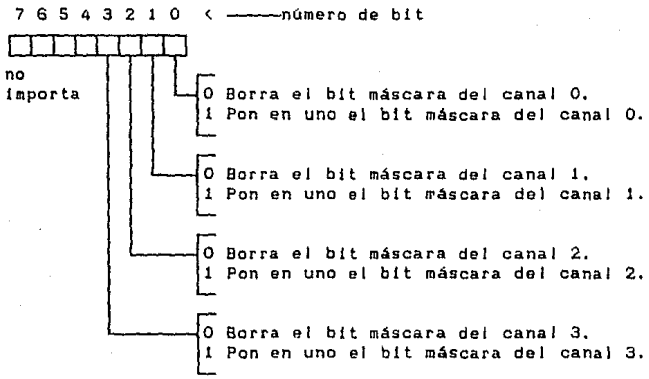
Registro de Petición.- El 8237 puede responder a las peticiones de servicio de DMA, las cuales son iniciadas por Software así como por la señal DREQ. Cada señal tiene un bit de petición asociado con éste en el registro de Petición de 4 bits. Estos son no mascarados y sujetos a prioritización por la red de Prioridad. Cada bit del registro es puesto a uno o cero separadamente bajo control de Software, o es borrado por la generación de la señal TC o la externa EOP. El registro es borrado por un RESET. Para poner a uno o cero un bit, el Software lo cargará desde la Palabra de Datos:



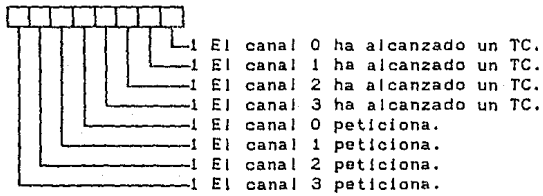
Registro Mascara.- Cada canal tiene asociado un bit máscara, el cual puede ser usado para reconocer o no las peticiones DREQ. Cada uno de los 4 bits de registro Máscara pueden ser activados o borrados bajo control de un programa. El registro es ajustado por un RESET, el cual deshabilita todas las peticiones DMA hasta que una instrucción de borrado del registro Máscara permite que esta ocurra. La instrucción para activar o borrar los bits màscara es similar a la forma usada en el registro de Petición:



Los 4 bits del registro Máscara pueden ser también escritos con un simple comando:



Registro de Estado.- Este registro es disponible para ser leído fuera del 8237 por el microprocesador. Este contiene información acerca del estado de los dispositivos hasta este punto. Esta información incluye cuáles canales han alcanzado un TC y qué canales tienen pendientes peticiones de DMA:



Registro Temporal.- Es usado para retener datos durante transferencias de Memoria a Memoria. Seguido de una transferencia completa, la última palabra movida puede ser leída por el microprocesador en la condición de un programa. El registro Temporal siempre contiene el último byte transferido en la previa operación de Memoria a Memoria, a no ser que sea borrado por una señal RESET.

ii) Definicion de Códigos de Registros.

REGISTRO	OPERACION	SEÑALES						
		CS	IOR	IOW	A3	A2	A1	A0
Comando	Escritura	0	1	0	1	0	0	0
Modo	Escritura	0	1	0	1	0	1	1
Petición	Escritura	0	1	0	1	0	0	1
Máscara	Set/Reset	0	1	0	1	0	1	0
Máscara	Escritura	0	1	0	1	1	1	1
Temporal	Lectura	0	0	1	1	1	0	1
Estado	Lectura	0	0	1	1	0	0	0

iii) Comandos de Software.

Estos son comandos adicionales de Software los cuales pueden ser ejecutados en la condición de un programa. Ellos no dependen de algún bit de especificación en el bus de datos. Los 3 comandos de Software son:

Borrado primer/último flip-flop.- Este comando es ejecutado antes de escribir o leer nuevas direcciones o palabras contadoras en el 8237. Este inicializa el flip-flop para conocer el estado y así los subsecuentes accesos a los registros por el microprocesador serán direccionados por bytes altos y bajos en la secuencia correcta.

Borrado Maestro.- Esta instrucción de Software tiene algunos efectos como el Reset del Hardware. Los registros Comando, Estado, Petición y Temporal así como el primer/último flip-flop son borrados y el registro máscara es activado. El 8237 entrará en el ciclo ocioso.

Borrado del Registro Mascara.- Este comando borra los bits máscara de los cuatro canales, habilitando estos para aceptar peticiones DMA.

CODIGOS DE COMANDOS DE SOFTWARE						
SEÑALES						OPERACION
A3	A2	A1	A0	IOR	IOW	
1	0	0	0	0	1	Lee el Registro de Estado
1	0	0	0	1	0	Escribe el Registro de Comandos
1	0	0	1	0	1	Ilegal
1	0	0	1	1	0	Escribe el Registro de Petición
1	0	1	0	0	1	Ilegal
1	0	1	0	1	0	Escritura a un bit del Registro Máscara
1	0	1	1	0	1	Ilegal
1	0	1	1	1	0	Escritura al Registro Modo
1	1	0	0	0	1	Ilegal
1	1	0	0	1	0	Borra byte apuntado por el flip-flop
1	1	0	1	0	1	Lee el Registro Temporal
1	1	0	1	1	0	Borrado Maestro
1	1	1	0	0	1	Ilegal
1	1	1	0	1	0	Borrado del Registro Máscara
1	1	1	1	0	1	Ilegal
1	1	1	1	1	0	Escribe a todos los bits del R. Máscara

i) Interconexión del Circuito en una PC.

i) Mapa de Puertos para el DMA.

RANGO HEXADECIMAL	9	8	7	6	5	4	3	2	1	0	DISPOSITIVO
00-0F	0	0	0	0	0	0	A3	A2	A1	A0	Registros R/W DMA
80-83	0	0	1	0	0	X	X	X	A1	A0	Registros Página DMA

ii) Direcciones de los registros de lectura escritura.

El 8237 posee una serie de registros de lectura/escritura, implementados en una PC como puertos I/O en el rango 000H a 00FH, divididos en dos grupos. Las direcciones 000H a 0007H son los registros de escritura que contendrán para cada canal la dirección de memoria inicial, la dirección actual para la siguiente transrerencia, el contador base y contador actual de bytes a transferir. El otro grupo comprendido entre las direcciones 0008H a 000FH, contiene entre otros los registros de control y estado que definen la operación de cada canal:

NOMBRE	TAMAÑO	No.DE REGISTROS	dir. I/O
Dirección en Curso	16 bits	4	0000H
Palabra Contadora en Curso	16 bits	4	a
Dirección Base	16 bits	4	.
Palabra Contadora Base	16 bits	4	0007H
Dirección Temporal	16 bits	1	
Contador Temporal	16 bits	1	
Temporal	8 bits	1	
Estado	8 bits	1	0008H
Comando	8 bits	1	0008H
Petición	4 bits	1	0009H
Máscara (simple)	4 bits	1	000AH
Modo	6 bits	4	000BH
Máscara	4 bits	1	000FH

iii) Comandos de Software.

Los siguientes comandos, independientes de los datos, son ejecutados durante el programa de inicialización para facilitar la escritura o lectura de los registros de 16 bits, tal como se expone a continuación.

Flip-Flop Puntero.- Los registros de dirección y conteo son de 16 bits, sin embargo, sólo disponemos de un puerto de 8 bits. La escritura de cualquier dato en el puerto 000CH ordena el cambio de un Flip-flop interno que apunta a los bits de bajo nivel de los registros de 16 bits. La siguiente operación de escritura o lectura cambia el Flip-flop apuntando a los 8 bits de mayor nivel. Esta técnica en definitiva efectúa el intercambio de información en dos pasos.

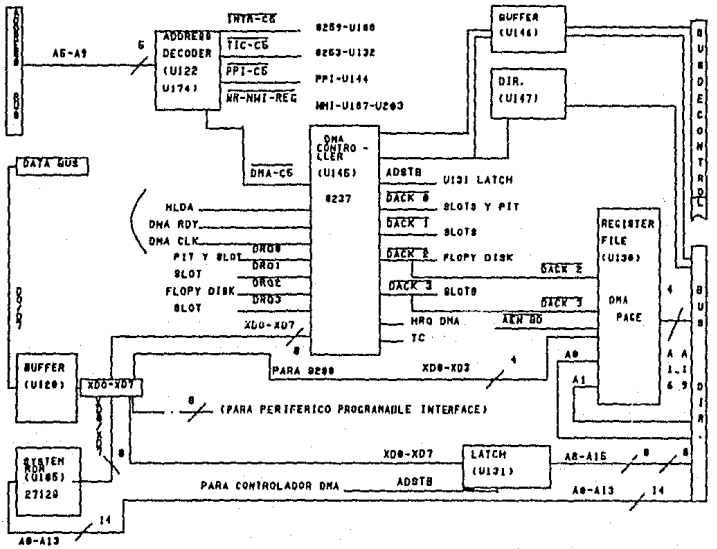
Borrado Maestro.- Podemos realizar una función de limpieza del controlador escribiendo en la dirección puerto I/O 000DH produciendo el mismo efecto que un RESET. El controlador entra en un estado inactivo, requiriendo una inicialización después de la orden del borrado maestro.

Registro Página DMA.- Los registros página de cuatro bits, accesibles en una PC, se encuentran en las direcciones siguientes:

CANAL DMA	DIRECCION PUERTO I/O
0	80H
1	83H
2	81H
3	82H

las cuales permiten formar direcciones de 20 bits, accediendo a un espacio de memoria de 1 MB. Se supera de esta manera la limitación de 64 KB que soporta el DMA. En un ciclo DMA, el contenido del registro página correspondiente es inyectado en el bus de direcciones como bits de mayor nivel.

La interconexión física del 8237 en una PC se muestra a continuación:



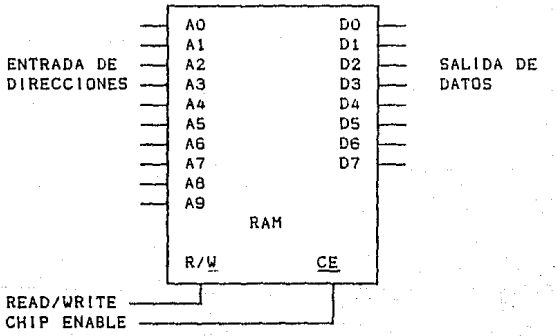
II.3 MEMORIAS RAM Y ROM.

Memorias RAM Estaticas y Dinamicas.

En la actualidad existe un gran número de memorias empleadas, de tal forma que para cada marca de PC puede decirse que emplea una memoria de marca y caracteristicas distintas. Es por eso que desarrollar el tema para alguna memoria sería sólo un caso especial o aislado.

RAM (random access memory o memoria de lectura-escritura de acceso aleatorio). Una memoria RAM almacena palabras en forma binaria. La RAM estática esencialmente es una matriz de flip-flops, por lo tanto, se puede escribir una nueva palabra de datos en una localidad RAM en cualquier tiempo, por medio de la aplicación de la palabra a las entradas de datos del flip-flop y modificando las condiciones del reloj. La palabra de datos guardada permanecerá en las salidas flip-flop en tanto haya energia en el sistema. Este tipo de memoria es volátil debido a que los datos se pierden cuando no hay energia, es decir cuando está desconectada.

La figura muestra el símbolo esquemático de una RAM común.



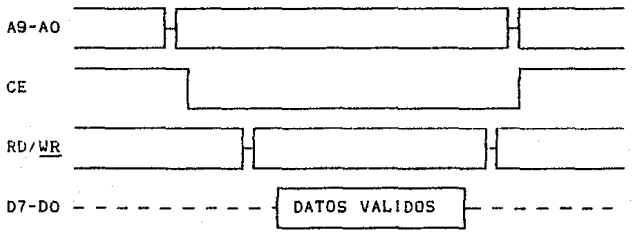
Esta tiene 10 líneas de direccionamiento A0-A9, de modo que guarda 1024 palabras binarias. Las 8 líneas de datos indican que la RAM guarda palabras de 8 bits. Cuando se está leyendo una palabra en la RAM estas líneas funcionan como salida, cuando se está escribiendo una palabra en la RAM estas líneas funcionan como entradas. La entrada CE (chip-enable) es usada para habilitar el dispositivo para una lectura o para una escritura.

La entrada R/W será colocada en ALTO si se desea leer la memoria o puesta en BAJO si se quiere escribir una palabra en la memoria. Para escribir en la RAM se aplica la dirección deseada en las entradas de direccionamiento, se coloca la entrada CE en BAJO para encender el dispositivo, y colocamos la entrada R/W en BAJO para indicarle a la RAM lo que se quiere leer de ella. Para una operación de lectura los buffers de salida en las líneas de datos deberán ser habilitados y la palabra de datos direccionada estará presente en las salidas.

En las RAMs dinámicas (DRAMs), los ceros y unos binarios son guardados como una carga o no carga eléctrica en un capacitor pequeño. Dado que estos capacitores ocupan mucho menos espacio que el que ocupa un flip-flop en un circuito, una RAM dinámica puede almacenar mucho más bits que una RAM estática del mismo tamaño.

La desventaja de las RAM dinámicas es que la carga se pierde de los capacitores pequeños. El estado lógico guardado en cada capacitor debe ser refrescado cada 2 milisegundos o más. Un dispositivo llamado controlador de refresco para RAM dinámicas puede ser usado para refrescar un gran número de estas RAMs en un sistema.

El diagrama de tiempos general de acceso a una RAM se muestra a continuación.

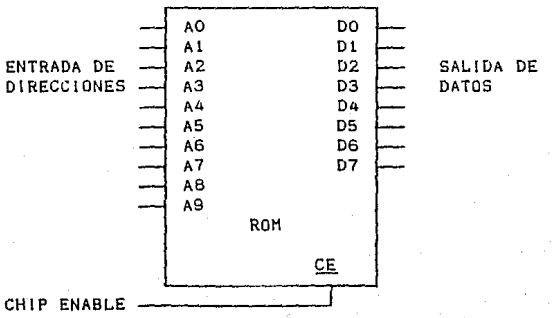


Originalmente, la PC contaba con memoria RAM de 256 KB x 9 bits. Acomodados en forma matricial, contando para esto con cuatro bancos (renglones) y 9 columnas, cada circuito era de 64K x 1 bit.

Memoria ROM.

El término ROM significa Read-Only Memory, memoria de sólo lectura.

Hay varios tipos de ROM que pueden ser escritas, leídas, borradas, y escritas con nuevos datos, pero la principal característica de las ROMs es que son no volátiles. Esto significa que la información almacenada en ella no se pierde cuando se le deja de suministrar voltaje. Las salidas de datos de la ROM están representadas por las terminales D0-D7, esta ROM almacena palabras de datos de 8 bits. Las salidas de datos son salidas tres estados.



Esto significa que cada salida puede ser un estado lógico bajo, un estado lógico alto o una alta impedancia, es decir en estado indefinido. En el estado de alta impedancia una salida es esencialmente desconectada de cualquier cosa que esté conectada a ésta. Si la entrada CE de la ROM no se dá, todas las salidas de la ROM estarán en el estado de alta impedancia, si la señal de CE se dá, el dispositivo estará preparado y la salida del buffer estará

habilitado. Las salidas estarán en un estado lógico normal bajo o alto.

Cada palabra binaria almacenada en la ROM tiene asociado un número que la identificará, llamada dirección. Si se quiere obtener una palabra en particular a través de las salidas de la ROM, se tienen que hacer dos cosas, se tiene que colocar la dirección de la palabra en las entradas de dirección, A0-A7 de tal forma que cuando se dé la señal de CE, exista en las salidas la información correspondiente a la palabra direccionada. El número de palabras binarias almacenadas en la ROM pueden determinarse por el números de entradas de dirección. El número de palabras es 2 elevado a la N , donde N es el número de lineas de dirección.

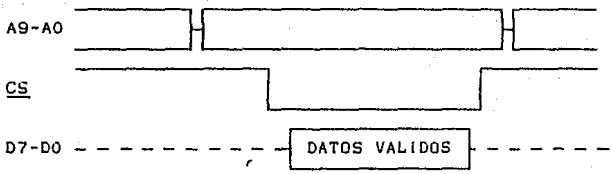
Los diferentes tipos de ROM son las siguientes:

ROM con Máscara Programada.- Programada durante su fabricación. No puede ser alterada.

EPROM.- Eléctricamente programable por el usuario, y borrada por medio de luz ultra-violeta.

EEPROM.- Electricamente programable por el usuario, y en vez de borrarse con luz ultra-violeta su información es borrada por medio de señales eléctricas.

El diagrama general de acceso a una ROM es el siguiente:



Originalmente en la PC se contaba con una EPROM de 64K x 8 bits, en la cual residia el BIOS. En este se encuentran varias rutinas para examinar los dispositivos de entrada y salida, este tipo de pruebas a los dispositivos proveia en muchos casos a los

técnicos para localizar las anomalías en las microcomputadoras.

Se tenía también una memoria ROM en la que se almacenaba el intérprete del lenguaje BASIC, que se cargaba automáticamente si no se encontraba nada en el drive A.

Para la expansión de memoria se tienen dos métodos. En el primero, los fabricantes previeron que las PCs requerirían de más memoria, es por eso que dejaron espacios vacíos para poder insertar bancos de memoria. En el segundo método se hace uso de los slots para expansión, con lo que se puede conseguir hasta un megabyte. Cuando se realiza cualquier tipo de expansión se debe reconfigurar el circuito por medio de los micro-switchs.

El mapa de memoria para una PC es el siguiente:

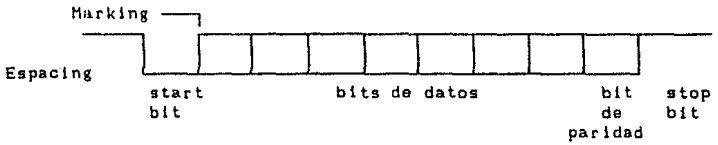
ROM BIOS	MEMORIA ALTA
AREA DE USO MULTIPLE PARA EL USUARIO	
PROGRAMAS RESIDENTES	
DOS	
AREA DE DOS	
AREA DE BIOS	
VECTORES DE INTERRUPCION	MEMORIA BAJA

II.4 CIRCUITO DE COMUNICACION ASINCRONA SERIE 8250.

a) Introducción.

El principal concepto del protocolo serial es que todos los datos e información de control necesarios para transmitir y recibir un carácter de información deben moverse sobre una sola línea de datos, un bit en un tiempo. Este camino se presenta en la siguiente figura:

Formato de los datos seriales asíncronos.



Si se observa la figura de arriba, se puede imaginar que los bits de datos fluyen como si éstos viajaran bajo un solo cable. El ancho de cada bit es determinado por la velocidad de transmisión de datos, que es medida en bits por segundo. Esta velocidad es llamada el baud rate, si los datos son transmitidos bajo la línea de comunicación a 300 bits por segundo, entonces la velocidad de transmisión es de 300 bauds.

Cuando un dato no es transmitido, la línea es puesta a un estado lógico '1' o a un estado marking. Cuando se quiere transmitir el carácter de un dato, el primer bit a ser transmitido es el start bit o bit inicial. El start bit es representado por un estado lógico '0' o un estado spacing, sobre la línea. La duración del start bit es determinado por el baud rate.

Cuando se sabe que se va a recibir un carácter de datos la línea cambia desde un estado marking a un estado espacing, en otras palabras, se presenta el start bit. Los bits de datos que arriba forman el carácter de información son transmitidos en el momento, inmediatamente después del start bit. El número de bits de datos pueden ser 5, 6, 7 u 8, pero cada carácter deberá contener el mismo número de bits de datos en la misma

transmisión.

El número de bits de datos no es fijo, ya que el número de bits que forma un caracter de información, cuando se quiere transmitir es menor a ocho en muchas ocasiones. Debido a que no existe un tamaño fijo del dato transmitido, la velocidad de la tranmisión de datos puede ser optimizada. Los bits de datos son trasmitidos en el primer bit menos significativo y son juntados de nuevo sobre el que recibe al final, formando el carácter de datos que fué transmitido.

Un bit de paridad opcional sigue inmediatamente de los bits de datos. El tipo de paridad seleccionada deberá ser consistente a lo largo de la misma transmisión; si la paridad par es seleccionada, entonces, el número de bits lógicos '1', que forman el bit de datos y el bit de paridad deberá ser par; si la paridad impar es seleccionada, entonces el número de bits lógicos deberán ser impar. Como se verá más adelante el bit de paridad permite al receptor detectar ciertos tipos de error de transmisión.

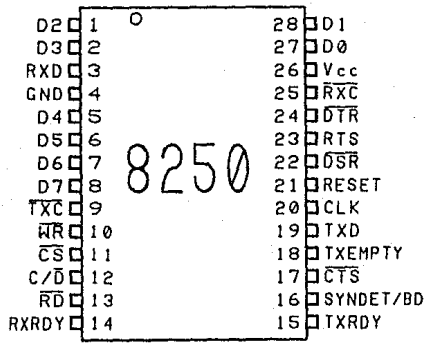
Hay 1, 1.5 ó 2 bits de marca (1 lógico) siguiendo al bit de paridad (o bits de datos si no está presente el bit de paridad). estos bits son llamados los bits stop. Los bits stop representan el mínimo valor de tiempo que la línea deberá estar en una condición de marca antes que el siguiente star bit pueda aparecer. El número de stop bits deberá ser consistente a lo largo de la misma transmisión, si hay otros caracteres a ser transmitidos, entonces inmediatamente seguirán los stop bits con sus asociados start bit.

Si otro carácter no está inmediatamente disponible para ser transmitido, entonces la línea permanecerá en una condición marking hasta que exista otro carácter para ser transmitido.

Porque los caracteres de datos pueden principiar y parar en cualquier tiempo, la comunicación del protocolo discutido en este momento es un protocolo asíconc; después que los stop bits del último carácter son recibidos, el receptor no es inicializado para aceptar más datos, hasta que un start bit llegue, hasta este momento, la línea permanecerá en una condición marking, esto es diferente desde la comunicación de un protocolo serial síncrono, en que los caracteres de datos son siempre transmitidos sobre la línea de comunicación; un protocolo asíncrono, tiene que sincronizar con el transmisor cuando el start bit de un nuevo caracter es recibido; entonces el baud rate del transmisor y el

receptor son actualizados para ser el mismo, el receptor está listo para arrancar los bits de start, paridad y stop, desde la línea como una función del tiempo; el inicio del start bit es usado como el sincronizamiento del evento. Si hay una diferencia pequeña entre los relojes del transmisor y receptor, un error no puede ser posible, porque el receptor y el tranmisor son resincronizados al inicio de cada carácter; el transmisor y el receptor deberán también ser actualizados por el mismo número de bits de datos, el mismo tipo de paridad, y el mismo número de stops bits por la recepción de datos a ser producidos.

b) Diagrama de Terminales.



c) Descripción de Terminales.

D7-D0 (Data Bus). Líneas bidireccionales mediante las cuales el 8250 recibe y envía datos al CPU.

C/D (Control/Data). Línea que indica al 8250 si el siguiente byte es dato a transmitir o un byte de control.

RD (Read). Línea de entrada que indica operación de lectura de un dato desde el 8250 al CPU.

WR (Write). Línea de entrada al 8250 que le indica escritura de un dato o comando.

CS (Chip Select). Linea de entrada al 8250 que indica habilitación de este circuito, para realizar una lectura o escritura en el.

CLK (Clock). Terminal de entrada que proporciona la señal de reloj a emplearse en las transmisiones, generalmente conectada a un dispositivo TTL.

RESET. Señal que coloca al 8250 en estado inhabilitado, por lo que el CPU debe de inicializarlo. Esta terminal se conecta a la señal RESET del sistema.

TXC. Terminal de salida mediante la cual se envía el reloj de la transmisión.

TXD. Terminal de salida mediante la cual se envían los datos de la transmisión.

RXC. Terminal de entrada en la que se recibe el reloj de la transmisión desde un dispositivo externo al sistema.

RXD. Terminal de entrada en la cual se recibe el dato de la transmisión.

RXRDY. Terminal de salida que indica "receptor preparado", esta señal indica al 8086 que puede leer un dato desde el 8250.

TXRDY. Terminal de salida que indica "transmisor preparado", con lo cual el 8086 puede ceder otro dato al 8250 para transmitir.

DSR. Terminal de entrada que indica datos preparados para enviarse al 8250.

DTR. Terminal de salida que indica que el 8250 está listo a recibir datos desde la fuente de transmisión.

SYNDET/BD. Terminal de entrada para indicar la detección de SYNC o de ruptura.

RTS. Terminal de salida empleada por el 8250 para solicitar el envío de datos.

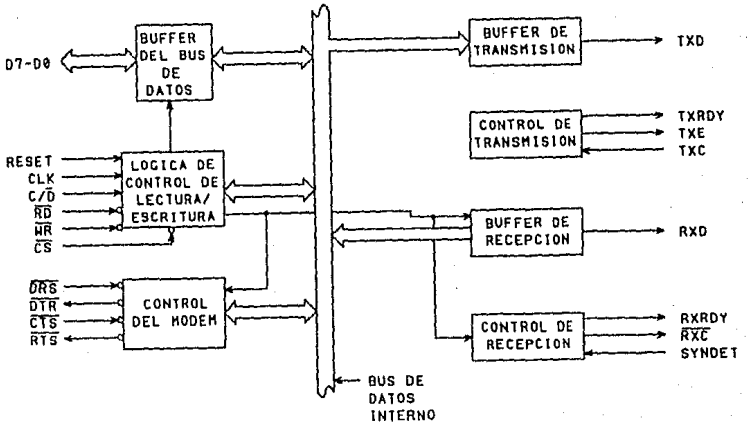
CTS. Terminal de entrada enviada al 8250 para solicitarle el envío de un dato.

TXE. Terminal de salida que indica que el 8250 está libre (vacío) de datos.

Vcc. Alimentación de + 5 Volts.

GND. Tierra eléctrica.

d) Diagrama de Bloques Interno.



e) Descripción del Diagrama de Bloques Interno.

BUFFER DEL BUS DE DATOS.

Este buffer permite almacenar tanto los datos a transmitir como los datos recibidos en una transmisión.

LOGICA DE CONTROL DE LECTURA/ESCRITURA.

Esta lógica es la que permite al 8250 decodificar los bytes de control y realizar las operaciones de lectura/escritura que se indican en estos bytes.

CONTROL DEL MODEM.

Este bloque es el encargado de generar las señales de control necesarias para el control de un modem.

BUFFER DE TRANSMISION.

En este buffer se retienen el carácter recibido en el 8250 desde el CPU, para enviarlo en forma serial en una trasnmisión.

CONTROL DE TRANSMISION.

Este bloque genera las señales de control necesarias para la transmisión de los datos.

BUFFER DE RECEPCION.

En este buffer se colocan los bits que se van recibiendo en forma serial, para conformar el byte que será enviado el CPU.

CONTROL DE RECEPCION.

Este bloque permite emplear las señales recibidas desde el dispositivo transmisor y con ello obtener el dato que se está recibiendo con información adicional (bits de stop, inicio y paridad).

f) Función General del Circuito.

Parecerá complejo para un programa para transmitir y recibir caracteres de datos en una comunicación asincrona serial con el medio ambiente. Afortunadamente existe un microprocesador llamado Universal Asynchronous Receiver Transmitter, o UART, que es usado para ejecutar conversiones de protocolo serial. El UART en la PC es el 8250, elemento de comunicación asincrono. EL UART nos permite transmitir y recibir datos serialmente, sin tener que ejecutar cualquier conversión paralelo a serie o serie a paralelo. Antes que el UART pueda ser usado, éste deberá primero recibir (vía la instrucción OUT del 8086) el baud rate, el número de bits de datos, el tipo de paridad y el número de stops bits. Una vez hecho esto, el UART tendrá que ser programado con las características del protocolo serial, éste puede ser usado para transmitir y recibir datos en serie.

Cuando necesitamos transmitir un caracter de dato se checa el status del UART (por medio del 8088, con una instrucción IN) para observar si el registro transmisor está vacío, si es así podemos enviar un byte de dato al UART, y este byte es destinado en el registro transmisor, cuando el UART tiene completada la transmisión del carácter previo o este generalmente no está transmitiendo cualquier dato (las líneas de salida están a un '1' lógico), el contenido del registro transmisor es destinado al

registro de cambio de transmisión (transmitter shift register). El registro transmisor entonces está disponible para otra salida de un carácter desde la computadora. Sólo los bits de bajo orden del byte en el registro transmisor son usados si el protocolo llamado es menor a los ocho bits de datos. El UART agrega los bits apropiados de start, paridad y stop para el carácter en el registro de cambio de transmision y entonces envía los string enteros de bits de salida sobre la línea de comunicación serial.

Una vez inicializado el UART lo que se tiene que hacer para realizar la salida de un byte de datos a través del UART, es enviárselo y el carácter es automáticamente transmitido con todas las características apropiadas. Mientras la transmisión es llevada a su destino, la computadora puede hacer otra tarea, periódicamente checar para ver si el registro transmisor está listo para aceptar otro byte de salida. El UART destina cualquier entrada serial recibida dentro del "receiver shift register", después que el número apropiado de stop bits son recibidos y habiendo checado errores; el carácter es destinado en el "receiver data register", si menos de ocho bits son usados, entonces los bits de bajo orden del registro receptor de datos son validos. El UART entonces actualiza estos status para mostrar que el receptor de datos está listo. La computadora sabe que puede leer otro byte de datos desde el UART cuando el status del receptor de datos es correcto. Después que los datos son leídos por la computadora, el UART no muestra otra vez listo su receptor de datos, hasta que éste tiene destinado otro carácter dentro del registro receptor de datos. El UART determina si existe algún error en el caracter de entrada, cuando el status del UART es checado por el receptor de datos listo, si el dato está listo, el status reflejará cualquiera de las posibles condiciones de error de entrada, si el UART debe destinar un segundo carácter a través del registro receptor de datos, antes de que el primer carácter sea leído por la computadora, entonces resulta un error de corrida (overrun). Este error puede ocurrir si la computadora no está monitoreando el status del receptor, si el protocolo en el momento usado requiere de siete bits de datos, paridad y un stop bit que dividiendo el baud rate por diez puede calcular el máximo número de caracteres por segundo que el UART necesitará para destinarlos al registro de datos receptor

g) Programación del Circuito.

La siguiente tabla contiene la información necesaria para implementar un simple programa de comunicación serial:

via de invocar	Requerimientos de entrada	Registros Alterados	Salidas/ Resultados
INT 14H	AH=0, BX=0 AL=Inicialización de parámetros.	AX	Inicializa el puerto serial con parámetros especificados en AL. Retorna el estatus serial en AX.
	AH=1, BX=0 AL=Carácter	AH	Trasmite el carácter del dato especificado en el registro AL. Retorna el status en AH, excepto el bit 7 de AH es puesto, si el carácter no pudo ser transmitido
	AH=2, BX=0	AX	Espera el carácter a ser recibido y éste lo regresa en el registro AL. Retorna el status en AH. excepto los bits 5 y 6 de AH que son siempre cero.
	AH=3, BX=0	AX	Retorna el status en AX. AH contendrá la comunicación de las lineas de status y AL contendrá el status del modem.

Todas las llamadas al BIOS por parte del puerto serial I/O usan la INT 14H, de entrada, AH es usado para diferenciar las diferentes funciones de las llamadas, si más de una adaptación de comunicación es instalada, BX puede ser 0 ó 1 de entrada, para nuestro propósito éste será 0.

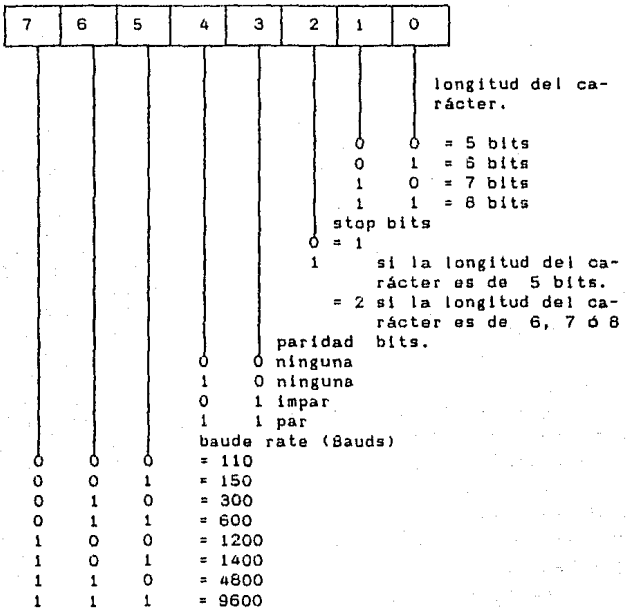
Cuando se necesita inicializar la comunicación del puerto con toda la información necesaria que el UART requiere para funcionar en forma apropiada se usa la INT 14H con AH=0. Esta función de llamada no nos está permitida para ejecutar la comunicación serial con interrupciones.

Cuando se necesita transmitir un carácter se usa la INT 14H con AH=1. Esta llamada activa las señales de control del modem "data terminal ready" y "request to send" y no transmitirá un carácter si no están correctas las señales de control del modem recibidas por "data set ready" y "clear to send". El carácter deberá enviarse al UART cuando el registro transmisor esté vacío, si las anteriores condiciones no son encontradas dentro del presente periodo de tiempo, el tiempo fuera de condición deberá ser puesto como parte del status retornado en AH. Cuando se quiere recibir un carácter se usa la INT 14H con AH=2, esta llamada activa la señal del control del modem "data terminal ready" y no tomará en cuenta el carácter recibido hasta que la señal de control llegue en forma correcta por "data set ready", si la señal no es recibida dentro del presente periodo de tiempo, el BIOS retornará en AH un error de fuera de tiempo. En otro caso el control no es retornado hasta que un carácter es recibido por el UART y destinado a AL.

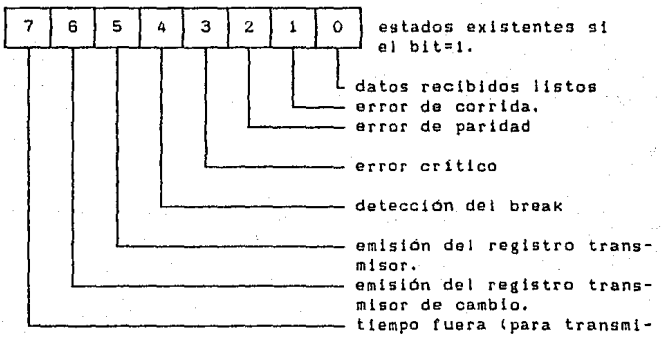
El status de la línea de comunicación y del modem pueden ser obtenidos, usando la INT 14H con AH=3.

Los registros están representados de la siguiente manera:

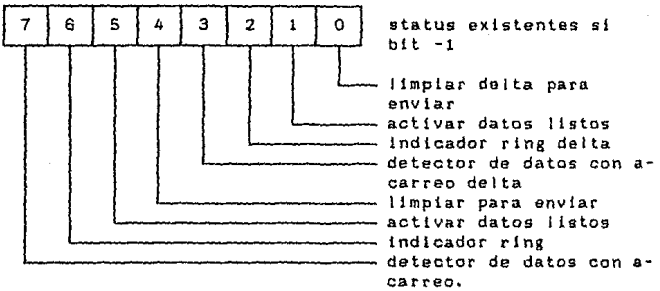
A) AL= Parámetros de inicialización:



B) AH= comunicación con las líneas de status.



c) AL = STATUS DEL MODEM.



Los bits de orden alto del registro de control de línea deberán ser puestos a '1' en orden para acceso de los registros "baud-rate-divisor". Este bit será '0' en todos los otros tiempos.

Cinco de los registros deberán ser programados por el 8086 con una instrucción OUT para inicializar el 8250. estos registros tendrán que ser inicializados, y ellos pueden ser ignorados por el resto del tiempo que el 8250 es usado.

Estos registros son:

Baud-rate Divisor(LSB).

Baud-rate Divisor(MSB).

Line-Control Register.

Modem-control Register.

Interrup-enable Register

Durante el transcurso del programa de comunicación el registro del status de líneas es usado por el 8086 para determinar cuando se transmite o recibe un carácter. Los caracteres a ser transmitidos son salidas para el registro transmisor. Los caracteres recibidos son entradas desde el registro de datos receptor.

Si las señales de control del modem son importantes, el registro de status del modem puede ser usado para monitonear estas señales o cambiarlas, si las señales de control del modem no son usadas, estos registros pueden ser ignorados.

Si el programa de comunicación usa interrupciones con el 8250 el registro de identificación de interrupción es usado para identificar la causa de la interrupción. La acción apropiada puede

ser tomada. Si las interrupciones no son usadas inicialmente, estos registros pueden ser ignorados.

INICIALIZANDO AL 8250.

El 8250 se deberá ser propiamente inicializado, como sabemos hay parámetros que deberán se puestos a un valor, dependiendo de los parámetros del sistema con el que nos estamos comunicando.

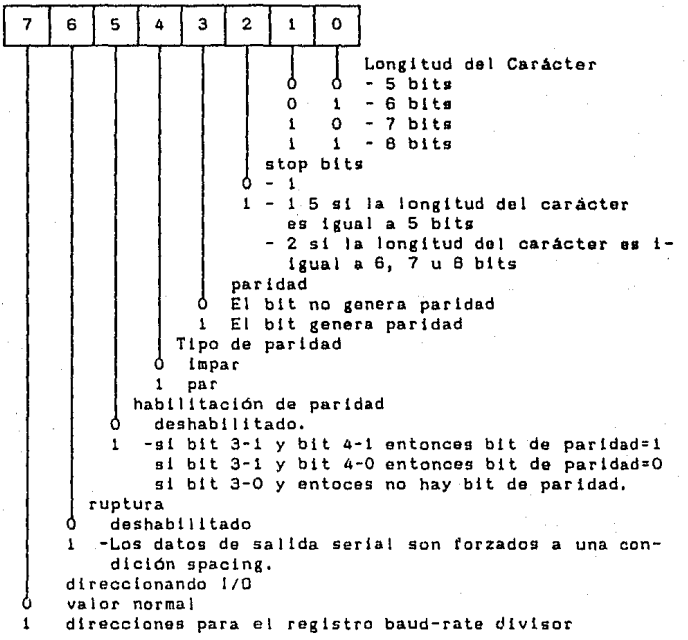
El primer parámetro que deberá ser inicializado es el Baud-rate divisor. Este valor es usado para dividir una salida de frecuencia alta de reloj a través de una señal de reloj que representa el baud rate que se usa para la transmisión y recepción de datos en serie. La siguiente tabla contiene los valores de MSB y LSB del Baud-rate necesitado para obtener un baud-rate conocido:

Baud Rate deseado	valor de los registros baud-rate divisor	
	MSB	LSB
50	09H	00H
75	06H	00H
110	04H	17H
134,5	03H	59H
150	03H	00H
300	01H	80H
600	00H	C0H
1200	00H	60H
1800	00H	40H
2000	00H	3AH
2400	00H	30H
3800	00H	20H
4800	00H	18H
7200	00H	10H
9600	00H	0CH

Para inicializar el baud-rate divisor, se deberá primero poner el bit de orden alto del registro de control de línea a '1' con un OUT para la dirección 3FBH.

Después que el baud-rate divisor es inicializado. El registro de control de línea deberá ser inicializado. Este registro determina la longitud del carácter, el número de stops bits, y el tipo de paridad a ser usada en la transmisión serial. En la siguiente figura se mostrará de que manera se puede inicializar

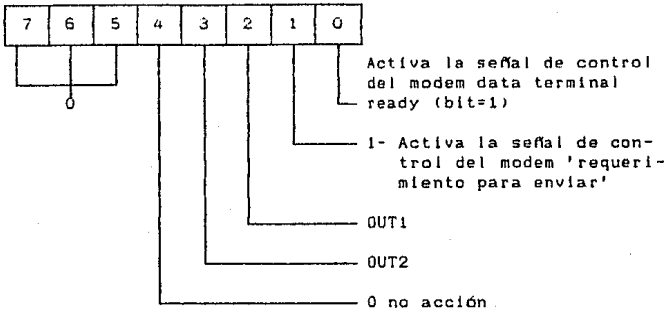
este registro de control:



Note que los cinco bits de orden-bajo de este registro son los mismos, como el parámetro de entrada de la llamada del BIOS usada para inicializar la comunicación del puerto.

Normalmente los tres bits de orden alto de este registro deberán ser puestos a '0' si no se accesa más el registro baud-rate-divisor. El bit de ruptura será puesto a 1, sólo si se dá la condición de ruptura a la salida sobre la línea. El bit de paridad deberá de ser puesto a '1' sólo si se desea que el bit de paridad sea un valor constante.

El siguiente registro a ser inicializado deberá ser el registro del control del modem. La siguiente figura muestra como se inicializa este registro:

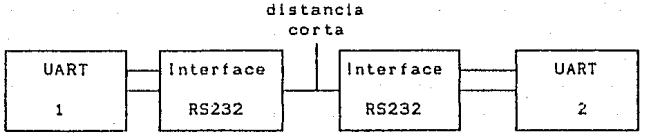


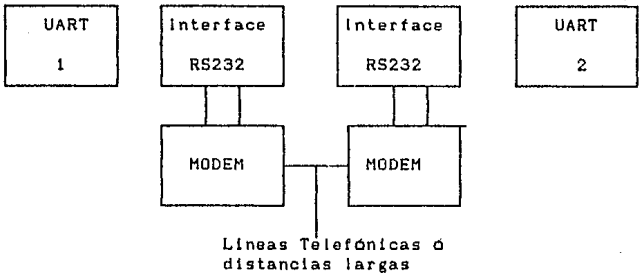
Normalmente este registro será puesto a 03H. Este valor dará la salida correcta para las señales de control del modem 'data terminal ready' y 'request to send'. Si no existe modem en el sistema, éste no podrá dañarse y se detendrá. Si se intenta usar interrupciones el bit OUT2 deberá ser puesto a 1, éste permite la interrupción que el 8250 genera para ser pasada a lo largo del bus del sistema PC, donde eventualmente pasará al circuito 8259 Controlador de Interrupciones.

h) Interconexión del Circuito en una PC.

EL MODEM.

La interface RS232 electrónicamente está conectada al UART para el mundo exterior. Este puede ser directamente conectado a otra interface RS232 sólo a distancias cortas. Si se necesitan dos computadoras que sean capaces de comunicarse la una con la otra sobre distancias largas o sobre líneas telefónicas la salida electrónica de la interfase RS232 no puede ser usada sin un equipo adicional de interface. Como se muestra en la siguiente figura este equipo adicional es llamado Modem.





La salida de la interface RS232 es un voltaje eléctrico que no puede ser destinado directamente sobre las líneas telefónicas.

El Modem convierte los voltajes que representan los unos y ceros lógicos a diferentes tonos que puedan ser transmitidos sobre las líneas telefónicas. El Modem sobre el otro fin de la línea telefónica convierte los tonos de nuevo a unos y ceros eléctricos, que la interface RS232 puede entender. Con este método los unos y ceros lógicos pueden ser transmitidos sobre distancias largas sin ningún problema.

