ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

"INTERFACE ENTRE MICROPROCESADOR Y DISCOS FLOPPY"

TESIS PREVIA A LA OBTENCION DEL TITULO
DE INGENIERO EN LA ESPECIALIZACION DE
ELECTRONICA Y TELECOMUNICACIONES

DIEGO MARCELO VALDEZ VITERI

JULIO 1982

Certifico que el presente trabajo ha sido elaborado en su totalidad por el Señor Marcelo Valdez Viteri.

ING. HERBERT JACOBSON
Director de Tesis

D E D I C A T O R I A

A MIS PADRES

A G R A D E C I M I E N T O

Al Ing. Herbert Jacobson por su acertada dirección y colaboración en el desarrollo y construcción de esta tesis.

I N D I C E

" INTERFACE ENTRE MICROPROCESADOR Y DISCOS FLOPPY "

INTRODUCCION

Dentro de los últimos años, el uso de discos "floppy" ha llegado a constituírse en uno de los medios más populares de almacenamiento -en línea- para sistemas pequeños de computación. Su tiempo de acceso rápido, gran confiabilidad y bajo costo por bit de información hacen que el disco "floppy" sea la solución para el almacenamiento no volátil de grandes cantidades de información en sistemas cuyo funcionamiento se basa en el uso de microprocesadores.

El hecho de que los discos pueden ser removidos del aparato que los lee o escribe y que la información almacenada en ellos no es volátil, permite la archivación de los discos para su uso futuro sin necesitar de un consumo de energía para retener la información en ellos escrita. Además, la facilidad de utilización y la versatilidad que presentan al poder ser escritos y leídos por máquinas diferentes, lo que permite el intercambio de datos o programas en forma sencilla y a un bajo costo, ha llevado a la rápida popularización de este sistema.

Debido a que este tipo de discos fueron creados por la Compañía IBM, las características de los mismos se convirtieron en un standard, por lo que la forma en que se conectan las lectoras/escritoras de este tipo de discos al sistema principal no varía mucho de un producto a otro, de forma que el usua

rio puede sustituír una lectora/escritora de un fabricante por la de otro, sin que esto represente grandes modificaciones en su sistema (1).

En la presente tesis se van a desarrollar los sistemas electrónicos necesarios para controlar el funcionamiento de lectoras/escritoras de este tipo de discos y para realizar los procesos de interface de estos sistemas con un equipo cuyo funcionamiento se basa en el uso de un microprocesador. Se desarrollarán además, los programas que sirvan de soporte para el correcto funcionamiento del sistema completo.

El sistema terminado tendrá todas las ventajas anteriormente indicadas y será sin duda, una gran ayuda para la creación y almacenamiento de programas que sirvan de base a trabajos futuros que se realicen en torno a este equipo, permitiendo además una mayor claridad en la enseñanza de la materia de microprocesadores. De esta manera se podrá hacer efectivo el uso de equipos que hasta ahora, a pesar de tenerlos disponibles practicamente no han sido utilizados debido a su dificultad de operación.

En el futuro se podrán adquirir discos previamente grabados, compatibles con el sistema que aquí se diseña, que contengan programas como editores, ensambladores, sistemas operativos y otras clases de programas que expandan la capacidad del equipo, faciliten su manejo y ayuden a la complementación de la enseñanza.

C A P I T U L O P R I M E R O

"INTRODUCCION A LOS DISCOS FLOPPY"

1. INTRODUCCION A LOS DISCOS FLOPPY.

1.1. LOS DISCOS FLOPPY.

El disco floppy o diskette es el medio que retiene la información, está recubierto por una funda plástica, delgada y flexible, que protege al disco de las partículas de suciedad.

La funda contiene internamente un material especialmente tratado para minimizar la fricción y las descargas de electricidad estática. Consta de tres aperturas principales, la primera permite que la cabeza de grabación pueda ponerse en contacto con la superficie del disco, la segunda asegura el acceso de eje que hace girar al disco dentro de su funda y la tercera permite la detección del agujero índice. En la mayoría de los casos se tiene otro agujero en el extremo de la funda, que posibilita la escritura de información en el disco dependiendo si está recubierto o no, pudiendo tener de esta forma discos "protegidos contra escritura".

El disco está formado por una película de Mylar de forma circular con su superficie recubierta por óxido magnético, la información es por tanto grabada magnéticamente en él. Su superficie es dividida en pistas concéntricas separadas una de otra, por aproximadamente 0,02 pulgadas (48 pistas por pulgada). Cada pista ocupa unas 0,012 pulgadas (2) y normalmente es subdividida en sectores de igual longitud.

El disco tiene un agujero central al cual se fija el eje que lo hace girar, cerca a éste se tiene otro llamado agujero índice que, al alinearse con la apertura de la funda, provee de un ángulo de referencia en el que se define el comienzo de una pista. Para la identificación del comienzo de cada sector se han adoptado dos maneras diferentes, ya sea por medio de la escritura de información adicional en cada pista o por medio de agujeros -uno por cada sector- localizados en la circunferencia que contiene al agujero índice. Estas dos formas de identificar el comienzo de cada sector han separado a los discos floppy en dos clases principales: "soft sectored" y "hard sectored" respectivamente.

El tamaño externo de los discos se ha estandarizado y por el momento existen dos tipos cuyas dimensiones son: 8 × 8 pulgadas y 5¼ × 5¼ pulgadas, teniendo cada uno de ellos las características anteriormente indicadas.

1.2. LAS LECTORAS/ESCRITORAS DE DISCOS FLOPPY

Las lectoras/escritoras de discos floppy contienen todas las partes mecánicas y eléctricas que permiten guardar o recibir información del disco. Aquí nos limitaremos a dar una breve descripción de las partes que no tengan que ver con las señales de interface a las cuales se les dará una mayor importancia debido a que su comprensión es indispensable para el desarrollo de la presente tesis. Mayor información se puede en

contrar en los apéndices y bibliografía al final de este trabajo.

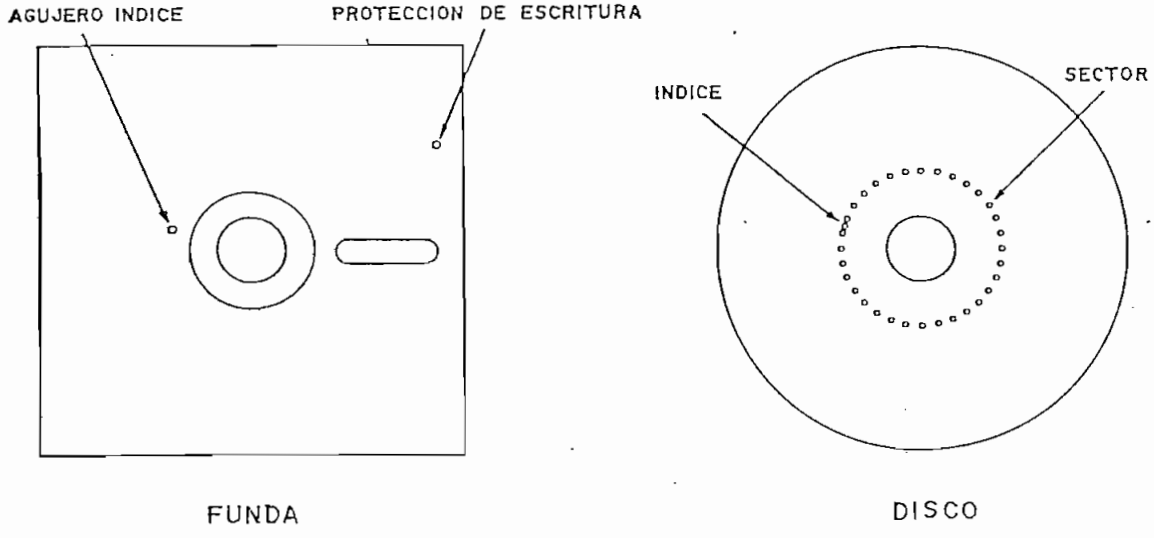


FIGURA 1

Las principales partes mecánicas tienen las siguientes funciones:

- Sujetar al disco en posición
- Girar al disco con una velocidad constante
- Mover la cabeza de lectura/escritura en pasos de igual longitud, ya sea hacia pistas internas (cercanas al centro del disco) o externas.
- Empujar a la parte expuesta del disco sobre la cabeza de lectura/escritura (este proceso debe entenderse como "bajar la cabeza")

El hecho de que se necesite bajar la cabeza para realizar un proceso de lectura o escritura prolonga la duración del

disco y de la cabeza de grabación pues no estarán en contacto a menos que esto sea requerido por el proceso que se esté desarrollando, esta característica extiende la vida útil de la cabeza a unos cinco años (Diez mil horas de contacto con el disco) y la del disco a unos dos años de mucho uso. (3),(4).