El Modem mas popular y de menos costo puede transmitir datos binarios sobre una línea telefónica ordinaria ya sea a 300 ó 1200 bauds. Los Modems que permiten que ambas computadoras puedan transmitir y recibir datos al mismo tiempo son llamados Modems Full-Duplex. Los Modems que puede adquirir una computadora casual son Full-Duplex. Algunos Modems tienen la capacidad de conmutar con el Modem de una computadora remota y establecer una conexión bajo el control de un programa, estos son llamados Modems Autodial. Los Modems Autoanswer tienen la capacidad para detectar cuando al teléfono al que ellos están conectados es resonante (ringing) y puede responder el teléfono para establecer una conexión con la computadora. Todo el control de información necesariamente al transmitir y recibir un carácter de dato serialmente está contenido en el bit que es enviado sobre una sola línea, la interface RS232 contiene mucho más líneas de control que son usadas en algún tiempo para interactuar con el Modem. Estas señales de control son accesibles bajo un programa de control a través del UART. Las líneas de control extra no representan un problema de cableado porque el Modem está físicamente dentro de la

computadora. Las señales de control del Modem son descritas en la siguiente tabla:

Señales para el Modem desde la Computadora.	
DTR.- Data Terminal Ready	Bajo el control de un programa. Usado para determinar al modem que la computadora está encendida y lista.
RTS.- Request To Send	Bajo el control de un programa. Usado para determinar al modem que la computadora necesita enviar los datos.
Señales desde el Modem para la Computadora.	
DSR.- Data Set Ready	Usado para que la computadora determine que el modem está listo y encendido.
CTS.- Clear To Send	Usado para que la computadora determine si el modem está listo para transmitir datos.
DCD.- Data Carrier Detect	Usado para que la computadora determine que el modem tiene establecida una conección con el modem sobre el otro fin de la línea telefónica.
RI.- Ring Indicator	Usado para que la computadora determine que el teléfono no conectado al modem es (ringinng) resonante.

Dada la comunicación con el programa, el deseo de usar alguna señal de control es dependiendo de la situación. Por ejemplo, si una computadora está siempre conectada a un modem que está siempre en contacto con otro modem, este no está necesariamente para checar cualquiera de las señales de control del modem, si la conección se rompe será obvio para el operador que el problema está el programa actual, a través del monitoreo de las señales de control del modem el programa podrá detectar el rompimiento de una conexión e informar al operador.

Esto pudo ser deseable en muchas situaciones, para este propósito se activan las salidas de "data terminal ready" y

"request to send" sobre la inicialización, olvidándose casi de ello.

LA INTERFACE FISICA.

La comunicación adaptada por la IBM PC provee una interface Standard RS232 para el mundo exterior. El conector que viene fuera del adaptador es un standard macho de 25 terminales tipo "D".

La interface RS232 convierte las señales electricas desde el UART a niveles de voltaje standard que son presentados sobre el conector físico. Un cero lógico que es una condición spacing, está representada por +15 volts. Un 1 lógico que es una condición marking, es representada por -15 volts. La interface también convertirá los voltajes de entrada a las señales binarias eléctricas correctas por el UART. La función de las terminales físicas del conector RS232 se muestra en la siguiente tabla:

Número de terminal	Dirección	Función
2	Salida	Datos Transmitidos.
3	Entrada	Datos Recibidos
4	Salida	Requerimiento para enviar.
5	Entrada	Limpiar para enviar.
6	Entrada	Datos listos
7		Tierra.
8	Entrada	Detecta un dato acarreado.
20	Salida	Terminal de datos listos.
22	Entrada	Indicador Ring

Con la comunicación adaptada e inicalizada se estará listo para medir un voltaje negativo entre las terminales 2 y 7, esto corresponde a la condición marking, en que en las líneas no se están transmitiendo datos. Esta interface puede ser obtenida directamente dentro de un modem con no cambiar la correlación de las terminales.

El 8250 tiene 10 registros que pueden ser accesados por el 8088, más algunos de ellos pueden ser ignorados, cuando el 8250 es inicializado. Una lista de los registros del 8250 y sus direcciones asociadas aparecen en la siguiente tabla:

I/O Direccion de puertos	salidas/entradas	Registro seleccionado
3F8H .	SALIDA	Registro Transmisor
3F8H .	ENTRADA	Registro de Datos Receptor.
3F8H +	SALIDA	Baud-Rate Divisor(LSB)
3F9H +	SALIDA	Baud-Rate Divisor(MSB)
3F9H .	SALIDA	Registro-Habilitador de Interrupción.
3FAH	ENTRADA	Registro Identificador de Interrupción.
3FBH	SALIDA	Registro de Control de Línea.
3FCH	SALIDA	Registro de Control del Modem
3FDH	ENTRADA	Registro del status de la línea.
3FDH	ENTRADA	Registro del status del modem

. El bit 7 del registro de control de linea = 0

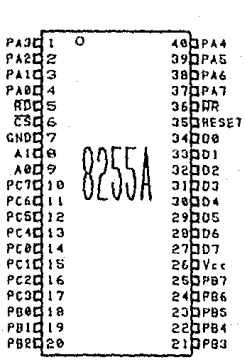
+ El bit 7 del registro de control de linea = 1

11.5. INTERFAZ PERIFERICA PARALELA PROGRAMABLE 8255A/8255A-5.

a) Introduccion.

El 8255A de Intel es un dispositivo de I/O programable de proposito general, diseñado para usarse con los microprocesadores Intel. Tiene 24 terminales de I/O, las cuales pueden programarse individualmente en dos grupos de 12 y usarse en tres modos principales de operacion. En el primer modo (modo 0), cada grupo de 12 terminales de I/O pueden ser programados en juegos de 4 que sean entrada o salida. En el modo 1, cada grupo puede ser programado para tener 8 lineas de entrada o salida. De las 4 terminales restantes, 3 son usadas para apertura o interrupcion, de las señales de control. El tercer modo de operación (modo 2) es un modo de bus bidireccional, el cual usa 8 lineas para un bus bidireccional, y 5 lineas de uno de los otros grupos, para protocolos (handshaking).

b) Diagrama de terminales.



c) Descripción de Terminales.

CS (Chip Select). Un BAJO en esta terminal de entrada habilita la comunicación entre el 8255A y el CPU.

RD (Read). Un BAJO en esta terminal posibilita al 8255A para enviar información de estado o datos, en el bus de datos, al CPU. En esencia, le permite al CPU "leer a partir" del 8255A.

WR (Write). Un BAJO en esta terminal habilita al CPU para escribir palabras de control o datos en el 8255A.

A0 y **A1** (Port Select 0 and Port Select 1). Estas señales de entrada, en conjuncion con las entradas RD y WR, controlan la seleccion de uno de los tres puertos o de los registros de palabra de control. Estas señales están conectadas normalmente a los bits menos significativos del bus de dirección (A0 y A1).

Reset. Un ALTO en esta entrada limpia el registro de control y todos los puertos (A,B,C) son puestos en el modo de entrada.

D0-D7 (Data Bus). Bus de datos bidireccional que interconecta al circuito 8255A con el CPU del sistema.

PA0-PA7, PB0-PB7, PC0-PC7 (Port I/O). Buses bidireccionales mediante los cuales se permite la comunicación de dispositivos periféricos con el bus del sistema.

d) Diagrama de bloques interno.

e) Descripción del Diagrama de Bloques Interno.

El 8255A es una interface periférica programable (PPI) diseñada para usarse en los sistemas de Microcomputadoras INTEL. Su función es la de un componente de I/O de propósito general que sirve de interface entre el equipo periférico y el bus del sistema de la microcomputadora. La configuración de funcionamiento del 8255A está programada por el software del sistema, de manera que no es necesaria una lógica externa para poder estar en interface con dispositivos o estructuras externas.

BUFFER DEL BUS DE DATOS.

Este buffer de 8 bits bidireccional de 3 estados es usado para hacer que el 8255A esté en interface con el bus de datos del sistema. Los datos son transmitidos o recibidos por el buffer como instrucciones de entrada o salida para el CPU. Asimismo, a través del buffer del bus de datos se transfieren palabras de control e información de estado.

LOGICA DE CONTROL Y LECTURA/ESCRITURA.

La función de este bloque es manejar todas las transferencias internas y externas de datos y palabras de control de estado. Acepta entradas de direccionamiento del CPU y buses de control y, a su vez proporciona comandos hacia grupos de control.

GRUPOS DE CONTROL A Y B.

La configuración de funcionamiento de cada puerto está programada por el software del sistema. El CPU envía una palabra de control hacia el 8255A. La palabra de control contiene información tal como "modo", "bit set", etc, la cual inicia el funcionamiento del 8255A.

Cada uno de los bloques de control (Grupo A y Grupo B), aceptan comandos de la lógica de control Lectura/Escritura, reciben palabras de control del bus de datos interno y proporcionan los comandos apropiados a sus puertos asociados.

Grupo de control A - Puerto A y Puerto C superior (C7-C4).
Grupo de control B - Puerto B y Puerto C inferior (C3-C0).

El registro de palabra de control solamente puede ser escrito. No se permite ninguna operación de lectura del registro de palabra de control.

PUERTOS A,B Y C.

El 8255A contiene tres puertos de 8 bits (A,B y C). Todos pueden ser configurados en una gran variedad de características de funcionamiento por el software del sistema, pero cada uno tiene sus propias características especiales para optimizar el poder y la flexibilidad del 8255A.

Puerto A. Es una salida de datos latch/buffer de 8 bits y un latch de entrada de datos de 8 bits.

Puerto B. Es una salida/entrada de datos latch/buffer y un buffer de entrada de datos de 8 bits.

Puerto C. Es una salida de datos latch/buffer de 8 bits y un buffer de entrada de datos de 8 bits (sin latch para entrada). Este puerto puede ser dividido en dos puertos de 4 bits bajo el modo de control. Cada puerto de 4 bits contiene un latch de 4 bits y puede ser usado para las señales de control y entradas de señales de estado en conjunción con los puertos A y B.

f) Función General del Circuito.

SELECCION DE MODO.

Hay tres modos de operación básicos que pueden ser seleccionados por el software del sistema:

Modo 0. Entrada/Salida Básicas.
Modo 1. Entrada/Salida (Strobed).

Modo 2. Bus bidireccional.

Cuando la entrada de reset está en ALTO todos los puertos son puestos en el modo de entrada. Después, cuando la señal reset va a alto, el 8255A puede permanecer en el modo de entrada sin que se requiera inicialización adicional. Durante la ejecución del programa del sistema puede ser seleccionado cualquiera de los otros modos usando una sencilla instrucción de salida. Esto permite que un solo 8255A sirva a una variedad de dispositivos periféricos con una simple rutina de mantenimiento de software. Los modos para los puertos A y B pueden ser definidos separadamente, mientras que el puerto C está dividido en dos porciones según sea requerido para la definición de los puertos A y B. Todos los registros de salida, incluyendo los flip-flops de estado, serán restablecidos (reset) siempre que el modo sea cambiado. Los modos pueden ser combinados de modo que su definición de funciones pueda ser construida para cualquier estructura de I/O.

Esto es, el grupo B puede ser programado en modo 0 para monitorear los cierres de switches sencillos, o para desplegar resultados computacionales, el grupo A puede ser programado en el modo 1 para monitorear un tablero o lectora de cinta en una fase de interrupción/transporte.

En el diseño del 8255A se han tomado en cuenta cosas tales como una disposición de PC y completa flexibilidad de funcionamiento para soportar casi cualquier dispositivo periférico sin lógica externa. Lo que se traduce en un uso óptimo de las terminales disponibles.

Característica Bit Set/Reset sencilla.

Cualquiera de los 8 bits del puerto C pueden ser conectados o desconectados usando una instrucción de salida sencilla. Esta característica reduce los requerimientos de aplicaciones basadas en control.

Cuando el puerto C está siendo usado como estado/control para el puerto A o B, estos bits pueden ser conectados o reconectados por medio del uso de una operación set/reset de bit como si se tratará de puertos de salida.

Funciones de Control de Interrupcion.

Cuando el 8255A está programado para operar en el modo 1 o en el modo 2, se proporcionan señales de control que pueden ser usadas como entradas de requerimiento de instrucción hacia el CPU.

Las señales de requerimiento de interrupción generadas a partir del puerto C, pueden ser inhibidas o habilitadas por medio de la conexión o reconexión del flip-flop asociado INTE, usando la función set/reset de bit del puerto C.

Esta función permite que el programador evite o permita que un dispositivo I/O especifico interrumpa al CPU sin afectar cualquier otro dispositivo en la estructura de interrupción.

Definición del flip-flop INTE:

(BIT-SET)-INTE está encendido - Interrupción habilitada.
(BIT-RESET)-INTE está apagado - Interrupción deshabilitada.

MODOS DE OPERACION.

MODO 0. (Entrada/Salida básica). Esta configuración proporciona operaciones simples de entrada y de salida para cada uno de los tres puertos. No se requiere apertura, pues simplemente los datos son escritos o leídos de un puerto específico.

Definiciones de funcionamiento básico del modo 0:

- Dos puertos de 8 bits y dos de 4 bits.
- Cualquier puerto puede ser entrada o salida.
- Las entradas no son conmutadas.
- En este modo hay 16 configuraciones I/O diferentes posibles.

A		B		GRUPO A			GRUPO B	
D4	D3	D1	D0	PUERTO A	PUERTO C (P. ALTA)	#	PUERTO B	PUERTO C (P. BAJA)
0	0	0	0	SALIDA	SALIDA	0	SALIDA	SALIDA
0	0	0	1	SALIDA	SALIDA	1	SALIDA	ENTRADA
0	0	1	0	SALIDA	SALIDA	2	ENTRADA	SALIDA
0	0	1	1	SALIDA	SALIDA	3	ENTRADA	ENTRADA
0	1	0	0	SALIDA	ENTRADA	4	SALIDA	SALIDA
0	1	0	1	SALIDA	ENTRADA	5	SALIDA	ENTRADA
0	1	1	0	SALIDA	ENTRADA	6	ENTRADA	SALIDA
0	1	1	1	SALIDA	ENTRADA	7	ENTRADA	ENTRADA
1	0	0	0	ENTRADA	SALIDA	8	SALIDA	SALIDA
1	0	0	1	ENTRADA	SALIDA	9	SALIDA	ENTRADA
1	0	1	0	ENTRADA	SALIDA	10	ENTRADA	SALIDA
1	0	1	1	ENTRADA	SALIDA	11	ENTRADA	ENTRADA
1	1	0	0	ENTRADA	ENTRADA	12	SALIDA	SALIDA
1	1	0	1	ENTRADA	ENTRADA	13	SALIDA	ENTRADA
1	1	1	0	ENTRADA	ENTRADA	14	ENTRADA	SALIDA
1	1	1	1	ENTRADA	ENTRADA	15	ENTRADA	ENTRADA

MODO 1. (Entrada/Salida Strobed). Esta configuración proporciona un medio para transferir datos de I/O hacia o de un puerto especifico en conjuncion con señales por paquete. En el modo 1, los puertos A y B usan las lineas en el puerto C para generar o aceptar estas señales de apertura.

Definiciones de funcionamiento básico del modo 1.

- Dos grupos (Grupo A y B)
- Cada grupo contiene un puerto de datos de 8 bits uno de control/datos de 4 bits.
- El puerto de datos de 8 bits puede ser de entrada o de salida. Ambas entradas y salidas están conmutadas.
- El puerto de 4 bits se usa para control y estado del puerto de datos de 8 bits.

Definicion de la señal de control de entrada.

STB (Strobe Input).

Un BAJO en esta entrada carga el dato en el latch de entrada.

IBF (Input Buffer Full FF).

Un ALTO en esta señal indica que los datos han sido cargados en el latch de entrada, en esencia, se conecta (set) un reconocimiento de IBF por medio de la entrada STB siendo BAJO, y es reconectada por el disparo hacia arriba de la señal de entrada RD.

INTR (Interrupt Request).

Un ALTO en esta salida puede ser usado para interrumpir al CPU cuando un dispositivo de entrada está requiriendo servicio.

INTR está conectado cuando STB es un "UNO", IBF es un "UNO" e INTE es un "UNO".

Es reconectado por medio del disparo hacia abajo de la señal de RD. Este procedimiento permite que un dispositivo de entrada pueda requerir servicio del CPU sólo enviando sus datos, por paquete (strobing) hacia el puerto.

Definición de la señal de control de salida.

OBF (Output Buffer Full FF).

La salida OBF se pondrá en BAJO para indicar que la CPU ha escrito datos hacia el puerto especificado. La OBF F/F será conectada por el disparo hacia arriba de la señal de entrada WR y reconectada cuando la entrada ACK sea BAJO.

ACK (Acknowledge Input).

Un bajo en esta entrada informa al 8255A que los datos del puerto A o del puerto B han sido aceptados. En esencia, es una respuesta del dispositivo periférico indicando que han sido recibidos los datos de salida por el CPU.

INTR (Interrupt Request).

Un ALTO en esta salida puede ser usada para interrumpir al CPU cuando un dispositivo de salida ha aceptado datos transmitidos por el CPU. INT está conectado cuando ACK es un "UNO", OBF es un "UNO" e INTE es un "UNO". INTR es apagada (reset) cuando la señal WR se dispara hacia abajo.

MODO 2. (Strobed bidirectional Bus I/O).

Esta configuración proporciona un medio para comunicarse con un dispositivo o estructura periféricos en un bus sencillo de 8 bits tanto para transmitir, como para recibir datos (Bus I/O bidireccional). Las señales de protocolo de apertura son proporcionadas para mantener una disciplina de bus apropiada en una manera similar a la del MODO 1. Las funciones de generación y habilitar/deshabilitar también están disponibles.

Definiciones de funcionamiento básico del modo 2.

- Solamente se usan en el grupo A.
- Puerto de bus bidireccional de 8 bits (puerto A) y puerto de control de 5 bits (puerto C).
- Ambas, las entradas y las salidas están conmutadas.
- El puerto de control de 5 bits (puerto C) es usado para control y estado del puerto del bus bidireccional de 8 bits (puerto A).

Definición de la señal del bus I/O bidireccional.

INTR (Interrupt Request).

Un ALTO en esta salida puede ser usado para interrumpir al CPU en operaciones tanto de entrada como de salida.

Operaciones de salida.

OBF (Output Buffer Full).

La salida OBF se pondrá BAJO para indicar que el CPU ha escrito datos hacia el puerto A.

ACK (Acknowledge).

Un BAJO en esta salida permite que el buffer de salida de 3 estados envíe hacia afuera los datos. Es decir que, el buffer de salida estará en el estado de alta impedancia.

Operaciones de entrada.

STB (Strobe Input).

Un BAJO en esta entrada carga datos en el latch de entrada.

IBF (Input Buffer Full F/F)

Un ALTO en esta salida indica que los datos han sido cargados en el latch de entrada.

Lectura del estado del puerto C.

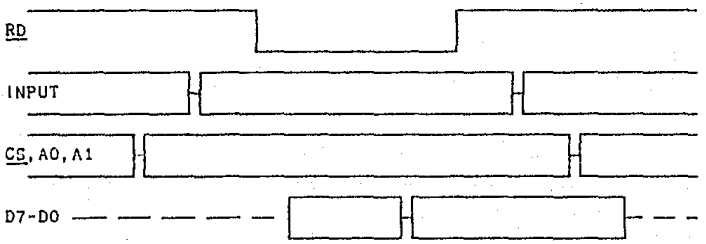
En el modo 0, el puerto C transfiere datos hacia o del dispositivo periférico. Cuando el 8255A está programado para funcionar en los modos 1 y 2, el puerto C genera o acepta señales de protocolo de apertura con el dispositivo periférico.

La lectura del contenido del puerto C permite al programador probar o verificar el estado de cada uno de los dispositivos periféricos y cambiar el flujo del programa según sea conveniente.

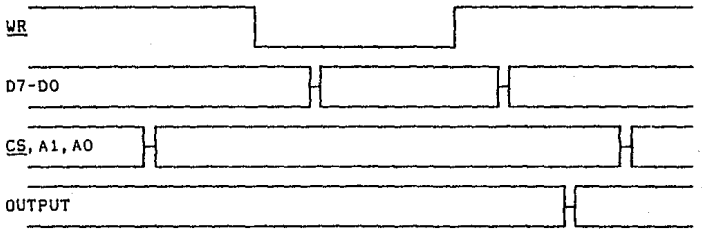
No hay una instrucción especial para leer la información de estado del puerto C. Una operación de lectura normal del puerto C es ejecutada para realizar esta función.

g) Diagramas de tiempos.

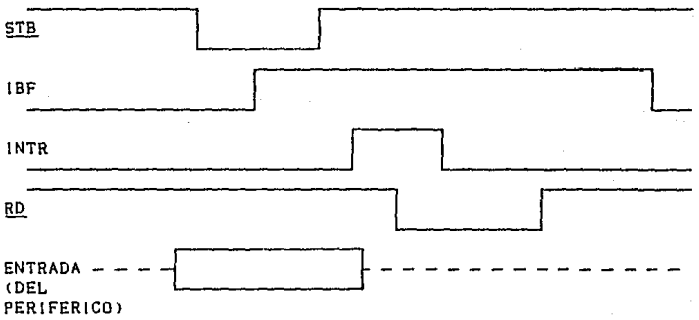
MODO 0. (Entrada Básica)



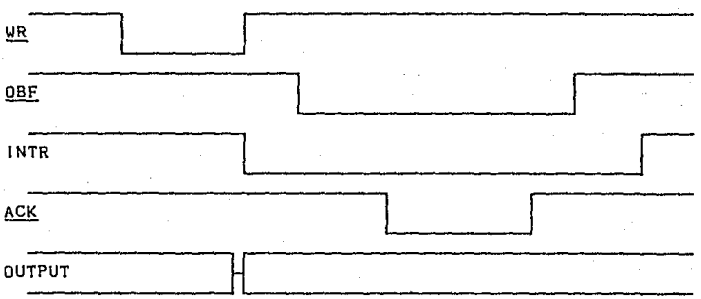
MODO 0. (Salida Básica)



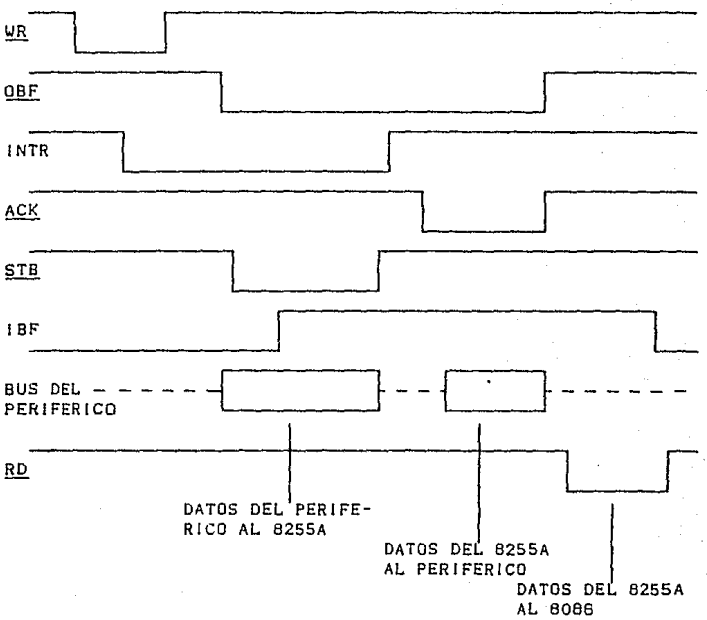
MODO 1. (Strobed Input)



MODO 1. (Strobed Output)

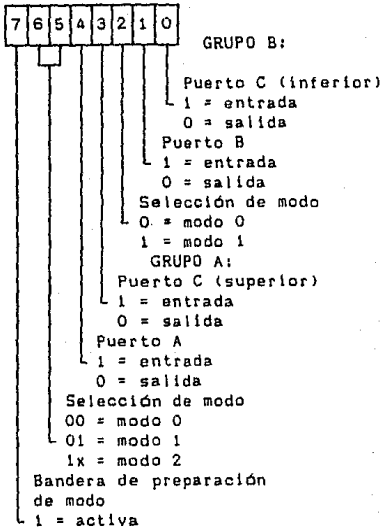


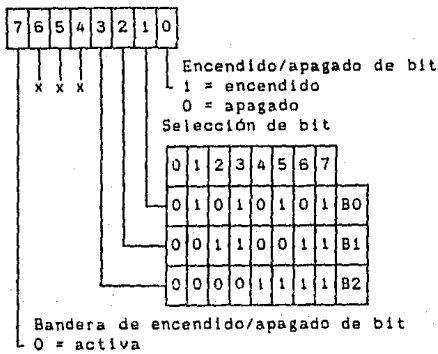
MODO 2. (Bidireccional)



h) Programación del Circuito.

El circuito 8255 se programa con las siguientes palabras de control:





En la figura anterior se muestran los formatos de las dos palabras de control del 8255. El MSb de la palabra de control le indica al circuito cuál palabra de control se está enviando. El formato de la palabra de control de definición de modo se usa para indicarle al dispositivo en que modos se desea operar los puertos,

para la palabra de control de definición de modo se coloca un 1 en el MSb. El formato de la palabra de control de encendido/apagado de bit se utiliza cuando se desea encender o apagar la salida de una terminal del puerto C, o cuando se desea habilitar las señales de salida de interrupción para transferencias de datos con protocolo. El MSb es 0 para esta palabra de control. Ambas palabras de control son enviadas a la dirección del registro de control por el 8255A.

Como es usual, la contrucción de una palabra de control consiste en figurarse lo que se debe colocar en los ocho bits.

i) Interconexión del Circuito en una PC.

El 8255 es un circuito de interface I/O de propósito general que puede ser configurado de muy diversas maneras. Es usado en la tarjeta del sistema para soportar una variedad de dispositivos y señales. Estos incluyen el teclado, el altavoz, los switches de configuración y algunas otras señales.

El circuito contiene tres puertos, llamados PA, PB y PC, los cuales son mapeados en las direcciones de I/O 60H, 61H y 62H, respectivamente. Adicionalmente, existe un registro de comando de un byte en el circuito, accesado mediante el puerto 63H. Durante el encendido, el BIOS inicializa este circuito enviando un valor de 99H al registro de comandos. Esto configura al 8255 de tal modo que PA y PC son considerados puertos de entrada y PB es considerado un puerto de salida.

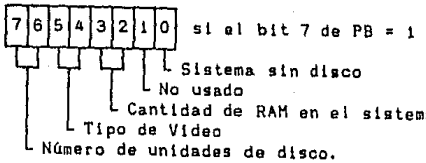
PUERTO PA (ENTRADA) 60H

Scan code del teclado

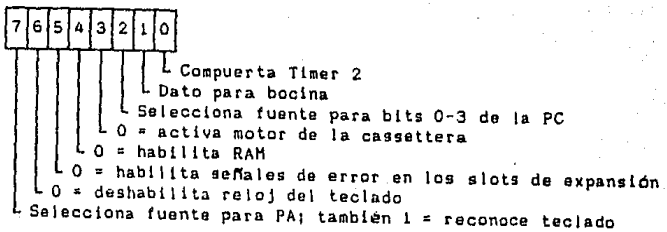
7	6	5	4	3	2	1	0

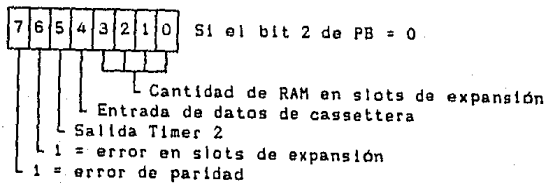
si el bit 7 de PB = 0

Switches de configuración.

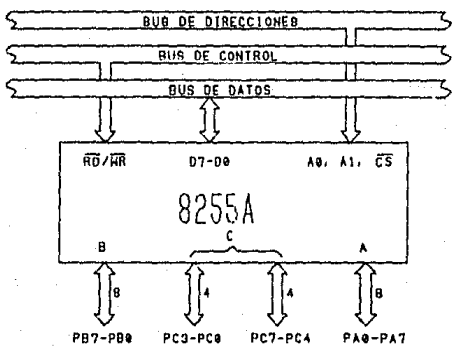


PUERTO PB (SALIDA) 61H





El diagrama general de interconexión del 8255 se muestra a continuación.



II.6 TECLADO.

a) Introducción.

El funcionamiento del teclado depende de la acción en conjunto de varios circuitos, además del teclado propiamente dicho.

Estos circuitos son los siguientes:

Controlador del teclado (8048). Este controlador es propiamente un convertidor matriz/scan code, que proporciona, a partir de la posición de la tecla presionada, el scan code asociado a ella.

Controlador de interrupciones (8259A). Por medio de este circuito el controlador del teclado interrumpe al CPU (interrupción 9), enviándole el scan code de la tecla presionada.

Puerto paralelo (8255). Al mismo tiempo que el controlador del teclado interrumpe al CPU, también proporciona al 8255 el scan code de la tecla presionada, depositándolo en el puerto A del mismo.

Circuitos lógicos. Estos circuitos básicamente son compuertas, registros de corrimiento (para convertir los datos seriales enviados desde el teclado, en un byte para el 8255), y flip-flops (utilizados para la lógica de interrupción al CPU).

b) Funcionamiento del teclado.

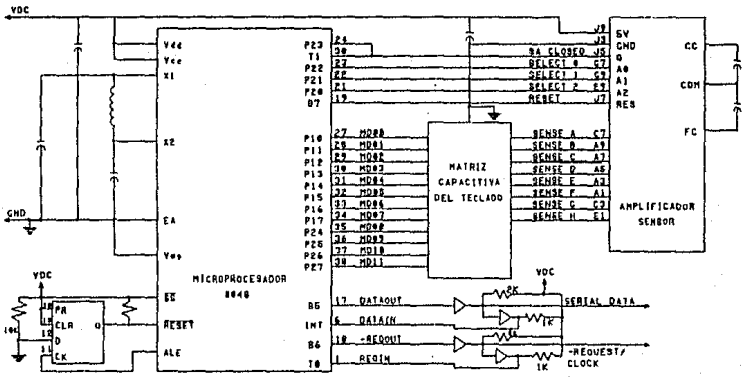
Cuando se presiona una tecla en el teclado, el controlador del mismo envía serialmente el scan code asociado a la tecla, a la vez que genera las señales necesarias para la interrupción al CPU a través del 8259A. El scan code es convertido en un byte y depositado en el puerto A del 8255, para su posterior lectura, cuando la interrupción de servicio del teclado (Int 9) sea ejecutada. Esta interrupción se encarga de convertir el scan code en un carácter, colocándolo en el buffer del teclado. Además, la interrupción también le indica al 8259 y al 8048 (a través del puerto B del 8255) que la interrupción está siendo atendida,

impidiendo con esto que otras interrupciones de menor prioridad sean bloqueadas.

Si la tecla presionada es de control (CTRL, ALT, SHIFTs, etc) no se genera el carácter ASCII asociado, en su lugar se modifica un byte en el área del BIOS correspondiente al status del teclado. Este byte puede ser accesado a través de la interrupción 16H.

c) Diagrama Básico del Teclado.

La interconexión de los circuitos contenidos en el teclado es la siguiente:

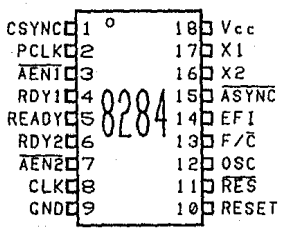


II.7. RELOJ 8284.

a) Introducción.

El 8284 es un generador de reloj y controlador para los procesadores iAPX 86,88 de la familia Intel. El modelo 8284 proporciona 5 MHz y el modelo 8284A 10 MHz. Utiliza un cristal o una señal TTL como fuente de frecuencia. Contiene señales de sincronización Local Ready y Multibus Ready. Requiere de una fuente de polarización única de 5 Volts. Es posible colocar diversos 8284 sincronizados.

b) Diagrama de terminales.



c) Descripción de terminales.

AEN1, AEN2 (Adress Enable). Señales de entrada que sirven para identificar su respectiva señal Bus Ready (RDY1 o RDY2). AEN1 valida RDY1, mientras AEN2 valida RDY2. Las dos señales AEN son usadas en sistemas que permiten al procesador el acceso a dos Multi-Master System Buses.

RDY1, RDY2 (Bus Ready). Señales de entrada, cada cual es una indicación de un dispositivo localizado en el bus de datos, que indican datos recibidos o bus listo.

ASYNC (Ready Sinchronization Select). Es una señal de entrada del modo de sincronización de la lógica de READY. Cuando ASYNC es

baja, dos etapas de la sincronización de READY se proporcionan. Cuando ASYNC es alta una etapa sencilla de la sincronización READY se proporciona.

READY. Señal de salida que es sincronizada con la señal de entrada RDY. READY es baja de nuevo después de mantenerse por un tiempo fijo al procesador.

X1, X2 (Cristal In). Son las terminales de entrada en las cuáles se conecta al cristal. La frecuencia del cristal es tres veces la frecuencia deseada para el procesador.

F/C (Frecuency/Crystal Select). Señal de entrada. Cuando es baja permite que la señal de reloj del procesador sea generada por el cristal. Cuando es alta la señal de reloj es generada a partir de la señal obtenida en la terminal EFI.

EFI (External Frecuency). Señal de entrada. Cuando un voltaje alto se aplica a F/C se obtendrá la señal de reloj a partir de la señal que se presente en esta terminal. La señal cuadrada de entrada debe ser tres veces de mayor frecuencia que la deseada.

CLK (Processor Clock). Terminal de salida, que será usada como reloj del procesador y todos aquellos dispositivos directamente colocados al bus del procesador local. La salida proporciona 4.5 Volts (Vcc=5 Volts) para manejar dispositivos MOS.

PLCK (Peripheral Clock). Señal de salida. Usada por dispositivos periféricos TTL. La frecuencia de salida es un medio de la obtenida en CLK.

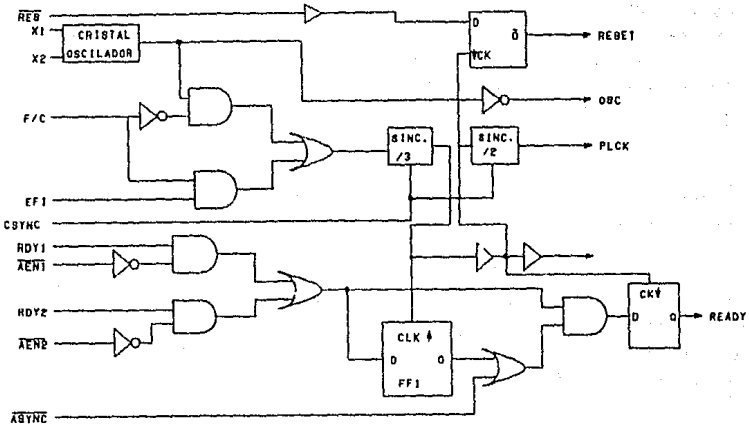
OSC (Oscillator Output). Es una salida con señal TTL del oscilador interno del circuito. Su frecuencia de salida es igual a la del cristal.

RES (Reset In). Señal de entrada que es usada para generar una señal de RESET. Generalmente se conectan a esta terminal un arreglo de resistencia y capacitor, empleados para estabilizar el reset al encender el sistema.

RESET (Reset). Terminal de salida, usada por los procesadores de la familia 8086. Sus características de tiempo son determinadas por RES.

CSYNC (Clock Synchonization). Terminal de entrada que permite a múltiples 8284 ser sincronizados y proporcionar relojes en fase. Cuando CSYNC es alta los contadores internos son inicializados. Cuando va a bajo los contadores internos empiezan a contar. CSYNC necesita ser sincronizado externamente por EFI. Cuando se emplean los osciladores internos CSYNC debe conectarse a tierra.

d) Diagrama de Bloques Interno.



e) Descripción del Diagrama de Bloques Interno.

OSCILADOR.

El circuito oscilador del 8284 está diseñado principalmente para emplearse con circuitos resonantes, fundamentalmente cristales, de los cuales se derivan frecuencias.

La frecuencia del cristal debe seleccionarse para ser tres veces mayor a la señal para el procesador. X1 y X2 son las dos entradas del cristal. Para una operación estable de la salida del circuito oscilador (OSC) se recomienda conectar dos resistencias de 510 ohms. La salida del oscilador es retenida momentáneamente y amplificada en corriente, por lo que pueden colocarse múltiples circuitos.

GENERADOR DE RELOJ.

Consiste de un contador divisor por tres síncrono con una entrada especial de limpieza que inhibe la cuenta. Esta señal

(CSYNC) permite que se sincronice las salida CLK con eventos externos.

SALIDAS DE RELOJ.

La salida CLK es un driver MOS con un ciclo de trabajo del 33%, necesarios por los procesadore iAPX 86,88. PLCK es una señal de reloj cuya frecuencia es la mitad de CLK. PLCK tiene un ciclo de trabajo de 50%.

LOGICA DE RESET.

Contiene entradas Schmitt trigger y un flip-flop sincronizado para sincronizar el frente de caida de CLK.

SINCRONIZACION DE READY.

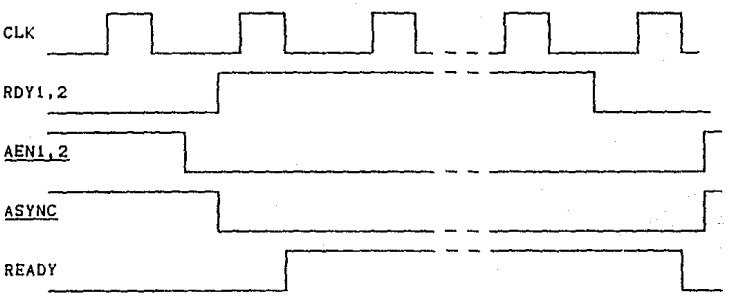
Se tienen dos entradas READY para accionar dos buses Multi-Master. Cada entrada es identificada con las señales AEN, que valida su respectiva señal de RDY. Si no existe un sistema Multi-Master la terminal AEN debe conectarse a bajo.

f) Función General del Circuito.

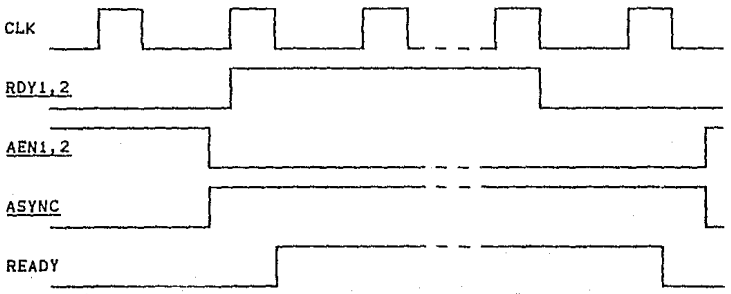
El 8284/8284A es un circuito que contiene un generador y un driver de reloj para los procesadores iAPX 86,88. El circuito contiene un oscilador controlado por cristal, un contador divisor por tres, sincronización completa en Multibus "Ready" y lógica de reset.

g) Diagramas de tiempos.

Señales Ready (Para circuitos asíncronos).



Señales Ready (para circuitos síncronos).

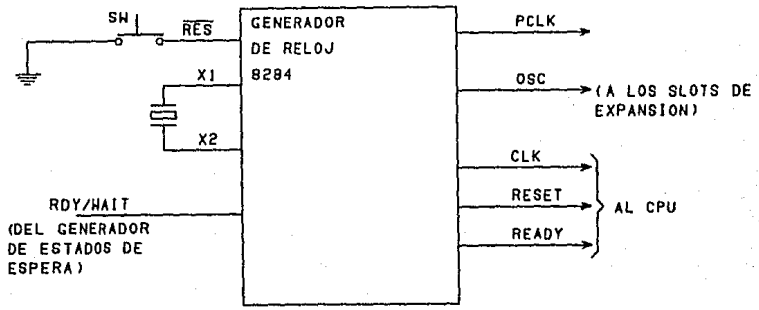


h) Interconexión del Circuito en una PC.

En las PCs originales este circuito proporcionaba la frecuencia del reloj mediante la colocación de un cristal que oscilaba a una frecuencia de 14.31818 MHz, la cual dividida por tres produce la frecuencia de 4.77 MHz necesaria para el procesador. La frecuencia del cristal se dividía por cuatro para producir 3.58 MHz que se alimentaban a los monitores de color.

Actualmente se utilizan cristales de mayor frecuencia para producir mayores frecuencias de operación.

La conexión del circuito se muestra a continuación:

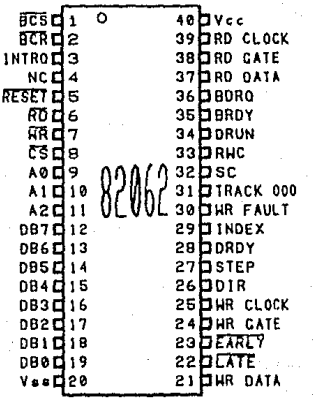


11.8. CONTROLADOR DE DISCO WINCHESTER 82062.

a) Introducción.

El circuito 82062 es un Controlador de Disco Winchester (CDW) que realiza la interface entre el microprocesador y el disco Winchester, que emplea interface de tecnología Seagate ST506/ST412. Estos podrían ser Seagate ST506 y ST412, Shugart SA604 y SA606, Tandem 600, y Computer Memories MS206 y CM5412. Este circuito convierte los datos paralelos provenientes del microprocesador a cadenas de bytes codificadas en MFM, a una tasa de 5 Mbit/seg. Proporciona toda la lógica de control para el drive y además las señales de control que simplifican el diseño de una malla cerrada de fase externa, y un circuito precompensado de escritura. El 82062 está diseñado para servir de interface entre el controlador local y un sector externo que sirva de buffer.

b) Diagrama de Terminales.



c) Descripción de Terminales.

BCS (Buffer Chip Select). Terminal de salida usada para habilitar la lectura o escritura de un sector externo empleado como buffer.
BCR (Buffer Counter Reset). Terminal de salida que es habilitada por el CDW para operar una lectura/escritura. Esta terminal emite un pulso siempre que la terminal BCS cambia de estado. Puede ser usada opcionalmente para limpiar la dirección del contador de la memoria buffer.
INTRQ (Interrupt Request). Terminal de salida que genera una interrupción a la terminación de un comando. Es limpiada cuando se lee el registro de estado.
RESET. Terminal de entrada que inicializa el controlador y borra todas la banderas de estado.
RD (Read). Terminal de entrada/salida. Como entrada controla el estado de la transferencia de información desde el CDW al procesador. Actúa como salida cuando el CDW está leyendo datos al buffer sector.
WR (Write). Terminal de entrada/salida. Como entrada controla la transferencia de comandos o información de la tarea a ejecutar en el archivo de registros del CDW. Es una salida cuando el CDW está escribiendo datos al buffer sector.
CS (Chip Select). Terminal de entrada que habilita a RD y WR como entradas.
A0-A2 (Adress). Terminales de entrada usadas para seleccionar un registro del archivo de registros de tareas.
DB7-DB0 (Data Bus). Terminales de entrada/salida que forman el bus de datos de 8 bits.
WR DATA (Write Data). Terminal de salida que indica la salida de datos codificados en MFM a una tasa determinada por la entrada Write Clock.
LATE. Terminal de salida usada para indicar un retraso empleado para una precompensación a la escritura. Válida cuando la salida WR GATE es alta.
EARLY Terminal de salida usada para obtener un retraso, empleado para una precompensación a la escritura. Válida cuando la salida WR GATE es alta.
WR GATE (Write Gate). Terminal de salida, alta cuando un dato escrito es válido. WR GATE es baja si la entrada WF es alta. Esta

salida es usada por el drive para habilitar la cabeza de escritura activa.

WR CLOCK (Write Clock). Terminal de entrada empleada para obtener la tasa de escritura de datos. Frecuencia = 5 MHz para la interface ST506, 4.34 MHz para la interface SA 1000.

DIR (Direction). Terminal de salida, un voltaje alto indica al driver mover la cabeza hacia el interior (incrementando el número de cilindro). La señal es determinada por el comando proporcionado al CDW.

STEP. Terminal de salida que proporciona pulsos de 8.4 microsegundos para mover la cabeza del drive a otro cilindro.

DRDY (Drive Ready). Terminal de entrada, conectada al drive. Si va a un estado bajo todos lo comandos son desactivados.

INDEX. Terminal de entrada. Esta señal proviene del drive e indica el comienzo de una pista. Es usada por el CDW durante el formateo y para contar intentos repetitivos.

WR FAULT (Write Fault). Terminal de entrada, indica una condición de error al CDW. Si WR FAULT proveniente del drive va a bajo todos los comandos son desactivados.

TRACK 000. Terminal de entrada usada por el comando Restore para verificar que la cabeza se encuentra en el cilindro más externo.