La cabeza de lectura/escritura no es más que una pequeña bobina con núcleos de ferrita usada para leer, grabar o borrar información del disco. La superficie de contacto con el disco está recubierta por una película de vidrio y ha sido diseñada para máxima transferencia de señal con un mínimo desgaste de la cabeza y el disco, contiene además elementos para borrar el espacio entre pistas de forma que el nivel de señal a ruido no se degrade al utilizar discos grabados en otra lectora/escritora. (6).

La rotación del disco se consigue por medio de un motor que lo hace girar a velocidad angular constante, este motor es usualmente de corriente alterna en discos de 8" mientras que en discos de 5¼" se usan preferentemente los de corriente continua, siendo estos últimos más estables a variaciones de voltaje de la línea ya que su velocidad es controlada electrónicamente por medio de un servomecanismo.(5).

Un segundo motor mueve la cabeza de lectura/escritura a las diferentes pistas en el disco, el movimiento se realiza "por pasos" de igual longitud, este tipo de motores son llama

dos "Stepper Motors".

La parte electrónica de las lectoras/escritoras se encarga de controlar el funcionamiento del sistema, así como de los procesos de interface con el sistema principal. Las funciones principales que se realizan en esta parte se describen a continuación:

- Detección del agujero índice, proceso que se realiza en base a un emisor de luz (usualmente un LED infrarrojo) y un detector luminoso, localizados en lados opuestos al disco de forma que cada vez que el agujero índice se alinea con el de la funda, exista una transmisión de luz desde el emisor al detector, éste último convierte esta señal luminosa en un pulso eléctrico llamado pulso índice.

- Generación de las diferentes fases eléctricas para hacer girar un cierto ángulo (siempre constante) al motor que mueve la cabeza en uno u otro sentido de acuerdo a una señal externa que indica la dirección deseada y otra que indica el momento en que debe producirse el giro.

- Comprobación de que el disco esté listo para cualquier proceso, es decir, que la puerta de acceso del disco esté cerrada, los niveles de voltaje y la velocidad angular del disco sean los correctos.

- Detección de que la cabeza se encuentra en la pista Ø ó

de que se trata de un disco protegido contra escritura, condiciones que son detectadas por medio de switches o sensores ópticos. Estas informaciones son enviadas al sistema principal en forma de señales eléctricas para su interpretación y uso.

- Activación del pulsador de bajado de la cabeza (usualmente por medio de un relé) de acuerdo a una señal externa.

- Especialmente en el caso de discos de 5¼", control electrónico de la velocidad del disco.

- Control de la cabeza de lectura/escritura de forma que se pueda guardar información o recobrarla del disco. Esta parte debe contener todos los amplificadores y filtros necesarios para este objeto, debe deshabilitar los circuitos de escritura cuando un disco está protegido y controlar los procesos de lectura o escritura de acuerdo a señales externas.

- En algunos casos, generación de voltajes regulados para uso de la lectora/escritora.

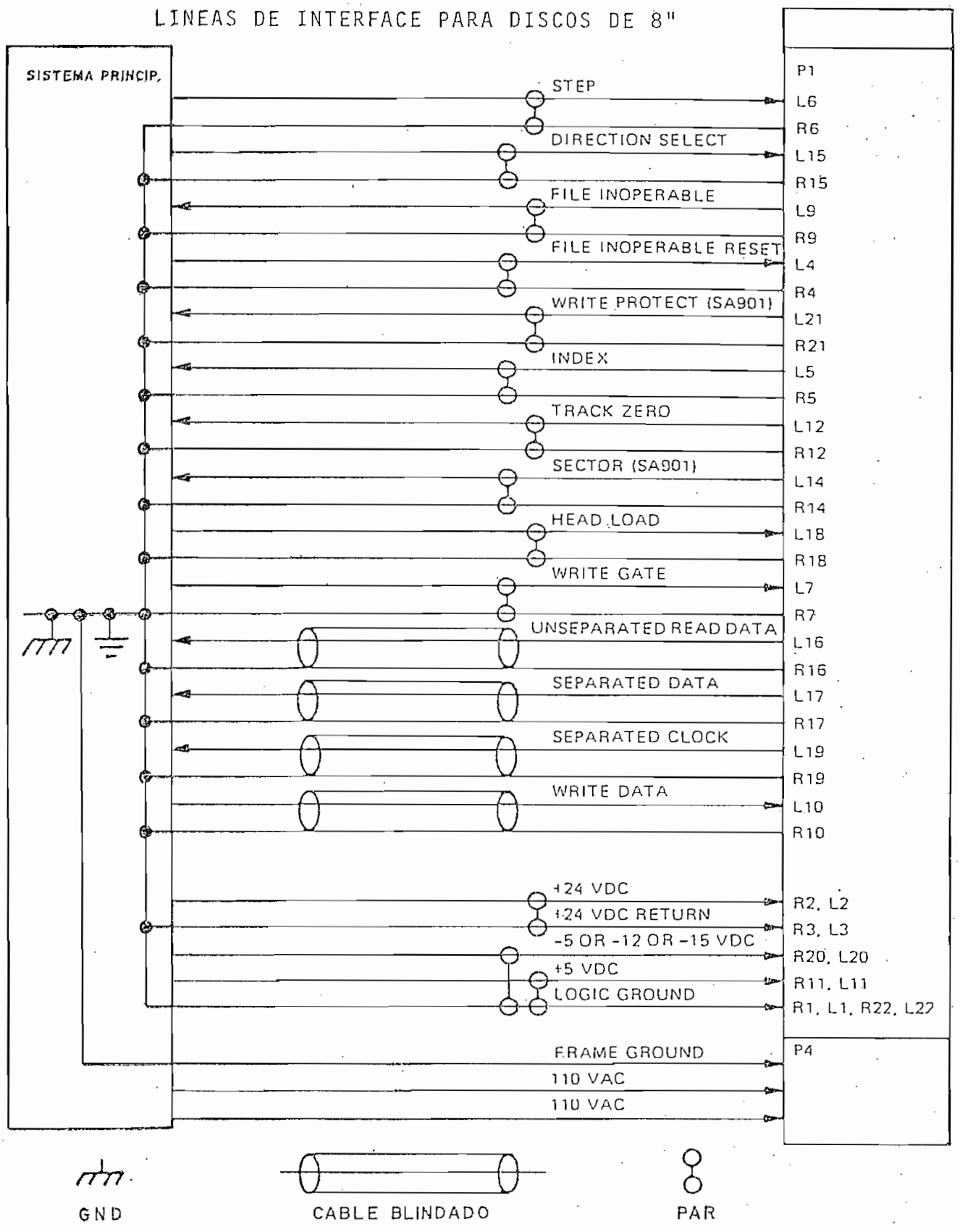
Las figuras 2 y 3 muestran sistemas típicos de interface para discos de 8" y 5¼" respectivamente, el significado de cada una de estas señales se explica a continuación.