SC (Seek Complete). Terminal de entrada proveniente del drive que indica que la lectura y escritura puede realizarse.

RWC (Reduce Write Current). Terminal de salida que va a alto para todos los números de los cilindros arriba del valor preprogramado por el registro Write Precomp Cylinder. Es usado por la lógica de precompensación y por el drive.

DRUN (Data Run). Terminal de entrada, busca un string de ceros o unos en los datos leídos, indicando el inicio de un campo ID. Si se detectan los ceros RD GATE cambia a alto.

BRDY (Buffer Ready). Terminal de entrada usada por la memoria buffer para indicar al controlador que está lista para una lectura (llena) o para una escritura (vacía). BRDY es checada durante los comandos Read y Write.

BDRQ (Buffer Data Request). Terminal de salida, activada opcionalmente durante los comandos Read y Write si BRDY es alta. Puede ser usada por una línea de petición al DMA.

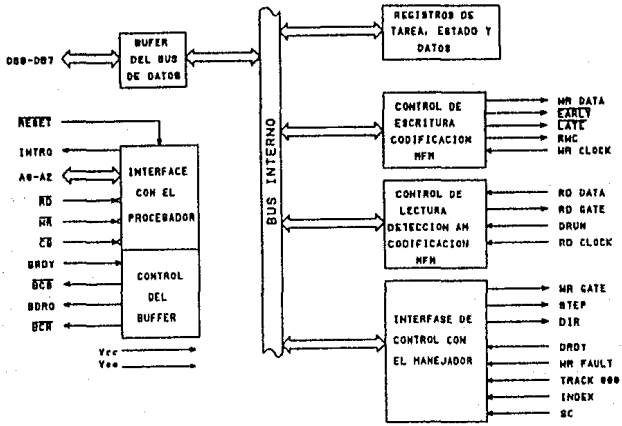
RD DATA (Read Data). Terminal de entrada, permite datos en MFM desde el drive.

RD GATE (Read Gate). Terminal de salida. Alta para campos de datos

o ID.

RD CLOCK (Read Clock). Terminal de entrada de una señal de reloj derivada de circuitos externos de recuperación de datos.

d) Diagrama de Bloques Interno.



e) Descripción del Diagrama de Bloques Interno.

CONTROLADOR DEL PLA.

El PLA interpreta los comandos y realiza todas las funciones de control. Es sincronizado con WR CLOCK.

COMPARADOR DE MAGNITUDES.

Un comparador de 10 bits se emplea para calcular el incremento, la posición actual y la deseada de la cabeza en el drive.

LOGICA DE CRC.

Genera y checa los caracteres de CRC (Cyclic Redundancy Check) añadidos a los campos de datos e ID. El polinomio usado

es:

$$x^{16} + x^{12} + x^5 + 1$$

CODIFICACION/DECODIFICACION MFM.

Codifica y decodifica los datos en MFM para ser escritos/leídos por el drive. La codificación en MFM opera con el WR CLOCK, el cuál tiene un reloj equivalente en frecuencia a la tasa de transmisión.

DETECCION DE MD.

El detector de marca de dirección checa el flujo de los datos obtenidos, buscando un reloj implícito (Data=A1h, Clock=0Ah), usado en cada campo .

CONTROL DE INTERFACE (CI) HOST/BUFFER.

La lógica de CI Host/Buffer contiene todos los circuitos necesarios para comunicarse con el bus de 8 bits del procesador del sistema.

CONTROL DE INTERFACE CON EL DRIVE.

La lógica de CI con el drive controla y checa todas las líneas provenientes del drive, con la excepción de los datos leídos o escritos.

f) Función General del Circuito.

El circuito de Intel 82062 es un controlador de disco Winchester que integra mucha de la lógica necesaria para implementar un subsistema controlador de disco Winchester. Provee un codificador de datos en MFM y todas las líneas de control requeridas por un disco duro que emplee interfaces estándar con tecnología Seagate ST506 o Shugart Associates SA1000. Generalmente, muchos de los discos Winchester de 5 1/4 y 8 pulgadas usan estas interfaces.

Debido a una de las tasas de transmisión requeridas por estos drives (1 byte cada 1.6 µseg) el 82062 se diseñó para servir de interface entre el CPU del sistema o un controlador de I/O y un buffer externo RAM. El CDW 82062 tiene cuatro terminales que minimizan la lógica necesaria para diseñar una interface con el

buffer.

El CDW es controlado por el CPU del sistema usando seis comandos:

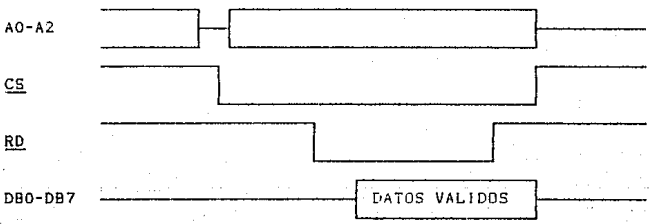
Restore
Seek
Read Sector
Write Sector
Scan ID
Write Format.

Estos comandos usan información almacenada en seis registros de tarea. La ejecución de los comandos comienza inmediatamente después de que el registro de comandos es cargado (a menudo los comandos requieren solo un byte desde el CPU despues de que el CDW ha sido inicializado).

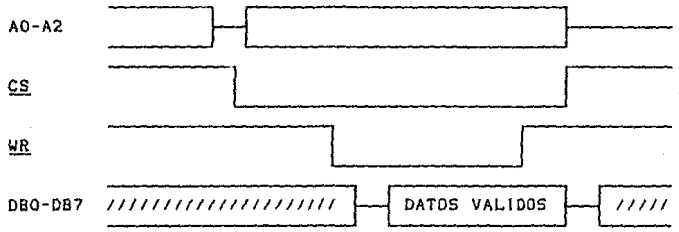
El 82062 añade todo el formateo de pistas a los campos de datos, incluyendo dos bytes de CRC. Opcionalmente, estos dos bytes pueden reemplazarse por siete bytes de información ECC para corrección externa de errores.

g) Diagramas de tiempos.

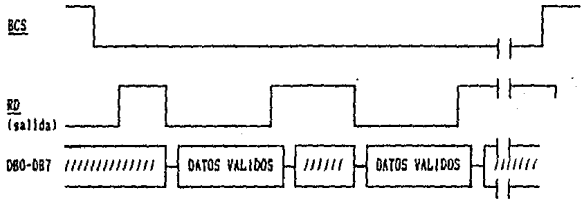
Lectura por el CPU del sistema.



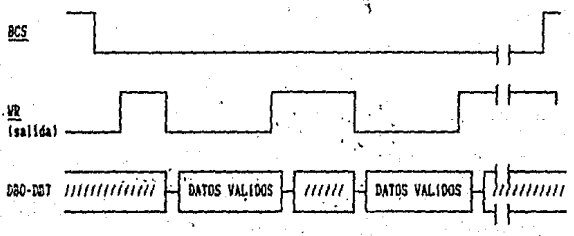
Escritura por el CPU del sistema.



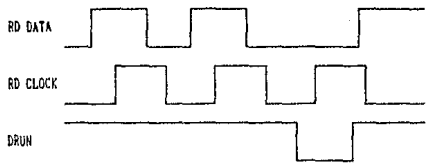
Lectura de Buffer (Comando Write Sector).



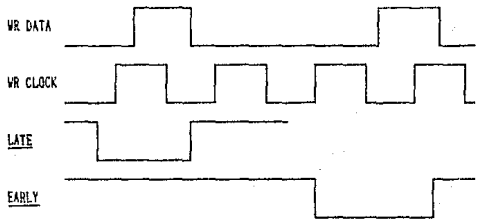
Escritura a Buffer (Comando Read Sector).



Lectura de Datos.



Escritura de datos.



h) Programación del Circuito.

Archivo de registros de tareas.

Es un banco de registros usados para mantener parámetros de información perteneciente a cada comando. Estos registros y su direccionamiento son:

A2	A1	A0	Lectura	Escritura
0	0	0	(Bus en tercer estado)	(Bus en tercer estado)
0	0	1	Banderas de error	Menor corriente escritura
0	1	0	Contador de sector	Contador de sector
0	1	1	Número de sector	Número de sector
1	0	0	Cilindro, parte baja	Cilindro, parte baja
1	0	1	Cilindro, parte alta	Cilindro, parte alta
1	1	0	SDC	SDC
1	1	1	Registro de estado	Registro de comando

REGISTRO DE ERROR.

Este registro de sólo lectura contiene el estado del error específico después de completar un comando. Sus bits se definen como sigue:

7	6	5	4	3	2	1	0
BBD	CRC	-	ID	-	AC	TK000	DM

BIT 7 - Bad Block Detect.

Este bit enciende cuando en un campo ID se ha encontrado una marca de bloque defectuoso. Se usa para mapear sectores defectuosos.

BIT 6 - CRC Data Field.

Este bit enciende cuando un error en el campo de datos CRC ha ocurrido o la marca de dirección del dato no se ha encontrado. El sector puede ser leído pero tendrá errores.

BIT 5 - Reservado, no usado.

Siempre es uno.

BIT 4 - ID Not Found.

Este bit enciende cuando el cilindro, cabeza, sector o tamaño deseados no ha sido encontrado después de 8 revoluciones del disco, o si ocurre un error en el CRC del ID.

BIT 3 - Reservado, no usado.

Siempre es uno.

BIT 2 - Aborted Command.

Este bit enciende si un comando se estaba ejecutando mientras DRDY (terminal 28) o WR FAULT (terminal 30) es baja. Este bit también encenderá si se escribe un comando indefinido en el registro COMMAND, pero se ejecuta una búsqueda.

BIT 1 - Track 000.

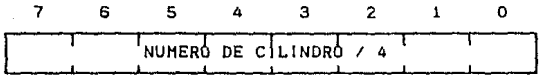
Este bit enciende sólo con el comando RESTORE. Indica que la terminal TRACK 000 (terminal 31) no ha sido activada después de ejecutar 1024 pulsos de posicionamiento.

BIT 0 - Data Address Mark.

Este bit enciende durante un comando READ SECTOR si la marca de dirección del dato no se encuentra después de leer el sector ID apropiado.

REGISTRO DE MENOR CORRIENTE ESCRITURA.

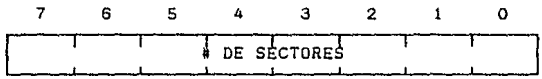
El registro es usado para definir el número de sector donde la terminal RWC (terminal 33) es válida:



El valor (0-255) cargado en este registro es multiplicado internamente por 4 para especificar el cilindro actual en el cuál RWC es válida. Así, un valor de 01H implica que RWC se activará en el cilindro 4, 02H implica en el registro 8 y así sucesivamente. Los puntos de activación son entonces los cilindros 0,4,8..1020. La terminal RWC será activada cuando el cilindro actual es mayor o igual que el cilindro indicado es este registro.

REGISTRO DEL CONTADOR DE SECTORES.

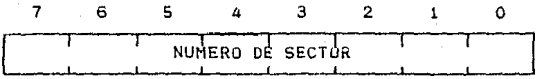
Este registro es usado para definir el número de sectores que necesitan ser transferidos al buffer durante los comandos READ MULTIPLE SECTOR o WRITE MULTIPLE SECTOR:



El valor contenido en este registro es decrementado después de que cada sector es transferido de/hacia el buffer. Un cero representa 256 sectores a transferir, un uno 0 sectores a transferir, etc. Este registro no tiene validez cuando se especifica un comando de sectores individuales.

NUMERO DE SECTOR.

Este registro mantiene el número de sector deseado.

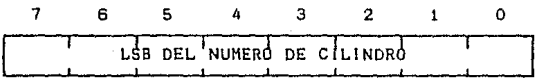


Para un comando de múltiples sectores, especifica el primer sector a ser transferido. Se decrementa después que cada sector es transferido de/hacia el buffer. Este registro puede contener cualquier valor de 0 a 255.

También es usado para programar el largo de la ranura (Gap1 y Gap 3) a ser usada al formatear un disco. Una explicación mejor se tendrá al hablar del comando WRITE FORMAT.

REGISTRO DE CILINDRO, PARTE BAJA.

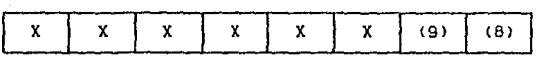
Este registro mantiene el byte menos significativo del número de cilindro deseado:



Es usado junto con el registro de cilindro, parte alta para especificar un rango de 0 a 1023.

REGISTRO DE CILINDRO, PARTE ALTA.

Este registro contiene los dos bits más significativos del número de cilindro deseado:

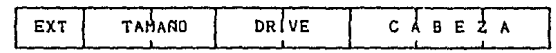


Interno al CDW 82062 hay otro par de registros que contienen la posición actual donde se encuentran localizadas las cabezas de lectura/escritura. Los registros de cilindro, partes alta y baja pueden ser considerados como el número de cilindro destino de los comandos de búsqueda y otros. Después de que estos comandos son ejecutados, el registro interno con la posición del registro es igual a los registros de número de cilindro, parte baja y alta. Si se detecta un cambio en el número proveniente del drive con un nuevo comando, se lee automáticamente un campo ID para actualizar

el registro interno de la posición de las cabezas. Esto afecta a todos los comandos, excepto RESTORE.

REGISTRO DE SECTOR/DRIVE/CABEZA.

El registro SDC contiene el tamaño del sector deseado, número de drive y el número de la cabeza. En la forma siguiente.



6	5	TAMAÑO DEL SECTOR
0	0	256
0	1	512
1	0	1024
1	1	128

Los bits de drive y cabeza contienen el número de drive y cabeza a emplear. El número de cabeza y tamaño de sector son comparados con el campo ID de los datos provenientes del disco. Las líneas selectoras de cabeza y drive no están disponibles como salidas del 82062 y deben ser generadas externamente.

El bit 7 (Extensión) es usado para extender el campo de datos a siete bytes cuando se usan códigos ECC. Cuando EXT=1, el CRC no es añadido al final del campo de datos, el campo de datos contiene "tamaño del sector +7" bytes de longitud. El CRC es checado en el campo ID, de acuerdo al estado de EXT. Note que los bits del tamaño del sector (6 y 5) son escritos en el campo ID durante el comando de formateo. El byte SDC escrito en el campo ID es diferente al contenido del registro SDC. El byte SDC no tiene escrito el número de drive (DRIVE) pero tiene una marca de bloque defectuoso en el bit 7.

REGISTRO DE ESTADO.

Es un registro de sólo lectura que informa al CPU del sistema de ciertos eventos que se presentan en el CDW 82062, también reporta el estado de de las líneas de control del drive. El formato es:

7	6	5	4	3	2	1	0
BUSY	READY	WF	SC	DRQ	-	CIP	ERROR

BIT 7 - Busy.

Este bit enciende siempre que el CDW está accesando al disco. No deben escribirse comandos cuando este bit esté encendido. Busy enciende cuando es escrito un comando en el CDW y apaga al final de los comandos, excepto READ SECTOR, Busy apaga después de que el sector se ha encontrado. Cuando el bit Busy está encendido, los otros bits de estado deben tomarse como válidos.

BIT 6 - Ready.

Este bit normalmente refleja el estado de la línea DRDY (terminal 28). Cuando se genera una interrupción por un error de un comando abortado, el bit Ready es mantenido para posteriores revisiones por parte del CPU del sistema. Después de la lectura del registro STATUS este bit seguirá reflejando el estado de la línea DRDY.

BIT 5 - Write Fault.

Este bit refleja el estado de la línea WR FAULT (terminal 30). Siempre que WR FAULT va a alto se genera una interrupción. El bit Write Fault es retenido, como el bit Ready (bit 6).

BIT 4 - Seek Complete.

Este bit refleja el estado de la línea SC (terminal 32). Ciertos comandos esperan hasta que el bit Seek Complete enciende. Este bit es retenido como el bit Ready.

BIT 3 - Data Request.

Este bit (DRQ) refleja el estado de la línea BDRQ (terminal 36). Enciende cuando el sector debe cargarse con datos o ser leído por el CPU, dependiendo del comando. El bit DRQ y la línea BDRQ permanecen altas hasta que se detecta BDRY, indicando que la operación ha terminado. BDRQ puede usarse para interfaces con DMA, mientras DRQ puede usarse para transferencias de entrada/salida.

BIT 2 - Reservado.

Forzado a cero.

BIT 1 - Command in Progress.

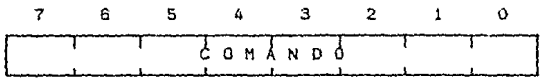
Cuando este bit enciende, un comando se está ejecutando y no debe cargarse otro nuevo hasta que se apague el bit. Aunque puede ejecutarse un comando, el sector no está listo para ser accesado por el CPU.

BIT 0 - Error.

Este bit enciende siempre que cualquier bit del registro ERROR enciende. Es el "or" lógico de los bits en el registro de error y puede usarse por el procesador para chequeos rápidos de la terminación sin errores de comandos. Este bit apaga cuando un nuevo comando es escrito en el registro COMMAND.

REGISTRO COMMAND.

Este registo es del tipo de sólo escritura, y contiene el comando deseado:



El comando empieza a ejecutarse inmediatamente después de cargarse. Este registro no debe cargarse mientras los bits Busy o Command in Progress están encendidos en el registro de estado. Si la línea INTRQ (terminal 3) está alta, será apagada con la escritura del registro COMMAND.

CONJUNTO DE INSTRUCCIONES.

El conjunto de instrucciones del CDW consta de seis comandos. Primeramente el CPU debe actualizar el archivo de registros con la información necesaria para el comando. Excepto con el registro COMMAND, los registros pueden cargarse en cualquier orden. Si se está ejecutando un comando, las escrituras al registro COMMAND serán ignoradas hasta la ejecución del actual comando.

COMANDO:	7	6	5	4	3	2	1	0
RESTORE	0	0	0	1	R3	R2	R1	R0
SEEK	0	1	1	1	R3	R2	R1	R0
READ SECTOR	0	0	1	0	1	M	0	T
WRITE SECTOR	0	0	1	1	0	M	0	T
SCAN ID	0	1	0	0	0	0	0	0
WRITE FORMAT	0	1	0	1	0	0	0	0

R3 - R0 = Tasa de intercambio

Para 5 MHz en la terminal WR CLOCK

R0-R3 =	0000	-	≈35 µs
	0001	-	0.5 ms
	0010	-	1.0 ms
	0011	-	1.5 ms
	0100	-	2.0 ms
	0101	-	2.5 ms
	0110	-	3.0 ms
	0111	-	3.5 ms
	1000	-	4.0 ms
	1001	-	4.5 ms
	1010	-	5.0 ms
	1011	-	5.5 ms
	1100	-	6.0 ms
	1101	-	6.5 ms
	1110	-	7.0 ms
	1111	-	5.5 ms

T = Habilita intentos sucesivos

T = 0 Habilita intentos sucesivos
T = 1 Desabilita intentos sucesivos

M = Bandera para Múltiples Sectores

M = 0 Transfiere 1 sector
M = 1 Transfiere múltiples sectores

I = Habilita interrupciones

I = 0 Interrumpe al cumplirse BDRQ
I = 1 Interrumpe al final del comando

RESTORE.

Este comado se usa generalmente después de encender el sistema. La tasa actual de incrementos usada por este comando es determinada por el tiempo de búsqueda completa. Un pulso es esperado en el CDW por la terminal SC para calcular el tiempo hasta el siguiente pulso en esa terminal. Si después de 1024

pulsos de posicionamiento la línea TRACK 000 no se activa, el CDW enciende el bit TRACK 000 en el registro de ERROR y habilita INTRQ. También ocurre una interrupción si se activa WR FAULT o se desactiva DRDY en cualquier tiempo durante la ejecución.

La tasa de transmisión especificada (R0-R3) se almacena en un registro interno para usos futuros, en comandos que impliquen búsquedas.

SEEK.

Para todas las peticiones de búsqueda se puede emplear este comando, con el que se pueden traslapar diferentes búsquedas en múltiples drives. La tasa actual de transmisión se almacena en un registro interno para futuros usos. Si se activa DRDY o WR FAULT al ejecutar la búsqueda, el comando termina y se activa la línea INTRQ.

La dirección y número de posicionamientos necesitados es calculado comparando los registros de cilindro, partes alta y baja y el par de registros internos. Después de esto se actualizan los comandos internos y termina el comando. La línea SC (Seek Complete) no se checa al comienzo o fin del comando.

Si se presenta una búsqueda, el 82062 hará la búsqueda hasta que SC va a alto.

READ SECTOR.

Este comando se usa para transferir uno o más sectores de datos del disco al buffer. Cuando recibe este comando el 82062 checa el par de registros de número de cilindro, partes alta y baja con el par interno de la posición actual, para decidir la dirección y número de posicionamientos. Las líneas WR FAULT y DRDY se monitorean al ejecutarse este comando.

En caso de que los datos dentro del ID no corespondan a los buscados, se checa el estado del bit T y se decide un nuevo intento en el mismo sector, hasta en 8 revoluciones del disco.

Si se colocó el bit M en 1 se realiza la primera transferencia de un sector, se decrementa el contador de sectores y se transfiere el siguiente sector, sucesivamente hasta que el contador es cero.

WRITE SECTOR.

Este comando se usa para escribir uno o más sectores de datos al disco desde el buffer. Se checan los cilindros deseados y actual, y de ser necesario se hace el posicionamiento. Cuando se encuentra la señal SC alta, la señal BDRQ se activa y el CPU descarga el buffer. Cuando se sensa BDRY alta, se busca el campo ID con las características deseadas.

El bit T en 1 indica intentos de hasta 8 revoluciones del disco si los parámetros buscados en el ID no se encuentran. El bit M en 1 indica escrituras en múltiples sectores, por lo que se hace uso de nueva cuenta del contador de sectores.

SCAN ID.

Este comando se usa para actualizar los registros de sector/drive/cabeza y número de cilindro, partes alta y baja..

Después de cargar este comando la línea SC es muestreada hasta que es válida. Las líneas DRDY y WR FAULT también se monitorean al ejecutarse el comando. Cuando se encuentra el primer campo ID, su información es cargada en los registros SDC, número de sector y número de cilindro. Los registros de posición de cilindro internos también se actualizan. Si se detecta un bloque malo, el registro BAD BLOCK se enciende. Si ocurre un error en el CRC el 82062 intenta hasta en 8 revoluciones del dico encontrar un ID libre de errores.

WRITE FORMAT.

Se usa para formatear una pista usando el archivo de registros de tarea y el buffer. Durante la ejecución de este comando el buffer es usado para información de parámetros adicionales del sector de datos.

Como en otros comandos existe una búsqueda implícita, si el número del drive ha cambiado un campo ID se busca para el posicionamiento en el cilindro deseado. Si no puede leerse un campo ID, se genera un error de ID no encontrado y el comando WRITE FORMAT aborta. Esto puede ser prevenido usando el comando RESTORE antes de formatear.

El registro contador de sectores se usa para mantener el número total de sectores a ser formateados (FFh=255), mientras que el registro de número de sector contiene el número de bytes menos tres usados por Gap 1 y Gap 3; por ejemplo, si el registro

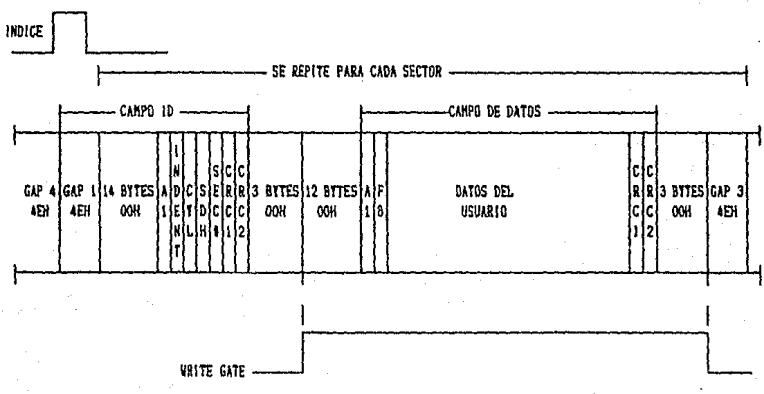
contador de sectores tiene un valor de 02h y el registro de número de sector tiene 00h, entonces 2 sectores son escritos y tres bytes de 4Eh se escriben en Gap 1 y Gap 3. Los campos de datos son llenados con FFh y el CRC es generado automáticamente y añadido. El bit de extensión de sectores en el registro SDC no se encenderá. Después de que el sector es escrito, la pista se llena con 4Eh.

El valor de Gap 3 se determina por la variación de la velocidad del motor del drive, el largo del sector y el factor de intermedios. Este factor es muy importante cuando se escoge de 1:1. La fórmula para determinar el valor mínimo del largo de Gap 3 es:

Gap 3 = (2 * M * S) + K + E

Donde: M = variación de la velocidad del motor $\pm$ 3%
S = largo del sector en bytes.
K = 25 por el factor de intermedios por 1
K = 0 para cualquier otro valor del factor
E = 7 si el factor será extendido

Como todos los comandos, una condición de WR FAULT o drive no listo termina la ejecución del comando WRITE FORMAT. La siguiente figura muestra el formato que el 82062 escribe en el disco.



CAMPO ID:
A1 = A1H CON RELOJ A0H
IDENT = MSB DEL NUMERO DE CILINDRO

BYTE SDH = BITS 1,1,2 = NUMERO DE CABEZA
BITS 3,4 = 0
BITS 5,6 = TAMAÑO DEL SECTOR

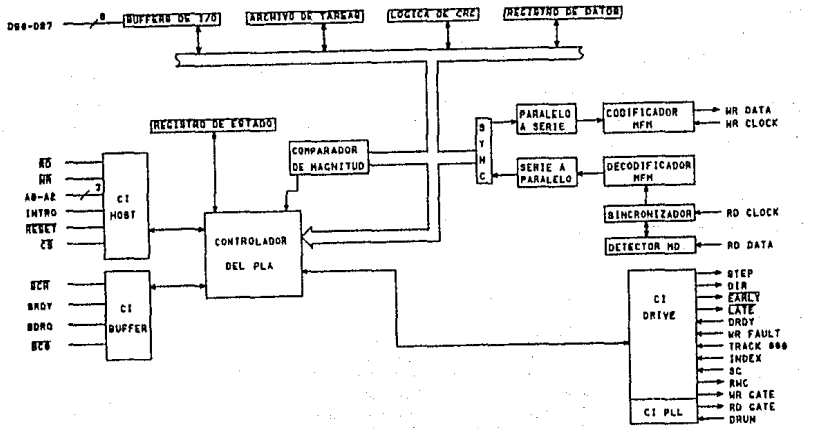
CAMPO DE DATOS
A1 = A1H CON RELOJ A0H
F8 = MARCA RDIRECCION DE DATOS

FE = 0-255 CILINDROS BIT 7 = MARCA DE BLOQUE MALO USUARIO = CAMPO DE 128-1024 BYTES
FF = 256-511 CILINDROS
FC = 512-767 CILINDROS SEC # = NUMERO DE SECTOR LOGICO
FD = 768-1023 CILINDROS

i) Interconexión del Circuito en una PC.

Los registros mencionados anteriormente se encuentan localizados de la localidad 300H a 307H del mapa de puertos en una PC.

La conexión física del circuito se muestra en la siguiente gráfica:



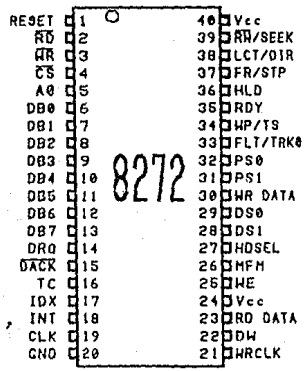
II.9. CONTROLADOR DE DISCO FLEXIBLE (FDC) 8272

a) Introducción.

El 8272 es un circuito integrado LSI que tiene asignada la tarea de transferencia de datos entre el microprocesador y cualquiera de hasta cuatro unidades de disco flexible (DD). El 8272 soporta dos densidades de grabación de datos para los discos: la densidad sencilla (utilizando FM para grabar datos) y la densidad doble (utilizando MFM). Esto permite que el 8272 sea soportado además en los sistemas IBM 3740 e IBM 34.

El 8272 funciona rodeado de varios circuitos más, tales como el PLL para la separación de datos, un precompensador de escritura de datos, etc. Además, es posible que realice la transferencia de datos utilizando al DMA o sin el.

b) Diagramas de Terminales.



c) Descripción de Terminales.

RST. Esta señal se encuentra en la terminal 1, es de entrada y se conecta directamente al microprocesador (µP). La señal de Reset coloca al FDC en estado de espera, desactivando todas las señales de salida.

RD. Esta señal ocupa la terminal 2, es de entrada y se conecta al µP. La señal de Read habilita la transferencia de datos del FDC al bus de datos.

WR. Esta señal está en la terminal 3, es de entrada y se conecta al µP. La señal de Write habilita la transferencia de datos del bus de datos al FDC.

CS. Esta señal está en la terminal 4, es de entrada y se conecta al µP. La señal de Chip Select habilita al FDC. Ninguna lectura o escritura se realiza hasta que ocurra esta señal.

AO. Esta señal ocupa la terminal 5, es de entrada y se conecta al µP. La señal AO selecciona el Data Register o el Main Status Register para entrada o salida, en conjunto con las señales RD y WR.

DB0-DB7. Estas señales ocupan las terminales 6 al 13, son de entrada/salida y se conectan al µP. Estas señales son las líneas del Bus de Datos de 8 bits.

DRQ. Esta señal se encuentra en la terminal 14, es de salida y se conecta al DMA. Esta señal constituye la petición de transferencia para el DMA.

DACK. Esta señal ocupa la terminal 15, es de entrada y se conecta al DMA. Esta señal es el reconocimiento por parte del DMA de la petición de transferencia.

TC. Esta señal se encuentra en la terminal 16, es de entrada y se conecta al DMA. Esta señal proviene de la señal TC/EOP del DMA y sirve para terminar un comando que el FDC esté ejecutando. Si no se utiliza DMA para la transferencia de datos, el µP debe suplirla de alguna forma.

IDX. Esta señal ocupa la terminal 17, es de entrada y se conecta al DD. La señal Index indica la detección de la marca física del disco flexible para el inicio de un track.

INT. Esta señal se encuentra en la terminal 18, es de salida y se conecta al µP. La señal de interrupción realiza la petición de interrupción por el FDC.

CLK. Esta señal ocupa la terminal 19, es de entrada. La señal Clock es el reloj para el FDC y tiene una frecuencia de 8 MHz, con un ciclo de trabajo del 50%.

GND. Esta terminal, número 20, se conecta a la tierra eléctrica.

WR CLK. Esta señal ocupa la terminal 21 y es de entrada. La señal de Write Clock es dada por un reloj adicional y es utilizada por el FDC para la grabación de datos. Para el formato FM WR CLK vale 500 kHz y para MFM vale 1 MHz. Esta señal debe estar presente en todo momento.

DW. Esta señal se encuentra en la terminal 22, es de entrada y se conecta al PLL. La señal Data Window permite al FDC realizar la separación de datos al leer del disco.

RD DATA. Esta señal ocupa la terminal 23, es de entrada y se conecta al DD. La señal de Read Data es la entrada de datos en forma serial proveniente del DD.

VCO. Esta señal se encuentra en la terminal 24, es de salida y se conecta al PLL. La señal de VCO habilita al PLL para sincronizarlo con la entrada de datos del DD.

WE. Esta señal ocupa la terminal 25, es de salida y se conecta al DD. La señal de Write Enable habilita al DD para la escritura.

MFM. Esta señal se encuentra en la terminal 26, es de salida y se conecta al PLL. La señal de modo MFM habilita el modo de grabación MFM. Cuando esta señal no se da, el modo FM es el seleccionado.

HDSEL. Esta señal ocupa la terminal 27, es de salida y se conecta al DD. La señal Head Select selecciona la cabeza 0 ó 1 del DD.

DS1,DS2. Estas señales se encuentran en las terminales 28 y 29, son de salida y se conectan al DD. Las señales de Drive Select seleccionan uno de los cuatro DD posible.

WR DATA. Esta señal se encuentra en la terminal 30, es de salida y se conecta al DD. La señal de Write Data es la salida de datos serial a escribir en el disco.

PS1,PS2. Estas señales ocupan las terminales 31 y 32, son de salida y se conectan al DD. Las señales de Precompensation controlan la precompensación durante el modo MFM de grabación.

FTL/TRK0. Esta señal se encuentra en la terminal 33, es de entrada y se conecta al DD. La señal de Fault/Track 0 sensa la

condición de falla del DD en el modo de Lectura/Escritura, y el Track 0 en el modo de Búsqueda.

WP/TS. Esta señal ocupa la terminal 34, es de entrada y se conecta al DD. La señal de Write Protect/Two-Sided sensa la condición de protección contra escritura en el modo de Lectura/Escritura, y la condición de doble lado del disco en el modo de Búsqueda.

RDY. Esta señal se encuentra en la terminal 35, es de entrada y se conecta al DD. La señal de Ready sensa la condición que indica que el DD se encuentra listo para operar.

HDL. Esta señal ocupa la terminal 36, es de salida y se conecta al DD. La señal de Head Load indica al DD que posicione a la cabeza de Lectura/Escritura en contacto con el disco.

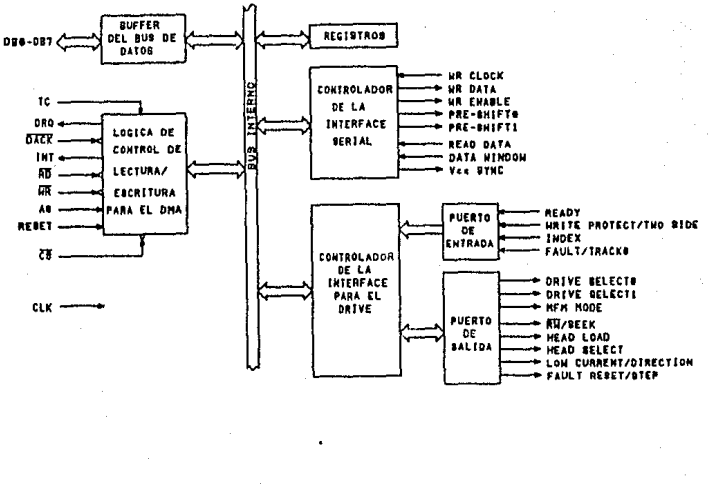
FR/STP. Esta señal se encuentra en la terminal 37, es de salida y se conecta al DD. La señal de Fault Reset/Step coloca en cero el flip flop de falla en el DD en el modo de Lectura/Escritura. En el modo de Búsqueda provee al DD los pulsos necesarios para desplazar a la cabeza de un cilindro a otro.

LCT/DIR. Esta señal ocupa la terminal 38, es de salida y se conecta al DD. La señal de Low Current/Direction en el modo de Lectura/Escritura indica que la cabeza del DD sea posicionada sobre los cilindros internos (44-77) del disco. En el modo de Búsqueda determina la dirección hacia la cual se moverá la cabeza con los pulsos dados por **FR/STP** (un uno lógico indica un movimiento hacia adentro del disco, un cero lógico indica un movimiento hacia afuera del disco).

RW/SEEK. Esta señal se encuentra en la terminal 39, es de salida y se conecta al DD. La señal de Read, Write/Seek selecciona, con un uno lógico, el modo de Búsqueda, y con un cero lógico el modo de Lectura/Escritura.

VCC. Esta terminal, el numero 40, se conecta a la fuente de alimentación de +5 Volts.

d) Diagrama de Bloques Interno.



e) Descripción del Diagrama de Bloques Interno.

Buffer del Bus de Datos. Este bloque se encarga de la transferencia de datos o de comandos provenientes del bus general del sistema, hacia el bus interno del 8272.

Lógica de Control de Lectura/Escritura para el DMA. Como su nombre lo indica, este bloque se encarga de la comunicación entre el DMA y el 8272 para la transferencia de datos.

Controlador de la Interface Serial. Este bloque maneja la recepción y el envío de datos desde o hacia el DD. La función de este bloque también consiste en recibir la ventana de datos para posibilitar la separación de datos y pulsos de reloj.

Controlador de la Interface para el Drive. Este bloque se encarga de ordenar al DD las acciones específicas que debe realizar (selección del Drive, Reset, etc), así como recibir la condición del DD mediante la habilitación de diversas líneas.

Registros. Este bloque contiene información interna del 8272, tal como su condición, el número de track actual, etc.

Puerto de Entrada. Mediante este puerto el DD comunica al

8272 su condición actual, así como los resultados de las operaciones enviadas por el 8272.

Puerto de Salida. Mediante este puerto, el 8272 comunica al DD las acciones que debe de realizar.

f) Función General del Circuito.

El 8272 puede realizar diferentes acciones sobre el DD dependiendo de los comandos recibidos desde el microprocesador. Básicamente, las acciones que puede realizar son las siguientes:

Leer datos de un disco flexible.

Escribir datos en un disco flexible.

Realizar búsquedas de información en un disco flexible.

Formatear un track.

Cuando el CPU desea realizar una transferencia desde o hacia el disco flexible, debe mandar una secuencia de bytes que identifiquen al comando a realizar. Acompañando a estos bytes, se debe de suministrar información referente a los parámetros del comando, tales como: la dirección del cilindro (0-76), el DD sobre el que se opera (0-3), la cabeza a utilizar (0-1), etc. Con toda esta información, el 8272 ya puede ejecutar las funciones para llevar a cabo el comando.

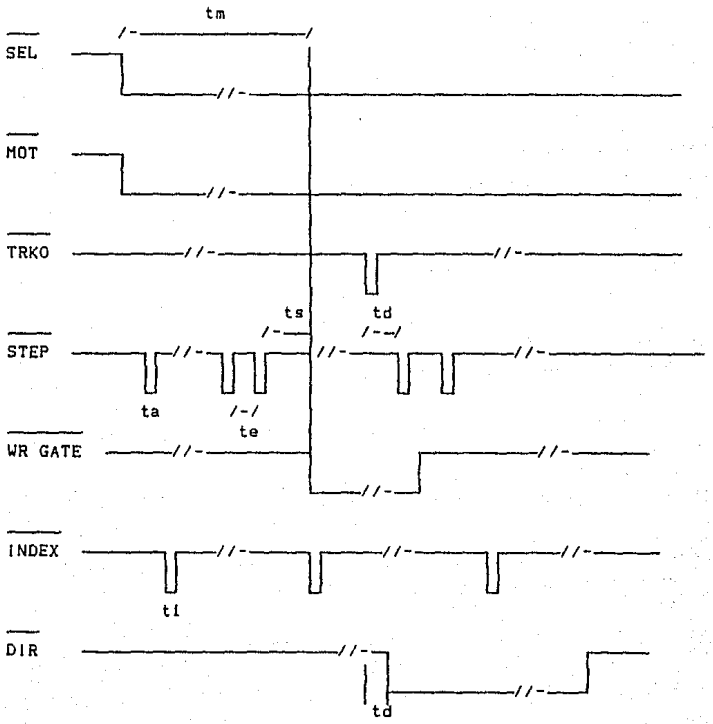
La ejecución de un comando involucra varias fases. La primera de ellas (command phase) es la transferencia de información hacia el FDC para la ejecución del comando (después de recibir un Reset). La segunda fase (execution phase) es la de ejecución del comando por el FDC, terminando hasta la completa transferencia de información o en caso de haber ocurrido un error. La última fase (result phase), consiste en el reporte del resultado de la operación ordenada.

Para realizar las operaciones de transferencia hacia o desde el DD, el 8272 utiliza dos tipos de formatos, dependiendo del tipo de grabación. Ambos formatos tienen en común el uso de la intercalación de pulsos de reloj con los pulsos de la información. Así, cuando se escribe o se lee información en un disco, ésta debe de estar acompañada de pulsos proporcionados por un reloj independiente al del sistema. En el formato FM, cada bit de información está flanqueado por un pulso de reloj, siendo la duración total de pulsos e información de 4 µs. En el formato MFM

el bit de información no siempre está flanquedo por pulsos de reloj, sólo en el caso de que el bit de información anterior y el presente sean igual a cero. La duración de pulsos e información es de 2 µs.

La grabación de los datos junto con los pulsos apropiados de reloj se realiza por un circuito externo al 8272: el precompensador de escritura. Asimismo, la separación de datos, de los pulsos de reloj, es realizado por un circuito exterior llamado PLL.

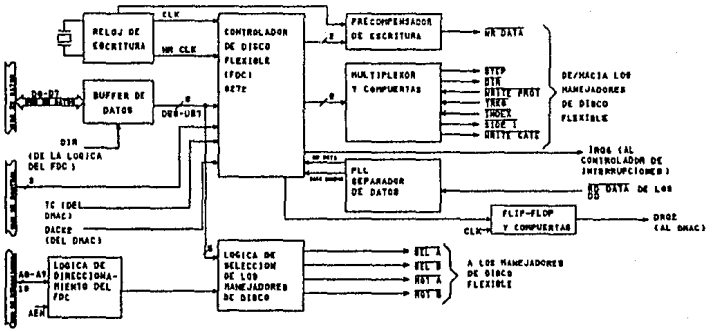
g) Diagramas de tiempos.



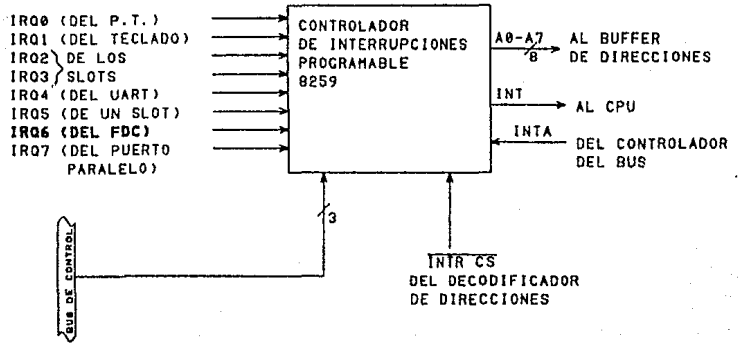
ts >= 35ms
ta = 1ms
te = 10ms
ti > 1ms y < 4ms
tm >= 0.5 s
td >= 1ms

h) Interconexión del Circuito en una PC.

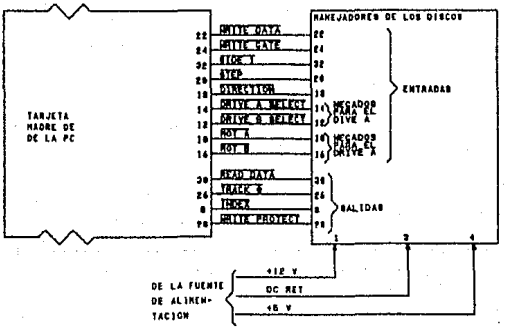
El 8272 se relaciona de varias maneras con múltiples circuitos. El subsistema que soporta al FDC está formado por un reloj independiente al del sistema, circuito de precompensador de datos, PLL, selector de DD, buffer de datos y diversos elementos lógicos. Estos circuitos realizan funciones adicionales a las del 8272. El conjunto se muestra a continuación.



Para la comunicación con el exterior, y para reportar resultados al CPU, el 8272, posee una línea propia para él, con la que puede interrumpir al procesador, a través del controlador de interrupciones 8259A. El diagrama de esta conexión se muestra a continuación.



Por último, a causa de que el 8272 se encuentra en la tarjeta madre de la PC, existen varias líneas de conexión con los DDs a través de un conector y un cable plano que parte de éstos. Las líneas de conexión se muestran a continuación.



g) Programación del Circuito.

Como se indicó más arriba, el 8272 puede ser controlado mediante el envío de varios comandos desde el CPU. Los comandos que el 8272 reconoce son los siguientes:

Especificación. Este comando permite especificar diversas constantes de operación para el FDC y el DD. Estas constantes son:

El tiempo de carga de la cabeza de Lectura/Escritura del DD.

El tiempo de descarga de la cabeza del DD.

El tiempo entre un pulso y el siguiente para realizar un acceso de un track al siguiente track.

Sensar la condición del DD. Este comando puede ser utilizado para que el CPU obtenga información de la condición del DD, mediante el acceso al registro de estado.

Sensar la condición de interrupción. Mediante este comando, el CPU puede conocer la causa por la que fue interrumpido por el FDC. El código que identifica la causa se encuentra en el registro de estado.

Búsqueda. Este comando permite que la cabeza del DD se coloque en un cilindro previamente especificado. El FDC determina la diferencia entre el actual cilindro y el deseado y genera los pulsos necesarios para el movimiento de la cabeza.

Recalibración. Mediante este comando se ubica a la cabeza del DD en el track 0.

Formatear un track. Este comando inicializa un cierto track, con los campos y las marcas de sincronía necesarias.

Leer datos. Con este comando se realiza la lectura de datos en un cierto sector.

Escritura de datos. Con este comando se transfiere datos hacia el disco flexible.

Lectura de datos borrados. Este comando opera de manera similar al de Lectura de datos, con la diferencia de que si, al realizarse la lectura se encuentra una marca que indica datos borrados, el proceso de lectura no se detiene, si no que continúa.

Escritura datos borrados. Este comando es similar al de Escritura de datos, con la diferencia de que los datos se escriben incluyéndoles una marca de datos borrados.

Lectura de un track. Con este comando se realiza la lectura de un track completo.

Lectura del ID. Este comando realiza la lectura del primer campo de identificación correcto en el actual track.

Comandos de búsqueda. Estos comandos realizan la comparación entre datos provenientes del CPU o del DMA y datos provenientes del DD, hasta que se cumpla alguna de las siguientes condiciones: el dato del DD es menor o igual al dato externo, o, el dato del DD es igual al dato externo, o, el dato del DD es mayor o igual al dato externo. La búsqueda se realiza hasta que la condición se cumpla, o hasta que el último sector del trakc actual sea leído.

Para llevar a cabo alguno de estos comandos, el CPU debe de obtener información del estado del FDC. Esta información se encuentra en el registro principal de estado, que tiene la siguiente descripción:

bit	descripción
0	Indica que el DD número 0 está ocupado o en modo de búsqueda.
1	Indica que el DD número 1 está ocupado o en modo de búsqueda.
2	Indica que el DD número 2 está ocupado o en modo de búsqueda.
3	Indica que el DD número 3 está ocupado o en modo de búsqueda.
4	Indica que el FDC está ocupado.
5	Indica que no se está utilizando DMA.
6	Indica la dirección Lectura/Escritura para el registro de datos.
7	Indica que el registro de datos puede ser leído o escrito.

Además de este registro, existen otros registros de estado (ST0, ST1, ST2, ST3), que sólo pueden ser leídos en la fase de resultados (result phase), y que reportan el resultado del comando ejecutado.

CAPITULO III

DESARROLLO DE LAS RUTINAS PARA DIAGNOSTICO

III.1. MODULO DEL CPU.
III.2. MODULO DEL DMA.
III.3. MODULO DE LAS MEMORIAS.
III.4. MODULO DEL TECLADO.
III.5. MODULO DEL DISCO DURO.
III.6. CONJUNCION DE LAS RUTINAS DE DIAGNOSTICO.

III. DESARROLLO DE LAS RUTINAS DE DIAGNOSTICO.

III.1 MODULO DEL CPU

a) Discusión y Análisis.

La rutina para diagnosticar al CPU tiene por objetivo determinar el buen estado general, tanto de los componentes internos del CPU como de los externos con los que interactúa (por ejemplo, los componentes lógicos que separan direcciones de datos). Este diagnóstico se puede realizar mediante la ejecución de instrucciones, sabiendo de antemano el resultado que deben producir (por ejemplo, la carga de registros o el acceso a una cierta localidad de memoria).

No obstante lo dicho arriba, es muy probable que la condición general del CPU casi siempre sea satisfactoria, por ello, el módulo del CPU debe ser pensado para que además proporcione información de la configuración del sistema e indirectamente del estado del mismo. Con esto se cumple el objetivo de proporcionar información para el mantenimiento preventivo, además de la proporcionada para el mantenimiento correctivo.

Un último punto de este análisis consiste en determinar los requerimientos del módulo a ser desarrollado. Debido a que se empleará en computadoras PCs compatibles, el módulo deberá ser desarrollado en una de ellas, y en un lenguaje que permita el acceso directo a los componentes del CPU (registros, ALU, etc.).

b) Diseño.

El módulo del CPU constará de 4 rutinas básicas, con la siguiente descripción para cada una de ellas:

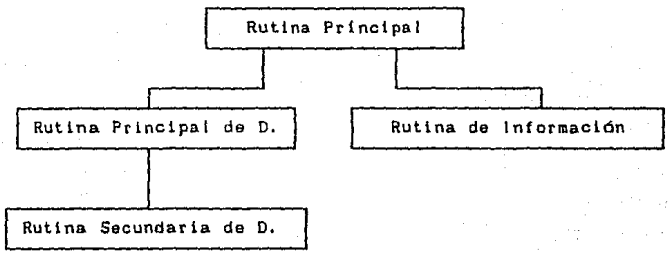
Rutina principal. Esta rutina controlará a las restantes, para que, en forma de menú, se dé la opción al usuario entre verificar al CPU y obtener información de la configuración del sistema.

Rutina principal de diagnóstico. Esta rutina realiza las instrucciones cuyos resultados se conocen de antemano, para ser verificados por la rutina secundaria de diagnóstico.

Rutina secundaria de diagnóstico. Esta rutina verifica los resultados de las intrucciones ejecutadas, registrando el o los errores si se produjeron.

Rutina de información. Esta rutina despliega la configuración del sistema, consistente en lo siguiente: número de puertos serie y paralelo, tamaño de la memoria de usuario, número de drives de disco flexible y número tipo de tarjeta de video instalada.

Diagrama Estructurado



Pseudocódigos.

```
Rutina Principal
     DespliegaLetreros('Qué opción desea?', 'Información',
'Verificar CPU').
     LeeOpción
     Si opción = Información Entonces
          Llama a Información
     SiNo
          Llama a Rutina_Principal_de_Diagnóstico
Fin Rutina_Principal.
```

```
Rutina de información
     ObténConfiguración
     DespliegaInformació
Fin Rutina_de_información.


Rutina Principal de Diagnóstico
     ax=Valor
     bx=Valor
     cx=Valor
     dx=Valor
     si=Valor
     di=Valor
     es=Valor
     bp=Valor
     Llama a Rutina_Secundaria de_Diagnóstico
     Memoria=Valor
     Si Memoria = Valor Entonces
          Transferencia_a_memoria=correcta
     SiNo
          Transferencia_a_memoria=incorrecta
Fin Rutina_Principal_de_Diagnóstico.


Rutina Secundaria de Diagnóstico
     Si ax=Valor Entonces
          RegAx=correcto
     SiNo
          RegAx=incorrecto
     Si bx=Valor Entonces
          RegBx=correcto
     SiNo
          RegBx=incorrecto
     Si cx=Valor Entonces
          RegCx=correcto
     SiNo
          RegCx=incorrecto
     Si dx=Valor Entonces
          RegDx=correcto
     SiNo
```

```
            RegDx=incorrecto
        Si si=Valor Entonces
            RegSi=correcto
        SiNo
            RegSi=incorrecto
        Si di=Valor Entonces
            RegDi=correcto
        SiNo
            RegDi=incorrecto
        Si es=Valor Entonces
            RegEs=correcto
        SiNo
            RegEs=incorrecto
        Si bp=Valor Entonces
            RegBp=correcto
        SiNo
            RegBp=incorrecto

    Fin Rutina_Secundaria_de_Diagnóstico.
```

c) Implantación.

La implantación de la rutina correspondiente al diagnóstico del CPU se realizará en un lenguaje de alto nivel, tal como lo es Pascal, en su versión Turbo para PC.

Para el despliegue de la información del sistema se requerirá utilizar la interrupción 11H del ROM BIOS que proporciona la configuración del sistema.

III.2. MODULO DEL DMA.

a) Discusión y análisis.

El papel que desempeña el DMA en una PC es el refresco de memoria (en el canal 0 del DMA) y, los accesos directos a memoria desde o hacia el Floppy Disk Controller (por el canal 2 del DMA).

Por lo tanto, el buen funcionamiento del DMA queda determinado por:

*El buen estado de los registros de lectura/escritura.

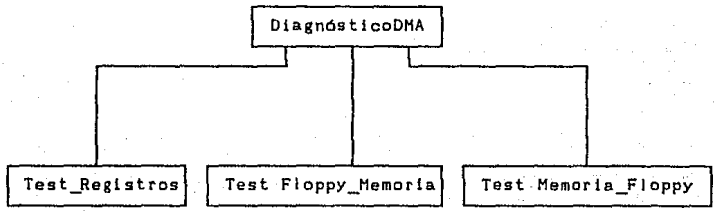
*El buen funcionamiento de la transferencia Memoria->Flopy.

*El buen funcionamiento de la transferencia Floppy->Memoria.

Para desarrollar estas tres subrutinas diagnóstico se hace necesario de un ensamblador o en el mejor de los casos un compilador que permita accesos de bajo nivel.

b) Diseño.

La rutina diagnosticadora del DMA estará formada por 3 subrutinas principales:



Módulo Test_Registros:

```
Empieza
Inicializa_Registros(Nombres);
Dibuja una pantalla adecuada;
```

```
     Desde Registro=$02 hasta $07 haz
          Empieza
          Despliega Nombre(Registro);
          Respaldo=Puerto(Registro);
          Puerto(Registro)=DatoPruebaEscritura;
          DatoPruebaLectura=Puerto(Registro);
          Puerto(Registro)=Respaldo;
          Si DatoPruebaEscritura=DatoPruebaLectura Haz
               Despliega "Registro en buen estado"
          Si No
               Despliega "Registro en mal estado"
          Fin Si;
          Termina
     Fin Desde;
     Termina
Fin Módulo;


Módulo Test_Floppy_Memoria:
     Empieza
     Asigna(Piloto,"A:Piloto");
     Abre(Piloto);
     LeeBloque(Piloto,Bloque);
     Cierra(Piloto);
     Dibuja_Memoria_Floppy;
     Anima_Transf_Floppy_Mem(Bloque);
     Anima_Comparación(Bloque);
     Termina
Fin Módulo;


Módulo Test_Memoria_Floppy:
     Empieza
     Asigna(Piloto,"A:Piloto");
     Bloque=Código ASCII;
     Abre(Piloto);
     EscribeBloque(Piloto,Bloque);
     Cierra(Piloto);
     Anima_Transf_Mem_Floppy(Bloque);
     Abre(Piloto);
```

```
    LeeBloque(Piloto,Bloque);
    Cierra(Piloto);
    Anima_Transf_Floppy_Mem(Bloque);
    Anima_Comparación(Bloque);
    Termina
Fin Módulo;
```

c) Implantación.

Para el caso de la implantación de este módulo se hará del mismo modo que en las rutinas anteriores en un lenguaje de alto nivel, para el caso presente se eligió tambien Turbo Pascal.

III.3 MODULO DE LAS MEMORIAS.

a) Discusión y análisis.

Dentro del hardware básico de un sistema PC se encuentra la memoria ,en donde se almacenan tanto datos como programas.

Existen dos clases de memoria: ROM y RAM.Como se explicó en el capitulo dos ,la memoria ROM (Read Only Memory - Memoria de sólo lectura) se utiliza para almacenar programas y datos que de forma permanente han de residir en la PC, es decir el contenido de la memoria ROM es fijo; y una vez que se ha fabricado un chip de ROM no puede cambiarse.

La memoria RAM (Random access memory - Memoria de acceso aleatorio) se utiliza para almacenar la mayor parte de los programas y datos empleados en la ejecución de un programa. El contenido de la memoria RAM si se pierde si se apaga la PC.

La PC puede tener cantidades variables de ROM y RAM.
La memoria está organizada en unidades llamadas bytes, generalmente se tienen 40 KB (Kbytes) de memoria ROM, y 640 KB de memoria RAM.El microprocesador de la PC es capaz de direccionar un total de 1,048,576 bytes de memoria (un Megabyte),incluyendo ROM y RAM.Cada byte se asocia a una sola dirección de memoria.
Las direcciones de memoria están escritas en notación hexadecimal,y pueden variar desde 00000 a FFFFF, este margen de direcciones se conoce como el espacio de direcciones de la PC, y el mapa de memoria de la PC indica como está dividido dicho espacio, el diagrama correspondiente está incluido en el capitulo dos de este trabajo.
El Espacio de dirección de la PC está dividido en bloques de 64 KB denominados segmentos.Una posición especifica de la memoria está direccionada con una direccion de segmento y un valor denominado desplazamiento (offset).
Si los segmentos son de 64 KB de longitud, los desplazamientos pueden variar desde 0 a 65535 (en decimal) o FFFF (en hexadecimal).Una vez descrita la organización de la memoria,la forma de diagnosticarla va a ser la siguiente:

Para la memoria ROM se hará la lectura de su contenido, se guardará en un archivo intermedio, se realizará de nuevo la lectura del contenido de la ROM y se comparará dicho contenido con el archivo intermedio. Si ambos son iguales la memoria ROM estará en buen estado. Se decidió hacerlo de esta manera dado que como ya se mencionó anteriormente el contenido de esta memoria es fijo.

Para la memoria RAM y la memoria denominada de Video se guardará el dato que contenga inicialmente. A continuación se va a realizar la escritura de un dato conocido en cada uno de los bytes posibles a direccionar. El dato a escribir contendrá una plantilla de unos y ceros intercalados, como se muestra a continuación:

1	0	1	0	1	0	1	0

Este dato será leído y posteriormente se comparará con lo que fué escrito. De ser iguales el dato leído y escrito, a la plantilla de unos y ceros se le aplicará un not aritmético, de tal forma que a continuación se escribirá en el siguiente byte direccionado el dato negado.

El proceso anterior se repetirá en el total de la memoria RAM de datos así como en la memoria de video.

En caso que existan diferencias entre los datos escritos y leídos se desplegará un mensaje de error en el cual se indicará el segmento y el offset del byte correspondiente.

En caso de no existir errores en la memoria se concluirá que la memoria RAM y la memoria de video se encuentran en buen estado.

b) Diseño.

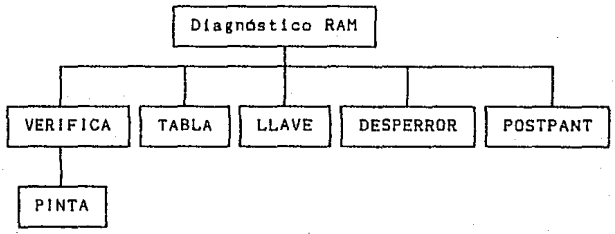
El programa de diagnóstico para la memoria RAM y para la Memoria de video está formado por los siguientes módulos:

- Una rutina principal que es la encargada de inicializar variables y llamar a cinco módulos más :
- Módulo Verifica.
- Módulo Tabla.
- Módulo Llave.
- Módulo Despliega Error.

- Módulo Pantalla Posterior.

Dentro del módulo de verifica se hace el diagnóstico de la memoria RAM y de la Memoria de video ,a su vez llama a otro módulo PINTA que es el encargado de dibujar ambas memorias, los demás módulos básicamente se utilizan para dar una presentación adecuada al diagnóstico , desplegar errores o indicar que la memoria está en buen estado.

El diagrama funcional para estos módulos es el siguiente :

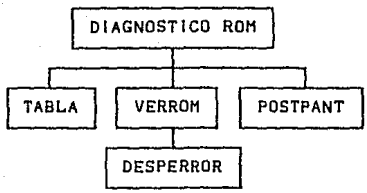


El programa Principal para el diagnóstico de la memoria ROM consta de los siguientes módulos :

- Una rutina Principal que se encarga de llamar a tres módulos más:
- Módulo VerROM.
- Módulo Tabla.
- Módulo Pantalla Posterior.

El módulo de VERROM es el que realiza el diagnóstico de la memoria ROM, a su vez llama a una rutina que despliega los errores en caso que los haya(DESPERROR), las otras rutinas son para dar la presentación adecuada al diagnóstico, ya que dibujan la memoria y despliegan el mensaje correspondiente al buen funcionamiento de la memoria.

El diagrama funcional para estos módulos es el siguiente :



c) Implantación.

La implantación de los módulos anteriores se hará en un lenguaje de programación de alto nivel que en este caso será Turbo Pascal,con sus servicios de interrupciones correspondientes a la memoria, así como los servicios de graficación para la presentación de este diagnóstico.

III.4 MODULO DEL TECLADO.

a) Discusión y Análisis.

El teclado es sin duda el dispositivo de entrada de datos en línea más importante en una PC. Aún en caso de que el mouse llegase a popularizarse, por su relativo bajo precio, por el número que existe actualmente y por su rapidez de manejo el teclado será todavía por un tiempo largo de importancia vital.

Sin embargo, el teclado es un dispositivo de comunicación unidireccional. Esto es, la computadora puede obtener datos e instrucciones a través de él, pero la única instrucción al teclado está restringida a la reinicialización (RESET).

Esta reinicialización puede ser por hardware (forzando a alto la señal RESET que va al teclado) o por software (vía un comando que le indique que se consideren nuevas condiciones).

Debido a esta restricción (comunicación virtualmente unidireccional del teclado), la forma en que se puede diagnosticar es mediante la ayuda del usuario. Se puede indicar al usuario un patrón o secuencia de teclas que debe pulsar, la computadora compara que el código que esperaba de esa tecla es el que le está llegando, entonces la comunicación es correcta.

Una variante es que el usuario pulse las teclas que crea están funcionando mal. La computadora le mostrará (vía el monitor) la tecla correspondiente al código que está recibiendo. Con esto el usuario verá como decodifica la computadora las teclas y sabrá si la comunicación es correcta. En este trabajo se prefirió el segundo tipo de verificación.

En el momento en que se pulsa una tecla, el teclado (a través del circuito 8048) genera una interrupción y coloca en el bus de datos un código (llamado comunmente scan code), de acuerdo a la tecla pulsada. Esta interrupción es la número 09h de software. Lo más conveniente es interceptar esta interrupción, y de acuerdo al scan code hacer saber al usuario la tecla que la computadora hace corresponder a ese código.

Existen actualmente una gran variedad de tipos de teclados. Muchos de ellos contienen muchas más teclas que las que contenía

el teclado original de la PC. Entre estas podemos citar las teclas de Pause, Sys Req, Break, un juego adicional de teclas para movimiento del cursor, etc.

Al no ser estas teclas del juego original de la PC, en algunos casos se presentan problemas de interpretación del scan code, es por eso que se optó en este trabajo por omitirlas, y en caso de que el usuario las oprimiera, estas teclas no serán desplegadas en el monitor.

b) Diseño.

Como se acostumbra, se requiere de un módulo que realice las tareas de inicialización, a este se le llamó INICIO. Al estar trabajando con el lenguaje de alto nivel Turbo Pascal, estas inicializaciones corresponden a la apertura de archivos que contienen rutinas de servicio. El pseudocódigo no es representativo, por tratarse de instrucciones muy específicas.

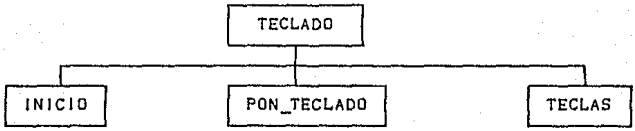
Para obtener la imagen de las teclas que el usuario pulsará se hizo necesario utilizar otro módulo, llamado PON_TECLADO. Este módulo realiza el posicionamiento en un área específica del monitor y coloca un carácter, de acuerdo a la disposición del teclado de una PC.

El tercer módulo es TECLAS, que realiza lo siguiente:

- Intercepta la interrupción 09h.
- Interpreta el scan code interceptado.
- De acuerdo a este código muestra en pantalla la tecla correspondiente al mismo.
- Regresa el control a la interrupción 09h, para que se realice lo que corresponda.
- Reinicializa el teclado para eliminar el "encolamiento" de teclas pulsadas, que traería situaciones indefinidas después que terminara este módulo.

El módulo principal (TECLADO) es el que realiza las llamadas a estos otros módulos. La forma en que se termina de emplear es presionar dos veces consecutivas la tecla Scroll Lock.

El diagrama funcional de estos módulos es:



c) Implantación.

De acuerdo a lo mencionado anteriormente, las rutinas que componen el módulo de diagnóstico del teclado se implantarán en lenguaje de programación de alto nivel Turbo Pascal, con interfaces de ensamblador para deshabilitar las rutinas de tratamiento de las interrupciones que manejan el funcionamiento del teclado.

Es de hacer notar que el funcionamiento de las rutinas es bastante robusto, esto es, las instrucciones empleadas son de uso específico, y lo único que se requiere es realizar las interfaces de graficación para mostrar las teclas que van siendo interpretadas por el programa.

III.5. MODULO DEL DISCO DURO.

En este inciso se ofrecen tres módulos, dos de los cuales son rutinas de utilería para el disco y la tercera es una rutina de mantenimientodel mismo:

a) Compresión y descompresión de archivos.

b) Encriptamiento y desencriptamiento de archivos.

c) Chequeo de la FAT (File Alocation Table).

a) Compresión y descompresión de archivos.

Para la compresión de archivos es necesario encontrar un código que permita guardar los caracteres de tal forma que ocupen menos espacio en bits que los que corresponden a cada carácter (8 bits en código ASCII).

Se hará uso por tanto, del método de optimización de Hamming, el cual permite encontrar este código óptimo basándose en la frecuencia con que aparece cada carácter. Esto es, si un carácter aparece muchas veces es necesario codificarlo con pocos bits, mientras que un carácter que pocas veces aparece debe codificarse con un mayor número de bits (posiblemente más de 8).

En un principio se hará un rastreo en el archivo a comprimir para encontrar la frecuencia con que se presentan los caracteres. Una vez hecho esto se encuentra el código óptimo para ese juego de caracteres.

Con este nuevo código se leerá de nuevo el archivo, pero al ir tomando un carácter se escribirá en otro archivo con el nuevo código equivalente a ese carácter.

El nuevo archivo debe tener un formato especial que permita saber que es un archivo que ha sido comprimido. Se propone el siguiente formato:

- Los tres primeros bytes serán el carácter nulo, esto indicará que se trata de un archivo que ha sido comprimido.

- El siguiente carácter será el correspondiente al número de compresión que se ha hecho a ese archivo, ya que es posible que a un mismo archivo se le aplique más de una compresión.

- Los siguientes 256 pares de bytes serán los correspondientes a las frecuencias de los caracteres ASCII que se dieron en el archivo original. Esto es necesario para poder descomprimir el archivo, debido a que el método de Hamming se basa en probabilidades que corresponderán a cada archivo.

- De ahí en adelante se encontrará el archivo codificado en el nuevo código.

Es de hacer notar que en este módulo se tomará la decisión de si es conveniente realizar la compresión, dependiendo de si ahorra o no espacio en disco.

En un principio se puede probar el módulo con archivos en Pascal u otro lenguaje, así como en bases de datos, ya que es notoria la frecuencia con que aparece el carácter 32 (espacio en blanco).

Como se mencionó anteriormente, el método de optimización de código de Hamming es de tipo estadístico. La forma práctica de obtener dicho código se muestra en el siguiente ejemplo:

Supóngase la probabilidad de aparición de los caracteres siguientes:

- El carácter "a" aparece en 45% de las veces.
- El carácter "b" aparece en 25% de las veces.
- El carácter "c" aparece en 12% de las veces.
- El carácter "d" aparece en 10% de las veces.
- El carácter "e" aparece en 6% de las veces.
- El carácter "f" aparece en 2% de las veces.

Los caracteres se colocan de manera descendente, dependiendo de la frecuencia con que aparecen; a los dos caracteres de menor frecuencia se les suman sus frecuencias y se vuelve a acomodar la lista de caracteres, tomando la unión anterior como si se tratara del mismo carácter:

a 45%	a 45%
b 25%	b 25%
c 12%	c 12%
d 10%	d 10%
e 6%	ef 8%
f 2%	
antes del paso anterior	después del paso anterior

Se realiza una nueva asociación de los caracteres de menor frecuencia y se repite esto hasta que se tiene la unión de todos los caracteres.

a 45%	a 45%	a 45%	a 45%	defcb 55%	defcba 100%
b 25%	b 25%	b 25%	defc 30%	a 45%	
c 12%	c 12%	def 18%	b 25%		
d 10%	d 10%	c 12%			
e 6%	ef 8%				
f 2%					

Se hace un proceso repetitivo inverso, en el cual se deshace la cadena final, desglozándola en sus dos últimos componentes. A la parte que queda posicionada en la parte superior se le asigna un cero y a la que queda en la parte inferior se le asocia un uno:

defcba	defcb 0	a 1	a 1	a 1	a 1
	a 1	defc 00	b 01	b 01	b 01
		b 01	def 000	c 001	c 001
			c 001	d 0000	d 0000
				ef 0001	e 00010
					f 00011

Con lo anterior, la forma óptima de codificar la letra "a" será con un bit, de valor uno. De forma contraria, para codificar un carácter que aparece muchas veces se utiliza un número mayor de bits, como en el caso de la letra f, para la cual se necesitarán cinco bits.

De esta forma se generará un código equivalente al código ASCII que será con el que se leerá el archivo a comprimir, se tendrá entonces el carácter equivalente y será este el que se escriba en el nuevo archivo.

Para la descompresión de archivos se realizará la lectura previa del archivo, para ver si es un archivo comprimido. de ser así se leerá la frecuencia de los caracteres, esta información se encuentra en el encabezado de un archivo que ha sido comprimido y se usará para generar el código equivalente que se obtuvo en a la compresión.

Una vez que se ha identificado el archivo como comprimido, se leerá un carácter y se tratará de encontrar su equivalente en el código ASCII. En cuanto se encuentre este código ASCII

equivalente, se escribirá en un archivo. Esto continuará hasta terminar de descomprimir el archivo.

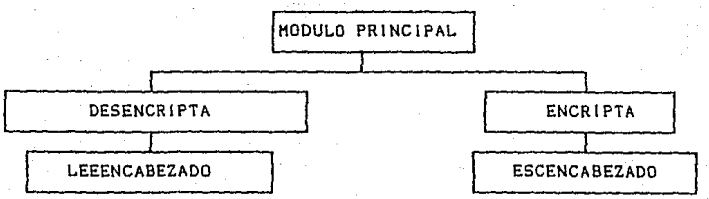
b) Encriptamiento y desencriptamiento de archivos.

Esta rutina tiene por objetivo realizar un encriptamiento o desencriptamiento de archivos, de manera masiva o selectiva. El usuario podrá seleccionar el disco o archivo que quiere encriptar o desencriptar.

El proceso de encriptamiento es muy sencillo: se elige al azar los elementos de una llave o clave cuya longitud tambien es aleatoria, y se realiza con esta clave la operación XOR sobre todos y cada uno de los bytes del (los) archivo(s) a encriptar. Además de esto, en el archivo resultante se agrega una cierta información consistente en la descripción del tamaño de la clave y de todos los elementos de la misma, este agregado se encripta a su vez siguiendo la misma técnica pero con una clave de encriptamiento que es constante. Mediante esta información es posible llevar a cabo el proceso de desencriptamiento.

El proceso de desencriptamiento es muy similar al del encriptamiento, debido a las propiedades de la operación XOR. En primer lugar se averigua si el archivo fue realmente encriptado o no (lo cual también es parte de la información agregada en el encriptamiento); si es un archivo encriptado, se lee la información adicional que se agregó en el encriptamiento, lo cual permite conocer la clave usada y realizar la operación XOR nuevamente sobre todos los bytes del (los) archivo(s). El resultado es la obtención del (los) archivo(s) original(s).

Diagrama de estructura de la rutina de encriptamiento y desencriptamiento.



Descripción de rutinas.

Las rutinas de EscEncabezado y LeeEncabezado genera y lee, respectivamente la información utilizada en el encriptamiento y que es requerida para el desencriptamiento.

Por otro lado, las rutinas de Encripta y Desencripta realizan las funciones arriba descritas apoyándose en las dos anteriores rutinas.

c) Verificación de la Tabla de Alojamiento de Archivos (FAT).

Esta utileria se basa en lo siguiente:

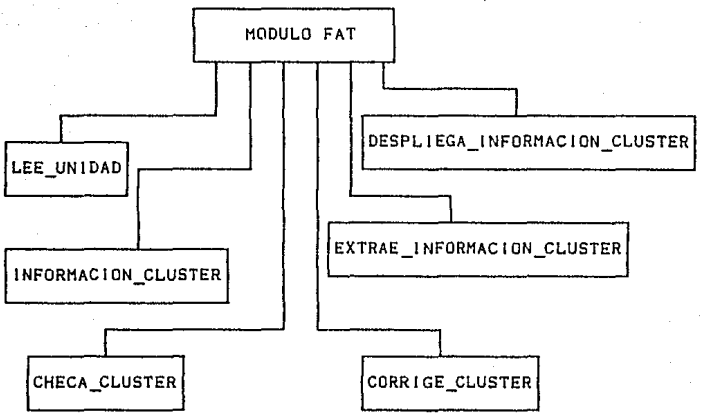
-Se extrae la información del disco a través de una lectura al sector cero y mediante la interrupción 21H con el servicio de la función 1CH. Dicha información es procesada y proporcionada al usuario.

-Se hace un recorrido de la FAT checando el estado de cada cluster y en caso de encontrar uno marcado como dañado se le aplica el siguiente procedimiento: primero se checa que efectivamente el cluster está dañado, esto se hace a través de una escritura y lectura absoluta a los sectores de dicho cluster; y en segundo lugar se le indica al usuario el cluster marcado como dañado así como la factibilidad de su recuperación.

Descripción del programa.

Este programa esta desarrollado en un lenguaje de alto nivel (Turbo Pascal) con accesos al lenguaje ensamblador (Turbo Asambler).

El cual consta de las siguientes rutinas:



Descripción de los procedimientos:

MODULO FAT: Esta rutina se encarga de coordinar el funcionamiento de los demás procedimientos que le preceden.

LEE UNIDAD: Este procedimiento se encarga de preguntarle al usuario la unidad en que se va a trabajar y depositar el resultado en la variable global UnidadDisco.

INFORMACION CLUSTER: Esta función recibe como entrada el número de cluster y da como salida el estado de dicho cluster (16 bits).

CHECA CLUSTER: Esta función recibe de entrada el número de cluster y después hace escrituras y lecturas absolutas a los sectores correspondientes a dicho cluster en donde determina si realmente el cluster está dañado. Esta función regresa una varible booleana en donde indica la posible recuperación del cluster.

CORRIGE CLUSTER: Este procedimiento se encarga de corregir en la FAT el buen estado del cluster diagnosticado.

EXTRAE INFORMACION CLUSTER: Este procedimiento tiene como función el obtener las características principales del disco auxiliandose del sector cero y con la interrupción 21H (función 1CH).

DESPLIEGA INFORMACION CLUSTER: Este procedimiento despliega al usuario la información obtenida por el módulo anterior.

III.6. CONJUNCION DE LAS RUTINAS DE DIAGNOSTICO.

Para el control de las rutinas de diagnóstico se desarrollará un programa de tipo interactivo y amigable al usuario. Dicho programa se realizará en Turbo Pascal y se denominará DIAGVMS, por medio del cual se podrá tener acceso a cada una de las rutinas de diagnóstico, de acuerdo a un menú gráfico.

Para la información del manejo y características del programa refiérase al apéndice B "Manual de usuario del sistema de Diagnóstico".

CAPITULO IV

DESARROLLO DEL HARDWARE PARA EL DIAGNOSTICO

IV.1. DISCUSION DEL PROBLEMA.
IV.2. DISEÑO DE LA TARJETA DE DIAGNOSTICO.
IV.3. DESARROLLO DEL SOFTWARE PARA EL CONTROL DE LA TARJETA.

IV. DESARROLLO DEL HARDWARE PARA EL DIAGNOSTICO.

IV.1 DISCUSION DEL PROBLEMA

a) Necesidad de la tarjeta

Para diagnosticar el funcionamiento de una PC se requiere verificar que los componentes de ésta realicen su función correctamente, es decir, generen las señales adecuadas en el tiempo correcto para realizar el procesamiento requerido de la información. Existen dos maneras principales para verificar que esto realmente está sucediendo. La primera manera consiste en hacer operaciones mediante rutinas (software) que impliquen el uso de varios de estos elementos, para posteriormente comparar los resultados obtenidos contra resultados que se esperarían obtener. Precisamente este es el objetivo de las rutinas diseñadas en el capítulo III.

La segunda forma de verificar el funcionamiento del hardware de la computadora implica el manejo de las señales físicas generadas por éste, ya no tanto basándose en los resultados obtenidos de las operaciones efectuadas. Para manejar señales del hardware es necesario crear dispositivos apropiados para ello. Estos dispositivos pueden ser analógicos (por ejemplo, un osciloscopio), o digitales, en cuyo caso se estaría hablando de un hardware diagnosticador. La ventaja de poseer a éste consiste en el hecho de que es posible diagnosticar sin intermedio alguno, y en el nivel más bajo posible, a un cierto dispositvo o componente de la PC, situación que no se da si el diagnóstico es por software únicamente.

b) Funciones específicas de la tarjeta

La tarjeta dignosticadora tiene las siguientes funciones:

1) Determinar la frecuencia exacta del reloj con la que una PC está trabajando.

ii) Determinar el funcionamiento correcto del puerto paralelo Centronics para impresora, verificando tanto la correcta transmisión de los datos, como la correcta secuencia de señales que deben producirse para realizar la transmisión de los datos.

iii) Determinar el funcionamiento correcto del puerto serial RS-232C, verificando tanto la correcta transmisión y recepción de los datos, como la correcta secuencia de señales que deben producirse para realizar la transmisión y recepción de datos.

iv) Determinar el funcionamiento correcto del manejador del disco flexible, verificando tanto la correcta transferencia de datos desde y hacia el periférico, como la secuencia de señales que deben producirse para realizar la transferencia de datos.

c) Listado global de los componente de la tarjeta

La tarjeta estará compuesta de varios componentes descritos a continuación:

Para realizar la comunicación con el exterior se poseerán 4 puertos conectados al bus de datos de la PC, uno de ellos será de sólo escritura, mientras que los otros 3 serán de sólo lectura.

Se requieren memorias RAM y ROM para la ejecución de los programas de diagnóstico.

Asociada a los puertos y a las memorias se necesita de una lógica de decodificación y una lógica de control.

El controlador interno de la tarjeta (puesto que no debe de depender del procesador de la PC), será un CPU sencillo elegido para ese fin.

La tarjeta contendrá el hardware necesario para realizar el manejo específico de las señales cuando se efectue el diagnóstico del reloj, puertos serie y paralelo, y controlador disco flexible.

Asociada al hardware de manejo de señales se tendrá una lógica de control.

d) Comunicación con el exterior (interface Tarjeta-PC)

Por hardware la comunicación se hará a través de puertos conectados al bus de datos. Estos puertos serán 4, de los cuales uno será de sólo escritura, a través del cual la PC enviará el comando necesario para que la tarjeta efectúe una operación determinada de diagnóstico. Los otros 3 puertos serán de sólo lectura, y permitirán a la PC obtener información del estado de la tarjeta y del resultado de la operación pedida (puerto de Estado), e información complementaria de la operación (puertos de Datos).

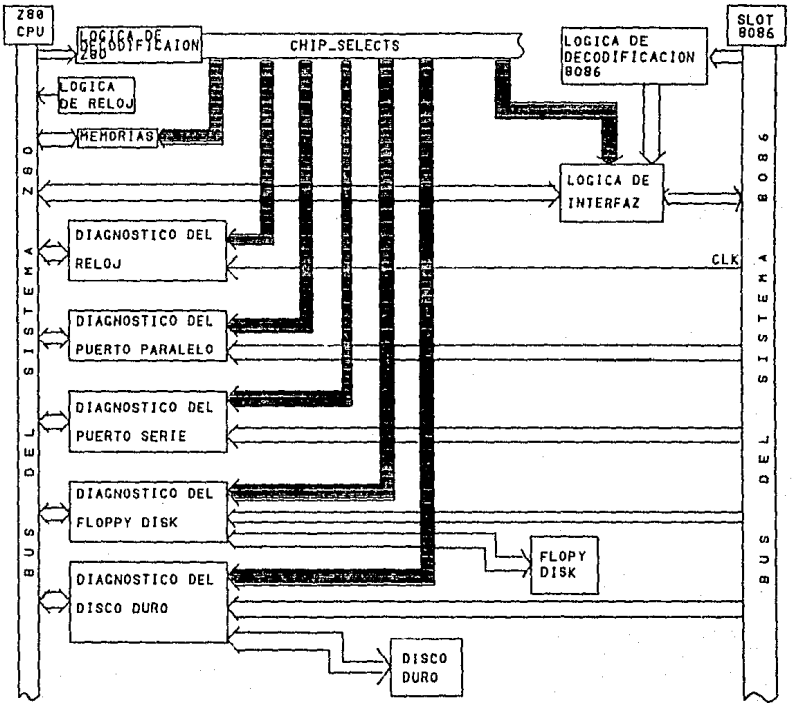
En cuanto al software se refiere, deberán de existir rutinas para ser ejecutadas por la PC, que permitan enviar comandos de operación a la tarjeta, recibir la información proporcionada por ella y procesarla para transmitir al usuario un resultado final del diagnóstico.

e) Control interno

Para realizar el control interno de la tarjeta se anexarán en ella diferentes lógicas de control para los diferentes hardwares de diagnóstico y de comunicación, estas lógicas de control serán comandadas por el CPU de la tarjeta mediante diferentes rutinas en ensamblador almacenadas en una memoria ROM.

IV.2. DISEÑO DE LA TARJETA DE DIAGNOSTICO.

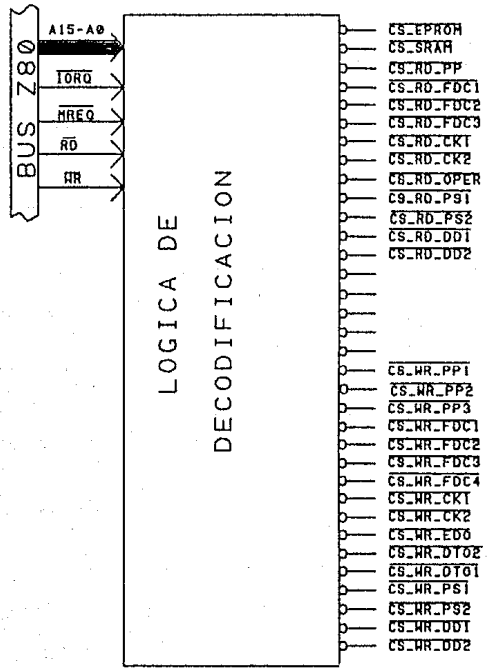
DIAGRAMA DE BLOQUES DE LA TARJETA DE DIAGNOSTICO



LOGICA DE DECODIFICACION Z80

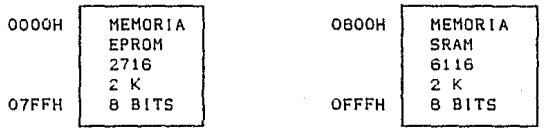
Esta lógica recibe como entrada al bus de direcciones y las señales (IORQ, MREQ, RD, WR) del Z80CPU, dando como salidas a un conjunto de lineas (chip selects) que se encargan de seleccionar a los circuitos perifericos del sistema.

Diagrama a Bloques:



Está lógica de decodificación define el mapa de memoria y puertos de la tarjeta diagnosticadora:

MAPA DE MEMORIA:



MAPA DE PUERTOS:

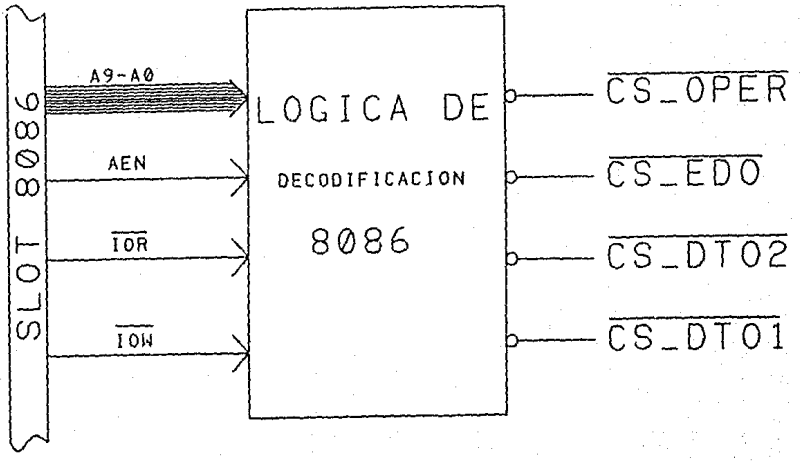
PUERTOS DE LECTURA		
PTO	SEÑAL	DESCRIPCION
00H	CS_RD_PP	Utilizado por el diagnóstico del puerto paralelo
01H	CS_RD_FDC1	Utilizado por el diagnóstico del floppy disk
02H	CS_RD_FDC2	Utilizado por el diagnóstico del floppy disk
03H	CS_RD_FDC3	Utilizado por el diagnóstico del floppy disk
04H	CS_RD_CK1	Utilizado por el diagnóstico del reloj
05H	CS_RD_CK2	Utilizado por el diagnóstico del reloj
06H	CS_RD_OPER	Utilizado por la lógica de interfaz
07H	CS_RD_PS1	Utilizado por el diagnóstico del puerto serie
08H	CS_RD_PS2	Utilizado por el diagnóstico del puerto serie
09H	CS_RD_DD1	Utilizado por el diagnóstico del disco duro
0AH	CS_RD_DD2	Utilizado por el diagnóstico del disco duro
0BH	ninguna	Libre
0CH	ninguna	Libre
0DH	ninguna	Libre
0EH	ninguna	Libre
0FH	ninguna	Libre

PUERTOS DE ESCRITURA		
PTO	SEÑAL	DESCRIPCION
00H	CS WR PP1	Utilizado por el diagnóstico del puerto paralelo
10H	CS WR PP2	Utilizado por el diagnóstico del puerto paralelo
20H	CS WR PP3	Utilizado por el diagnóstico del puerto paralelo
30H	CS WR FDC1	Utilizado por el diagnóstico del floppy disk
40H	CS WR FDC2	Utilizado por el diagnóstico del floppy disk
50H	CS WR FDC3	Utilizado por el diagnóstico del floppy disk
60H	CS WR FDC4	Utilizado por el diagnóstico del floppy disk
70H	CS WR CK1	Utilizado por el diagnóstico del reloj
80H	CS WR CK2	Utilizado por el diagnóstico del reloj
90H	CS WR EDO	Utilizado por la lógica de interfaz
A0H	CS WR DTO2	Utilizado por la lógica de interfaz
B0H	CS WR DTO1	Utilizado por la lógica de interfaz
C0H	CS WR PS1	Utilizado por el diagnóstico del puerto serie
D0H	CS WR PS2	Utilizado por el diagnóstico del puerto serie

LOGICA DE DECODIFICACION 8086

La lógica de decodificación del 8086 está conformada por circuitos lógicos combinacionales cuya función es proporcionar líneas para seleccionar 4 puertos de la PC. Dichos puertos serán utilizados por la lógica de interfaz.