1.2.1. SEÑALES DE INTERFACE ENVIADAS A LA LECTORA/ESCRITORA.

DIRECTION:

FIGURA 2

LINEAS DE INTERFACE PARA DISCOS DE 8"



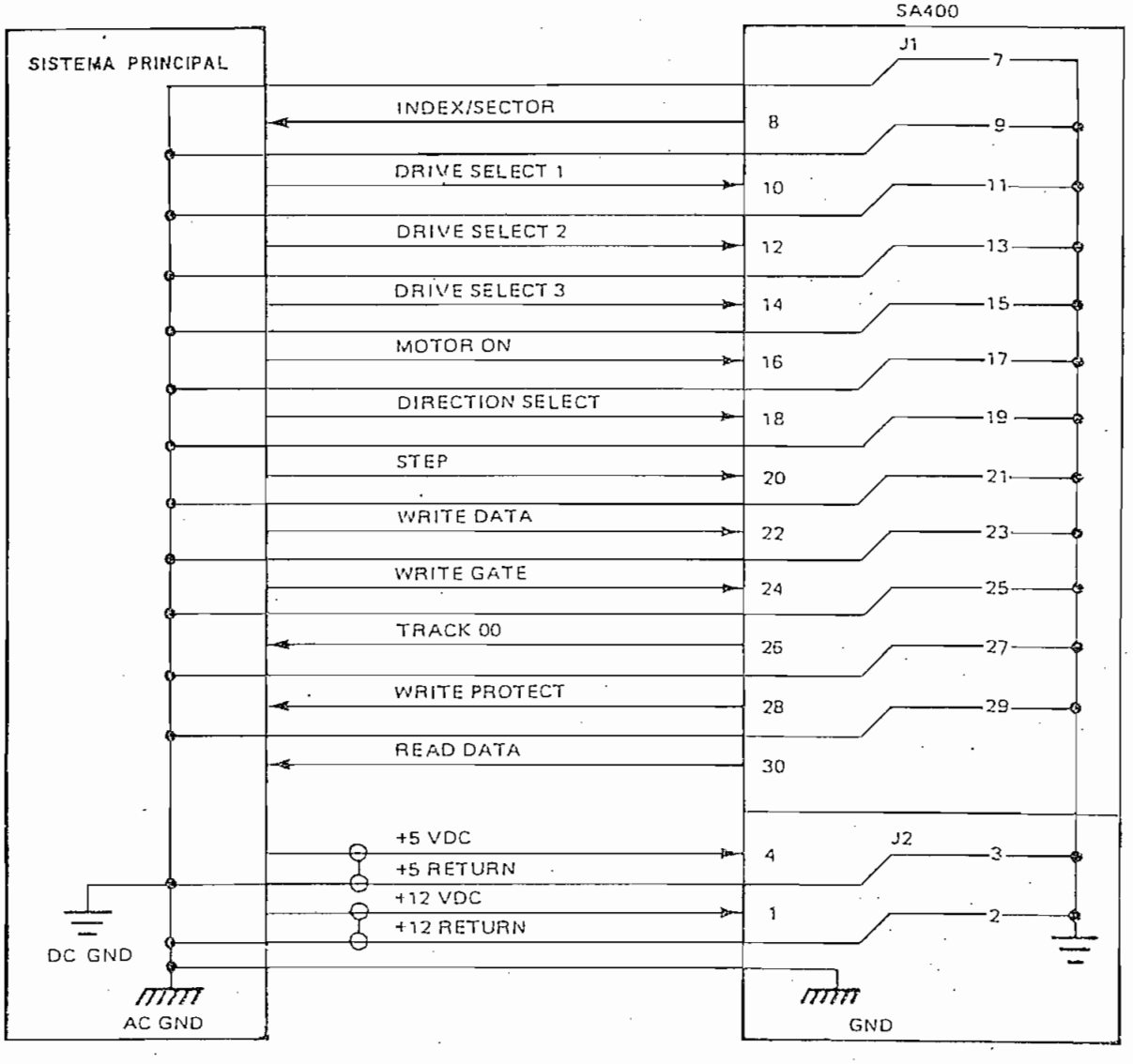


FIGURA 3

LINEAS DE INTERFACE PARA DISCOS DE 5¼"

Esta señal define la dirección de movimiento de la cabeza de lectura/escritura cuando se manda un pulso de "STEP". Un uno lógico define la dirección como "hacia afuera" y si un pulso es aplicado a la línea de STEP, la cabeza se moverá hacia pistas cada vez más distantes del centro del disco, lo contrario sucederá si el pulso es aplicado cuando la señal $\overline{\text{DIRECTION}}$ está en cero lógico. Hay que anotar que esta convención no siempre se mantiene y puede variar de un fabricante a otro.

$\overline{\text{STEP}}$

Esta señal de control produce el movimiento de la cabeza de lectura/escritura de acuerdo a la señal $\overline{\text{DIRECTION}}$. El movimiento de la cabeza empieza con el final del pulso aplicado a la línea de $\overline{\text{STEP}}$, es decir, con una transición positiva de esta señal. La duración de los pulsos aplicados, el tiempo mínimo entre pulsos y el tiempo mínimo que debe transcurrir entre la estabilización de la señal $\overline{\text{DIRECTION}}$ y la aplicación de los pulsos son condiciones que hay que tener en cuenta para el correcto funcionamiento del sistema, por esta razón están siempre especificados por el fabricante.

$\overline{\text{HEAD LOAD}}$

Esta señal controla el pulsador de bajado de la cabeza, un uno lógico en ella desactiva el pulsador mientras que un cero lógico lo activa, haciendo que la parte expuesta del disco se ponga en contacto con la cabeza, en algunos casos también se usa esta señal para desconectar parte de la alimentación de vol

taje al motor que mueve la cabeza de forma que se evite el calentamiento excesivo de este motor. En algunos sistemas de interface como el ilustrado en la figura 3, puede no existir la señal HEAD LOAD en cuyo caso el activador de bajado de la cabeza es activado cada vez que se encienda el motor que mueve al disco o cuando la lectora/escritora en cuestión sea seleccionada, el uso de una u otra condición para que se produzca el bajado de la cabeza es usualmente controlado por el usuario.

DRIVE SELECT

Usualmente son varias las líneas de DRIVE SELECT, estas son utilizadas especialmente en casos en que el sistema principal se conecta a varias lectoras/escritoras, cada una de las cuales tiene asignada una línea de DRIVE SELECT de forma que si el sistema principal manda un cero lógico por una de estas líneas será solamente la lectora/escritora asignada a esa línea que entre en funcionamiento aceptando y enviando señales de interface del o hacia el sistema principal, las demás lectoras/escritoras se desconectan del sistema es decir, no envían señales y hacen caso omiso a las que reciben. En algunos casos esta señal también indica que la lectora/escritora debe bajar la cabeza.

MOTOR ON

Esta señal es utilizada principalmente en discos de 5¼" su función es la de encender al motor que gira el disco, esto

tiene el objeto de extender la vida útil de este motor así como la de las partes asociadas a él, un cero lógico en esta señal de interface pone en movimiento este motor y aunque esto constituye una ventaja en lo que se refiere a la duración del motor, tiene también una desventaja pues se necesita esperar un cierto tiempo -especificado por el fabricante- para iniciar cualquier proceso en el disco, en discos de 8" en los que se prefiere tener un menor tiempo de acceso se mantiene normalmente encendido al motor en todo momento.

TRACK GRATER THAN 43

Esta señal informa a la lectora/escritora que su cabeza está posicionada en una pista mayor a la pista 43. Su función es controlar la corriente enviada a la cabeza de lectura/escritura en procesos de escritura, esto se debe a que en pistas internas hay una mayor agrupación de la información debido a que su circunferencia se ha reducido considerablemente en relación a la de pistas externas.

SIDE SELECT

Esta señal es utilizada en lectoras/escritoras que pueden leer o escribir discos por sus dos lados, un cero en esta línea indica que se ha seleccionado el lado Ø del disco (el único que puede ser utilizado en discos de un solo lado) mientras que un uno lógico selecciona el lado opuesto a este.

WRITE DATA

Por esta línea de interface el sistema principal manda una señal que debe ser grabada en el disco en forma de cambios de polarización magnética, la señal no es más que pulsos correspondientes a los datos serializados más señales de reloj usadas para sincronizar la recepción de los mismos. La forma de esta señal será mejor comprendida cuando se estudien los diferentes formatos de grabación.

WRITE GATE ó WRITE ENABLE

Esta línea de interface controla la escritura de información en el disco. Un uno lógico en esta línea impide la escritura de datos en el disco, mientras que un cero lógico a la vez que habilita la escritura, inhabilita a los circuitos que controlan el movimiento de la cabeza.

1.2.2. SEÑALES DE INTERFACE PROVENIENTES DE LA LECTORA/ESCRITORA.

TRACK Ø

Esta señal se pone en cero lógico cuando se ha llegado a la pista cero del disco a la vez que impide que el motor que mueve la cabeza continúe moviéndola hacia pistas más externas. Esta señal es de gran importancia pues provee de un punto de referencia para encontrar las demás pistas en el disco.

READY

Un cero lógico en esta línea de interface indica que el

disco ha sido introducido correctamente, que la puerta está cerrada y que los niveles de voltaje y la velocidad del disco son correctos, un uno lógico indicará que cualquiera de estas condiciones no se cumple.

WRITE PROTECT

Un cero lógico en esta señal indicará que un disco protegido contra escritura ha sido introducido en la lectora/escritora, de cumplirse esta condición no se podrá escribir información sobre el disco aún cuando el sistema principal la mande.

INDEX

Esta señal indica que se ha detectado el agujero índice en el disco, la indicación se la hace enviando un pulso de corta duración a través de esta línea, la misma que permanece en uno lógico hasta la siguiente detección del agujero índice. Esta señal provee al sistema principal de una referencia que indica el comienzo de una pista y como es lógico se tendrá un pulso por cada revolución del disco.

READ DATA ó UNSEPARATED READ DATA

Por esta línea de interface se envía la señal que se lee del disco, la misma que tendrá la misma forma que aquella que se envía al disco para su escritura por la línea de WRITE DATA es decir, que contendrá tanto señales de datos como señales de reloj como se explica en la parte de formato de escritura.

SEPARATED DATA

Esta señal no siempre es entregada por las lectoras/escritoras y solamente se la tiene disponible en aquellas que tienen un separador de datos y señal de reloj interno, de esta forma, la señal enviada por la línea de SEPARATED DATA no es más que la transmisión serial de los datos escritos en el disco. Debido a la gran variedad de circuitos separadores de datos y señal de reloj se ha asignado todo un capítulo al estudio de estos circuitos, allí se podrá encontrar mayor información sobre esta señal.

SEPARATED CLOCK

Por esta línea de interface se envía la señal de reloj proveniente del separador de datos y como sucede con la señal de SEPARATED DATA, no siempre es proveída por la lectora/escritora.

1.3. FORMATOS DE ESCRITURA

1.3.1 FORMAS DE CODIFICACION DE LA INFORMACION PARA SU ALMACENAMIENTO EN EL DISCO.

Antes de explicar los diversos formatos de escritura utilizados, se deben conocer las formas en que son codificados los datos para su escritura en el disco. Como se dijo anteriormente, la señal que se escribe en el disco está formada por dos señales distintas, una proveniente de la serialización de los

datos y otra que corresponde a una señal de sincronismo, el uso de esta última señal se debe a que la velocidad angular del disco puede variar en pequeños porcentajes por diversos factores, por lo que, de no existir esta señal se perdería el sincronismo en el circuito que recobra los datos, esto sucedería especialmente en el caso de leer largas cadenas de ceros las cuales no habrán producido cambios de polarización magnética en la superficie del disco cuando fueron escritos.

Debido a lo explicado anteriormente, el uso de una señal de reloj se hace imprescindible para recuperar la información escrita en el disco. Puesto que la forma más sencilla de lograr esto es la de enviar un tren de pulsos de reloj entre los cuales se intercalan los bits de información, fue esta la forma de codificación que se utilizó originalmente y es conocida como FM por ser, como se muestra en la figura 4, una forma

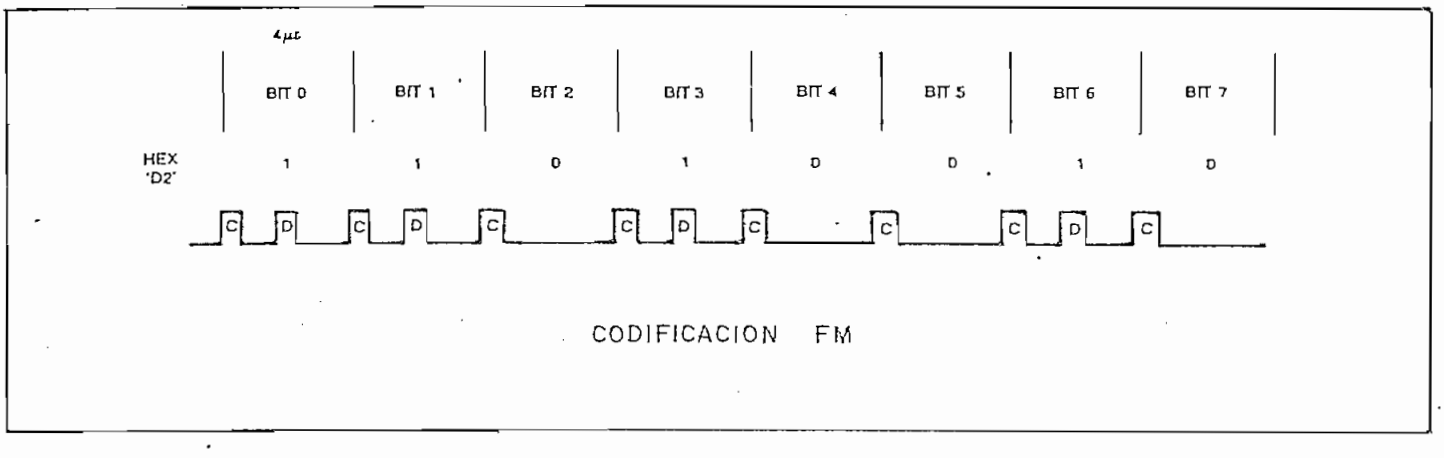


FIGURA 4

de modulación en frecuencia. De forma que no se pierda completamente el espacio del disco ocupado por estos pulsos de reloj, se manda también información por ellos, que sirve para facilitar la detección de ciertas áreas específicas de cada pista en el disco como se explicará al estudiar los formatos de grabación, este hecho convierte a los pulsos de reloj en "bits" de reloj".

El sistema FM ha sido utilizado con mucho éxito y lo sigue siendo, sin embargo, debido a que los bits de reloj ocupan tanto espacio en el disco como los de datos, se buscó un mejor sistema de codificación, al cual se lo ha denominado MFM (FM modificado), una señal típica en este sistema ha sido ilustrado en la figura 5, su característica es la de que los bits de reloj son escritos sólamente en el caso de que tanto el último bit de datos escrito, como el que será escrito a continuación, sean ceros, de esta forma se puede escribir el doble de información

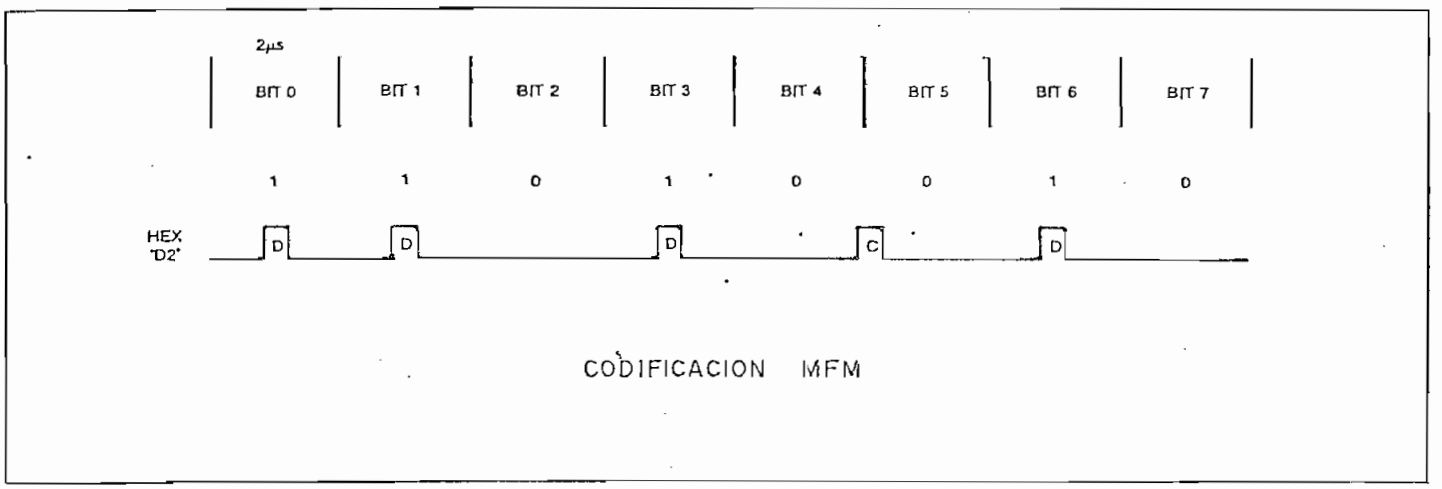


FIGURA 5

en el disco que utilizando el sistema FM sin que aumente el número máximo de cambios de polarización magnética por unidad de longitud en la superficie del disco. Este sistema es también llamado de "densidad doble" y, si bien permite almacenar mayor cantidad de información, dificulta la separación de las señales de datos y reloj de la señal leída del disco, para facilitar este proceso se utiliza la precompensación de escritura que será explicada a continuación.