Diagrama a Bloques:



Observando cuidadosamente el mapa de puertos de una PC se determinan como libres al 220h, 221h, 222h y 223h, los cuales están definidos de la siguiente forma:

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
1	0	0	0	1	0	0	0	*	*	220H..223H

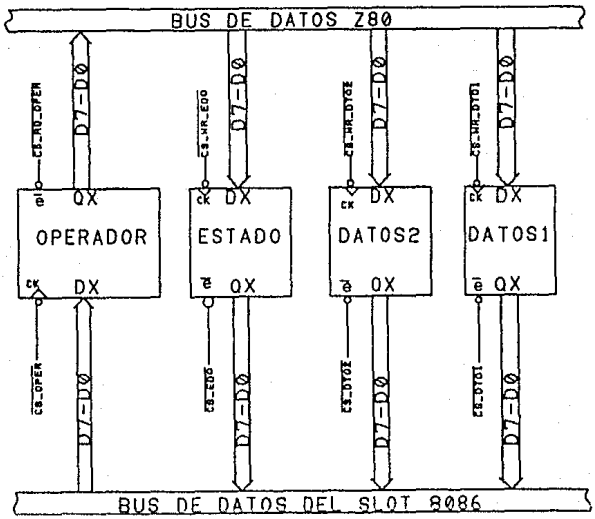
Definiendose entonces el siguiente mapa de puertos del 8086:

PTO	SEÑAL	DESCRIPCION
220H	CS_OPER	Define la operación que desarrollará la tarjeta
221H	CS_EDO	Defina el diagnóstico del dispositivo
222H	CS_DTO2	Dato relacionado con el estado del dispositivo
223H	CS_DTO1	Dato relacionado con el estado del dispositivo

LOGICA DE INTERFACE

La lógica de interfaz está formada por cuatro circuitos de tipo latch, de 8 bits y con salidas tres estados. Su función es comunicar el Z80CPU con el 8086.

Diagrama a Bloques:

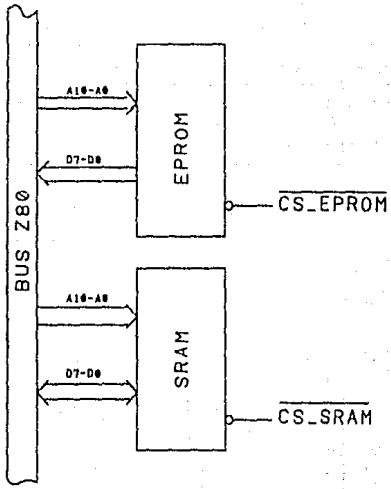


MEMORIAS

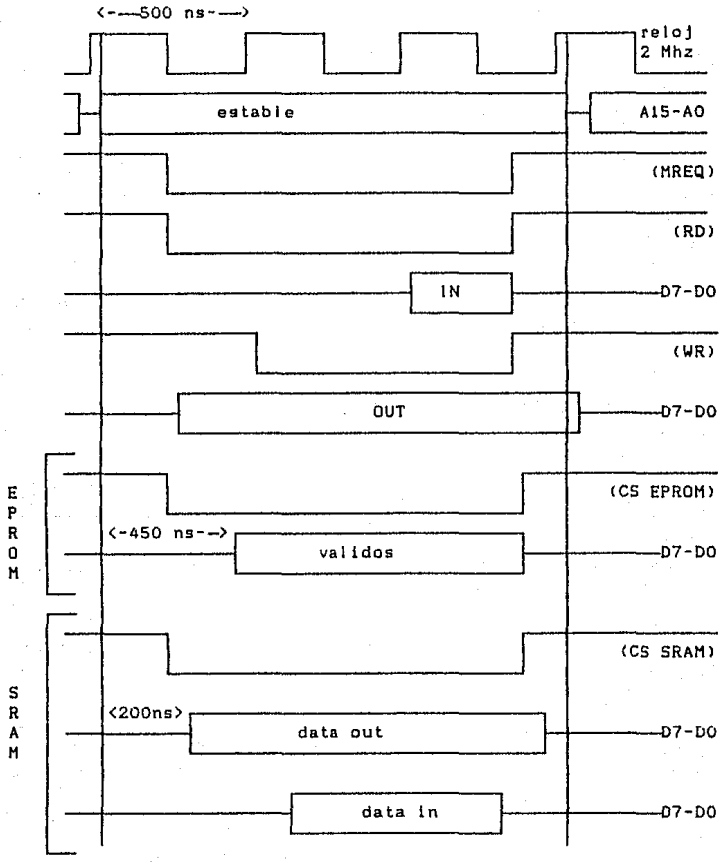
Este bloque está constituido por 2 memorias:

* Memoria EPROM 2716, cuya organización es de 2K por 8 bits.
* Memoria SRAM 6116, cuya organización es de 2K por 8 bits.

Diagrama a Bloques:



ANALISIS DINAMICO:



En base al diagrama anterior se determina que las memorias ya mencionadas se acoplan perfectamente al Z80CPU, sin la necesidad de insertar estados de espera.

a) Diseños parciales del hardware.

i) Puerto Serie.

Diagnóstico de transmisión.

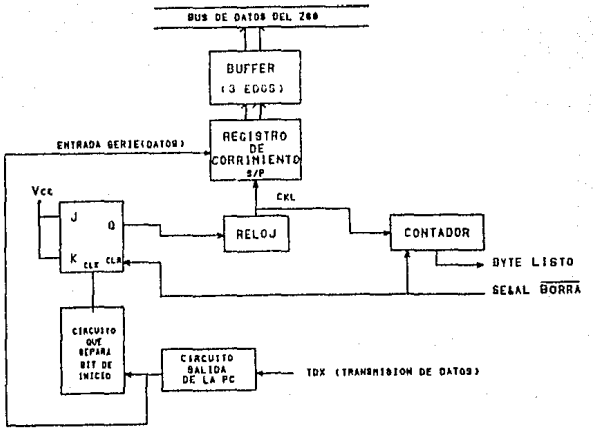
El funcionamiento de este puerto fue descrito en el capitulo segundo del presente trabajo. Básicamente el diagnóstico de transmisión consiste en enviar un dato conocido desde la interfaz RS-232 hacia la tarjeta de diagnóstico.

- El 8086 indicará a la tarjeta de diagnóstico que el puerto serie va a ser diagnosticado, posteriormente se inicializará el puerto serie y se preparará para transmitir un dato.

. - El 8086 esperará una señal del Z80 que le indique que ya recibió el dato.

- En el momento en que el Z80 reciba el dato, será colocado en el puerto de resultados para que sea cotejado posteriormente por el 8086.
- Una vez hecha esta comparación, se determinará si el estado del puerto serie es el correcto.

- Se tendrá el siguiente diagrama de hardware para realizar el diagnostico del puerto serie (Transmisión):



Las señales que se manejarán por medio del hardware son:

TXD : Terminal de salida mediante la cual se envían los datos de la transmisión.

RXD: Terminal de entrada en la cual se recibe el dato de la transmisión.

GND: Tierra.

Se reciben los datos de la interface RS-232 por medio de la señal TDX, que va directamente al circuito de salida, dicho circuito convierte un voltaje de -12 v a 5 v y de 12 v a 0 v. Existe a continuación una lógica que permitirá al bit de inicio activar el reloj, el cual maneja una velocidad de transmisión de 4800 bauds, este reloj activará un contador y a la vez a un registro de corrimiento (Serie / Paralelo), con lo que se empiezan a cargar los datos en el registro de corrimiento, y el contador indicará cuando se haya completado el dato enviado (8 bits), lo cual permitirá que el Z80 obtenga el dato.

Descripción del programa en Z80.

Los puertos asignados para el diagnóstico del puerto serie son:

07H y 08H (puertos de lectura)
C0H y D0H (puertos de escritura).

El programa realizará lo siguiente:

- El Z80 esperará el dato enviado por el 8086.
- Cuando se dé la señal de "byte listo" (puerto de edos), se habrá recibido el dato del 8086.
- El Z80 indicará que ha recibido el dato, y una vez recibido lo colocará en el puerto de resultados.
- Fin del programa.

Descripción del programa en Pascal.

Los puertos para la operación de ambos programas son:

0220 H (Puerto de Comandos)
0221 H (Puerto de Estado).
0223 H (Puerto de Resultados 1).
0222 H (Puerto de Resultados 2).

El programa realizará lo siguiente :

Se inicializa el puerto serie y se envía po él un dato conocido.
Se recibe un código que indica si llegó o no el dato al Z80.
Si el 8086 nunca recibe dicho código envía un mensaje de error.
Se pregunta si el dato recibido es igual al dato enviado.
Si no es el correcto, envía mensaje de "ERROR DE TRANSMISION".
Si es el correcto, el puerto serie se encuentra funcionando correctamente.
Si no ha sido instalada la tarjeta de diagnóstico se despliega un mensaje indicándolo.
Fin del programa.

Diagnóstico de Recepción.

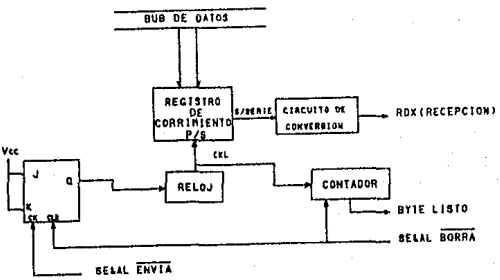
El diagnostico de recepción consiste en enviar un dato conocido desde la tarjeta de diagnóstico hacia la interface RS-232.

- El 8086 indicará a la tarjeta de diagnóstico que el puerto serie va a ser diagnosticado, posteriormente se inicializará el puerto y se preparará para recibir un dato.

- El 8086 verificará si hubo o no algun error en la recepción

- En el momento en que el 8086 reciba el dato, cotejará si el dato recibido es igual al dato transmitido (Z80) y se determinará el correcto estado del puerto serie.

- Se tendrá el siguiente diagrama de hardware para realizar el diagnostico del puerto serie (Recepción):



El dato se cargará al registro de corrimiento (Paralelo / Serie), por medio de la habilitación de la señal ENVIA, previamente con la señal BORRA, se han inicializado las condiciones para que el circuito empiece a funcionar correctamente, el contador indicará cuando se haya completado el dato a transmitir (8 bits)- señal de " Byte Listo "- con lo que la tarjeta de diagnóstico considera que ya se ha hecho la transmisión del dato, este pasará a través de un circuito de conversión de voltaje, que directamente irá a la señal RDX de la interface RS-232.

Descripción del programa en Z80.

Los puertos asignados para el diagnóstico del puerto serie son:

07H y 08H (puertos de lectura)
C0H y D0H (puertos de escritura).

El programa realizará lo siguiente:

- El Z80 mandará un dato al 8086.
- Cuando se dé la señal de "byte listo" (puerto de lectura), el Z80 habrá realizado la transmisión.
- El Z80 avisará al 8086 que esta parte del diagnóstico ha terminado.
- Fin del programa.

Descripción del programa en Pascal.

Los puertos para la operación de ambos programas son:

0220 H (Puerto de Comandos)
0221 H (Puerto de Estado).
0223 H (Puerto de Resultados 1).
0222 H (Puerto de Resultados 2).

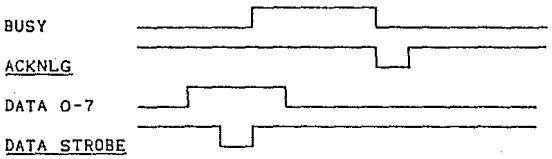
El programa realizará lo siguiente :

Se inicializa al puerto serie y se programa para recibir un dato.
El 8086 conoce el dato que va a ser transmitido por la tarjeta de diagnóstico.
El 8086 verificará si existió o no error en la recepción del dato.
Si no hubo error el 8086 verificará si el el dato recibido fue igual al transmitido por la tarjeta de diagnóstico.
Si no es el correcto, envía mensaje de "ERROR DE RECEPCION ".

Si es el correcto, el puerto serie se encuentra funcionando correctamente.
Si no ha sido instalada la tarjeta de diagnóstico se despliega un mensaje indicándolo.
Fin del programa.

ii) Puerto Paralelo.

A la salida del puerto paralelo, cuando se maneja el envío de un solo dato se tiene un diagrama de tiempos como el siguiente:



Las señales BUSY y ACKNLG van del dispositivo externo a la computadora y las señales DATA 0-7 y DATA STROBE en sentido inverso. La señal BUSY puede colocarse en bajo si sólo se va a recibir un dato, pues su objetivo es indicar a la computadora que la memoria de almacenamiento del periférico está llena (con la señal BUSY en alto). La señal ACKNLG a su vez indica a la computadora que el dato fue recibido. La señal DATA STROBE indica al dispositivo que puede obtener un dato a través de las líneas DATA 0-7.

La forma de diagnosticar el puerto paralelo será enviar un dato conocido, recibirlo en la tarjeta de diagnóstico e indicar al procesador de la computadora el dato recibido.

Aunado a esto, se pueden introducir atenuaciones en la línea, con el fín de conocer que tan inmune es el puerto paralelo a estas atenuaciones.

Los pasos a seguir para el diagnóstico son los siguientes:

- El 8086 indicará a la tarjeta que se quiere diagnosticar el puerto paralelo, para ello colocará un dato en el puerto de comandos descrito en la sección IV.1.

- El Z80 guardará el estado de un flip-flop T, para que en cuanto éste cambie de estado sepa este microprocesador que se recibió el dato .

- El Z80 eligirá un grado de atenuación a producir en la línea e indicará al 8086 que está listo para recibir el dato.

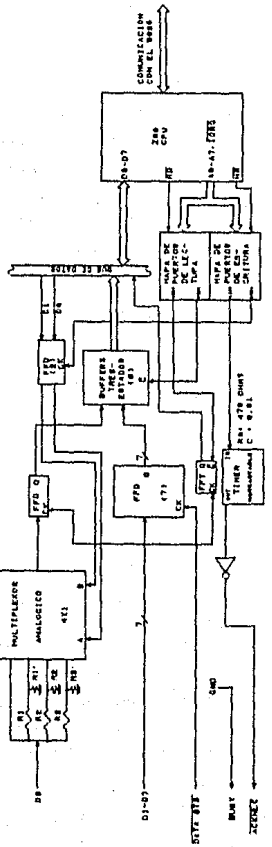
- Hasta que no reciba el dato (mediante el cambio de estado del flip-flop T), el Z80 estará verificando la recepción y el

8086 el puerto de comandos, para saber si ha transcurrido un segundo, lo que indicaría que la transmisión ha tardado más de lo normal.

- En el momento en que detecte que el dato se ha recibido lo enviará al 8086 para que sea procesado.

- Si no se han realizado cuatro chequeos de la línea, se procederá de nuevo desde el paso dos.

El diagrama del hardware para el diagnóstico del puerto paralelo es el siguiente:



De acuerdo al anterior diagrama, para elegir el grado de atenuación el Z80 direccionará al puerto 00h de su mapa de salida y colocará en sus líneas de datos D0 y D1 uno de los cuatro grados de atenuación que se desee. Como se aprecia en el diagrama, la línea D0 que se recibe del puerto paralelo deberá ser siempre 1, para que opere la atenuación.

Si direcciona al puerto 00h de lectura, el Z80 estará obteniendo el dato contenido en los flip-flop D. Esto se realizará una vez que el microprocesador sense un cambio de estado en el flip-flop T que indica transmisión recibida. Como se puede ver en la anterior figura, una entrada del multiplexor analógico es la entrada D0 directa de la terminal Centronix, esto con el fin de verificar D0 en la forma como se está obteniendo.

El puerto 08h de lectura le sirve al Z80 para obtener el estado del flip-flop T que indica transmisión recibida.

El puerto 01h de salida sirve para generar la señal ACKNLG (el cual debe tener de 5 a 6 µs de duración) que indica a la computadora que el dato fué recibido. Esta señal será obtenida por medio de un circuito Timer 555 en modo monoastable (la fórmula para el ancho del pulso es: Tw=1.1 Ra C).

Como se mencionaba anteriormente, la señal BUSY se proporciona baja, pues sólo se recibirá un dato.

Programa para el 8086:

a) Reseña:

El programa se realizará en lenguaje de alto nivel (Pascal) y realizará lo siguiente:

1. Alertará al microprocesador Z80 contenido en la tarjeta de diagnóstico para que espere un mensaje.

2. Mandará el mensaje a través del puerto paralelo, el cual contendrá una cadena de ceros y unos intercalados, teniendo en cuenta que el bit D0 del puerto paralelo debe ser siempre uno (para poder diagnosticar con atenuaciones).

3. Esperará un tiempo razonable (un segundo) para que el Z80 tenga oportunidad de recibir y procesar el mensaje recibido.

4. Si el Z80 indica "transmisión recibida", se cotejará el resultado de la transmisión, esto es, si todos los bits de la transmisión fueron recibidos como se esperaba.

5. De no haberse detectado transmisión puede suceder que la tarjeta no haya sido instalada en la computadora o bien el Z80 la esperó pero nunca fue detectada. En cualquiera de estos dos casos el programa termina, después de emitir el mensaje apropiado.

6. Si se detectó la transmisión ahora se niega el mensaje (para cambiar los unos a ceros y viceversa) y se regresa al punto 1), recordando que se realizarán cuatro transmisiones, de las cuales tres son con atenuación.

Programa para el Z80:

Reseña:

El programa estará guardado en una memoria ROM contenida en la tarjeta de diagnóstico, y lo que realizará será lo siguiente:

1. Inicializará sus registros para mantener un control del número de diagnósticos que se han realizado y de la cadena de bits que deberán ser recibidos.

2. Leerá de un puerto el estado que guarda un flip-flop T, con el objeto de que al sensar un cambio en el estado del mismo se defina una recepción.

3. Elegirá un grado de atenuación, de acuerdo al número de diagnóstico presente.

4. Esperará hasta que el 8086 le indique que va a enviar el mensaje.

5. Colocará en el puerto de estado el comando de "espero una transmisión".

6. Tomará del puerto descrito en el inciso 2) el estado que se guarda en el flip-flop T, para determinar si se ha dado una recepción. De ser así se pasará al punto 8.

7. Verificará en el puerto de comandos si el 8086 ha esperado un segundo, esto indicará que la transmisión nunca llegará, pues el tiempo ha sido bastante sin obtener respuesta. El programa terminará después de ejecutar el siguiente inciso.

8. El Z80 mandará una señal ACKNLG para evitar posibles bloqueos en el 8086.

9. Enviará el byte recibido al 8086, vía el puerto 1 de resultados, que sirve para comunicar ambos microprocesadores.

10. Colocará el estado "byte recibido" en el puerto de status, para que el 8086 tome el byte del puerto 1 de resultados.

11. Si no se han dado los cuatro diagnosticos se pasará al inciso 3.

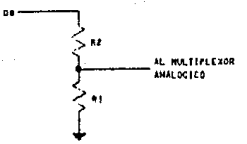
12. Termina el programa.

Atenuaciones.

Las atenuaciones que se darán a D0, discutidas anteriormente, estarán en función de los valores que se manejan como "alto" y "bajo" dentro de la lógica de la familia de circuitos TTL.

Un "alto" está definido dentro de los valores 2.4 a 5 volts, siendo el típico 3.5 v. El valor "bajo" está definido dentro de los 0 a 0.4 v., siendo el típico 0.2 v.

Como se menciono, uno de los diagnósticos se hará con la señal tal cual sale del puerto paralelo, que se supondrá será de 5 v. Para los siguientes diagnósticos se tendrá un arreglo como el que sigue:



De acuerdo a la fórmula del divisor de voltaje, el voltaje obtenido a la salida del multiplexor analógico será de:

$$V = \frac{R1}{R1 + R2} * 5$$

Para el segundo diagnóstico los valores de las resistencias serán: R1 = 2.2 kΩ, R2 = 1 kΩ. Por lo que el voltaje obtenido será de 3.44 v., valor cercano al "alto" típico. En este punto el diagnóstico deberá indicar que ese bit en particular no fué encontrado erroneo.

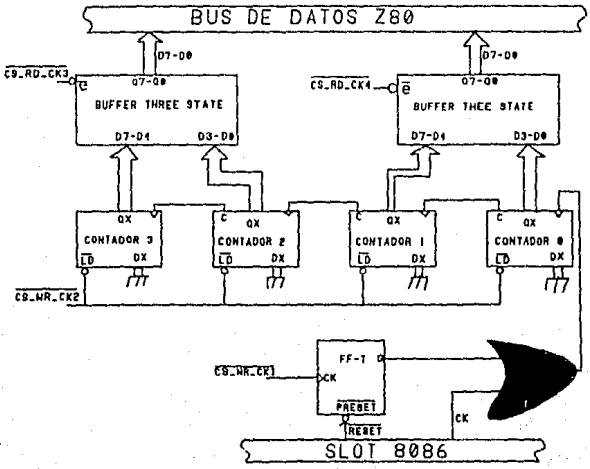
El tercer diagnóstico se hará con valores de R1 = R2 = 2.2 kΩ, que proporciona un voltaje de 2.5 v., por lo que el bit D0 debe encontrarse aún en un valor "alto".

El cuarto diagnóstico se hará con valores de R1 = 1 kΩ, R2 = 10 kΩ, para obtener un voltaje de 0.50 v., es hasta este valor en que el diagnóstico deberá indicar que el bit D0 está sensado como un "bajo".

iii) Reloj

En este bloque se contarán los pulsos del reloj 8284 utilizado por la PC, en donde el intervalo de conteo estará determinado por el propio reloj de la tarjeta diagnosticadora. Para lograr esto se cuenta con la siguiente arquitectura: cuatro contadores en cascada cuya función es contar los pulsos del 8284; dos buffers de 8 bits y tres estados, los cuales acoplarán a la salida de los contadores con el bus de datos del Z80 CPU; un flip flop tipo T y una compuerta OR, los cuales permitirán y bloquearán el paso de la señal de reloj del 8284.

Diagrama a Bloques:



Programa para el Z80:

Reseña:

Secuencia que seguirán las señales seleccionadoras:

* Activar $\overline{CS\ WR\ CK2}$;Pone en cero a los contadores
* Activar $\overline{CS\ WR\ CK1}$;Permite el paso de la señal CK del 8284
* Esperar T (ms) ;El Z80 CPU espera un tiempo determinado
* Activar $\overline{CS\ WR\ CK1}$;Bloquéa el paso de la señal CK del 8284
* Activar $\overline{CS\ RD\ CK4}$;El Z80 CPU recibe el dato menos significativo de la cuenta
* Activar $\overline{CS\ RD\ CK3}$;El Z80 CPU recibe el dato más significativo de la cuenta.

Programa para el 8086:

Reseña:

El programa está desarrollado en lenguaje de alto nivel (Turbo Pascal) y realizará lo siguiente:

-Desactivar las interrupciones
-Mandar al puerto 84h el operador 01h (diagnostico reloj)
-Esperar 3000 milisegundos
-Recibir el dato menos significativo del puerto 87h
-Recibir el dato más significativo del puerto 86h
-Calcular la frecuencia con la siguiente expresión:
   f = round((datoLSB * 256 + datoMSB)/TiempoMuestreo);
   En donde el Tiempo de Muestreo = 0.0001;
-Activar las interrupciones.

iv) Controlador de disco flexible.

El diagnóstico que se efectuará sobre el disco flexible (DF) abarcará únicamente al manejador del DF, en forma directa e indirectamente proporcionará, un diagnóstico del controlador en sí.

Para llevar a cabo el diagnóstico del manejador del DF se requiere desconectarlo ya sea de la tarjeta madre, o de la tarjeta del controlador, para así realizar la conexión en la tarjeta diagnosticadora.

Una vez hecha esta conexión se procederá a realizar el diagnóstico por medio del software y el hardware de la tarjeta.

Esta etapa se dividirá en dos partes, la primera de ellas será la descripción hablada y detallada de las funciones que deben ser cubiertas en el diagnóstico del manejador del DF. La segunda parte se refiere a la descripción a bloques del hardware mencionado y la enumeración y descripción de los elementos físicos que conlleva este hardware.

1) Diagnóstico del manejador del DF

Para realizar este diagnóstico será necesario ordenar al manejador que efectúe varias operaciones sobre un DF, verificando que los resultados obtenidos sean correctos. Estas operaciones se llevarán a cabo a través de un hardware diseñado exprésamente para ello, por lo cual se requiere que exista una conexión física entre el manejador y la tarjeta diagnosticadora.

Las operaciones a realizarse son básicamente la de un movimiento controlado de las cabezas y la detección de las señales sensoras del manejador.

Para la operación del movimiento de las cabezas es necesario llevar a cabo los siguientes pasos:

1) Encender el motor y seleccionar el manejador durante al menos 0.5 s para asegurar la estabilización de la velocidad del motor.
2) Seleccionar la dirección del movimiento hacia el track 0.
3) Generar pulsos (STEP) para que el motor de pasos del manejador mueva la cabeza hacia el track 0.
4) Detectar si la cabeza ya está en el track 0, si lo está, seguir con 5), si no, repetir 3).

5) Una vez que la cabeza ya está en el track 0, cambiar la dirección del movimiento hacia afuera y proporcionar el número de pulsos necesarios para su desplazamiento hasta el track 0 deseado.
6) Repetir desde 2) hasta 5) verificando que no exista una falla en el movimiento.
7) Apagar el motor y deseleccionar el manejador.

Para realizar la detección de la señal index continuar con los siguientes pasos:
1) Encender el motor y seleccionar el manejador.
2) Es necesario que se haya introducido un disco y cerrado la puerta del manejador.
3) Preguntar tantas veces sea necesario por la señal index, a menos que se llegue a un límite de tiempo, con lo que se supondría que no es posible la detección de esta señal.
4) Apagar el motor y deseleccionar el manejador.

Por último, para poder observar la velocidad del motor de polea del manejador seguir los pasos que a continuación se dan:
1) Encender el motor y seleccionar el manejador.
2) Preguntar por la señal index hasta que se dé (si no sucede esto en un tiempo razonable existe un error en la detección de la señal index).
3) Una vez que se dió index, preguntar de nuevo por ella, y contar el tiempo que transcurrió entre la primera detección del index y la segunda, ya que este es el tiempo que tarda una revolución del motor.
4) Apagar el motor y deseleccionar el manejador.

El hardware que efectuará estas operaciones deberá ser capaz de manejar físicamente las líneas de conexión con el manejador, regulando los tiempos de operación.

Asimismo, este hardware debe de proporcionar constantemente información al CPU acerca del estado del proceso que se está llevando a cabo.
Es posible construir diferentes pruebas para averiguar cuál falla se ha presentado o en su defecto cuál tiene más probabilidades de ser la falla.

Una vez presentada esta estructura inicial del diagnóstico del manejador del DF el siguiente paso lo constituye la descripción a bloques de este hardware diagnosticador, para posteriormente pasar a su implantación física con componentes

reales.

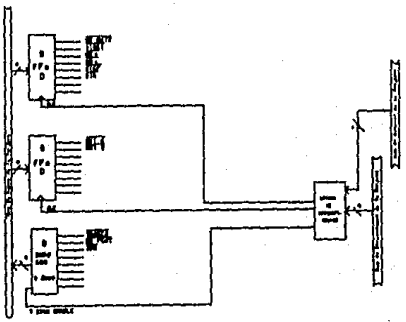
2) Descripción del hardware diagnosticador

El hardware de diagnóstico será básicamente el controlador de las líneas de desplazamiento y posicionamiento que maneja el drive, tales como: la habilitación del drive, el encendido del motor, la señal de avance de un track, la señal que indica la dirección del movimiento de la cabeza, los sensores que detectan la señal index y la señal de que la cabeza está sobre el track 0.

Los valores de estas señales serán proporcionados por el procesador de la tarjeta de diagnostico, y serán retenidos por varios registros a los cuales se conectará el conector del drive.

3) Implantación física del hardware

Como ya se describió, las señales de control y sensoras del drive serán propocionadas y leídas mediante un conjunto de registros. Para las señales de control, se utilizarán flip-flops D's que retengan la información proveniente del procesador, y para las señales sensoras se utilizarán buffers tres estados que permitan su lectura oportuna.



4) Diagnóstico por software

No estaría completo un diagnóstico del disco flexible si no se presentará un diagnóstico para la lectura/escritura verificando con ello un gran conjunto de circuitería (tal como el separador de datos). Para ello, se realizarán una serie de rutinas lenguajes de alto nivel, que, a través del controlador de DF y de sus circuitos anexos efectúen operaciones de lectura/escritura sobre un disco

flexible.

Este diagnóstico tendrá la forma comentada anteriormente, consistente en escribir a lo largo de todo el disco un conjunto datos conocidos, y posteriormente realizar la lectura sobre los mismos sectores y tracks donde se escribió para verificar que sea igual lo leído a lo escrito.

b) Conjunción de los diseños parciales.

Se conjuntarán los diseños parciales mediante la interconexión de los módulos por medio de las señales de control (señales de habilitación, inicio, paro, reset, etc), los buses (datos y direcciones) y líneas de alimentación (Vcc y GND).

IV.3. DESARROLLO DEL SOFTWARE PARA EL CONTROL DE LA TARJETA.

Para el control general de la tarjeta de diagnóstico se desarrollará una rutina en ensamblador Z80 que constantemente estará preguntando por el registro de comandos, en el cual el 8086 colocará el comando que identifique la acción que desea que el Z80 realice (la ejecución de la rutina en ensamblador Z80 correspondiente al diagnóstico solicitado por el usuario). Además, el programa de control posee una rutina que permite al 8086 conocer si la tarjeta de diagnóstico efectivamente está instalada.

Junto con esta rutina, se desarrollará una rutina en Pascal, version Turbo para PC, que permitirá ejecutar un Reset por software a la tarjeta para un mejor control de ésta y permitir al sistema una recuperación de errores, y además verificar la instalación efectiva de ella.

El restante software de diagnóstico elaborado en ensamblador Z80 y que se emplea para el control directo de los diferentes hardwares de diagnóstico se encuentra descrito en la sección correspondiente a cada módulo.

CAPITULO V

IMPLANTACION Y PRUEBAS DEL SISTEMA DE DIAGNOSTICO

V.1. DISEÑO DE CASOS DE PRUEBA.
V.2. RESULTADOS OBTENIDOS DE LAS PRUEBAS.

V. IMPLANTACION Y PRUEBAS DEL SISTEMA DE DIAGNOSTICO.

V.1. DISEÑO DE CASOS DE PRUEBA.

a) Casos de prueba para la rutina de diagnóstico del CPU.

Para probar a la rutina del CPU se pueden enumerar algunas pruebas:

a) Eliminar la acción del CPU sobre el sistema.
b) Modificar físicamente los registros del CPU.
c) Modificar lógicamente el valor de los registros del CPU.

a) La eliminación del CPU consistiría en su bloqueo, o ausencia del sistema. Esta situación haría imposible la ejecución del sistema de diagnóstico y por lo tanto de la rutina diagnosticadora.

b) Modificar los registros del CPU acarrearía el riesgo de afectar la ejecución del programa, sin embargo, es posible suponer la existencia de un CPU cuyos registros no mantengan su valor de manera estable; para esta situación y suponiendo una afección no muy grande sobre la ejecución del programa, es posible obtener resultados del diagnóstico.

c) La modificación lógica (por software) de los valores de los registros sí es posible, y por lo tanto se espera que la rutina de diagnóstico detecte una diferencia con lo esperado, indicando al o a los registro(s) defectuoso(s).

b) Casos de prueba para la rutina de diagnóstico del DMA.

En el capitulo III, se diseñó el diagnóstico para el DMA a través de software, el cual se basa en los siguientes tres pasos:

1) Verificación de los registros internos del DMA.
2) Transferencia de floppy disk a memoria.
3) Transferencia de memoria a floppy disk.

Con estos puntos se logra obtener una alta confiabilidad del buen funcionamiento interno del circuito.

Las pruebas que se realizarán para este módulo serán las siguientes:

a) Eliminar el DMA de la PC.
b) Trasladar el DMA de una PC dañada a otra en buen estado.

a) Esta prueba resultaría imposible de realizar, dado que no se podría ejecutar el sistema de diagnóstico.

b) Con esta prueba se pretende determinar que el circuito DMA se encuentra en buen estado y que por lo tanto no fue el causante del mal funcionamiento de la PC. No se podrá realizar esta prueba hasta que no se localice alguna PC en la que se suponga que el DMAC se encuentra dañado.

c) Casos de prueba para la rutina de diagnóstico de Memorias.

Para este diagnóstico se tienen dos casos de prueba :

a) Retirar directamente bancos de memoria de la PC.
b) Cambiar un dato.

a) La primera prueba que se realizará consistirá en quitar directamente de los bancos de memoria de la computadora un chip,esto ocasionará que se disminuya el tamaño original de la memoria y se detecte automáticamente el nuevo tamaño de la misma por medio de la interrupción (12H), que reporta el tamaño de la memoria.

b) Esta prueba consistirá en cambiar por medio de programación el dato que se va a escribir y leer en la memoria. Con esto se pretende confirmar que el dato escrito será el mismo que el dato leído, y que la memoria está en buen estado.

d) Casos de prueba para la rutina de diagnóstico del puerto serie.

A continuación se enumerarán los casos de prueba que para este puerto se diseñaron.

a) No hacer la conexión de la tarjeta de diagnóstico (conector DB-25).
b) Cambiar la velocidad de transmisión.
c) Modificar el dato a ser transmitido.
d) Modificar el dato a ser recibido.

a) Con esto se pretende que la tarjeta reconozca que no existe conexión y por lo tanto se tenga un "ERROR TOTAL".

b) El puerto serie está programado para que se reciban y transmitan datos a una cierta velocidad. Esta prueba consiste en cambiar dicha velocidad (frecuencia del reloj), modificándola directamente en la tarjeta de diagnóstico y que se obtenga como resultado que el dato transmitido es diferente al dato recibido.

c) Para el caso de la transmisión de un dato, se cambiará mediante programación el dato que se conoce, y se provocará un "ERROR EN LA TRANSMISION ", ya que el dato transmitido será diferente al recibido.

d) Para el caso de la recepción, se puede causar un "ERROR EN LA RECEPCION", cambiando el dato conocido que va a recibir el RS-232 mediante programación, el dato que se va a transmitir (Z80) es fijo, por lo tanto ambos datos serán diferentes.

e) Casos de prueba para la rutina de diagnóstico del puerto paralelo.

Para probar esta rutina se propondrán los siguientes casos:

a) Eliminar una señal de protocolo físico (hand shaking) entre el puerto paralelo y el dispositivo que se conecte a él.

b) Eliminar la señal de envío de dato.

c) Interceptar uno o varios de los datos enviados a través del puerto paralelo.

a) En esta rutina las señales enviadas a la computadora por la tarjeta (BUSY, PE, SLCT, ERROR) serán colocados en el estado lógico que indique a la PC de un error en la conexión. La rutina deberá sensar que las señales están erróneas y hacerlo saber al usuario. El mismo resultado se obtendrá si el conector que surge de la tarjeta de diagnóstico no es conectado al puerto paralelo.

b) La señal DATA STROBE será interceptada, con lo que la tarjeta de diagnóstico asumirá que esta señal está desconectada y se enviará esta información al CPU de la PC.

c) En esta prueba se pretende interceptar los datos enviados por el puerto paralelo y hacerle llegar otros a la tarjeta de diagnóstico, la incorrecta llegada de los datos deberá mostrarse en la pantalla de la PC.

f) Casos de prueba para la rutina de diagnóstico del teclado.

Para diagnosticar el teclado, se pueden realizar los siguientes casos:

a) Pulsar teclas de acuerdo a un patrón específico.
b) No pulsar alguna tecla.

a) Con esta prueba se propone sensar el correcto despliegue de las teclas que corresponde al patrón. Al momento de observar el erróneo despliegue de alguna tecla se concluye el mal funcionamiento de la misma. Con esta prueba también se pueden detectar aquellas teclas que por suciedad u otro motivo no estén proporcionando un funcionamiento adecuado.

b) Con esta prueba se propone detectar el mal funcionamiento de aquellas teclas que por motivos diversos permanecen enviando su código de rastreo (scan code) aún sin que sean pulsadas.

g) Casos de prueba para la rutina de diagnóstico del reloj.

En el capítulo IV, se diseñó el diagnóstico para el reloj (8284) por medio de hardware, el cual se basa en el siguiente procedimiento:

A través de una tarjeta diagnosticadora formada por un CPU Z80 y una lógica de contadores, se detectan y acumulan los pulsos emitidos por el reloj (8284) en los puertos de datos de esta tarjeta que posteriormente son trasmitidos a la memoria RAM de la PC en donde un programa de alto nivel se encarga de mostrar digital y analógicamente la señal del reloj 8284 en la pantalla de la microcomputadora.

Las pruebas que se proponen para este diagnóstico son las siguientes:

a) Eliminar físicamente la señal del reloj (8284).
b) Conmutar la frecuencia.

a) Esta prueba consiste en bloquear la señal del reloj(8284) y obtener digital y analógicamente una frecuencia nula.

b) Se cambiará la frecuencia de una PC de 4.7 Mhz a 10 Mhz. Con esta prueba se pretende que el diagnóstico registre este cambio y así lo indique.

h) Casos de prueba para la rutina de diagnóstico del controlador de disco flexible.

Para probar tanto el software como el hardware diagnosticador del disco flexible, se enumeran las siguientes pruebas:

a) Eliminar las señales de detección del manejador, tales como index y track 0.

b) Eliminar las señales de control del manejador, tales como dirección, step, motor y selección.

c) Desconectar la fuente de alimentación del disco flexible.

d) Estando todo correctamente conectado no cerrar la puerta del disco flexible, o en su lugar, no colocar un disco en el interior del manejador.

e) Para la detección de la velocidad, entorpecer el giro del motor del manejador.

f) Para el diagnóstico por software, es posible repetir el punto d).

g) Para el diagnóstico por software, es posible repetir el punto c), o en su lugar, desconectar el cable plano que va al manejador.

h) Para el diagnóstico por software, colocar en el interior del manejador un disco flexible dañado en un sector o en un track.

a) Al eliminar la señal index su no detección será reportada. En el caso de la señal track0, se espera detectarla cuando mucho después de 80 retrocesos de la cabeza de lect/escr, si no es así se mandará un error. La eliminación es muy sencilla de realizar, basta con remover el cable correspondiente a la señal, proveniente del manejador.

b) La eliminación de las señales de control causará que el manejador no opere o no lo haga adecuadamente, por lo que, para el caso de las señales de dirección y de step la inmovilidad o su movimiento erróneo se detectará como ubicación equivocada de la cabeza (tenga en cuenta que la cabeza no debe estar inicialmente en el track 0) o como la no detección de la señal index.

c) La eliminación de la alimentación al manejador se puede interpretar como la conjunción de los puntos a) y b).

d) Esta situación será detectada como una falla en la señal index, debido a la detección del agujero que poseen los discos flexibles.

e) Al producir esta situación, la velocidad calculada y desplegada variará con la misma intensidad con que se entorpece el giro del motor.

f) La rutina de diagnóstico debe mandar un error que indica que por alguna razón lo leído en el disco no es lo esperado.

g) Estas situaciones provocan la misma reacción descrita para el inciso f).

h) Teniendo en cuenta que el disco debe estar dañado en aquellos tracks sobre los cuales se va a escribir y leer, la rutina de diagnóstico mandará un mensaje de error solamente cuando realiza el diagnóstico sobre esos tracks en particular.

i) Casos de prueba para las rutinas del disco duro.

i.1) Módulo de compresión y descompresión.

Para este módulo el único caso de prueba es aplicar el módulo de compresión sobre un archivo y verificar que se genere un archivo con menor longitud, posteriormente descomprimirlo, sin que ello ocasione una alteración del archivo original.

i.2) Módulo de encriptamiento y desencriptamiento.

El único caso de prueba de este módulo consistirá también en aplicar el módulo de encriptamiento a un archivo, verificar que el archivo ha sido alterado en su contenido y no presenta los mismos datos que el original. Posteriormente se realizará el desencriptamiento del archivo y se cotejará que el archivo haya sido restaurado a su estado original.

i.3) Módulo de verificación de la FAT.

Para probar la rutina de la FAT se pueden enumerar algunas pruebas:

a) A través de un programa como Norton marcar un cluster con la marca FFF7H y posteriormente utilizar el diagnóstico FAT el cual detectará este cluster marcado como dañado y luego le permitirá al usuario su corrección.

b) Comparar la información vertida por este módulo para verificar discos flexibles que hayan sido detectados como dañados por los programas especiales, tales como FORMAT.COM.

V.2. RESULTADOS OBTENIDOS DE LAS PRUEBAS.

Los resultados obtenidos de las pruebas son los siguientes:

DIAGNOSTICOS	EXITO	FRACASO	PRUEBA NO POSIBLE
DIAGNOSTICO DEL CPU			
- Prueba tipo a			X
- Prueba tipo b			X
- Prueba tipo c	X		
DIAGNOSTICO DEL DMA			
- Prueba tipo a	X		
- Prueba tipo b			X 1
DIAGNOSTICO DE LAS MEMORIAS			
- Prueba tipo a	X		X 2
- Prueba tipo b	X		
DIAGNOSTICO DEL PUERTO SERIE			
- Prueba tipo a	X		
- Prueba tipo b	X		3
- Prueba tipo c	X		
- Prueba tipo d	X		
DIAGNOSTICO DEL PUERTO PARALELO			
- Prueba tipo a	X		
- Prueba tipo b	X		3
- Prueba tipo c	X		
DIAGNOSTICO DEL TECLADO			
- Prueba tipo a	X		
- Prueba tipo b	X		
DIAGNOSTICO DEL RELOJ			
- Prueba tipo a	X		
- Prueba tipo b	X		
DIAGNOSTICO DEL DISCO FLEXIBLE			
- Prueba tipo a	X		
- Prueba tipo b	X		
- Prueba tipo c	X		
- Prueba tipo d	X		
- Prueba tipo e	X		
- Prueba tipo f	X		
- Prueba tipo g	X		
- Prueba tipo h	X		

DIAGNOSTICOS	EXITO	FRACASO	PRUEBA NO POSIBLE
DIAGNOSTICO DEL DISCO DURO			
i) Compresión y descompresión			
- Prueba tipo a	X		
ii) Encriptamiento y desencriptamiento			
- Prueba tipo a	X		
iii) Verificación de la FAT			
- Prueba tipo a	X		
- Prueba tipo b	X		

NOTAS:

1. Aún no se ha detectado una computadora de la que se sospeche el mal funcionamiento del DMAC, para realizar esta prueba.

2. En algunas computadoras el hecho de retirar un banco de memoria provoca que el sistema deje de funcionar, al detectar un error de paridad en la memoria, por lo que en estas computadoras el diagnóstico es imposible.

3. Este diagnóstico no es posible en aquellas computadoras que carezcan de este puerto.

CAPITULO VI
CONCLUSIONES

VI.1. ALCANCES Y LIMITACIONES DEL SISTEMA DE DIAGNOSTICO.
VI.2. OTRAS ALTERNATIVAS DE SOLUCION.

VI. CONCLUSIONES.

VI.1. ALCANCES Y LIMITACIONES DEL SISTEMA DE DIAGNOSTICO.

a) Conclusiones del diagnóstico del CPU

El diagnóstico diseñado para el CPU tiene la limitación de que sólo detecta un conjunto pequeño de posibles fallas relacionadas con el CPU o con los componentes directamente ligados a él (tales como el separador de direcciones y datos, el transceptor de datos, etc.), debido principalmente a que fallas mayores imposibilitarían la ejecución confiable de la misma rutina de diagnóstico.

b) Conclusiones del diagnóstico del DMA.

El diagnóstico del DMA se limita a determinar el buen funcionamiento interno de dicho circuito, por lo tanto, no es posible determinar el correcto funcionamiento del mismo en su interacción con los circuitos a los que está conectado.

Un alcance de esta rutina es que puede auxiliar también en el diagnóstico de circuitos tales como el controlador de disco y el estado mismo del disco.

Esta rutina de diagnóstico está limitada a que el FDC (Controlador de Disco Flexible) tenga un correcto funcionamiento.

c) Conclusiones del diagnóstico de las memorias.

No se puede realizar una gran variedad de pruebas, dado que la memoria es un dispositivo del hardware fijo, al que el usuario no tiene acceso fácilmente.

En la parte de memoria en que la rutina de diagnóstico está trabajando no se hace la verificación de la memoria como se explicó en el capítulo tres del presente trabajo,se simula únicamente la verificación y se concluye que si estuviera en mal estado esta parte de la memoria, la rutina simplemente no se ejecutaría.