El uso de precompensación en la escritura se debe a que al grabar señales como las de las figuras 4 ó 5 en el disco, se producen desplazamientos en la posición de aquellos bits que no se encuentran a igual distancia del bit que los precede y el que los sigue, el desplazamiento está dirigido hacia el punto medio entre estos dos bits (7), por esta razón la precompensación produce un desplazamiento en sentido opuesto al que producirá la grabación de forma que los bits se encuentren en su posición nominal en la lectura.

Usualmente no se utiliza precompensación en el sistema FM o de densidad simple, mientras que en densidad doble es usada especialmente para discos de 8". Puesto que la precompensación de escritura es una función de la lectora/escritora utilizada, la cantidad de precompensación será usualmente especificada por el fabricante y a su valor típico está en el rango de los 100 - 300 nanosegundos. (8).

La decisión de si un bit debe ser escrito "antes" ó "después" de su posición nominal debe ser tomada por el circuito que crea la señal $\overline{\text{WRITE DATA}}$ en base a los criterios ya explicados.

En la figura 6 se ilustra un circuito de precompensación cuyo funcionamiento se basa en un generador de cuatro fases como elemento de retraso. Cuando un pulso es mandado por la lí

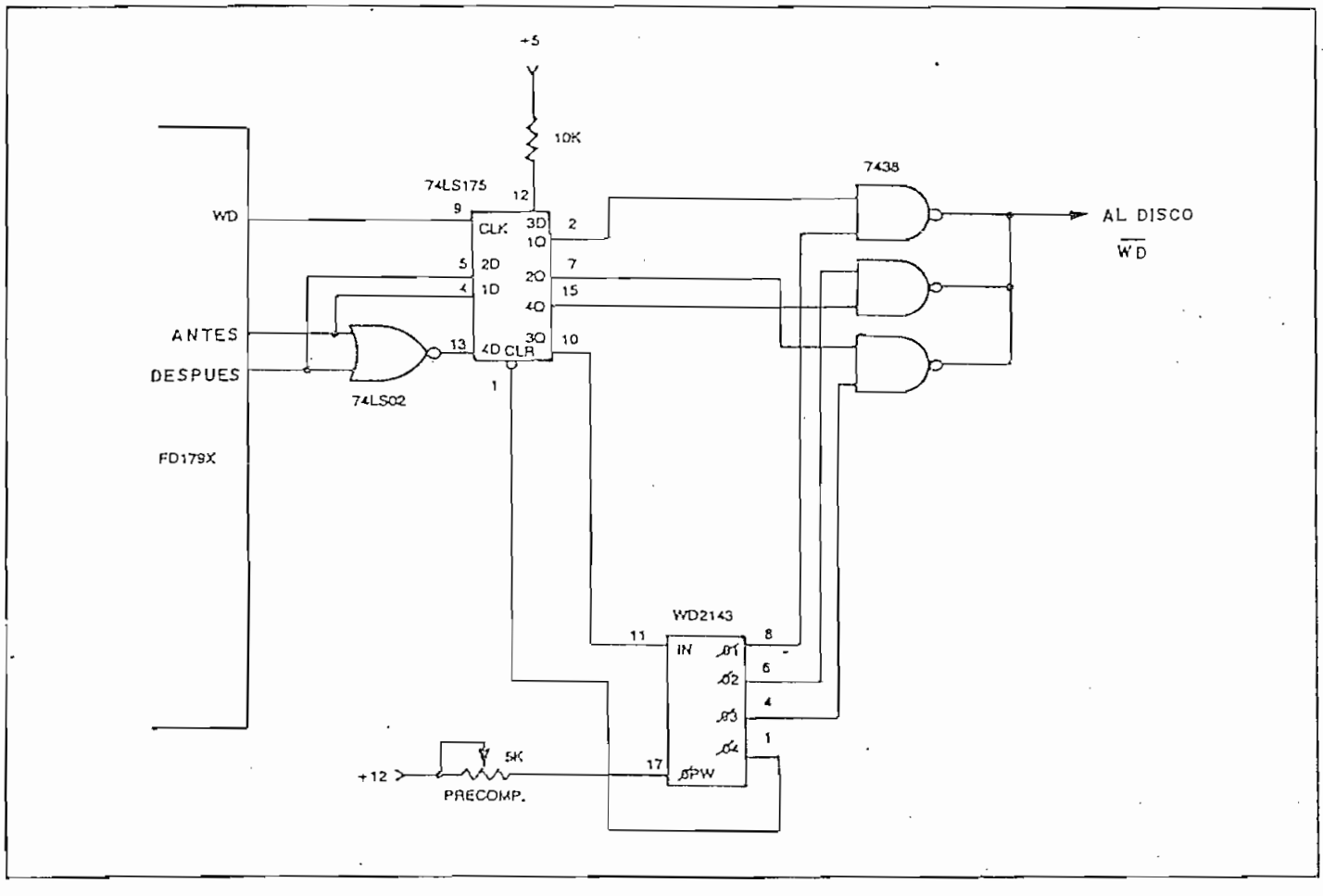


FIGURA 6

nea de WRITE DATA, las entradas de los flip-flops tipo D son copiadas en sus salidas, y ya que la entrada 3D está siempre en uno lógico, se producirá una transición positiva en 3Q que dará inicio a la generación de las fases como se indica en la figura 7, de esta forma si la señal "antes" estuvo en uno lógico será la fase número uno la que se escriba en el disco, si lo estuvo la señal "después" la fase tres será escrita, y si ninguna de estas señales estuvo en uno lógico entonces el bit es escrito en su posición nominal correspondiente a la fase dos. la fase

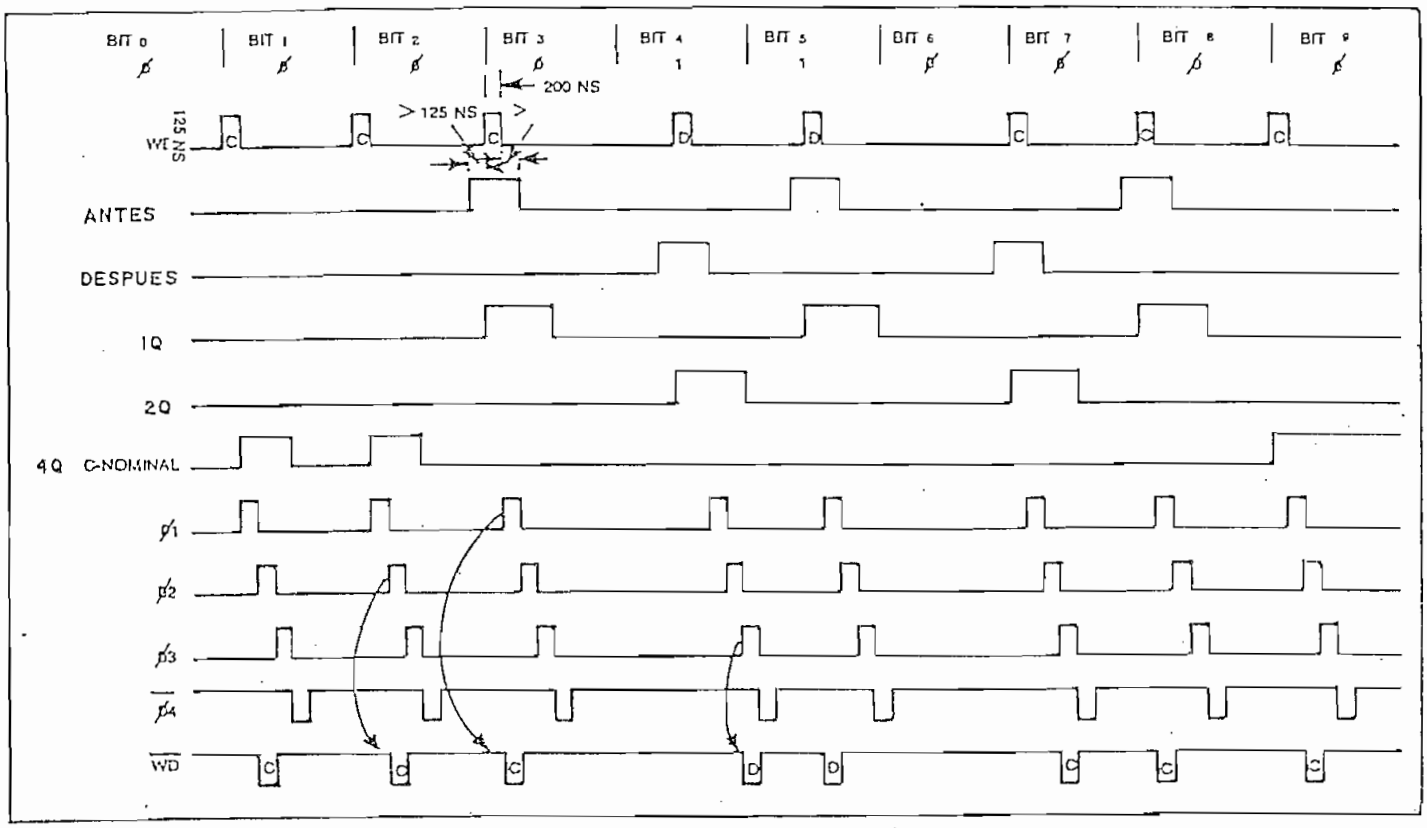


FIGURA 7

número cuatro es utilizada para poner en cero lógico todas las salidas Q de forma que se pueda producir una transición positiva en Q3 con el siguiente pulso en WRITE DATA así como impedir el envío de señales al disco. La ventaja de este método es la de que permite variar la cantidad de precompensación variando la duración de los pulsos de cada fase.

Otro método de precompensación es el digital, un circuito que utiliza este método se ilustra en la figura 8. La cantidad de precompensación en este caso no puede ser variada fácilmente y en el circuito indicado tiene un valor de 250 ns, u

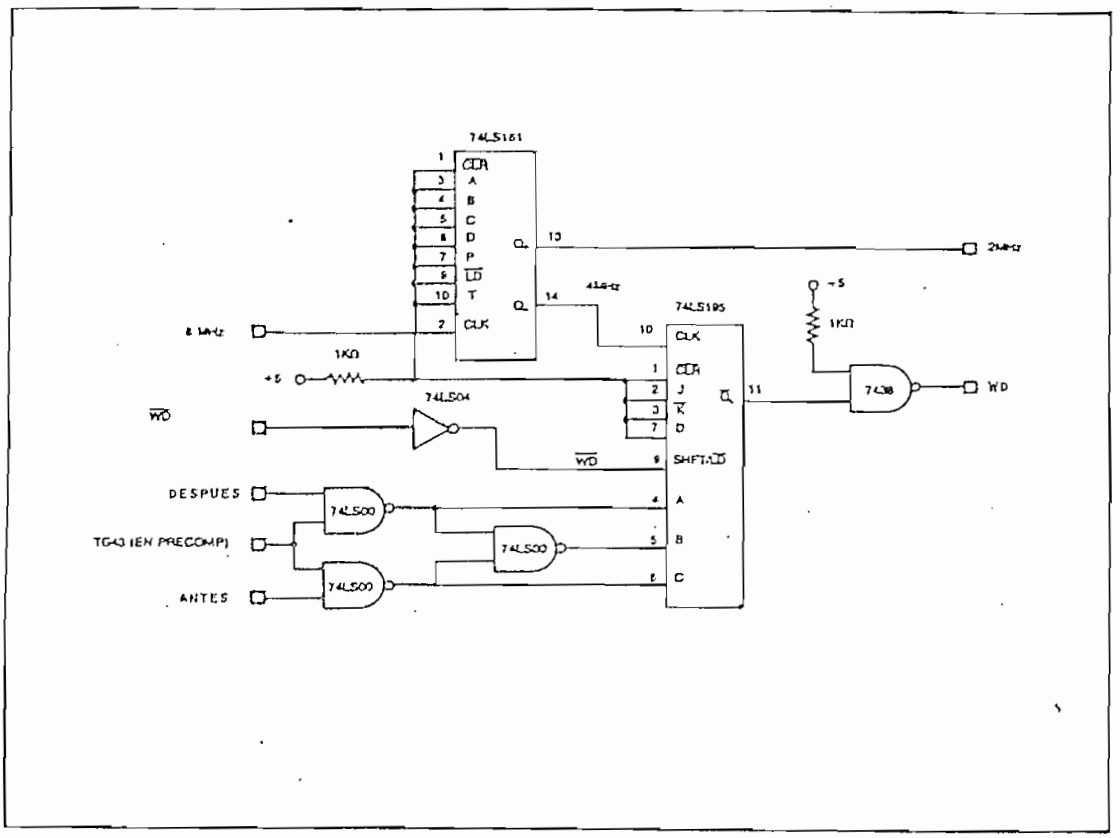


FIGURA 8

na señal de 4 MHz es utilizada por un registro de desplazamiento de cuatro bits como señal de reloj, mientras la señal $\overline{\text{WRITE DATA}}$ esté en uno lógico existirá solamente un desplazamiento de "unos" hacia la salida Q_D pues las entradas seriales J y $\overline{K}$ están en uno lógico también, al producirse un pulso en la señal $\overline{\text{WRITE DATA}}$ un cero lógico es cargado en una de las entradas paralelas A, B ó C de acuerdo a si el bit debe ser escrito antes de, en su posición nominal o después de ella, el cero lógico es luego desplazado por el registro hasta que alcanza la salida Q_D por lo que la demora introducida vendrá dada por el número de desplazamientos de que ha sido objeto. En el circuito indicado la precompensación se realiza solamente en pistas mayores a la 43 como se usa generalmente.

Habiendo estudiado ya la forma de la señal escrita en el disco, veamos ahora su capacidad de almacenamiento de datos. La capacidad de un disco depende de su tamaño, velocidad angular, número de pistas, tiempo entre un bit de datos y el siguiente y, el método de grabación utilizado: FM ó MFM. Como ya se indicó el sistema MFM permite almacenar el doble de información que el FM pues el tiempo entre bits de datos puede ser reducido a la mitad. La velocidad de rotación del disco varía también dependiendo del tamaño del disco, en los sistemas actuales se utilizan velocidades de 360 y 300 RPM para discos de 8 y 5¼ pulgadas respectivamente, el número de pistas por disco no es tampoco una constante, usualmente se tienen 77 pistas en discos de 8 pulgadas y 35 en los de 5¼ pulgadas. Todas las

características indicadas hacen que la capacidad de almacenamiento de datos varíe de la forma indicada en la Tabla I.

TABLA I

TAMAÑO (Pulgad.)	DENSIDAD	CAPACIDAD NOMINAL SIN FORMATO (Bytes) POR PISTA	POR DISCO	TIEMPO ENTRE BITS DE DATOS (Microsegundos)
5¼	SIMPLE	3.125	109.375	8
5¼	DOBLE	6.250	219.750	4
8	SIMPLE	5.208	401.016	4
8	DOBLE	10.416	802.032	2

NOTA: La capacidad nominal por disco ha sido calculada para 35 pistas en discos de 5¼ pulgadas y 77 para los de 8 pulgadas, en discos de doble lado la capacidad total por disco será el doble de la indicada.