Para este caso los resultados que se obtuvieron en los diferentes equipos fueron exitosos y satisfactorios, el diagnóstico funciona en todas las máquinas, que por su variedad presentaron diversos tamaños de memoria.

Es importante aclarar que al retirar un circuito del banco de memoria de la computadora,se tiene que tomar en cuenta la organización de la misma para no quitar alguna parte esencial.

Por último, nunca se presentaron problemas de que alguna parte de la memoria estuviera dañada,aún cambiando el dato conocido.

d) Conclusiones del diagnóstico del puerto serie.

Para el diagnóstico del puerto serie, el sistema cuenta con un convertidor de DB-25 a DB-9 para poder analizar el puerto independientemente del tipo de conector.

El diagnóstico está limitado a una velocidad de transmisión y recepción (4800 Bauds).

El usuario no tiene oportunidad de programar el puerto (datos, velocidad, paridad, etc).

Una vez realizada la etapa de pruebas, se puede concluir que que todos los resultados obtenidos fueron satisfactorios, ya que en la mayoría de los equipos en que se probó el hardware de diagnóstico se confirmó lo que se esperaba (que el sistema de diagnóstico pudiera determinar el funcionamiento del puerto). Sólo en un caso específico el RS-232 se diagnosticó fallando (Error Total). Esto ocurre cuando el RS-232 está definitivamente en mal estado, tanto para transmitir,como para recibir datos.

e) Conclusiones del diagnóstico del puerto paralelo.

Con la implantación del módulo de diagnóstico del puerto paralelo se logra checar el correcto funcionamiento de las señales de "handshaking", de la señal DATA STROBE y de las líneas por las que se envían los datos.

Se ha logrado por medio de este diagnóstico simular fallas y detectarlas, de tal foma que se interceptan las líneas de control y de datos y el sistema es capaz de sensar la falla en esas líneas.

No es posible indicar específicamente las líneas de estado (BUSY, ERROR, etc) que se encuentren en mal estado.

No se puede diagnósticar por ráfagas de datos.

f) Conclusiones del diagnóstico del teclado.

Con el módulo de diagnóstico del teclado se puede llegar a sensar aquellas teclas que están proporcionando un mal funcionamiento. El diagnóstico del teclado proporciona parámetros de comparación para aquellos usuarios que puedan estar dudando del correcto funcionamiento de su CPU. Esto es posible porque se muestra al usuario que los datos que están entrando son correctamente direccionados a su CPU.

Entre las limitantes de este diagnóstico se tienen:

No se permiten diferentes tipos de teclado.

No se diagnóstican otras características del teclado (por ejemplo repetición, bloqueo, etc).

g) Conclusiones del reloj.

Con este módulo se logra obtener una medición de la frecuencia del circuito 8284 que proporciona la señal de reloj al sistema con una exactitud alta (aproximadamente del 85%) debido a que se está midiendo la frecuencia de este circuito a través del reloj que se encuentra en la tarjeta de diagnóstico.

Un alcance de este diagnóstico es el detectar posibles distorsiones en la señal de reloj. Si la frecuencia excede un límite permisible (85%) del valor nominal de frecuencia del sistema se podrá conocer (indirectamente) que la señal se está proporcionando en forma anómala en el sistema.

Una limitante es que no muestra el nivel real de la señal del reloj, y por lo tanto no muestra las limitantes del mismo.

h) Conclusiones del diagnóstico del controlador de disco flexible.

El diagnóstico del Manejador del Disco Flexible tiene la posibilidad de localizar fallas en las líneas de control del Manejador, en las líneas de sensado, y en la circuitería para la escritura/lectura de datos, además de que indirectamente, detecta fallas en el Controlador del Disco Flexible. La precisión del diagnóstico varía de caso en caso, y el diagnóstico de lectura/escritura en particular sólo arroja resultados generales, debido a que el resultado de una correcta escritura/lectura implica el funcionamiento adecuado de una cantidad considerable de circuitos y otros elementos. Sin embargo, sí es posible obtener información valiosa para la detección de la(s) falla(s) que se presenten, mediante una inteligente combinación de diagnósticos y pruebas efectuadas.

El presente diagnóstico tiene la limitación de no poder realizar exteriormente la escritura/lectura mediante un hardware adicional, aunque en su lugar el diagnóstico involucra la circuitería normalmente utilizada por el sistema. No obstante, tener ese diagnóstico adicional por hardware podría reportar más información útil para la detección de la(s) falla(s) del sistema.

Una limitación más del diagnóstico es que varias de las señales diagnósticadas (por ejemplo, Track0), pueden fallar debido a diferentes causas, que no necesariamente tengan relación entre sí (por ejemplo, no existe alimentación o existe una falla en el motor de pasos o existe una falla interna en el Manejador), la distinción de la verdadera causa de fallas de las restantes estará a cargo del operador del sistema, quien debe tener la suficiente habilidad y conocimiento, así como los aparatos de medición que requiera, para aislar la falla real del conjunto de fallas posibles.

i) Conclusiones del disco duro.

a) Rutina de compresión y descompresión de archivos.

Esta rutina logra la compresión satisfactoria de archivos en cuanto a espacio de almacenamiento requerido.

Cabe mencionar que en cuanto a tiempo se refiere, la rutina realiza la compresión de una manera adecuada.

Algunas de las limitaciones de este módulo son:

- Para el caso extremo en que la capacidad en disco sea menor que el requerimiento para generar el nuevo archivo (el archivo comprimido) no se podrá llevar a cabo la compresión.

En la descompresión de archivos se tienen las mismas limitantes que el módulo de compresión, a excepción de que el tiempo en que se descomprime un archivo es mayor que en la compresión.

b) Rutina de encriptamiento y desencriptamiento de archivos.

Esta rutina logra el encriptamiento de los archivos de tal manera que el desencriptamiento por otros métodos resulta no factible debido a que la llave con la que se encriptan los archivos se encuentra a su vez encriptada.

En cuanto a limitantes se refiere se puede mencionar la siguiente:

- Si no existe suficiente espacio en disco para generar el archivo encriptado el proceso no se llevará a cabo.

c) Rutina de verificación de la Tabla de Alojamiento de Archivos (FAT).

Este procedimiento se limita a detectar y corregir en lo posible clusters marcados como dañados. En otra palabras, para cada cluster marcado como dañado se aplicará un diagnóstico para verificar que efectivamente el cluster estaba dañado o se encontraba en buen estado y la marca es falsa.

Un alcance de esta rutina es que se pueden corregir clusters en buen estado que están marcados como dañados por agentes externos, tales como virus informáticos, fallas de la energía en la PC y mal funcionamiento de los sensores de información del disco duro.

VI.2. OTRAS ALTERNATIVAS DE SOLUCION.

a) Otras alternativas para el diagnóstico del CPU.

Otra alternativa a este diagnóstico consiste en utilizar equipo analógico (osciloscopio por ejemplo) que permite observar el comportamiento directo del CPU, y posteriormente interpretar los resultados.

No obstante, es posible ampliar el diagnóstico si se coloca al CPU un hardware exterior que consistiría en una base para el CPU a diagnosticar y una circuitería (junto con el software apropiado) que probaría las diferentes características del circuito.

b) Otras alternativas para el diagnóstico del DMA.

Otra alternativa para el diagnóstico del DMA es la utilización de una tableta especial en donde se colocará el DMA que se sospecha está dañado y verificar el estado del circuito a nivel externo.

c) Otras alternativas para el diagnóstico de las memorias.

Para este diagnóstico se pueden verificar directamente los circuitos de las memorias, así como sus señales por medio de un dispositivo externo a la PC, señales tales como lectura y escritura. Este diagnóstico consiste en utilizar equipo analógico (osciloscopio) que permitiría observar el comportamiento directo de estos circuitos.

d) Otras alternativas para el diagnóstico del puerto serie.

Existe una alternativa que simplificaría la realización de este diagnóstico, se trata de conectar directamente las señales de recepción y transmisión del puerto, sin necesidad de utilizar la tarjeta de diagnóstico, esto no es recomendable ya que sólo se detectaría (en caso de falla) que hay algo mal,pero no se especificaría exactamente cuál es el error.

Si se desea hacer alguna mejora al hardware de diagnóstico, esta consiste en que el usuario tenga libertad para programar al puerto serie en velocidad, paridad, etc., aunque como se presenta y realiza este diagnóstico resulta amigable al usuario y cumple con la función principal de indicarle si el puerto está o no en buen estado.

e) Otras alternativas para el diagnóstico del puerto paralelo.

Algunas empresas fabricantes de PCs emiten para sus empleados de servicio manuales en los cuales se indica que el puerto paralelo puede ser diagnosticado con el uso de un conector el cual tiene interconectadas algunas de sus terminales.

En estos manuales se indica que el puerto paralelo es bidireccional, lo cual no es cierto en la mayoría de las marcas de PCs. Esto es, esta forma de diagnóstico no puede ser empleada en la totalidad de los modelos de las PCs.

Es de hacer notar que el conjunto de las interrupciones del ROM BIOS no contiene una instruccion para obtener un dato del puerto paralelo (análogamente al puerto serial). Esto implica que el circuito que forma el puerto paralelo (8255) puede ser programado para que trabaje en forma bidireccional, pero no puede ser explotada esta característica por la mayoría del software disponible en el mercado.

Otra forma de diagnosticar el puerto paralelo es conectarlo a un osciloscopio de dos o más canales y verificar que las señales (control y datos) se estén dando en forma correcta.

Una última alternativa consiste en conectar el puerto paralelo a un dispositivo(por ejemplo una impresora) del que se observe un correcto desempeño al estar conectado a otra computadora, posteriormente conectar el dispositivo en el puerto paralelo a diagnosticar y observar si dicho dispositivo continúa funcionando correctamente, de esta forma se sabrá si el puerto paralelo está en buen estado.

f) Otras alternativas para el diagnóstico del teclado.

Para el diagnóstico del teclado se propone la alternativa de realizar un conector especial, de tal forma que se proporcionarán al teclado las señales de alimentación de voltaje y la línea de datos (serial) se conecte a un equipo especial, que pueda registrar el código generado al pulsar alguna de las teclas.

Los distribuidores de sistemas de diagnóstico, sin embargo, solucionan el chequeo del teclado por un sistema bastante similar al que se propone en este trabajo.

g) Otras alternativas para el diagnóstico del reloj.

Otras alternativas para el diagnóstico del reloj es la utilización de un osciloscopio o frecuencímetro para detectar la señal de reloj emitida por el circuito 8284.

h) Otras alternativas para el diagnóstico del controlador de disco flexible.

Para realizar el diagnóstico del Manejador y del Controlador de Disco Flexible existen en el mercado diferentes productos que, al menos de los que se tiene conocimiento, realizan diferentes pruebas a través del control directo del Controlador, por medio de software. Es recomendable la utilización de estos productos cuando las fallas probables no están muy ligadas al hardware del sistema de Disco Flexible, puesto que en última instancia no son capaces de manejar las señales que usa el hardware a diagnosticar.

i) Otras alternativas para las rutinas del disco duro.

En relación a la compresion y descompresión de archivos, existen diferentes alternativas en el mercado tales como:

PKZIP y PKUNZIP

PKARC y PKXARC

PCTOOLS.

Con respecto al encriptamiento y al desencriptamiento de archivos existe una utileria de PCTOOLS que permite esto.

Para el módulo de diagnóstico de la FAT existen otras alternativas en el mercado tales como:

CHKDSK (comando externo del DOS)

NORTON

PCTOOLS.

APENDICE A

DESCRIPCION DEL HARDWARE ADICIONAL

A.1. PROCESADOR DE ENTRADA/SALIDA.
A.2. CONTROLADOR DE INTERRUPCIONES.
A.3. LATCH OCTAL.
A.4. BUS TRANSCEPTOR OCTAL.
A.5. CONTROLADOR DE BUS.
A.6. TIMER PROGRAMABLE.
A.7. SLOTS E INTERFACES.

APENDICE A. DESCRIPCION DEL HARDWARE ADICIONAL.

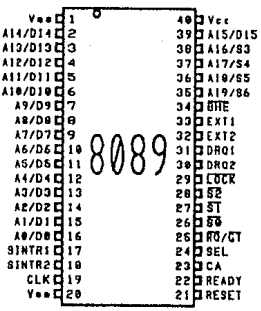
A.1. PROCESADOR E/S DE 8 Y 16 BITS HMOS 8089.

a) Introducción.

El 8089 es un concepto revolucionario en microprocesadores para el proceso de entrada/salida. Encapsulado en 40 terminales, el 8089 es un alto ejecutor de procesos implementado en canales N. El conjunto de instrucciones del 8089 está optimizado para alta rapidez, flexibilidad y eficiencia de E/S. Esto permite fáciles interfaces con los microprocesadores de Intel 8086 de 16 bits y el 8088 de 8 y 16 bits. En la configuración Remota, el 8089 es usado para permitir la compatibilidad con los microprocesadores de 8 y 16 bits de Intel así comos fáciles interfaces con el MULTIBUS (Multiprocessor System Bus Standard) de Intel.

El 8089 ejecuta funciones inteligentes como controlador de DMA. Este puede operar completamente en paralelo con un CPU, llevando aplicaciones intensivas de E/S en forma eficiente. El 8089 provee 2 canales de E/S, cada uno soporta transferencias de 1,25 mbyte/seg con el reloj estandar de 5 Mhz. La memoria base comunicacional entre el IOP y el CPU se enlaza en un sistema flexible y permite la modularidad del Software, permitiendo más formalidad y fácil desarrollo de sistemas.

b) Diagrama de Terminales.



c) Descripción de Terminales.

A0-A15/D0-D15 (Multiplexed Address and Data Bus): La función de estas líneas "I/O" es definida por los estados S0, S1 y S2. Las terminales estan indefinidas después de un RESET y cuando el bus no es adquirido A8-A15 son utilizadas en transferencias al bus de datos físico de 8 bits (como el del 8088), y son multiplexadas con datos en transferencias de 16 bits.

A16-A19/S3-S6 (Address and Status): Son 4 líneas de salida, se multiplexan las líneas de dirección más significativas y el estado de la información. Estas líneas de dirección son activadas sólo cuando se direcciona a Memoria. Por otro lado, las líneas de estado son activadas y codificadas como se muestra a continuación:

S6	S5	S4	S3	
1	1	0	0	Ciclo DMA en CH1
1	1	0	1	Ciclo DMA en CH2
1	1	1	0	Ciclo Non-DMA en CH1
1	1	1	1	Ciclo Non-DMA en CH2

BHE (Bus High Enable): Esta línea de salida es usada para habilitar operaciones de datos en la parte más significativa del bus de datos (D8-D15). La señal es activa baja cuando un byte es

transferido en la parte alta de el bus de datos,

Esta línea está indefinida después de un RESET y cuando el bus no es adquirido. BHE no tiene que ser activada.

S0,S1,S2 (Status): Estas tres líneas de salida están definidas durante actividades IOP en algún ciclo dado. Estas son codificadas como se muestra a continuación:

S2	S1	S0	
0	0	0	Instrucción fetch, espacio I/O
0	0	1	Fetch de datos,espacio I/O
1	0	1	Guardado de Datos, espacio I/O
1	1	0	No usada.
1	1	1	Instrucción fetch,Sistema de Memoria
1	0	1	Fetch de Datos, Sistema de Memoria
1	1	0	Guardado de Datos, Sistema de Memoria.
1	1	1	Pasivo.

Las líneas de status son utilizadas por el Bus de Control y el controlador de bus para activar toda la memoria y señales de control I/O. Estas señales cambian durante T4 y un nuevo ciclo es iniciado mientras el retorno a un estado pasivo en T3 o TW indican el fin del ciclo. Las terminales estan indefinidas después de un RESET y cuando el bus no es adquirido.

READY (READY): Esta señal de entrada es recibida desde el dispositivo de direccionamiento, indicando que el dispositivo está listo para la transferencia de datos. La señal es activa alta y es sincronizada por el generador de reloj 8284.

LOCK (LOCK): Esta señal de salida indica a través del Bus de Control que el Bus de datos es necesitado para más de un ciclo contiguo. Este es puesto por el registro Control de Canal, y durante la instrucción TSL. La terminal está indefinida después de un RESET y cuando el Bus no es adquirido. Esta salida es activa baja.

RESET (RESET): Esta línea de entrada y activa alta causa que el IOP suspenda todas las actividades y entre en un ciclo ocioso hasta que la atención a un canal es recibida. La señal debe ser activa desde los últimos cuatro ciclos de reloj.

CLK (Clock): Esta línea de entrada, activa alta, provee todo el tiempo requerido por las operaciones internas del IOP.

CA (Channel Attention): Esta línea de entrada llama la atención del IOP. En la bajada del borde de la señal, la terminal de entrada SEL es examinada para determinar información Maestro/esclavo o CH1/CH2.

SEL (Select): SEL es una terminal de entrada activa alta. La primer CA recibida después de un RESET del sistema informa al IOP a través de la línea SEL, si se encuentra como Maestro (SEL=0) o como Esclavo (SEL=1) y empieza la secuencia de iniciación. Durante algún otro CA la línea SEL significa la selección de CH1/CH2 (0/1 respectivamente).

DRQ1-2 (Data Request): Son dos líneas de entrada, las cuáles indican al IOP si un periférico está listo para transmitir/recibir datos usando los canales 1 o 2 respectivamente. Estas señales deben ser retenidas activas altas hasta que el apropiado fetch/strobe es iniciado.

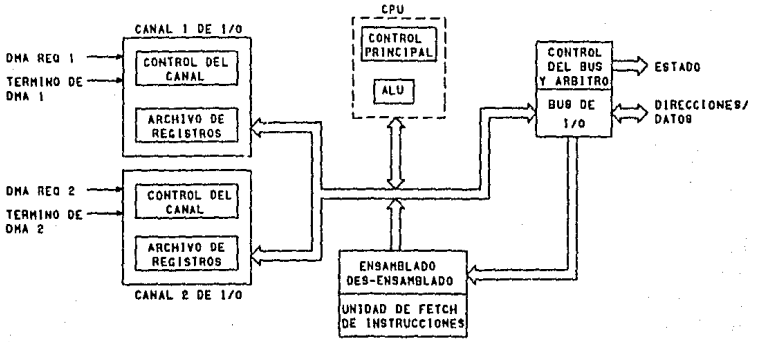
SINTR1-2 (Signal Interrupt): Son dos líneas de salida, una para el canal uno y la otra para el dos. La interrupción puede ser enviada directamente al CPU a través del 8259 (Interrupt Controller). Estas son usadas para indicar al sistema la ocurrencia de eventos definidos.

EXT1-2 (External Terminate): Son dos líneas de entrada , una para CH1 y otra para el CH2. Estas señales pueden causar la terminación de la operación de transferencia DMA en curso, si el canal es programado por el registro de Control de Canal. La señal debe ser retenida alta hasta que la terminación es completa.

Vcc (Voltage): Poder de entrada a +5 volts.

Vss (Ground): Tierra.

d) Diagrama de Bloques Interno.



e) Descripción del Diagrama de Bloques Interno.

El diagrama de bloques del IOP 8089 se compone de: Un ALU, el que se encarga de realizar las operaciones aritméticas y lógicas que se necesiten. Un bloque de Control Principal que ejecuta el conjunto de instrucciones diseñadas para este procesador. Dos Canales I/O, los cuales comunican al 8089 con los periféricos. Un bloque llamado I/O Bus, que proporciona la interfaz de direcciones y datos con el procesador maestro. Un Bus de Control y Arbitro, que indica el estado del 8089 al procesador maestro. Y por último una unidad de Fetch junto con un bloque de Ensamblaje y Desensamblaje, los cuales se encargan de dar un tratamiento adecuado a cada instrucción.

f) Función General del Circuito.

El IOP 8089 es utilizado para realizar o remover procesos de I/O, control y alta rapidez de transferencia desde el CPU. Incluye la inicialización y mantenimiento a componentes periféricos y un versátil soporte de DMA. Este aprovechamiento del DMA simplifica los tiempos del bus y proporciona compatibilidad con memoria y periféricos, asignando operaciones a ser desarrolladas en el bus de datos así como transferencias. Las operaciones deben incluir construcciones así como translaciones, en donde el 8089 busca vectores en una tabla y compara máscaras.

El 8089 es funcionalemente compatible con la familia de Intel 8086/88. Este circuito soporta combinaciones de 8 y 16 bits en los buses. En el modo Remoto puede ser usado para complementar otros procesadores de la familia de Intel. La arquitectura de hardware y comunicaciones es diseñada para proveer simples mecanismos para sistemas grandes.

La única comunicación directa entre el IOP y el CPU es el Canal de Atención y las líneas de Interrupción; parámetros y programas de tarea son pasados a través de bloques a diferentes partes de la memoria, simplificando interfaces del Hardware y Estructuras de Programación. El 8089 puede ser usado en aplicaciones tales como archivos y manejadores de buffer en el control del disco duro o flexible. Este también puede proveer rutinas de recuperación de errores y rastrear el control. El control del CRT, tales como el control de cursor y autoscrolling, es simplificado con el 8089. El control del teclado, control de comunicación y el control general I/O son típicas aplicaciones para el 8089.

MODOS LOCAL Y REMOTO:

En el Modo Local el 8086/88 es usado en modo máximo. El 8089 y el 8086/88 residen en algún bus local, compartiendo el mismo conjunto de buffers del sistema. Periféricos localizados en el bus del sistema pueden ser direccionados por el 8086/88 o el 8089. El 8089 pide el uso del bus local activando la línea RQ/GT. Esta desarrollará una función similar a las líneas HOLD y HLDA en el

8086/88 en modo mínimo. Este modo permite una configuración del sistema más económico así como una reducción en tamaño del mismo.

Una típica configuración Remota es donde el bus del IOP es físicamente separado del bus del sistema por medio de la utilización de buffers/latches. El IOP mantiene un bus local y puede operar fuera de éste, accesando el sistema de memoria.

La interface del bus del sistema contiene los siguientes componentes:

* Arriba de tres 8282 (buffer/latch) para atrapar las direcciones en el bus del sistema.

* Arriba de dos 8286 (buffer bidireccional) para el bus de datos del sistema.

* Un 8288 (controlador de bus) que suple el control de señales necesarias por operaciones del buffer así como señales MRDC (lectura a memoria) y MWTC (escritura a memoria).

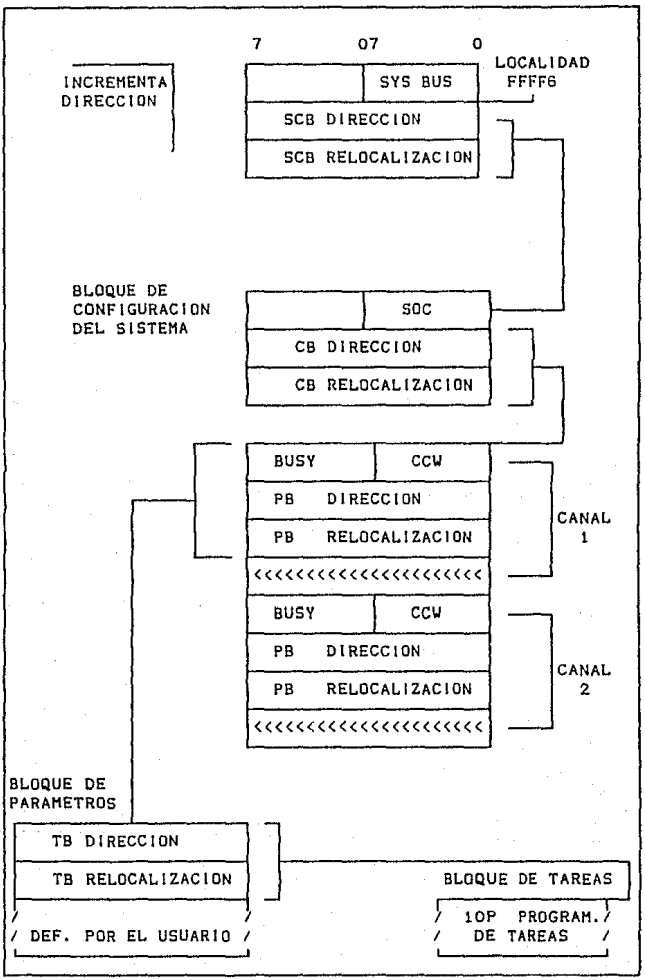
* Un 8289 (arbitro de bus), el cual desarrolla todas las funciones necesarias para coordinar el uso del bus del sistema. Este es usado en lugar de la lógica RQ/GT en el modo local. Este arbitro decodifica diferentes tipos de ciclos de información desde el estado de las líneas del 8089 para determinar si el IOP desea desarrollar una transferencia fuera del bus del sistema. Los dispositivos periféricos PER1 y PER2 son un soporte para el bus de datos y de direcciones.

MECANISMOS DE COMUNICACION:

Fundamentalmente, la comunicación entre el CPU y el IOP es desarrollada a través de mensajes preparados en porciones de la memoria principal. El CPU puede causar que el 8089 ejecute un programa para que el 8089 tome lugar en un espacio de memoria y/o direccione las líneas de control del 8089, accediendo a la señal (CA) "Canal de Atención" del IOP, activando el propio canal I/O. La terminal SEL le indica al IOP cuáles canales empiezan a direccionar. Comunicaciones desde el IOP hacia el microprocesador pueden ser ejecutadas de una manera similar a través de

interrupciones al sistema (SINTR 1,2), si el CPU tiene habilitado interrupciones para este propósito. Adicionalmente, el 8089 puede guardar mensajes en memoria con respecto al estado de algunos periféricos. Estos mecanismos de comunicación son soportados por datos estructurados jerárquicos para proveer un máximo aprovechamiento y flexibilidad de la memoria usada con la capacidad de adicionar múltiples IOPs.

En la siguiente figura se ilustra la Estructura de Datos Jerárquica para la comunicación, la cual existe dentro del IOP 8089:


7
07
0
LOCALIDAD FFFF8
INCREMENTA DIRECCION
SYS BUS
SCB DIRECCION
SCB RELOCALIZACION
BLOQUE DE CONFIGURACION DEL SISTEMA
SOC
CB DIRECCION
CB RELOCALIZACION
BUSY
CCW
PB DIRECCION
PB RELOCALIZACION
CANAL 1
BUSY
CCW
PB DIRECCION
PB RELOCALIZACION
CANAL 2
BLOQUE DE PARAMETROS
TB DIRECCION
TB RELOCALIZACION
DEF. POR EL USUARIO
BLOQUE DE TAREAS
1OP PROGRAM. DE TAREAS

En el primer CA (canal de atención), si el IOP es inicializado así como el BUS MASTER, 5 bytes de información son leídos dentro del 8089 empezando en la localidad FFFF6 (FFFF6, FFFF8, FFFFB) en donde el tipo de bus de sistema (16 bits u 8 bits) y apuntadores al bloque de configuración del sistema son obtenidos. Las demás direcciones son obtenidas a través de la Estructura de Datos Jerárquica. El 8089 determina las direcciones de la misma manera que el 8086/88. Después, la configuración de apuntadores al sistema es formada, en donde el 8089 accesa el Bloque de Configuración del Sistema (SCB), el cual sólo es usado durante el comienzo, después se apunta al Bloque de Control (CB) y se provee la configuración de datos al sistema IOP a través del byte SOC. Este byte inicializa el bus I/O del IOP con 8/16, y define uno de los dos modos de operación RQ/GT. Para el modo 0, el IOP es inicializado como esclavo y tiene la línea RQ/GT ligada al CPU maestro (típica configuración local). En este modo, el CPU normalmente tiene el control del bus, transfiere el control al IOP en caso necesario y, tiene el bus restaurado para completar las tareas del IOP (IOP petición, CPU transfiere, IOP dona). El modo 1 es usado sólo en el modo remoto entre dos IOPs, la designación Maestro/Esclavo es usada sólo durante la inicialización del bus de control: cada IOP peticiona y transfiere el bus necesitado (IOP1 peticiona, IOP2 transfiere, IOP2 peticiona, IOP1 transfiere). Así es, como cada IOP retiene el control del bus hasta que otra petición es hecha. La terminación de la inicialización es señalada por el borrado IOP de la bandera BUSY en el CB. Este tipo de comienzo permite al usuario tener la inicialización de apuntadores en ROM con el SCB en RAM. Permitiendo al SCB estar en RAM y llevando al usuario la flexibilidad para la habilitación del comienzo al inicializar múltiples IOPs.

El Bloque de Control suministra la inicialización al bus de control para la operación del IOP (CCW o Palabra del Control del Canal) y provee apuntadores al Bloque de ParAmetros o "datos". El byte CCW es codificado para determinar la operación del canal.

El Bloque de Parámetros contiene las direcciones del Bloque de Tareas y acciona un mensaje centrado entre el IOP y el CPU. ParAmetros o información de variables son pasados desde el CPU al IOP en este bloque para optimizar la interface de Software a los dispositivos periféricos. Este también es usado para transferir

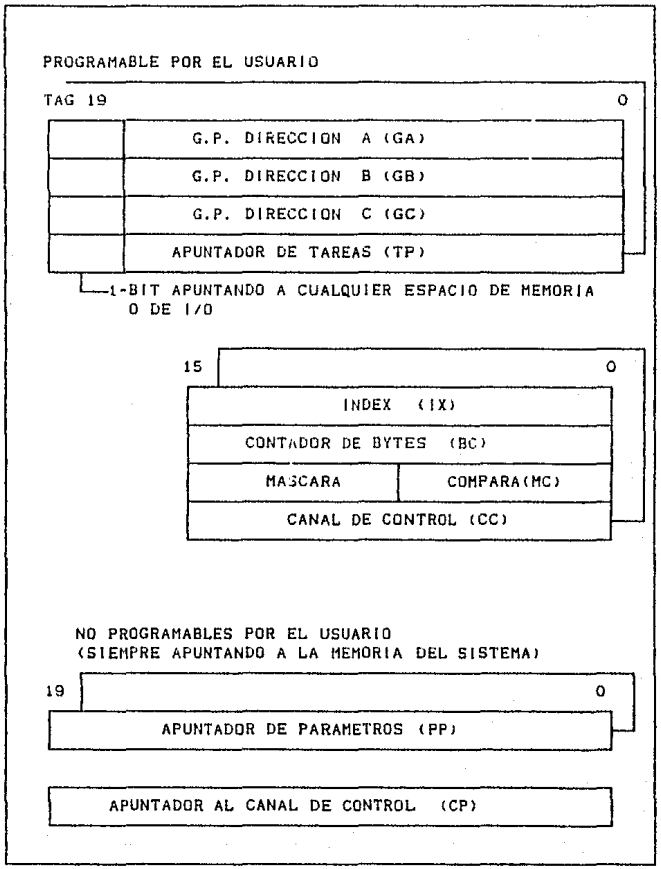
datos y estado de la información entre el IOP y el CPU.

El Bloque de Tareas contiene las instrucciones para el respectivo canal. Este bloque puede residir en el bus local del IOP, permitiendo al IOP operar concurrentemente con el CPU, o residir en la memoria del sistema.

La ventaja de este tipo de comunicación entre el procesador IOP y periféricos es que permite desde muchos métodos de borrado hasta la operación del sistema para atrapar rutinas I/O. Programas o "Bloque de Tareas" permiten la ejecución de rutinas I/O de propósito general con el estado y comandos de periféricos en donde la información empieza a pasar a través del Bloque de Parámetros ("datos", memoria). El Bloque de Tareas (o "programa", memoria) puede ser terminado o restaurado por el CPU, si es necesario.

CONJUNTO DE REGISTROS:

El 8089 mantiene registros separados desde dos canales de I/O así como algunos registros comunes:



Estos son suficientes registros en cada canal para sostener transferencias propias de DMA y procesos. Los registros básicos apuntadores (GA,GB, de 20 bits cada uno) del DMA, pueden apuntar al bus del sistema o al bus local, DMA fuente o destino, y pueden

ser autoincrementados. El registro (GC) puede ser usado para permitir traslaciones durante procesos DMA a través de una tabla vista. El registro de control de canal puede ser accesado sólo por un MOV o por un MOVI, el cual determina el modo de operación del canal. Adicionalmente, los registros son provistos de una máscara comparada durante la transferencia de datos y pueden ser activados para señalar una condición de terminación. Algunos registros pueden ser usados como registros de propósito general durante la ejecución de un programa, cuando el IOP no desarrolla ciclos DMA.

OPERACION DEL BUS:

El 8089 utiliza la misma estructura de bus que el 8086/88 en la configuración de modo máximo. Las direcciones son multiplexadas en tiempo con los datos de las primeras 16/8 líneas. Las líneas A16-A19 son multiplexadas en cuatro estados con las líneas S3-S6. Para los ciclos del 8089, las líneas S4 y S3 determinan que tipo de ciclo comienza a desarrollarse en los canales 1 o 2. S5 y S6 tienen un código asignado al IOP8089, habilitando al usuario para detectar cual procesador está desarrollando un ciclo de bus en un medio ambiente de multiprocesamiento.

El estado de las tres primeras líneas, S0-S2, son usados con el 8288 (controlador de bus) para determinar si alguna instrucción fetch o una transferencia de datos comienza a desarrollarse en un espacio de I/O o de memoria del sistema. Las transferencias DMA requieren dos ciclos de bus, los cuales están formados por un mínimo de 4 ciclos de reloj. Los ciclos de reloj adicionales son sumados si los estados de espera son requeridos. Estos dos ciclos simplifican considerablemente los tiempos de bus en el DMA. El 8089 optimiza las transferencias entre dos diferentes anchos del bus, usando tres ciclos de bus contra cuatro para transferir una palabra. Más de una lectura (escritura) es ejecutada cuando se mapea un bus de 8 bits en uno de 16 bits. Por ejemplo, una transferencia de datos desde un periférico de 8 bits a una localidad física de memoria es ejecutada primeramente por dos lecturas, con palabras ensambladas dentro del IOP en donde se ensambla un archivo de registros y entonces una escritura.

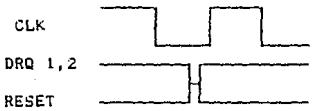
La siguiente tabla muestra el ancho de banda, el retardo y la utilización del bus del sistema:

	LOCAL		REMOTO	
	BYTE	WORD	BYTE	WORD
BANDWIDTH	830 KB/S	1250 KB/S	830 KB/S	1250 KB/S
LATENCIA	1.0/2.4µs*	1.0./2.4µs*	1.0./2.4µs*	1.0./2.4µs*
UTILIZACION DEL SYS BUS	2.4 µs por transf.	1.6µs por transf.	0.8µs por transf.	0.8µs por transf.

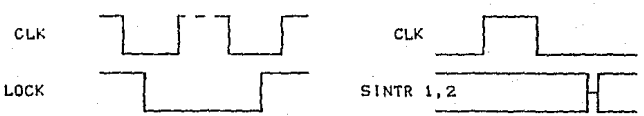
* 2.4 µs, si interactúa con otros canales y no tiene estados de espera. 1 µs, si el canal está esperando desde una petición.

g) Diagramas de Tiempos.

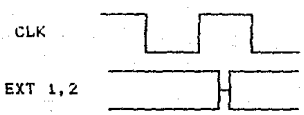
Reconocimiento de señal asincrona.



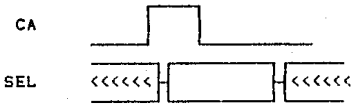
Señal LOCK del bus y SINTR.



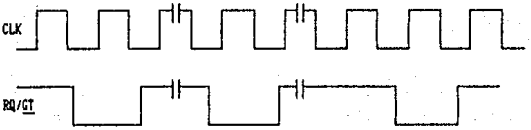
Terminación externa mediante tiempo.



Señales SEL y CA.

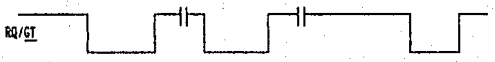


Secuencia para la señal REQUEST/GRANT.



(8089 como esclavo, en modo 0)

(8089 como maestro en modo 1)



(8089 como maestro, modo 0)

h) Programación del Circuito.

Transferencia de Datos:

INSTRUCCIONES DE APUNTADOR	OPCODE
LPD P.M Carga el apuntador PPP desde la localidad direccionada LPDI P.I Carga el apuntador PPP con los siguientes 4 bytes MOVP M.P Almacena el contenido del apuntador PPP en la localidad direccionada MOVP P.M Restaura el apuntador	7 0 7 0 PPP0 0AA1 | 1000 10MM PPP1 0001 | 0000 1000 PPP0 0AA1 | 1001 10MM PPP0 0AA1 | 1000 11MM
MOVIMIENTO DE DATOS	OPCODE
MOV M.M Mueve de fuente a Destino Fuente--- Destino--- MOV R.M Carga registro RRR desde la localidad direccionada MOV M.R Almacena el contenido del registro RRR con la localidad direccionada MOVI R Carga el registro RRR con el siguiente byte con extensión de signo MOVI M Mueve inmediato a la localidad direccionada	7 0 7 0 0000 0AAW | 1001 00MM 0000 0AAW | 1100 11MM RRR0 0AAW | 1000 00MM RRR0 0AAW | 1000 01MM RRRwb 00W | 0011 0000 000wb AAW | 0100 11MM

Control de Transferencias:

LLAMADAS (CALL)	OPCODE
	7 0 7 0
*CALL CALL UNICONDICIONAL	100 DD AAW | 1001 11MM

SALTOS (JUMP)	OPCODE
JMP Unicondicional JZ M salta en memoria = cero JNZ R salta en registro = cero JNZ M salta en memoria <> cero JNZ R salta en registro <> cero JBT prueba bit y salta si = 1 JNBT prueba bit y salta si = 0 JMCE salta si coincide mascara JMCNE salta si no coincide mascara	100 dd 00W | 0010 0000 000 dd AAW | 1110 01MM RRR dd 000 | 0100 0100 000 dd AAW | 1110 00MM RRR dd 000 | 0100 0000 BBB dd AA0 | 1011 11MM BBB dd AA0 | 1011 10MM 000 dd AA0 | 1011 00MM 000 dd AA0 | 1011 01MM

Intrucciones Aritméticas y Lógicas:

INCREMENTOS, DECREMENTOS	OPCODE
	7 0 7 0
INC M Incrementa localidad INC R Incrementa registro DEC M Decrementa localidad DEC R Decrementa registro	0000 OAAW | 1110 10MM RRR0 0000 | 0011 1000 0000 OAAW | 1110 11MM RRR0 0000 | 0011 1100

SUMAS (ADD)		OPCODE	
		7 0	7 0
ADDI	M.I ADD Inmediato a memoria	000 wb AAW	1100 00MM
ADDI	R.I ADD Inmediato a registro	RRR wb 00W	0010 0000
ADD	M.R ADD Registro a memoria	RRR0 0AAW	1101 00MM
ADD	R.M ADD Memoria a registro	RRR0 0AAW	1010 00MM
	AND	OPCODE	
ANDI	M.I AND Memoria e inmediato	000 wb AAW	1100 10MM
ANDI	R.I AND Registro e inmediato	RRR wb 00W	0010 1000
AND	M.R AND Memoria y registro	RRR0 0AAW	1101 10MM
AND	R.M AND Registro y memoria	RRR0 0AAW	1010 10MM
	OR	OPCODE	
ORI	M.I OR Memoria con inmediato	000 wb AAW	1100 01MM
ORI	R.I OR Registia con inmediato	RRR wb AWW	0010 0100
OR	M.R OR Memoria con registro	RRR0 0AAW	1101 01MM
OR	R.M OR Registro con memoria	RRR0 0AAW	1010 01MM
	NOT	OPCODE	
NOT	R Complementa registro	RRR0 0000	0010 1100
NOT	M Complementa memoria	0000 0AAW	1101 11MM
NOT	R.M Complementa memoria, coloca en registro	RRR0 0AAW	1010 11MM

Instrucciones de Manipulacion y Prueba de Bits:

MANIPULACION DE BITS	OPCODE
SET Enciende el bit seleccionado	BBBO OAAO | 1111 01MM
CLR Borra el bit seleccionado	BBBO OAAO | 1111 10MM
PRUEBA	**OPCODE**
TSL Prueba y enciende	0001 1AAO | 1001 01MM

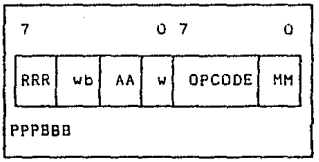
Instrucciones de Control:

CONTROL	OPCODE
	7 0 7 0
HLT Deten ejecución del canal	0010 0000 | 0100 1000
SINTR Ejecuta interrupción FF	0100 0000 | 0000 0000
NOP No Operacion	0000 0000 | 0000 0000
XFER Realiza transferencia DMA	0110 0000 | 0000 0000
WID Captura fuente, destino y bus de longitud S, D0=8, 1=16	1SD0 0000 | 0000 0000

* AA(en una instrucción de llamada, únicamente puede ser 00,01,10).
* OPCODE (es el segundo byte ejecutado).

Todas las instrucciones consisten de dos bytes, mientras algunas instrucciones pueden usar arriba de tres bytes adicionales para especificar literales y desplazamiento de datos. La

definición de los campos dentro de cada instrucción es de la siguiente manera:



MM Selecciona el apuntador base	
00	GA
01	GB
10	GC
11	PP

RRR (Campo Registro):

El campo RRR especificado con un registro de 16 bits es usado en la instrucción. Si GA, GB, GC o TP, son referenciados por el campo RRR, los cuatro bits más altos de los registros son cargados con el bit de signo (Bit 15). Los registros PPP son usados como apuntadores de dirección de 20 bits:

RRR	
000 r0 GA	
001 r1 GB	
010 r2 GC	
011 r3 BC ; contador de bytes	
100 r4 TP ; bloque de tareas	
101 r5 IX ; registro índice	
110 r6 CC ; canal de control (modo)	
111 r7 MC ; compara máscara	Ver notas 1 y 2

PPP	
000	p0 GA ;
001	p1 GB ;
010	p2 GC ;
100	p4 TP ; apuntador del bloque de tareas

Nota 1: Las instrucciones aritméticas y lógicas deben ser usadas para actualizar el registro CC (sólo las instrucciones MOV y MOVI pueden ser usadas).

Nota 2: Los registros de 20 bits (GA,GB,GC o TP) son inicializados como apuntadores de 16 bits en espacio I/O y deben ser salvados cuando se usan las instrucciones MOVP o CALL.

NOTAS:

BBB (campo selector de bits): Este campo reemplaza el campo RRR en instrucciones de manipulación de bits y es usado para seleccionar un bit a ser operado por esas instrucciones. El bit 0 es el bit más significativo.

wb:

01 1 byte literal

10 2 bytes(word) literales

dd:

01 1 byte desplazado

10 2 bytes(word) desplazados.

AA (campo):

00 El apuntador seleccionado contiene la dirección del operando.

01 La dirección del operando es formada adicionando 8 bits, conteniendo el offset en la instrucción del apuntador seleccionado. El contenido del apuntador permanece sin cambios.

10 La dirección del operando es formada adicionando el contenido del registro índice al apuntador de selección. Ambos registros permanecen sin cambios.

11 Excepto el registro índice es postautoincrementado (por 1 de 8 bits transferidos, por 2 de 16 bits transferidos).

W (campo de anchura):

0 El operando es seleccionado por 1 byte de longitud.

1 El operando es seleccionado por 2 bytes de longitud.

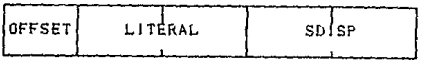
Bytes Adicionales:

OFFSET : 8 Bits sin señalar el offset.

SD!SP : 8/16 bits señalando el desplazamiento.

LITERAL: 8/16 bits literales. (32 bits para LDPI).