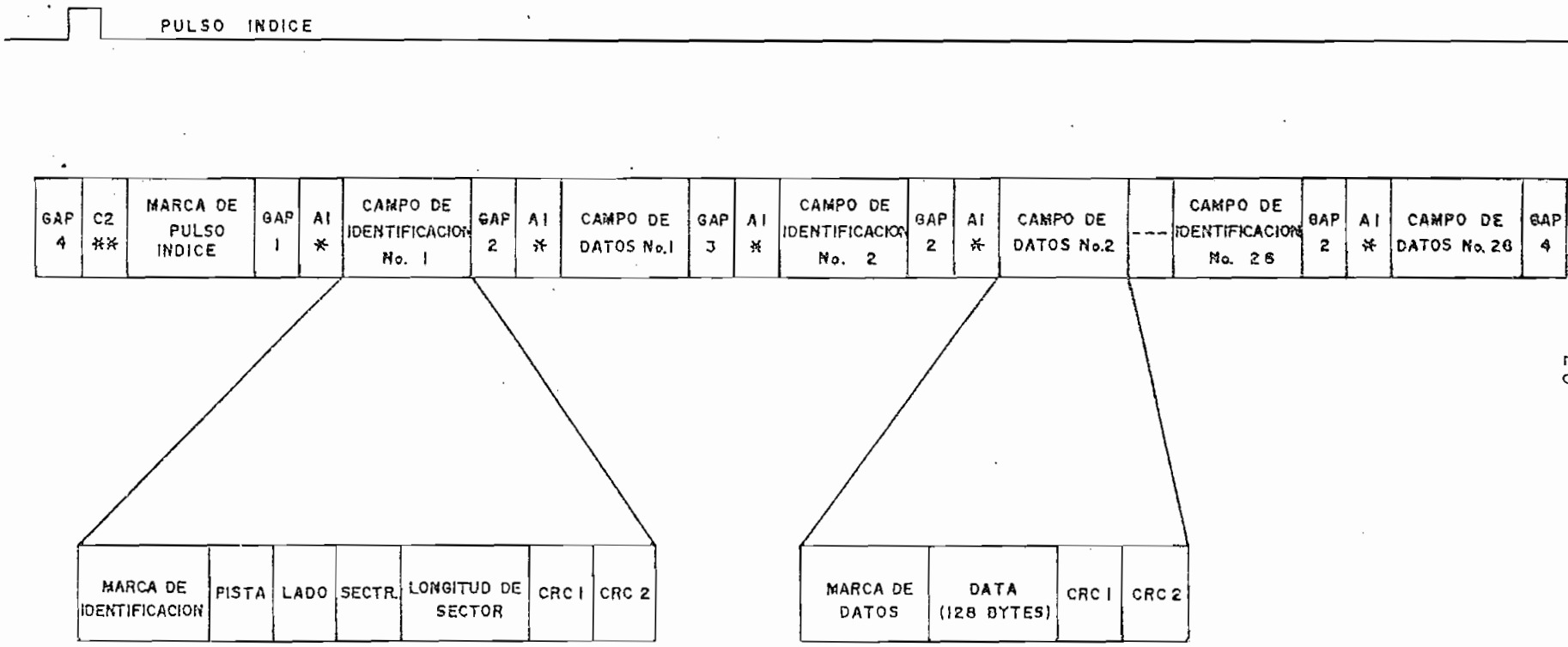
1.3.2 FORMATOS DE ESCRITURA.

Una vez comprendida la forma de grabación de la información en el disco y su capacidad de almacenamiento, podemos pasar a estudiar los formatos. La necesidad de utilizar formatos se debe a que la información que se almacena en el disco debe ser ordenada en cierta forma que permita su recuperación fácilmente. El primer paso a este ordenamiento es el de dividir a la superficie del disco en varias pistas como ya fue indicado pero esto no es suficiente pues la cantidad de información que puede ser escrita en una pista es mucho mayor que las necesida

des usuales, por esta razón se subdivide a cada pista en varios sectores, de esta forma el usuario puede identificar el lugar en que se almacena su información especificando la pista y el sector en que se encuentra. Para poder comprobar que la cabeza de grabación se encuentre sobre la pista y el sector especificados debe escribirse información adicional de forma que al ser leida se pueda identificar el inicio de una pista (en conjunto con el pulso índice) o de un sector, su número y especialmente si la información que se lee es de identificación o de datos. Esta información adicional en conjunto con ciertos espacios utilizados para sincronización de los circuitos de separación de datos y los espacios para los datos del usuario es lo que se conoce como formato.

En la figura 9 se indica la forma general del formato más utilizado en discos floppy de 8 pulgadas, se lo conoce como formato IBM por ser esta compañía la que lo utilizó para sus discos por primera vez. (1) Algo parecido se puede encontrar en la bibliografía adjunta, para discos de 5¼ pulgadas.

Los espacios de sincronización son también llamados "gaps" por su nombre en inglés, su función no es únicamente la de permitir la sincronización del circuito separador de datos sino que además producen una cierta demora para que el sistema que lee los datos pueda decidir cual será su siguiente paso, por último proveen de una separación entre sectores lo suficien



* PULSO DE RELOJ ELIMINADO ENTRE BITS 4 Y 5 (USADO SOLO EN MFM)

** PULSO DE RELOJ ELIMINADO ENTRE BITS 3 Y 4 (SOLO EN MFM)

FIGURA 9

FORMATO IBM

temente grande como para que pequeñas variaciones en la velocidad de rotación del disco no hagan que nuevos datos que se escriban puedan pasar hasta el inicio del siguiente sector.(10).

El comienzo de cada pista, campo de identificación ó campo de datos están precedidos por "marcas de identificación". En el sistema FM pueden ser facilmente detectadas pues, cada una tiene asignado un número hexadecimal, tanto en su señal de datos como en la de reloj, como se indica en la Tabla II.

La forma de la señal escrita en el disco para cada una de ellas puede ser entendida observando el ejemplo de la figura 10 en el que se representa la señal escrita para una marca de pulso índice; otros ejemplos pueden ser encontrados en los apéndices y bibliografía de la presente tesis.

En el sistema MFM no se puede enviar todo un byte por la señal de reloj pues algunos bits de esta señal no son escritos en el disco, por esta razón las marcas de identificación tienen asignado un número hexadecimal tan sólo para su señal de datos, su señal de reloj es escrita en la forma acostumbrada, además, antes de cada marca se escribe otro byte de datos en cuya correspondiente señal de reloj no se escribe uno de los pulsos que de acuerdo con el sistema MFM debería ser escrito, en la Tabla II pueden verse los códigos asignados a cada marca y a los bytes que las anteceden.