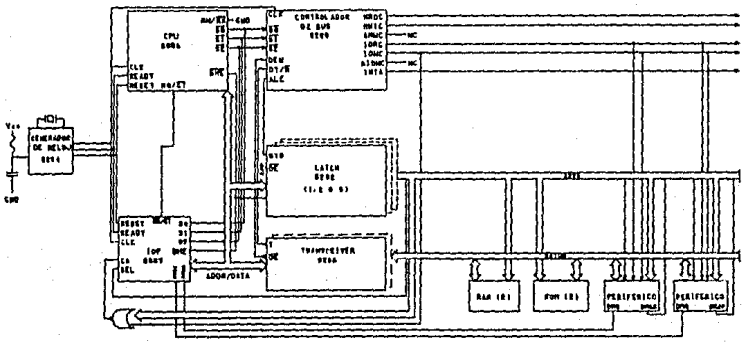
El orden en el cual los bytes opcionales son tratados por el IOP, es de la siguiente manera:



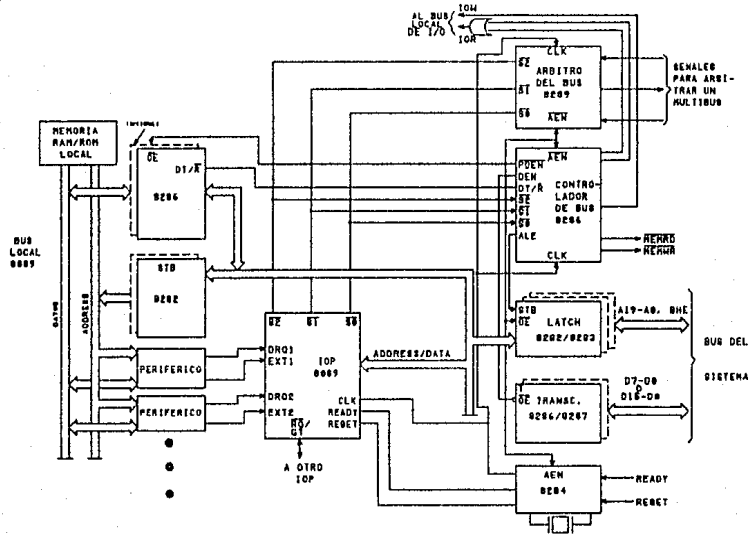
El offset es tratado como números sin cambiar. Literales y desplazamientos son señalados en complemento a 2.

i) Interconexión del Circuito en una PC:

El circuito 8089 puede ser implementado en una PC, colocando al 8086 en modo máximo, como se muestra a continuación:



En ocasiones se emplea al 8089 en modo remoto, como sigue:



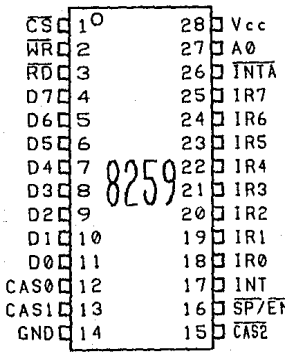
A.2. CONTROLADOR DE INTERRUPCIONES PROGRAMABLE (PIC) 8259A.

a) Introducción.

El 8259A esta diseñado para minimizar el software y el sobre paso de tiempo real, manejando prioridades de interrupciones en nivel múltiple. Tiene varios modos de programación, permitiendo la optimización en una variedad de sistemas que emplean interrupciones.

El 8259A es totalmente compatible con el INTEL 8259. El software originalmente escrito para el 8259 puede emplearse por el 8259A, así como sus modos equivalentes de programación. Este circuito es compatible con el sistema iAPX86/88, controla la prioridad de ocho niveles de interrupción, es expandible a 64 niveles de interrupción, colocando ocho 8259As en cascada, tiene la capacidad de manejar el requerimiento de máscara en forma individual, no requiere entrada de reloj, consta de 28 terminales, de las cuales una de ellas tiene doble función.

b) Diagrama de Terminales.



c) Descripcion de Terminales.

Vcc. +5 Volts de suministro.

GND. Tierra.

CS (Chip Select). Un bajo en esta terminal habilita la comunicación de lectura (RD) y escritura (WR) entre el CPU y el 8259A. Un bajo en esta entrada habilita al 8259A. No puede hacerse ninguna lectura o escritura si el circuito no está habilitado. Las funciones de INTA son independientes de CS.

WR (Write). Un bajo en esta terminal cuando CS está en bajo habilita al 8259A para aceptar palabras de control (ICWs y OCWs) desde el CPU.

RD (Read). Un bajo en esta terminal cuando CS está en bajo habilita al 8259A para enviar los estados del Registro de Requerimiento de Interrupción (IRR), Registro en Servicio (ISR), el Registro de Interrupción Mascarable (IMR), o el nivel de interrupción al bus de datos.

D7-D0 (Bus de Datos Bidireccional). Información de control, estado y vector de interrupción es tranferida por vía de este bus.

CAS0-CAS2 (Líneas en Cascada). Las líneas CAS forman un bus interno del 8259A para el control de una estructura múltiple del 8259A. Estas terminales son salidas, para un 8259A maestro y entradas para un 8259A esclavo.

SP/EN (Programa Esclavo/Habilitación del Buffer). Esta es una terminal de doble función. Cuando está en modo de almacenamiento temporal esta terminal puede ser usada como una salida de control para habilitar el buffer (EN). Cuando no está en modo de almacenamiento temporal esta terminal es usada como una entrada para designar si el 8259A es un maestro (SP=1) o un esclavo (SP=0).

INT (Interrupción). Esta terminal se colocará en alto en el momento en que la petición de una interrupción válida es aceptada. Esta salida va directamente a la línea de entrada de interrupción del CPU. El nivel bajo en esta línea está diseñado para ser totalmente compatible con los niveles de entrada del 8088A,8085A y 8086.

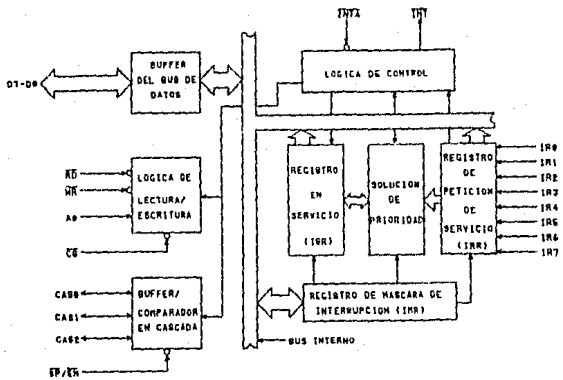
IR0-IR7 (Requerimiento de Interrupción). Son entradas asíncronas. Un requerimiento de una interrupción es ejecutada por el flanco positivo en una entrada IR, colocándose en alto hasta que es aceptada (modo de disparo por flanco), y sólo por un nivel alto en

una entrada IR (modo de disparo por nivel).

INTA (Interrupción Reconocida). Esta señal es usada para habilitar datos del vector de interrupciones del 8259A a traves del bus de datos, por una secuencia de pulsos de interrupciones reconocidas por el CPU. El formato de estos datos dependen del modo de operación actual del 8259A.

AO (Línea de Dirección). Esta señal actúa en conjunto con las señales CS, WR, RD. Es usada por el 8259A para descifrar varias palabras de comando escritas por el CPU y condiciones actuales del circuito que el CPU desea leer. Esta señal es generalmente conectada al CPU en su línea de dirección AO (A1 para IAPX 86, 88).

d) Diagrama de Bloques Interno.



e) Descripción del Diagrama de Bloques Interno.

Registro de requerimiento de petición (IRR) y Registro en Servicio (ISR). Las interrupciones en las líneas de entrada IR son manejadas por dos registros en cascada, el registro de requerimiento de interrupción (IRR) y el registro en servicio

(ISR). El IRR es usado para almacenar todos los niveles de interrupción cuando están requiriendo servicio; y el ISR es usado para almacenar todos los niveles de interrupción que están siendo atendidos.

Solución de Prioridad. Este bloque lógico determina las prioridades de los bits puestos en el IRR. La prioridad más alta es seleccionada y habilitada sobre el correspondinte bit de el ISR durante un pulso de INTA.

Registro de Interrupción Mascarable (IMR). El IMR almacena los bits de máscara de las líneas de interrupción a ser empleados. El IMR opera sobre el IRR. El enmascarar una entrada de alta prioridad no afectará los requerimientos de las líneas de interrupciones de prioridad inferior.

Buffer del Bus de Datos. Este es un buffer tres estados, bidireccional de 8-bits; es usado como interface entre el 8259A y el bus de datos del sistema. Las palabras de control y la información del estado son transferidas a través del buffer del bus de datos.

Lógica de Control Read/Write. La función de este bloque es aceptar los comandos de salida del CPU. Este contiene registros de palabras de comando de inicialización (ICW) y de operación (OCW), los cuales almacenan los diferentes formatos de control para la operación del dispositivo. Este bloque también permite que el estado del 8259A sea transferido al bus de datos.

Buffer/Comparador en Cascada. Este bloque almacena y compara los IDs (identificadores) de todos los 8259A usados en el sistema. Las tres terminales asociadas (CAS0-2) son salidas cuando el 8259A es usado como maestro, y son entradas cuando el 8259A es usado como esclavo. Como maestro, el 8259A manda el ID del dispositivo esclavo que está interrumpiendo sobre las líneas CAS0-CAS2. El esclavo que de esta forma es seleccionado enviará la dirección de la subrutina preprogramada al bus de datos durante el siguiente o siguientes dos pulsos de INTA consecutivos.

f) Función General del Circuito.

El Controlador de Interrupciones Programable (PIC) funciona como un manejador total, en el medio ambiente de un sistema que controla interrupciones. Acepta peticiones desde equipo periférico, determinando cuál de las peticiones que llegan es de más alta prioridad, resolviendo si la petición de llegada tiene una prioridad más alta que el nivel servido en ese momento, enviando una interrupción al CPU, basado en esta determinación.

Cada dispositivo periférico o estructura tiene un programa especial o rutina, la cual está asociada con su función especifica o petición operacional, referida como a una rutina de servicio.

Después de enviar una interrupción al CPU, la información de entrada a través del CPU, permite apuntar el Program Counter (PC) a la rutina de servicio que el dispositivo está requiriendo. Este apuntador es una dirección en la tabla de vectores, el cual podrá ser referido como un dato del vector respectivo.

La secuencia de interrupción que sigue el 8086 es la siguiente:

1. Una o más de las líneas de requerimiento de interrupción (IR7-0) son colocadas en alto poniendo los correspondientes bits en el interruptor Requirement Register (IRR).

2. El 8259A evalúa estas peticiones y envía una interrupción al CPU.

3. El CPU acepta la interrupción y responde con un pulso de INTA.

4. Un grupo de pulsos de INTA se reciben del CPU, el bit de prioridad más alta es colocado en el ISR y el correspondiente bit del registro IRR es apagado. El 8259A no controla al bus de datos durante este ciclo.

5. El microprocesador inicializará un segundo pulso de INTA. Durante este pulso, el 8259A colocará un apuntador de 8 bits en el bus de datos, que será leído por el CPU.

6. Esto completa el ciclo de interrupción. En el modo AEOI (Automatic End Of Interrupt) el bit del Registro en Servicio (ISR) es apagado, y termina el segundo pulso de INTA. En otro caso el bit del ISR permanece encendido hasta que un comando EOI (Fin De Interrupción) es enviado al final de la subrutina de interrupción.

A continuación se describirá la secuencia de interrupción de salida:

El primer ciclo de aceptación de una interrupción el 8259A lo usa para inutilizar internamente el estado de las interrupciones, para resolver su prioridad y como un maestro envia el código de interrupción en las líneas en cascada en el fin del pulso de INTA. En este primer ciclo no se envían los datos al procesador, y el buffer del bus de datos es inhabilitado.

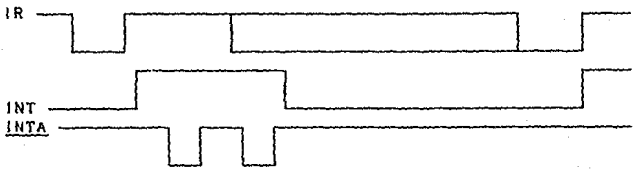
En el segundo ciclo de aceptación de una interrupción el maestro (o esclavo si así fue programado) envía un byte de datos al procesador con el código de la interrupción aceptada compuesta como sigue (NOTA: El estado del modo de control ADI es ignorado y las terminales de A5-A11 no son usadas):

Contenido del Vector de Interrupciones
para el sistema IAPX86

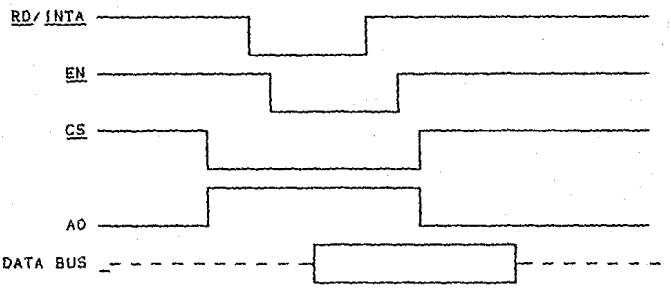
D7	D6	D5	D4	D3	D2	D1	D0	
T7	T6	T5	T4	T3	1	1	1	IR7
T7	T6	T5	T4	T3	1	1	0	IR6
T7	T6	T5	T4	T3	1	0	1	IR5
T7	T6	T5	T4	T3	1	0	0	IR4
T7	T6	T5	T4	T3	0	1	1	IR3
T7	T6	T5	T4	T3	0	1	0	IR2
T7	T6	T5	T4	T3	0	0	1	IR1
T7	T6	T5	T4	T3	0	0	0	IR0

g) Diagramas de Tiempo.

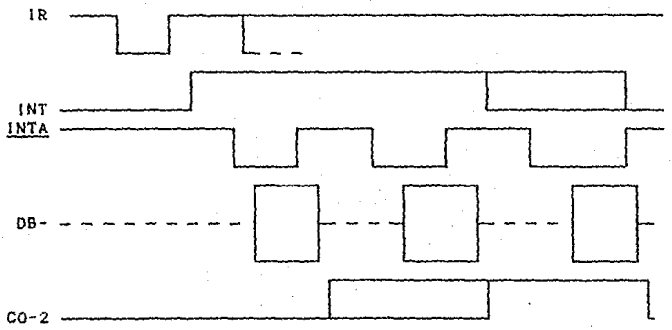
Requerimiento de Interrupción (IR).



Read/INTA

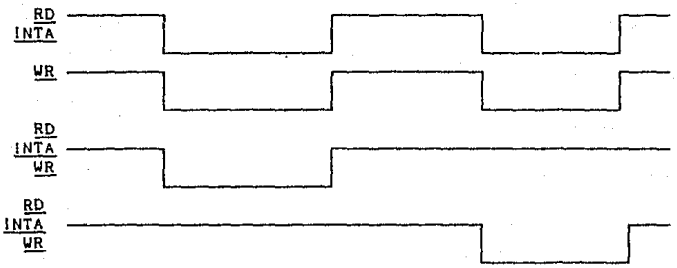


Secuencia INTA.



Nota: En el primer ciclo el bus de datos (DB) no es activado.

Otros Tiempos.



h) Programación del Circuito.

El 8259A acepta dos tipos de comandos generados por el CPU:

1.- Palabras de Comando de Inicialización. Antes de que comience la operación normal, cada 8259A en el sistema debe ser inicializado con una secuencia de 2 a 4 bytes sincronizados por pulsos de WR.

2.- Palabras de comando de operación. Estas son las palabras de comando que le indican al 8259A que opere en diferentes modos de interrupción. Estos modos son:

Modo de Anidamiento Completo.

El 8259A queda en este modo después de la inicialización, a menos que se le programe otro modo. Las peticiones de interrupción son ordenadas en forma de prioridades, de la 0 a la 7 (0 es la mas alta). Cuando una interrupción es reconocida la solicitud de mayor prioridad es elegida y su vector es colocado en el bus. Adicionalmente, un bit del registro ISR (ISR0-7) es encendido. Este bit permanece prendido hasta que el microprocesador manda un comando de fin de interrupción (EOI), inmediatamente antes de regresar de la rutina de servicio, o si el bit AEOI (Automatic End Of Interrupt) está prendido, hasta el flanco descendente del último INTA. Mientras el bit ISR está prendido, todas las

interrupciones posteriores de la misma o menor prioridad quedan inhibidas, mientras que los mayores niveles generan una interrupción (la cual será reconocida solamente si el flip-flop de habilitación interno del microprocesador ha sido encendido a través de software).

Después de la secuencia de inicialización, IR0 tiene la más alta prioridad e IR7 la más baja. Las prioridades pueden cambiarse como se explicará en el modo de prioridad rotante.

Modo de Prioridad Rotante.

El programador puede cambiar prioridades programando la más alta prioridad y entonces fijando todas las otras prioridades, por ejemplo, si IR5 se programa como el dispositivo de más baja prioridad, entonces IR6 tendrá la más alta. El comando para fijar la prioridad se proporciona en OCW2, donde: R=1, SL=1; L0-L12 es el código binario del nivel del dispositivo con la más baja prioridad.

Modo de Enmascaramiento Especial.

Algunas aplicaciones pueden requerir una rutina de servicio de interrupción que dinámicamente altere la estructura de prioridades del sistema durante su ejecución bajo control de software. Por ejemplo, la rutina podría querer inhibir las solicitudes de baja prioridad durante una porción de su ejecución pero habilitar alguna de ellas durante otra petición.

La dificultad aquí, es que si una solicitud de interrupción es reconocida y un comando EOI no apaga la señal IS, el 8259A inhibe todas las solicitudes de baja prioridad sin manera de que fácilmente las rutinas las habiliten.

Aquí es donde ayuda el modo de enmascaramiento especial. En este modo cuando se pone un bit de máscara en OCW1, inhibe las subsecuentes interrupciones a ese nivel y habilita las interrupciones de todos los otros niveles (mayores o menores) que no están enmascaradas.

Así, cualquier interrupción puede ser selectivamente habilitada cargando el registro de máscara.

El modo especial de enmascaramiento es seleccionado por OCW3 donde: SSMM=1; SMM=1; y deshabilitado cuando SSMM=1; SMM=0.

Modo de Encuesta.

En este modo la salida INT no es usada o el flip-flop de habilitación interno del microprocesador es apagado. El servicio a los dispositivos es realizado por software, utilizando el comando de encuesta.

El comando de encuesta es seleccionado poniendo P=1 en OCW3. El 8259A trata el siguiente pulso RD del 8259A como un reconocimiento de interrupción, prende el bit de IS apropiado si hay una solicitud, y lee el nivel de prioridad. La interrupción queda congelada desde WR hasta RD.

Este modo es útil si hay una rutina de comando común a varios niveles, de modo que la secuencia INTA no se necesita (ahorra espacio en ROM). Otra aplicación es usar el modo de encuesta para expander el número de niveles de prioridad a más de 64.

La palabra habilitada a través del bus de datos durante RD es:

D7	D6	D5	D4	D3	D2	D1	D0
I	-	-	-	-	W2	W1	W0

W0-W2 es el código binario de la prioridad más alta del nivel requiriendo servicio.

I : Es igual a 1 si hay una interrupción.

PALABRAS COMANDO DE INICIALIZACION (ICWs)

Siempre que un comando haya sido enviado con A0=1 Y D4=1 será interpretado como la primera palabra comando de inicialización (ICW1). ICW1 principia durante la secuencia de inicialización, ocurriendo automáticamente lo siguiente:

a) El circuito es reinicializado en la entrada de la petición de una interrupción (IR), tendrá que existir una transición de un bajo a un alto para generar una interrupcion.

b) El Registro Mascarable de interrupción es limpiado.

c) A la entrada IR7 se le asigna la prioridad 7.

d) La dirección de modo esclavo es puesto a 7.

e) El modo Especial Mascarable es limpiado y el estado de lectura es colocado al IRR.

f) Si IC4=0, todas aquellas funciones seleccionadas en la cuarta palabra comando de inicialización son puestas a cero.

A continuación se describirá cuál es el formato de cada palabra comando de inicialización:

Primera y Segunda Palabra Comando de Inicialización.

En el sistema iAPX86 los valores de las líneas A15-A11 son insertados en los 5 bits más significativos del byte vectorial, y el 8259A pone los bits menos significativos de acuerdo al nivel de interrupción. De la A10-A5 son ignorados y ADI (Address Interval) no tiene efecto.

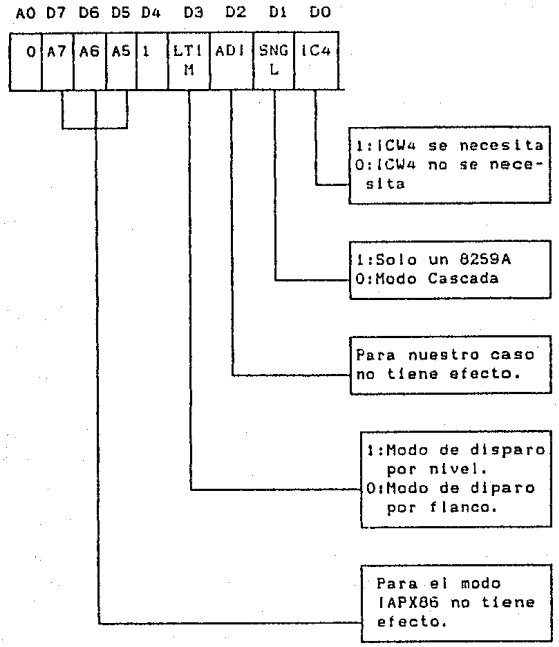
LTIM: Si LTIM =1 el 8259A opera en el modo de interrupción por nivel. La detección lógica de flancos en las entradas de interrupción será inhabilitada.

ADI: No tiene efecto para el microprocesador iAPX86.

SNGL: Si SNGL=1 la tercera palabra comando de inicialización no deberá enviarse, ya que este uno nos indica que sólo un 8259A se está utilizando.

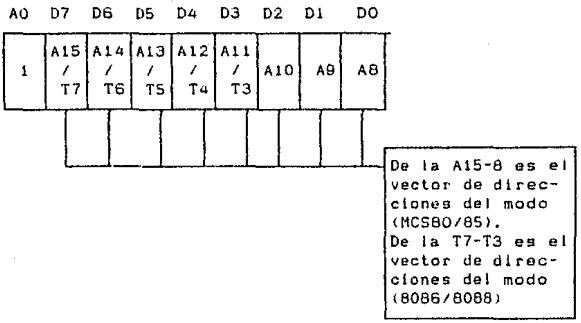
IC4: Si IC4=1 la cuarta palabra comando de inicialización tiene que ser leída. Si la cuarta palabra de inicialización ICW4 no se necesita, entonces IC4=0.

La primera palabra comando es interpretada como sigue:



La segunda palabra comando se interpreta de la siguiente manera:

ICW2



Tercera Palabra Comando de Inicialización (ICW3).

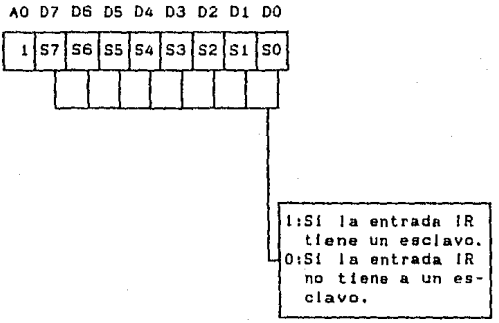
Esta palabra es leída, sólo cuando hay más de un 8259A en el sistema, y es usado en cascada en el caso en que SNGL=0. Se cargará el registro esclavo de 8-bits. El funcionamiento de estos registros es:

a) En el modo maestro (ya sea que SP=1 o M/S=1 en ICW4) un uno es colocado por cada esclavo en el sistema. El maestro habilitará el correspondiente esclavo para cargar 2 bytes a través de las líneas en cascada.

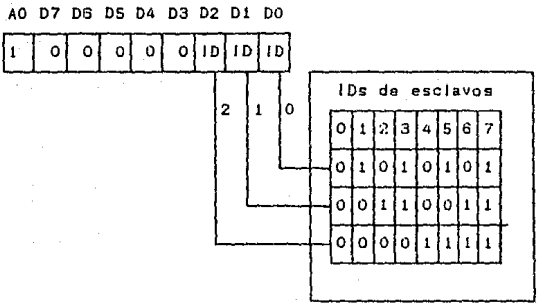
b) En el modo esclavo (ya sea cuando SP=1 o si BUFF=1 y M/S=0 en ICW4) los bits 2-0 identifican al esclavo. El esclavo compara las entradas en cascada con estos bits, y si son iguales, 2 bytes de la secuencia de llamada son cargados en el bus de datos.

La tercera palabra comando se interpreta de la siguiente manera:

ICW3. Dispositivo Maestro



ICW3. Dispositivo Esclavo



Cuarta palabra comando de inicialización (IWC4).

SFNM: Si SFNM=1 el modo de anidamiento fué programado.

BUF: Si BUF=1 el modo de almacenamiento temporal fue programado.

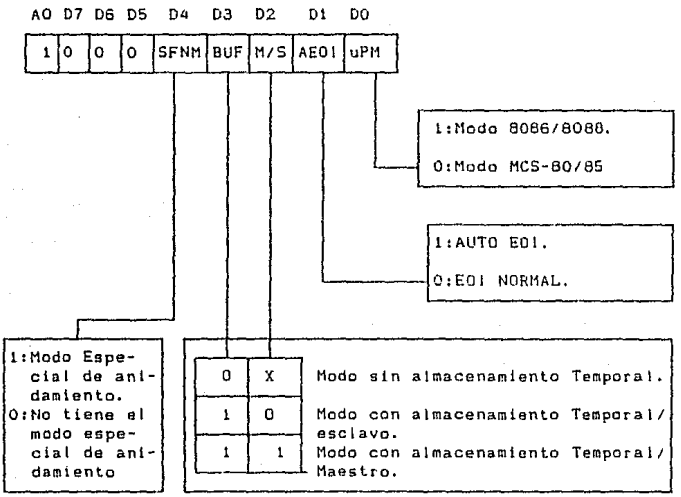
M/S: Si el modo de almacenamiento temporal fue seleccionado, M/S=1 si el 8259A está programado para ser maestro, M/S=0 si el 8259A está programado para ser esclavo. Si BUFF=0 la señal M/S no tiene función.

AEOI: Si AEOI=1 el modo automático del fin de una interrupción fué programado.

uPM: Modo del Microprocesador, si uPM=0 está encendido el 8259A está habilitado para operar con el sistema (MCS-

80, 85), si uPM=1 el circuito esta programado para operar con el sistema (iAPX86).

La cuarta palabra comando se interpreta de la siguiente manera:



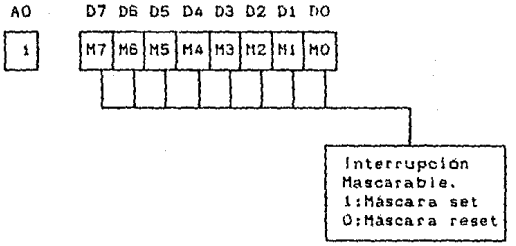
A continuación se describirán las Palabras Comando de Operación (OCWs):

Después que las Palabras Comando de Inicialización (ICWs) son proporcionadas al 8259A, el circuito está listo para aceptar requerimientos de interrupciones en sus líneas de entrada. Durante la operación del 8259A, una selección de algoritmos pueden ayudar al 8259A para operar en varios modos a través de las Palabras Comando de Operación (OCWS).

Primera Palabra Control de Operación (OCW1).

OCW1 prende y apaga los bits de máscara en el Registro Mascarable de Interrupción (IMR). M7-M0 representan los 8 bits mascarables. M=1 indica que el canal está mascarado (inhabilitado) M=0 indica que el canal está habilitado.

Su vector queda como sigue:

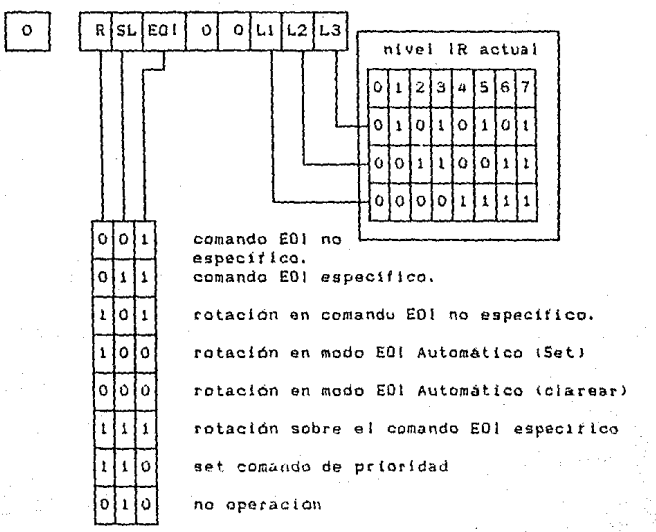


Segunda Palabra Control de Operación (OCW2).

R,SL,EOI.- Estos tres bits controlan los modos de rotación y fin de interrupción y la combinación de los dos.

L1,L2,L3- Estos bits determinan el nivel de interrupción actual, cuando el bit SL esta activo.

Su vector queda como sigue:



Las primeras dos combinaciones de R,SL,EOI corresponden al fin de una interrupción.

La tercera, cuarta y quinta corresponden a la rotación automática.

La sexta y septima corresponden a una rotación especifica.

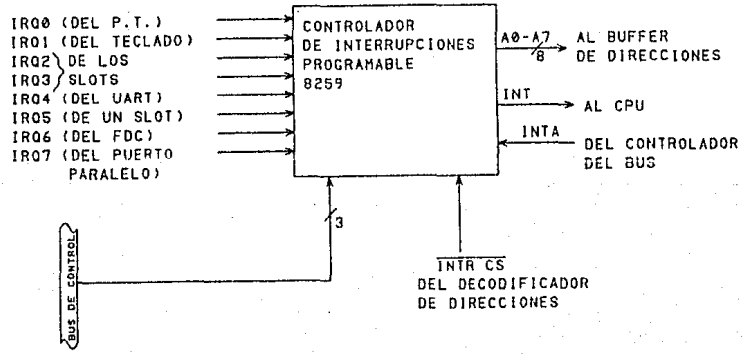
Tercera Palabra Control de Operación(OCW3).

ESMM- Habilita el modo especial Mascarable. Cuando este bit está puesto a uno, habilita el bit SMM para prender o apagar el modo especial mascarable. Cuando ESMM=0 el bit SMM no tiene efecto.

SMM- Modo especial Mascarable. Si ESMM=1 y SMM=1 el 8259A entrara en modo especial mascarable. Si ESMM=1 y SMM=0 el 8259A regresará al modo mascarable normal.

i) Interconexión del Circuito en una PC.

El 8259A controla prioridades y señales de servicio del requerimiento de interrupciones desde dispositivos e interfaces I/O, para que este circuito pueda realizar estas operaciones se apoya en los circuitos del medio ambiente como son el CPU 8086, el buffer de direcciones, controlador del bus, así como cada señal que venga de teclado, puerto serie (UART), puerto paralelo, disco duro y flexible, a continuación se presenta dicha conexión:



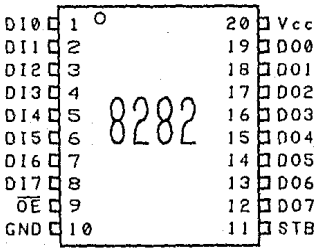
A.3. LATCH OCTAL 8282/8283 .

a) Introducción.

El 8283 y el 8283 son latches bipolares (8 bits) con buffers de salida 3 estados. Ellos pueden ser usados para implementar latches, buffers o multiplexores. El 8283 invierte los datos de entrada en sus salidas, mientras que el 8282 no lo hace.

De este modo, todas las funciones principales periféricas y de I/O de una microcomputadora, se pueden implementar con estos dispositivos.

b) Diagrama de Terminales.



c) Descripción de terminales.

STB (Strobe). STB es un pulso de control de entrada usado para habilitar datos en las terminales de entrada de datos (terminales A0 - A7) hacia los latches de datos.

Esta señal es activada en ALTO para admitir datos de entrada. El dato es guardado en la transición de ALTO a BAJO de STB.

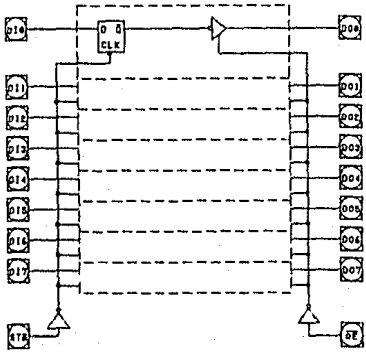
OE : (Output Enable). OE es una señal de control de entrada, cuando se activa en BAJO, permite que el contenido de los latches de datos pasen a la terminal de salida de datos (terminales BO - B7) .

OE estando inactiva en ALTO forza a los buffers de salida a su estado de alta impedancia.

DIO - DI7 (Input Data Pins). Los datos presentados en estas terminales satisfacen los requerimientos de arreglos de tiempo cuando STB es habilitado y guardado en los latches de entrada de datos.

DOO - DO7 (Output Data Pins). Cuando OE es verdadero, el dato en el latch de datos es presentado invertido (8283) o no invertido (8282) en las terminales de salida de datos.

d) Diagrama de Bloques Interno.



e) Función General del Circuito.

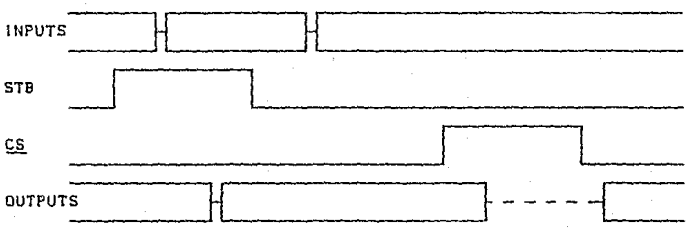
Habiendo satisfecho los requerimientos del arreglo de tiempo con los datos , se almacenan en los latches de datos, por medio de la habilitación de la línea STB de ALTO a BAJO.

Manteniendo la línea de STB en su estado ALTO hace que los latches aparezcan transparentes.

El dato es presentado en las terminales de salida de datos por medio de la activación de la línea de entrada OE. Cuando OE está inactivo ALTO, los buffers de salida están en su estado de alta impedancia.

Habilitando o deshabilitando los buffers de salida, no causará que transitorios negativos aparezcan en el bus de salida de datos.

f) Diagrama de Tiempos.

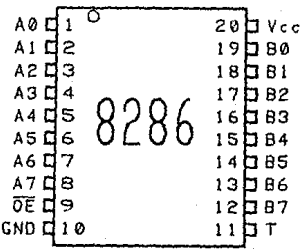


A.4. BUS TRANSCEPTOR OCTAL 8286/8287.

a) Introducción.

El 8286 y 8287 son transceptores bipolares de 8 bits con salidas de 3 estados. El 8287 invierte la entrada de datos en sus salidas, mientras que el 8286 no lo hace. Por lo tanto, se puede encontrar una variedad de aplicaciones de interconexión en los sistemas de microcomputadora.

b) Diagrama de Terminales.



c) Descripción de Terminales.

T (Transmit). T es una señal de control de entrada, usada para controlar la dirección de los transceptores. Cuando está en ALTO, configura los transceptores B0 - B7 como salidas y los A0 - A7 como entradas.

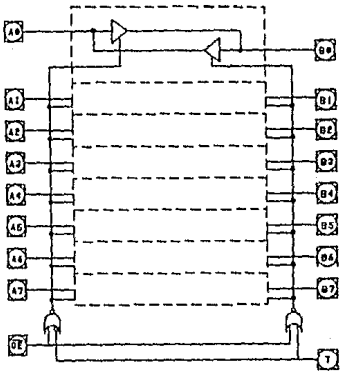
T en bajo configura A0 - A7 como salidas con B0 - B7 como entradas.

OE (Output Enable). OE es una señal de control de entrada usada para habilitar el transporte de salida apropiado (seleccionado por T) hacia el bus respectivo. Esta señal es activada en BAJO.

A0 - A7 (Local Bus Data Pins). Estas terminales sirven ya sea para presentar datos, como para aceptarlos, a partir del bus local del procesador dependiendo del estado de la terminal T.

B0 - B7 (System Bus Data Pins). Terminales que sirven tanto para presentar como para aceptar datos del bus del sistema, dependiendo del estado de la terminal T.

d) Diagrama de Bloques Interno.

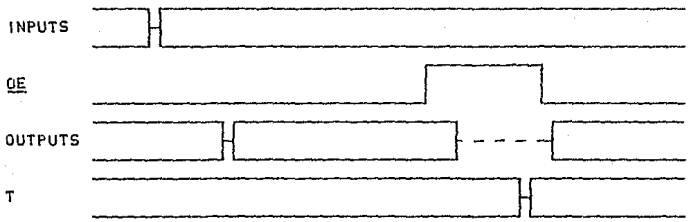


e) Función General del Circuito.

El 8286 y 8287 , son transceptores de 8 bits con salidas de alta impedancia. Con T activo en ALTO Y OE activo en BAJO , los datos en las terminales A0 - A7 son dirigidos hacia las terminales B0 - B7. Con T inactivo en BAJO y OE activado en BAJO, los datos en B0 - B7 son dirigidos hacia las terminales A0 - A7.

No ocurrirá ningún estado transitorio de salida siempre y cuando los transceptores estén entrando o saliendo del estado de alta impedancia.

f) Diagrama de Tiempo.



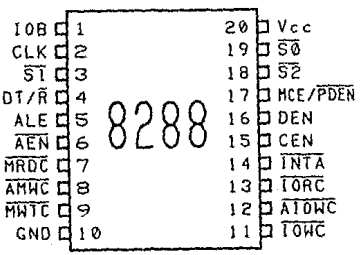
A.5. CONTROLADOR DE BUS 8288.

a) Introducción.

El controlador de bus 8288 de Intel es un componente bipolar de 20 terminales para usarse con los sistemas de procesado medianos y grandes iAPX 86,88.

El controlador de bus provee comandos y control de generación de tiempo, así como capacidad de manejar el bus bipolar, mientras que se optimiza el rendimiento del sistema.

b) Diagrama de terminales.



c) Descripción de Terminales.

Vcc. Poder; Suministro de + 5 V.

GND. Tierra.

S0,S1,S2 (Status Input Pins). Terminales son terminales de entrada de estado para los procesadores 8086,8088 u 8089.El 8288 decodifica estas entradas para generar comandos y señales de control en un tiempo apropiado. Cuando estas terminales no son usadas (estado pasivo) están todas en ALTO.

CLK (Clock). Es la señal de reloj, proveniente del generador 8284, sirve para establecer cuando las señales de control y comando son generadas.

ALE (Address Latch Enable). Esta señal sirve para habilitar una dirección dentro de los latches direccionadores. Es activa ALTO y el almacenamiento ocurre en la transición (ALTO -BAJO). ALE está pensada para usarse con latches tipo D, transparentes.

DEN (Data Enable). Esta señal sirve para habilitar los transceptores de datos en cualquier bus de datos local o bus del sistema. Esta señal es activa Alto.

DT/R (Data Transmit/Receive). Señal que establece la dirección del flujo de datos a través de los transceptores. Un ALTO en esta linea indica Transmision (escritura a I/O o memoria) y un BAJO indica recepción (lectura).

AEN (Address Enable). AEN habilita comandos de salida del controlador de bus 8288 en 115 ns después de haberlo activado (BAJO). AEN inactiva inmediatamente con 3 estados los comandos de transporte de salida. AEN no afecta las lineas de comando de I/O si el 8288 está en el modo BUS I/O. (IOB en ALTO)

CEN (Command Enable). Cuando esta señal está en bajo todas las salidas de comando del 8288 , DEN y las salidas de control PDEN son forzados a su estado inactivo. Cuando esta señal esta en ALTO, estas mismas salidas son habilitadas.

IOB (I/O Bus Mode). Cuando el IOB está en ALTO el 8288 funciona en modo BUS I/O. Cuando está en BAJO, el 8288 está en modo BUS del sistema.

AIOWC (Advanced I/O Write Command). El AIOWC permite que el comando de escritura I/O, en el ciclo inicial de máquina, dar a los dispositivos de I/O una indicación temprana o una instruccion de escritura. Su tiempo es el mismo al de una señal de comando de lectura . AIOWC es activa BAJO.

IOWC (I/O Write Command). Esta línea de comando instruye a un dispositivo I/O, para leer los datos del bus de datos. Esta señal es activa BAJO.

IORC (I/O Read Command). Esta línea de comando instruye a un dispositivo de I/O para que transporte sus datos hacia el bus de datos. Esta señal es activa en BAJO.

AMWC (Advanced Memory Write Command). El AMWC permite un comando de escritura de memoria inicial en el ciclo de máquina para darle

a los dispositivos de memoria una indicación temprana de instrucción de escritura. Su tiempo es el mismo de una señal de comando de lectura. AMWC es activa BAJO.

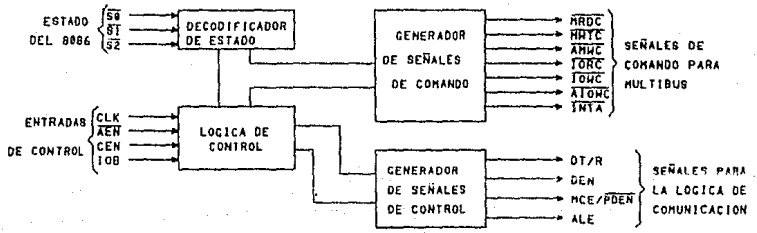
MWTC (Memory Write Command). Esta línea de comando instruye a la memoria para que grabe los datos presentes en el bus de datos. Esta señal es activa BAJO.

MRDC (Memory Read Command). Línea de comando que instruye a la memoria para que transporte datos a su bus de datos. Esta señal es activa BAJO.

INTA (Interrupt Acknowledge). Línea de comando que le indica al dispositivo que interrumpe, que su interrupción ha sido reconocida, y que él deberá transportar información vectorial hacia el bus de datos. Esta señal es activa BAJO.

MCE/PDEN (Master Cascade Enable/Peripheral Data Enable). (IOB esta en BAJO). La habilitación de cascada maestra ocurre durante una secuencia de interrupción y sirve para leer una dirección de cascada del PIC maestro (Controlador Prioritario de interrupción) hacia el bus de datos. Esta señal es activa ALTO. Esta señal habilita al transceptor del bus de datos de I/O, para que funcione el bus del sistema. PDEN es activo BAJO.

d) Diagrama de Bloques Interno.



e)Función General del Circuito.

COMANDO Y CONTROL LOGICO.

El comando lógico decodifica las 3 lineas de estado de CPU 8086,8088 u 8089. (S0,S1,S2) , para determinar que comando será usado.

El comando es usado en una de dos formas dependiendo del modo del controlador de BUS 8288.

MODO BUS I/O.

El 8288 está en modo de bus I/O, si la terminal IOB es activada ALTO. En el modo BUS I/O todas las lineas de comando I/O (IORC,IOWC,AIOWC,INTA) están siempre habilitadas. Cuando un comando I/O es inicializado por el procesador, el 8288 inmediatamente activa las lineas de comando usando PDEN y DT/R para controlar el bus transceptor de I/O.

Las lineas de comando I/O no deberán ser usadas para controlar el bus del sistema en esta configuración, porque no hay arbitraje presente.

Este modo permite que un controlador de bus 8288 maneje dos buses externos. Ninguna espera está involucrada cuando el CPU necesita ganar acceso hacia el bus I/O.

El acceso a memoria normal requiere una señal BUS LISTO (AEN BAJO) antes de proceder.
Es ventajoso usar el modo IOB, si I/O o periféricos dedicados a un procesador existen en un sistema multiprocesador.

MODO BUS SISTEMA.

El 8288 está en el modo de bus del sistema si IOB está en BAJO.
En este modo ningun comando está disponible hasta 115 ns despues de que la línea AEN está activada en BAJO.

Este modo asume que la lógica de arbitraje de bus informará al controlador de bus (en la línea AEN) cuando el bus está libre para usarse.

Ambos comandos de I/O y memoria esperan para el arbitraje de bus. Este modo es usado cuando solamente existe un bus. De aqui que ambos I/O y la memoria son compartidos por más de un procesador.

COMANDOS DE SALIDA.

Los comandos de escritura avanzados son para iniciar procedimientos tempranos en el ciclo de máquina.

Esta señal puede ser usada para prevenir que el procesador entre en un estado de espera innecesario.

Los comandos de salida son:

MRDC : Comando lectura memoria.
MWTC : Comando escritura memoria.
IORC : Comando lectura I/O.
IOWC : Comando escritura I/O.
AMWC : Comando avanzado escritura I/O .
INTA : Conocimiento de interrupción .

INTA actua como lectura I/O durante un ciclo de interrupción. Su proposito es informar a un dispositivo de interrupción que dicha interrupción está siendo conocida y que deberá colocar información vectorial en el bus de datos.

SALIDAS DE CONTROL.

Las salidas de control del 8288 son habilitación de datos (DEN), Transmisión/Recepción de datos (DT/$\overline{R}$) y habilitación de cascada maestra/habilitación de datos periféricos (MCE/$\overline{PEN}$).

La señal DEN determina cuando el bus externo debe habilitar el bus local y el DT/$\overline{R}$ determina la dirección de la transferencia de datos. Estas señales usualmente van a la dirección de las terminales de un transceptor.

La terminal MCE/$\overline{PDEN}$ cambia su funcion con los dos modos del 8288. Cuando el 8288 está en el modo IOB (IOB ALTO), la señal $\overline{PDEN}$ sirve como una señal dedicada habilitadora de datos para el bus del sistema periferico o I/O.

CONOCIMIENTO DE INTERRUPCION Y MCE.

La señal MCE es usada durante un ciclo de conocimiento de interrupción, si el 8288 está en el modo bus del sistema (IOB en BAJO). Durante cualquier secuencia de interrupción hay dos ciclos de conocimiento de interrupción que ocurren uno detrás de otro.

Durante el primer ciclo de interrupción, no hay transferencia de datos o direcciones.

La lógica debe poder enmascarar o desenmascarar MCE durante este ciclo. Justamente antes de que comience el segundo ciclo, la señal MCE da paso a que una cascada de controlador de interrupción prioritario (PIC) direccione hacia el bus local del procesador, donde ALE habilita hacia los latches de direcciones.

En el segundo ciclo de interrupción el PIC esclavo de direccionamiento da paso a un vector de interrupción hacia el bus de datos del sistema, donde se lee por el procesador.

Si el sistema contiene solamente un PIC, la señal no es usada. En este caso la señal de conocimiento de interrupción da paso al vector de interrupción hacia el bus del procesador.

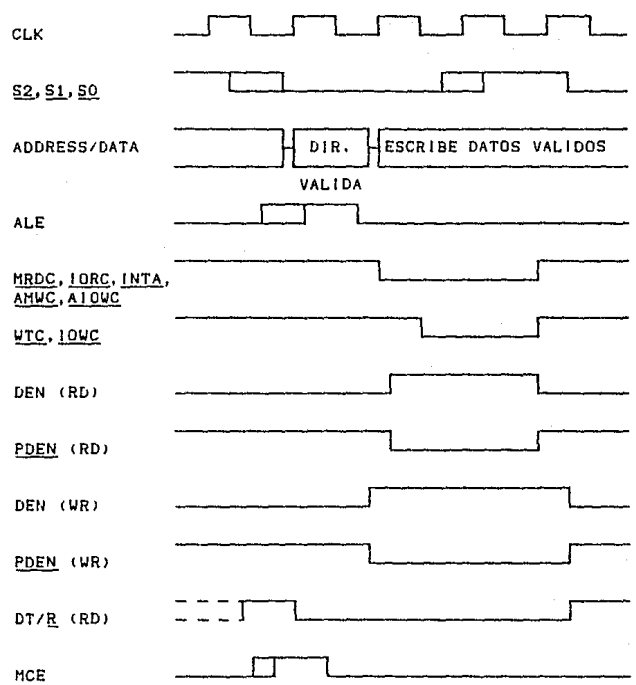
HABILITACION DEL LATCH DE DIRECCION Y ALTO.

ALE ocurre durante cada ciclo de máquina y sirve para habilitar la dirección de corriente en los latches direccionadores. ALE también sirve para habilitar el estado (S0, S1, S2) hacia un latch para la decodificación del estado ALTO.

COMANDO DE HABILITACION.

La entrada del comando de habilitación (CEN) actúa como un comando habilitador para el 8288. Si la terminal CEN está en ALTO el 8288 funciona normalmente. Si la terminal CEN está en BAJO, todas las líneas de comando son mantenidas en su estado inactivo (no en 3 estados). Esta característica puede ser usada para implementar particiones en memoria y para eliminar conflictos de dirección entre los dispositivos del bus del sistema y dispositivos del bus residente.

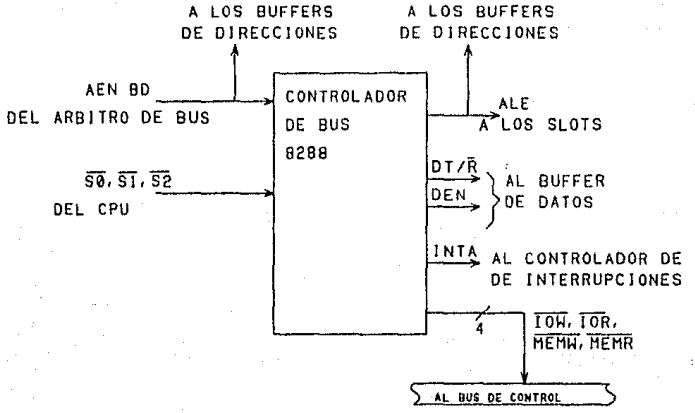
f) Diagramas de Tiempos.



g) Interconexión del Circuito en una PC.

El controlador de bus 8288 decodifica las líneas de estado S0, S1, S2 del CPU y genera las señales para controlar el bus. También genera las señales ALE (Address Latch nable), DEN (Data Enable) y DT/R (Data Transmit/Receive) que son usadas para demultiplexar los buses del CPU.

Su colocación física en una PC se muestra a continuación:


A LOS BUFFERS
DE DIRECCIONES
A LOS BUFFERS
DE DIRECCIONES
AEN BD
DEL ARBITRO DE BUS
CONTROLADOR
DE BUS
8288
ALE
A LOS SLOTS
S0, S1, S2
DEL CPU
DT/R
DEN
AL BUFFER
DE DATOS
INTA
AL CONTROLADOR DE
DE INTERRUPCIONES
4
IOW, IOR,
MEMW, MEMR
AL BUS DE CONTROL

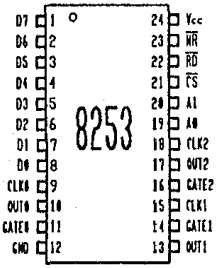
A.6. TIMER PROGRAMABLE 8253

a) Introducción.

El circuito empleado en las PCs es el 8253, el cual es un contador/timer con intervalos programables, diseñado para usarse como periférico a un microprocesador de la familia Intel. Posee tecnología nMOS, la cual requiere una fuente de voltaje de +5 Volts.

Está constituido por tres contadores independientes de 16 bits, cada uno tiene una tasa de hasta 2.6 MHz. Es programable en todos sus modos de operación mediante software.

b) Diagrama de Terminales.



c) Descripción de Terminales.

RD (Read). Un voltaje "bajo" en esta terminal indica al circuito que el CPU está aceptando un dato en forma de un valor de algún contador.

WR (Write). Un voltaje "bajo" en esta terminal de entrada indica al 8253 que el CPU está proporcionando un modo de operación o

cargando a los contadores.

A0, A1. Estas terminales de entrada son normalmente conectadas al bus de direcciones. Su función es seleccionar uno de los tres contadores a ser operado y direccionar el registro de la palabra de control, para seleccionar el modo de operación.

CS (Chip Select). Un voltaje "bajo" en esta entrada habilita al 8253. No ocurrirán lecturas o escrituras a menos que se dé la selección del circuito. El estado de la terminal CS no tiene efecto sobre la operación actual de los contadores.

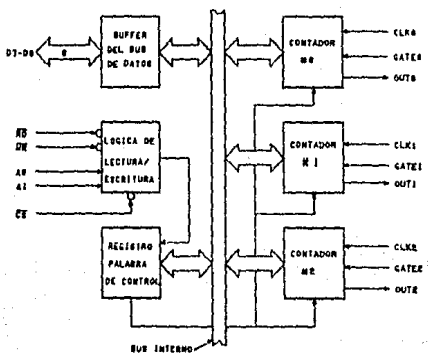
CLKn (Clock n). Terminal de entrada que sirve como reloj al contador n-ésimo. Esta terminal puede conectarse a un generador de pulsos para usarse como divisor de frecuencias, o a un dispositivo externo, sirviendo como contador de eventos.

GATEn. Terminal de entrada que se emplea para sincronizar y comenzar cuentas de los relojes con eventos externos al circuito.

OUTn. Terminal de salida, que proporciona pulsos cuya duración y frecuencia depende del modo de operación.

D7-D0. Terminales de entrada/salida que permiten leer cuentas del circuito o datos de operación del circuito.

d) Diagrama de Bloques Interno.



e) Descripción del Diagrama de Bloques Interno.

Buffer del bus de datos.

Este es un buffer tres estados, de ocho bits, usado como interface del 8253 al bus de datos. Los datos son transmitidos o recibidos mediante la ejecución de instrucciones INput y OUTput. Este módulo tiene tres funciones básicas:

1. Programar los modos de funcionamiento del 8253.
2. Carga de los registros de los contadores.
3. Lectura del valor de los contadores.

Lógica de Lectura/Escritura.

Esta lógica acepta entradas desde el bus de datos y las convierte en señales de control para la ejecución del 8253. Es habilitada o desabilitada por la terminal CS, por lo que no pueden realizarse operaciones en el circuito a menos que sea explícitamente seleccionado.

Registro de la Palabra de Control.

Este registro se selecciona cuando en las terminales A0 y A1 hay (en ambas) un uno. La información es tomada del buffer del bus de datos y se almacena en un registro. La información almacenada controla el modo de operación de cada contador, selecciona una cuenta en BCD o binario y la carga de cada registro de los contadores.

El registro de la palabra de control puede ser escrito, pero no es permitida una operación de lectura.

Contador #1, Contador #2, Contador #3.

Los tres bloques son idénticos, por lo que la explicación se hace extensiva a cualquiera de ellos. Cada uno es un contador descendente de 16 bits, pre-programable. El contador puede operar en modos BCD y binario y las líneas INPUT, GATE y OUTPUT son configuradas al seleccionar el modo, almacenado en el registro de la palabra de control.

Los contadores son totalmente independientes, y cada uno se puede trabajar en un modo distinto. La lectura del contenido de cada contador es posible con una operación de lectura. Esto último se realiza sin detener o inhibir la cuenta de los contadores.

f) Función General del Circuito.

Este circuito es un contador/timer de intervalos programables, diseñado para usarse en sistemas de microcomputadoras de la familia Intel. Es de propósito general, sus múltiples elementos pueden ser tratados como un arreglo de puertos de entrada/salida, y programarse como tal.

El 8253 resuelve el problema común de la generación de retardos programables bajo control de software. Pueden realizarse ciclos de cuenta para llenar este requerimiento. Puede programarse la cuenta de uno de los contadores e interrumpir al microprocesador cuando se termine la cuenta.

Otra de las funciones del circuito que no son propiamente retardos, pero que son ampliamente empleados en microcomputadoras son:

- Generador de tasa programable.
- Contador de eventos.
- Multiplicador de tasas binarias.
- Reloj en tiempo real.
- One-shot (disparo) digital.
- Controlador de motores complejos.

g) Programación del circuito.

Todos los modos de operación de los contadores se programan mediante software, usando operaciones simples de entrada/salida.

Cada contador del 8253 es programado individualmente escribiendo en el registro de la palabra de control (A0,A1=11). El formato de la palabra de control es:

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

SC significa contador seleccionado, de tal forma que los valores de estos bits indican:

SC1 SC0

0	0	Contador #0 seleccionado
0	1	Contador #1 seleccionado
1	0	Contador #2 seleccionado
1	1	Ilegal

Los bits RL (read/load) proporcionan:

RL1 RL0

0	0	Operación de retención del contador
0	1	Lee/carga sólo byte más significativo
1	0	Lee/carga sólo byte menos significativo
1	1	Lee/carga byte menos significativo, y a continuación byte más significativo

Los bits M (modo) implican:

M2	M1	M0	
0	0	0	Modo 0
0	0	1	Modo 1
*	1	0	Modo 2
*	1	1	Modo 3
1	0	0	Modo 4
1	0	1	Modo 5

Bit BCD:

0	Cuenta binaria, 16 bits
1	Cuenta en BCD (4 décadas)

Carga del contador:

El registro de cuenta es inicializado hasta que el valor de la cuenta es escrito (uno o dos bytes, dependiendo del modo seleccionado con los bits RL), seguido de un frente positivo y negativo del reloj. Cualquier lectura del contador anterior al

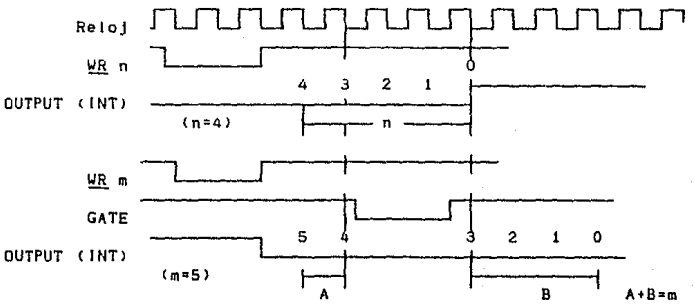
frente descendente del reloj puede producir un dato inválido.

MODOS DE OPERACION.

MODO 0: Interrumpe al terminar la cuenta. La salida es inicialmente baja, después de seleccionar el modo de operación. Después de colocar la carga en el registro del contador seleccionado, la salida permanece baja y el contador empieza a funcionar. Cuando termina la cuenta la señal se coloca alta y permanece así hasta que el registro del contador es cargado con otro modo o una nueva cuenta. El contador continua decrementando después de que la cuenta final ha sido alcanzada.

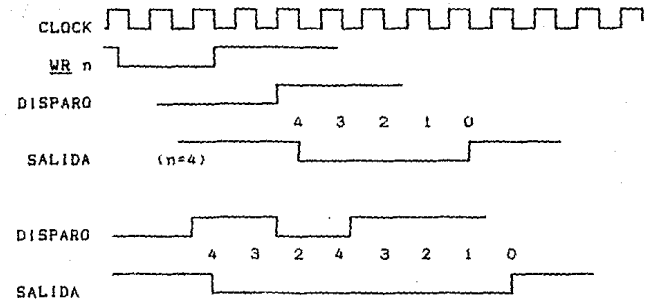
Sobre escribir un registro del contador durante la cuenta resulta en lo siguiente:

1) Escribir el primer byte detiene la cuenta actual.
2) Escribir el segundo byte empieza la nueva cuenta.



MODO 1: Un-disparo programable. La salida es baja en la cuenta seguida a un frente positivo en la entrada GATE.

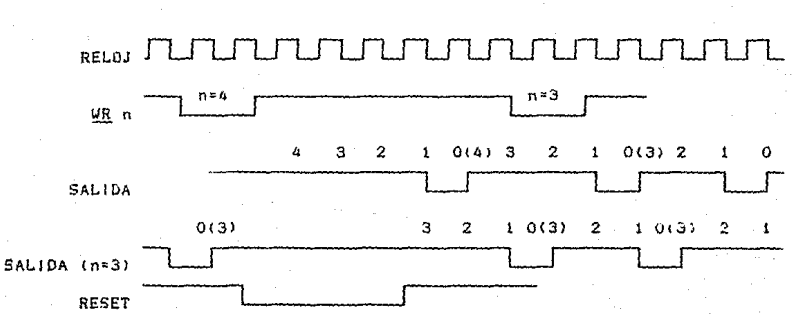
La salida va alta al terminar la cuenta. Si un nuevo valor de la cuenta es cargado mientras la salida es baja no afectará la duración del pulso único hasta el siguiente disparo. La cuenta actual puede leerse en cualquier tiempo sin afectar el pulso de salida.



MODO 2: Generador de frecuencias. Divide por N la cuenta. La salida es baja en un periodo del reloj de entrada. El periodo de un pulso de salida al otro es igual a la cuenta colocada en el registro de cuenta. Si el registro de cuenta es sobre escrito entre los pulsos de salida el presente periodo no es afectado, pero los periodos subsecuentes serán de acuerdo al nuevo valor.

La entrada GATE, cuando baja, forza la salida a alto. Cuando la entrada GATE va a alto el contador comienza desde la cuenta inicial. Así, la entrada GATE puede emplearse para sincronizar los contadores.

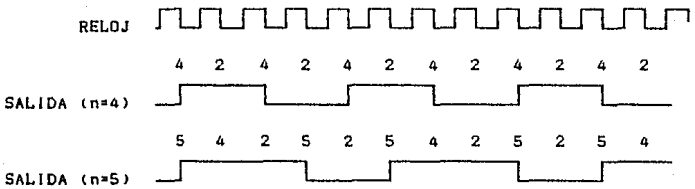
Cuando se selecciona este modo de operación, la salida es mantenida alta hasta que el registro de cuenta es cargado. La salida puede entonces ser sincronizada por software.



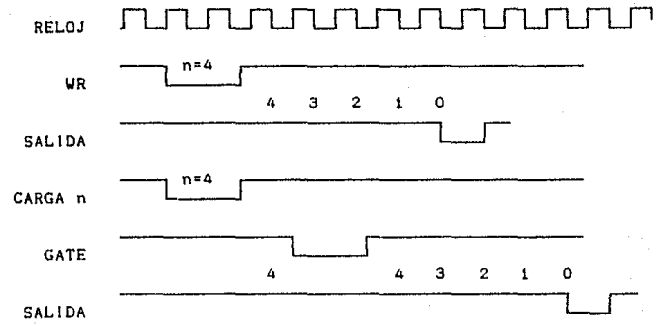
MODO 3: Generador de ondas cuadradas. Este modo es similar al 2, excepto que la señal permanece alta hasta que la mitad de la cuenta se completa (para números pares) y pasa a baja en la otra mitad de la cuenta. Esto se logra al decrementar de dos en dos la cuenta del contador con cada pulso negativo del reloj. Cuando la cuenta se termina el estado de la salida cambia y el contador es de nuevo cargado con el valor original.

Si la cuenta es non y la salida es alta, el primer pulso del reloj (después de la carga) decrementa el contador por 1. Los siguientes pulsos del reloj decrementan la cuenta por dos. Posteriormente la cuenta va a bajo y la cuenta completa es de nuevo cargada. El primer pulso (después de la recarga) decrementa el contador por tres, posteriores pulsos del reloj la decrementan por dos.

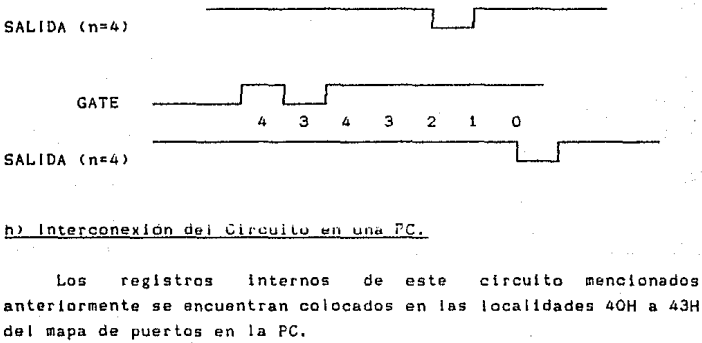
En los modos 2 y 3, si se emplea un reloj distinto al del sistema, la terminal GATE se activa inmediatamente despues de WR, al cargar un valor.



MODO 4: Disparo repetitivo vía Software. Después de seleccionado este modo, la salida es alta. Cuando la cuenta es cargada el contador empieza a contar. Al terminar, la salida va a bajo por un periodo del reloj y comienza de nuevo la cuenta.



MODO 5: Pulso repetitivo vía hardware. El contador empieza la cuenta después del flanco positivo de la entrada GATE y va a bajo por un ciclo de reloj cuando termina la cuenta. El contador es reprogramable. La salida no será alta hasta terminar completamente la cuenta después del flanco positivo de la entrada GATE.



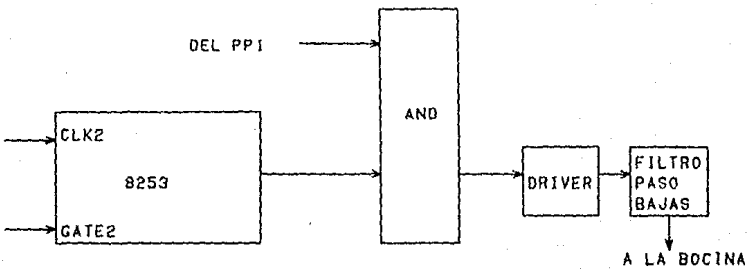
h) Interconexión del Circuito en una PC.

Los registros internos de este circuito mencionados anteriormente se encuentran colocados en las localidades 40H a 43H del mapa de puertos en la PC.

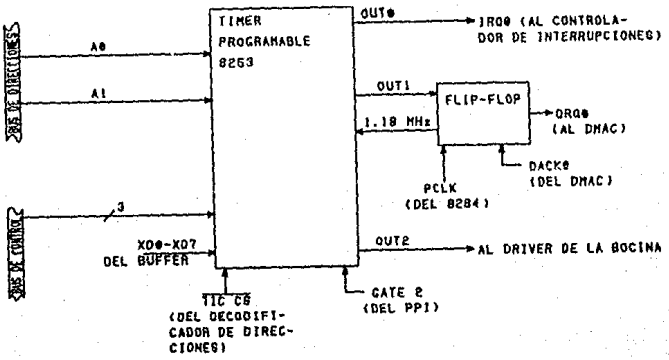
La salida OUT0 genera la petición de interrupción IORQ0, la cual actualiza el reloj de la hora del día. OUTi genera la señal

Data ReQuest DRQ0, que se emplea para generar el refresco de memoria para los circuitos colocados en los slots de expansión. OUT2 es empleado para generar la frecuencia que hará sonar la bocina de la PC.

El subsistema para la bocina se muestra a continuación.

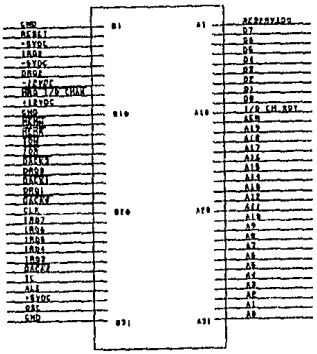


La conexión del 8253 con otros circuitos en una PC se muestra a continuación.

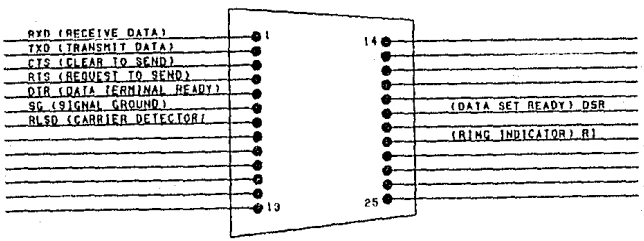


A.7. SLOTS E INTERFACES.

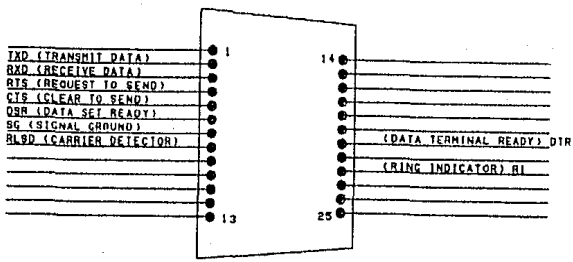
Para poder realizar el diseño de la tarjeta, se investigó las señales de las que se puede hacer uso en un slot de expansión de la PC. Estas señales son:



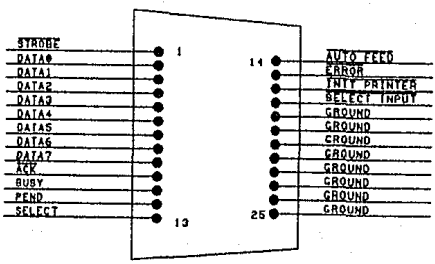
Los puertos paralelos y serie se diagnosticarán a través de sus interfaces. El conector empleado en las interfaces seriales, a conectar en un DCE (equipo de comunicación de datos) contiene las siguientes señales:



Mientras que el conector serial para conectar a un DTE (equipo terminal de datos) contiene las señales:



El conector paralelo (llamado comunmente Centronics) hace uso de las siguientes señales:



APENDICE B
MANUAL DEL USUARIO

B.1. INTRODUCCION.
B.2. INSTALACION DEL SISTEMA.
B.3. GUIA DEL USUARIO.

APENDICE B. MANUAL DEL USARIO

B.1. INTRODUCCION

El sistema de diagnóstico consta de dos partes:

1.a) Una tarjeta diagnosticadora.

1.b) Un programa.

La tarjeta diagnosticadora permite verificar los siguientes componentes:

- Puerto serie
- Puerto paralelo
- Reloj
- Controlador/manejador del disco flexible.

1.b) El programa de diagnóstico (DIAGVMS.EXE) sólo puede ser ejecutado en microcomputadoras IBM PC /XT /AT /PS2 y compatibles bajo el sistema operativo DOS, controla la tarjeta de diagnóstico y contiene una rutina de mantenimiento así como utilerías para el disco duro, además de verificar los siguientes componentes:

- CPU
- DMAC
- Memorias (RAM y ROM)
- Teclado

B.2. INSTALACION DEL SISTEMA

Los requerimientos del sistema son:

- La capacidad de memoria debe ser al menos de 512 kB.
- La tarjeta de video debe ser del tipo CGA, o en su defecto contar con un simulador de CGA.
- Tener al menos un slot de expansión disponible.
- Cuando menos un drive de disco flexible de 5 1/4".

a) Instalación de la tarjeta diagnosticadora

- Verifique que la computadora y sus elementos periféricos (monitor, impresoras, etc.) se encuentren apagados.
- Desconecte los cables de alimentación de la computadora.
- Remueva la cubierta o gabinete de la unidad central.
- Inserte en alguno de los slots disponibles la tarjeta de diagnóstico (de preferencia utilice los slots centrales).

b) Archivos del sistema

- DIAGVMS.EXE	Programa principal de diagnóstico
- CGA.BGI	Manejador de gráficos
- DISPLAYF.TXT	Archivo que contiene las pantallas gráficas de presentación
- PILOTO	Archivo empleado en el diagnóstico del DMAC
- AYUDA.DOC	Archivo que contiene la ayuda para el usuario en modo gráfico
- TEXTOS.TXT	Archivo que contiene la ayuda para el usuario en modo texto.

B.3. GUIA DEL USUARIO

Para ejecutar el sistema, teclee DIAGVMS seguido de <ENTER>.

Las dos primeras pantallas muestran la presentación del sistema. A continuación se desplegará un microcomputadora y una primera pantalla de ayuda. Para abandonarla presione <ESC> o bien presione <PgDn>/<PgUp> para desplegar la ventana siguiente o anterior de la ayuda. El programa contiene una ayuda de contexto sensitiva que puede ser accesada por medio de la tecla F1.

Por medio de las teclas del movimiento del cursor es posible posicionarse sobre alguna de las siguientes opciones:

a) Salida del sistema al DOS.

b) Accesar los diagnósticos de los elementos contenidos en la unidad central.

c) Ejecutar el diagnóstico del teclado.

a) Salida del sistema al DOS

Esta opción permite regresar al ambiente del sistema operativo, terminando la ejecución del sistema de diagnóstico.

b) Diagnósticos de los elementos de la unidad central

Al seleccionar esta opción se despliega la gráfica de la tarjeta madre, en la cual el usuario puede elegir alguno de los módulos de diagnóstico posicionándose (con las teclas del movimiento del cursor) sobre el dibujo correspondiente al diagnóstico elegido y presionando <ENTER>.

Los módulos que se pueden accesar son los siguientes:

- CPU
- DMAC
- Memorias (RAM y ROM)
- Puerto serie
- Puerto paralelo
- Reloj

- Controlador/Manejador del disco flexible
- Disco duro

Si el usuario desea abandonar esta pantalla deberá posicionarse en la opción "SALIR".

c) Diagnóstico del teclado

En la figura que muestra la microcomputadora posiciónese por medio de las teclas del movimiento del cursor sobre el teclado y presione <ENTER>.

Aparecen en la pantalla los caracteres correspondientes a las teclas de este dispositivo. Presionando alguna de las teclas se muestra en pantalla un letrero indicando qué tecla fué presionada.

Para abandonar este módulo bastará con presionar la tecla <SCROLL LOCK> cuando menos dos veces seguidas.

B.3.1 DIAGNOSTICO DE LOS COMPONENTES DE LA TARJETA MADRE

a) CPU

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente después se desplegará un menú con las siguientes opciones:

i) Información
ii) Diagnóstico del CPU
iii) Salida del módulo

i) Esta opción puede ser seleccionada presionando la tecla <F2>. Al hacerlo, se presentará en pantalla la información sobre la configuración del hardware de la microcomputadora. Para regresar al menú anterior presione cualquier tecla.

ii) Mediante esta opción se realiza el diagnóstico del CPU, que consiste en la verificación de los registros internos del mismo. Si el diagnostico encontró una falla se desplegará en cuál de los registros se encontró la anomalía.

iii) Presionando la tecla <ESC> se regresa a la pantalla de la tarjeta madre.

b) DMAC

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente después se desplegará un menú con las siguientes opciones:

i) Verificación de los registros
ii) Transferencia de disco flexible a memoria
iii) Transferencia de memoria a disco flexible
iv) Salida del módulo

i) Al accesar esta opción (presionando la tecla <F2>) se despliega una pantalla que muestra la verificación del estado de los registros del DMAC.

ii) Presionando la tecla <F3> se realiza una transferencia de información desde un archivo (PILOTO) en disco flexible hacia la memoria RAM, simulándose en pantalla.

iii) Presionando la tecla <F4> se realiza una transferencia de información desde memoria hacia un archivo (PILOTO) en disco flexible, simulándose en pantalla.

iv) Presionando la tecla <ESC> se regresa a la pantalla de la tarjeta madre.

c) Memoria RAM

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente después se realiza la verificación de la memoria RAM (para programas y de video) desplegándose en la pantalla una simulación que muestra los segmentos de memoria e indica aquél que se está verificando en ese momento.

En caso de que el programa detecte errores se mostrará el segmento y el offset correspondiente a la dirección de la localidad de memoria errónea (hasta un máximo de diez).

Este módulo permite la impresión de los errores que se hayan detectado, para ello se le presenta al usuario un aviso correspondiente.

d) Memoria ROM

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente después se realiza la verificación de la memoria ROM desplegándose en la pantalla una simulación que muestra los segmentos de memoria e indica aquél que se está verificando en ese momento.

En caso de que el programa detecte errores se mostrará el segmento y el offset correspondiente a la dirección de la localidad de memoria errónea (hasta un máximo de diez).

Este módulo permite también la impresión de los errores que se hayan detectado, para ello se le presenta al usuario un aviso correspondiente.

e) Puerto serie.

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente después se desplegará un menú con las siguientes opciones:

i) Ayuda
ii) Transmisión a través del RS-232
iii) Recepción a través del RS-232
iv) Salida del módulo

i) Esta opción se selecciona al pulsar la tecla <F1> y proporciona una explicación de los requerimientos para este diagnóstico.

ii) Esta opción se selecciona al pulsar la tecla <F2>, realizando la transmisión de un dato a través del RS-232 hacia la tarjeta de diagnóstico. Se presenta una pantalla en la que se simula esta acción.

iii) Esta opción se selecciona al pulsar la tecla <F3>, realizando la recepción de un dato a través del RS-232 proveniente de la tarjeta de diagnóstico. Se presenta una pantalla en la que se simula esta operación.

iv) Presionando la tecla <ESC> se regresa a la pantalla de la tarjeta madre.

f) Puerto paralelo

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente despues se desplegará un menú con las siguientes opciones:

i) Ayuda

ii) Diagnóstico del puerto paralelo

iii) Salida del módulo

i) Esta opción se selecciona al pulsar la tecla <F1> y proporciona una explicación de los requerimientos para este diagnóstico.

ii) Por medio de la tecla <F4> se puede accesar esta opción. En caso de que el sistema sense que el conector DB25 no se encuentra instalado se mostrará un aviso indicándolo, permitiendo después que se intente de nuevo el diagnóstico.

El diagnóstico se realiza en cuatro ocasiones, cada una de las cuales se muestra en pantalla, así como los datos obtenidos por la tarjeta de diagnóstico.

iii) Presionando la tecla <ESC> se regresa a la pantalla de la tarjeta madre.

g) Reloj

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente despues se desplegará una pantalla que contiene tres opciones, que pueden ser elegidas con las teclas de movimiento del cursor. Las opciones proporcionadas son:

i) Cambiar escala de tiempo

ii) Muestrear
iii) Salida del módulo

i) Con esta opción se permite cambiar la escala de tiempo con la que se mostrará la señal del reloj. Es necesario teclear tres números para esta escala.

ii) Esta opción realiza el muestreo de la señal del reloj de la PC. Si el usuario desea que se continue con otro muestreo, basta con oprimir la tecla de <ENTER>.

iii) Presionando la tecla <ESC> se regresa a la pantalla de la tarjeta madre.

h) Controlador/Manejador de disco flexible

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente después se desplegará un menú con las siguientes opciones:

i) Ayuda
ii) Diagnóstico de las líneas
iii) Diagnóstico de velocidad
iv) Diagnóstico de Escritura/Lectura
v) Salida del módulo

i) Esta opción se selecciona al pulsar la tecla <F1> y proporciona una explicación de los requerimientos de este diagnóstico.

ii) Presionando la tecla <F2> se inicia el diagnóstico de las líneas de control y sensado del manejador del disco flexible. Para esto tenga en cuenta que el manejador a ser diagnosticado ya debe estar conectado tal como lo indica la ventana de ayuda. El diagnóstico realizará el movimiento de las cabezas de lectura/escritura y el encendido del motor de polea del manejador (para este último punto coloque un disco flexible dentro del manejador y asegúrese de que la puerta de éste se encuentre cerrada). Al realizar el diagnóstico se presentará una simulación del mismo.

iii) Esta opción se selecciona con la tecla <F3> se determina la velocidad del motor de polea del manejador. Para esto asegúrese

de que se encuentre un disco flexible dentro del manejador y la puerta del mismo cerrada. La velocidad se desplegará en unidades de RPM (revoluciones por minuto).

iv) Este diagnóstico se selecciona presionando la tecla <F4> y verifica la escritura/lectura a través del controlador y circuitería anexa. Para esto el manejador tendrá que estar conectado al controlador (ya no a la tarjeta de diagnóstico) y en su interior deberá haber un disco flexible con la puerta cerrada (la información del disco flexible será destruida con este diagnóstico).

v) Presionando la tecla <ESC> se regresa a la pantalla de la tarjeta madre.

i) Disco duro

Para seleccionar esta opción posiciónese sobre ella mediante las teclas del movimiento del cursor y presione <ENTER>.

Inmediatamente después se desplegará un menú con las siguientes opciones:

i) Ayuda
ii) Compresion/Descompresión de archivos
iii) Encriptamiento/Desencriptamiento de archivos
iv) Verificación de la FAT
v) Salida del modulo

i) Esta opción se selecciona al pulsar la tecla <F1> y proporciona una explicación de los requerimientos para el diagnóstico.

ii) Al seleccionar esta opción (pulsando la tecla <F2>) se muestra en la pantalla un menú, que se recorre con las teclas de movimiento del cursor, posicionándose en la opción deseada y seleccionando por medio de la tecla <ENTER>. Se tienen las siguientes opciones:

a) Comprimir un archivo
b) Descomprimir un archivo
c) Cambiar el drive activo

a) Al seleccionar la opción de comprimir se presenta un nuevo menú, el cual contiene las opciones:

a.1) Comprimir los archivos de todo el disco. Con esta opción se podrán comprimir la totalidad de los archivos del disco que sean suceptibles de serlo, pudiendo detener la compresión de los siguientes archivos pulsando la tecla <ESC>. El pulsar esta tecla no detendrá la compresión del archivo que esté en curso.

a.2) Comprimir un archivo con ruta conocida. Esta opcion permite la compresión de un archivo, la ruta y el nombre de el son proporcionados por el usuario.

a.3) Comprimir un archivo con ruta desconocida. Esta opción proporciona al usuario una herramienta que permite seleccionar un archivo al irse ubicando en los diferentes subdirectorios. Se mostrarán los nombres de los archivos y subdirectorios (elija entre ellos con las teclas de movimiento izquierdo y derecho), en los que se podrá ubicar el cursor. En caso de existir más nombres que los mostrados se puede emplear la tecla <PgUp> y <PgDn> para ubicarse en ellos.

b) Al seleccionar la opción de descomprimir se presenta un nuevo menú, el cual contiene las opciones:

b.1) Descomprimir los archivos de todo el disco. Con esta opción se podrán descomprimir la totalidad de los archivos que se encuentren comprimidos en el disco, pudiendo detener la descompresión de los siguientes archivos pulsando la tecla <ESC>. El pulsar esta tecla no detendrá la descompresión del archivo que esté en curso.

b.2) Descomprimir un archivo con ruta conocida. Esta opción permite la descompresión de un archivo, la ruta y el nombre de el son proporcionados por el usuario.

b.3) Descomprimir un archivo con ruta desconocida. Esta opción proporciona al usuario una herramienta que permite seleccionar un archivo al irse ubicando en los diferentes subdirectorios. Se mostrarán los nombres de los archivos y subdirectorios (elija entre ellos con las teclas de movimiento izquierdo y derecho), en los que se podrá ubicar el cursor. En caso de existir más nombres que los mostrados se puede emplear la tecla <PgUp> y <PgDn> para ubicarse en ellos.

c) Al seleccionar la opción de cambiar el drive activo se presenta una ventana en la que se muestra el drive activo y se pregunta por el que se desea que pase a ser el nuevo drive activo.

Presione la tecla correspondiente a él. De ser inválido no se abandonará la opción, sino hasta capturar una opción válida.

iii) Al seleccionar esta opción (pulsando la tecla <F3>) se muestra en la pantalla un menú, que se usa con las teclas de movimiento del cursor, posicionando éste en la opción deseada y seleccionando por medio de la tecla <ENTER>. Se presentan las siguientes opciones:

a) Encriptar un archivo
b) Desencriptar un archivo
c) Cambiar el drive activo

a) Al seleccionar la opción de encriptar se presenta un nuevo menú, el cual contiene las opciones:

a.1) Encriptar los archivos de todo el disco. Con esta opción se podrán encriptar la totalidad de los archivos del disco, pudiendo detener el encriptamiento de los siguientes archivos pulsando la tecla <ESC>. El pulsar esta tecla no detendrá el encriptamiento del archivo que este en curso.

a.2) Encriptar un archivo con ruta conocida. Esta opción permite el encriptamiento de un archivo, la ruta y el nombre de él son proporcionados por el usuario.

a.3) Encriptar un archivo con ruta desconocida. Esta opción proporciona al usuario una herramienta que permite seleccionar un archivo al irse ubicando en los diferentes subdirectorios. Se mostrarán los nombres de los archivos y subdirectorios (elija entre ellos por medio de las teclas de movimiento izquierdo y derecho), en los que se podrá ubicar el cursor. En caso de existir más nombres que los mostrados se puede emplear la tecla <PgUp> y <PgDn> para ubicarse en ellos.

b) Al seleccionar la opción de desencriptar se presenta un nuevo menú, el cual contiene las opciones:

b.1) Desencriptar los archivos de todo el disco. Con esta opción se podrán desencriptar la totalidad de los archivos del disco suceptibles a serlo, pudiendo detener el desencriptamiento de los siguientes archivos pulsando la tecla <ESC>. El pulsar esta tecla no detendrá el desencriptamiento del archivo que esté en curso.

b.2) Desencriptar un archivo con ruta conocida. Esta opción permite el desencriptamiento de un archivo, la ruta y el nombre de él son proporcionados por el usuario.

b.3) Desencriptar un archivo con ruta desconocida. Esta opción proporciona al usuario una herramienta que permite seleccionar un archivo al irse ubicando en los diferentes subdirectorios. Se mostrarán los nombres de los archivos y subdirectorios (elija entre ellos con las teclas de movimiento izquierdo y derecho), en los que se podrá ubicar el cursor. En caso de existir más nombres que los mostrados se puede emplear la tecla <PgUp> y <PgDn> para ubicarse en ellos.

c) Al seleccionar la opción de cambiar el drive activo se presenta una ventana en la que se muestra el drive activo y se pregunta por el que se desea que pase a serlo. Presione la tecla correspondiente a él. De ser inválido no se abandonará la opción, sino hasta capturar una opción válida.

iv) Esta rutina de mantenimiento está conformada por dos opciones, las cuales se describen a continuación:

a) Información de la configuración del disco.
b) Verificación de los clusters.

a) Obtención de la configuración del disco duro, en donde se despliegan las características principales, tales como:

- Número de bytes por sector
- Número de FATs por disco
- Número total de clusters en el disco
- Número de sectores por cluster
- Número de sectores en el disco.

b) Mediante esta rutina se diagnosticará el estado de todos los clusters que conforman el disco duro y en caso de encontrar clusters marcados como dañados se le indicará al usuario lo ocurrido y se le dará la opción de tratar de corregirlos, en caso de escoger esta última alternativa se verificará que efectivamente el cluster está dañado y que en caso contrario se le permitirá al usuario la corrección de este cluster marcado como disponible.

v) Presionando la tecla <ESC> se regresa a la pantalla de la tarjeta madre.

BIBLIOGRAFIA.

1. Lemmons, Phil
   THE IBM PERSONAL COMPUTER
   Byte, 1981 Vol. 6 No. 10

2. Gens, Frank
   COULD 1'000,000 IBM PC USERS BE WRONG?
   Byte, 1983 Vol. 8 No. 11

3. Killen, Michael
   IBM FORECAST: MARKET DOMINANCE
   Byte, 1984 Vol. 9 No. 9

4. Freiberger, Paul
   MICROINFORMATICA: ORIGENES - PERSONAJES, EVOLUCION Y DESARROLLO
   Osborne/McGraw-Hill, 1986

5. Navarro, Fibla Miguel
   SISTEMA DE ACCESO DIRECTO A MEMORIA: EL 8237-5
   PC Word, 1986 No. 17

6. Morris, Mano M.
   LOGICA DIGITAL Y DISEÑO DE COMPUTADORES
   Prentice Hall, 1982

7. INTEL HANDBOOK
   Inter Corporation, 1986

8. Morgan, L. Christopher y Waite Mitchell
   INTRODUCCION AL MICROPROCESADOR 8086/8088 (16 BITS)
   McGraw-Hill, 1984

9. Nichols, Joseph C., Nichols Elizabeth A. y Rony Peter R.
   MICROPROCESADOR Z-80
   Publicaciones Marcombo, S. A., 1979

10. EL MUNDO DE LA COMPUTACION 1.
Ediciones OCEANO-EXITO, 1985

11. Norton, Peter y Wilton Richard
THE NEW PETER NORTON PROGRAMER'S GUIDE TO THE PC & PS/2
Prentice Hall, 1985

12. Martell, Alberto Torfer
MANANA ES 2000. ESTRATEGIA PARA EL FUTURO
Gráfica General, 1987

13. LS/S/TTL LOGIC DATABOOK
National Semiconductor Corporation, 1989

14. Douglas V. Hall
MICROPROCESSORS AND INTERFACING
McGraw-Hill, 1986

15. Duncan, Ray
ADVANCE MS-DOS
Microsoft Press