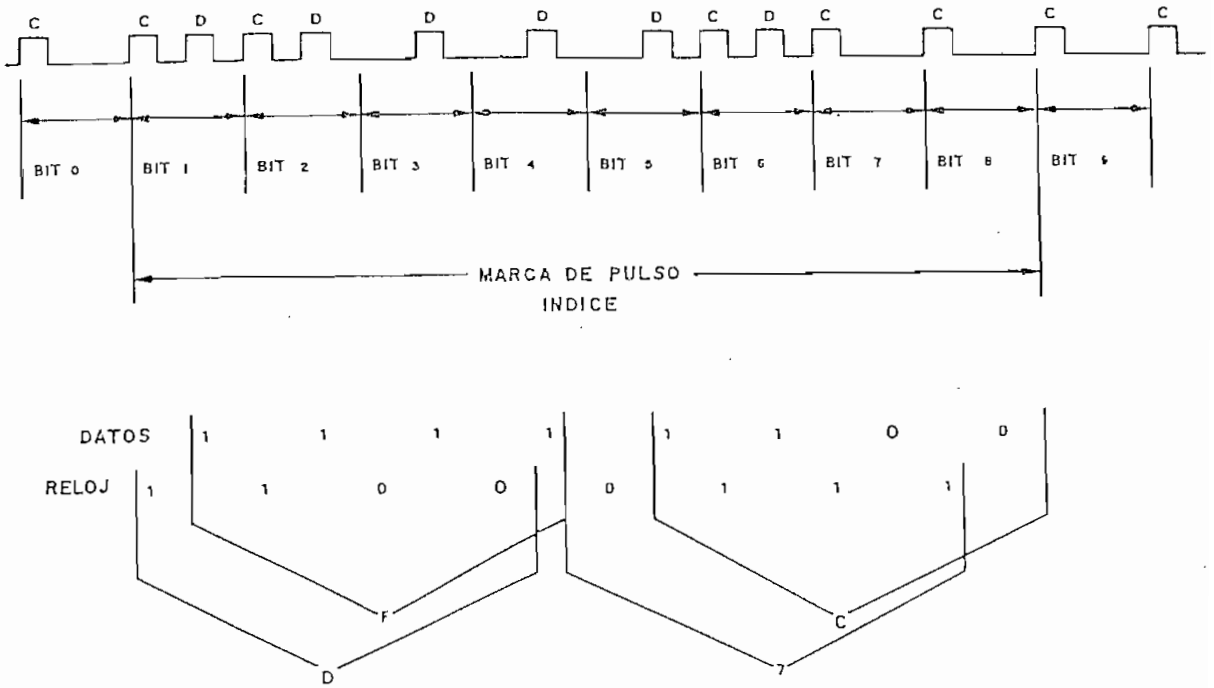


FIGURA 10

MARCA DE PULSO INDICE

TABLA II

FUNCION	CODIGO EN FM		CODIGO EN MFM	
	DATOS	RELOJ	DATOS	BYTE PRECEDENTE
Marca de pulso índice	FC	D7	FC	C2*
Marca de campo de identificación	FE	C7	FE	A1**
Marca de campo de datos	FB	C7	FB	A1**

* Pulso de reloj no escrito entre los bits 3 y 4 del byte de datos.

** Pulso de reloj no escrito entre los bits 4 y 5 del byte de datos.

En algunos casos especiales la marca del campo de datos puede ser cambiada por el código F8 para la señal de datos y C7 para la de reloj, a esta nueva marca se la denomina en inglés "Deleted Data Address Mark" e indica que el campo de datos escrito a continuación tiene alguna característica específica definida por el usuario.

Con el fin de detectar errores de lectura o escritura se escriben dos bytes adicionales al final de los càmpos de identificación y el de datos, estos bytes son conocidos como CRC (en inglés Cyclic Redundancy Check). Para generar los bytes de CRC se considera a la información como un polinomio, por ejemplo, si la información tiene la forma: 110010011 puede ser considerada como un polinomio: $M(X) = X^8 + X^7 + X^4 + X + 1$, si se desea incluir a continuación dos bytes de CRC podemos considerar que en conjunto con la información forman un solo mensaje es decir que el polinomio M(X) debe ser desplazado dieciséis posiciones a la izquierda para dar cabida a los dos bytes de CRC, lo que equivale a multiplicarlo por X^{16}. Si a este nuevo polinomio se lo divide -módulo dos- por otro llamado "polinomio generador" G(X) de orden 16, obtendremos un cuociente Q(X) y un residuo R(X). Puede demostrarse que $X^{16} M(X) + R(X)$ módulo dos, es exactamente divisible para G(X). (11). Cabe anotar que la división módulo dos se la realiza en base a restas módulo dos y que en este tipo de operaciones tanto la suma como la resta producen los mismos resultados. (12).

Para la grabación de información en discos floppy se utiliza el polinomio: $X^{16} + X^{12} + X^5 + 1$ como polinomio generador y ya que $X^{16} M(X)$ tiene sus últimos dos bytes iguales a cero, la suma módulo dos $X^{16} M(X) + R(X)$ tendrá sus últimos dos bytes iguales a R(X), de esta forma la información puede ser escrita normalmente en el disco pero aumentando a continuación de ella los dos bytes del residuo R(X) o bytes de CRC. Durante la lectura se puede hacer la división módulo dos de el conjunto, es decir, de la información y los bytes de CRC para el polinomio generador, en este caso el residuo debe ser igual a cero si no se ha producido algún error en la escritura o en la lectura.

Un circuito digital que realiza esta operación se ilustra en la figura 11, en este circuito se inicializa primero a todos los flip-flops con cero lógico, por medio de la señal de CLEAR, luego cada bit de datos en el campo, empezando desde el bit más significativo de la marca de identificación, son desplazados hacia la entrada del circuito enviando un pulso en la entrada de reloj por cada bit que se envía, para esto la señal de DATOS/CRC debe ser puesta en uno lógico, cuando todos los bits hayan sido pasados por el circuito los flip-flops quedarán con el valor de los bits de CRC correspondientes a la información enviada como se indica en la figura, si se está escribiendo en el disco se puede enviar a continuación cada bit de CRC poniendo la señal de DATOS/CRC en cero lógico y enviando cuatro pulsos a la señal de reloj. Si se está leyendo informa-

ción del disco, se puede realizar un proceso parecido esto es, poner todos los flip-flops en cero lógico por medio de la señal de CLEAR, y pasar a todos los bits de la información y los bits de CRC manteniendo la señal DATOS/CRC en uno lógico, en este caso el valor final almacenado en todos los flip-flops debe ser igual a cero si no se ha producido ningún error (lo que puede ser detectado por medio de una compuerta OR de cuatro entradas). (13). Cabe anotar que el mismo proceso puede ser realizado por medio de programación.

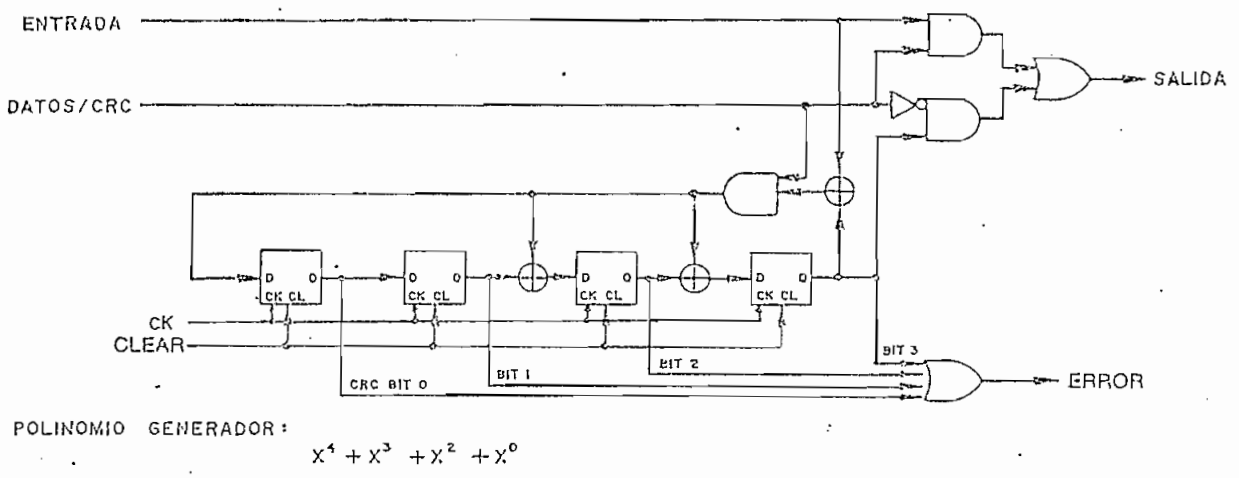


FIGURA 11

GENERADOR DE CRC.

El formato IBM no está limitado a 128 bytes por sector sino que sin perder compatibilidad se puede tener longitudes de 256, 512, 1024 bytes por sector, desde luego entre mayor sea la longitud del sector, menor será el número de sectores que

pueden entrar en una pista. Al tener 256 bytes por sector o más, se puede aprovechar en mejor forma la capacidad de almacenamiento del disco ya que con menos sectores por pista se eliminan muchos de los espacios entre sectores así como algunos bytes de identificación. La tabla III sumariza las longitudes de sector más utilizadas para cada tamaño de disco y la capacidad de almacenamiento de datos para cada caso, además se indica el tiempo que transcurre entre la escritura o lectura de dos bytes consecutivos. Cabe anotar que la capacidad de almacenamiento de datos puede ser duplicada utilizando discos de doble lado.

TABLA III

TAMAÑO (Pulg.)	DENSIDAD	BYTES POR SECTOR	NUMERO DE SECTORES	CAPACIDAD DE ALMACENAMIENTO POR PISTA (Bytes)	CAPACIDAD DE ALMACENAMIENTO POR DISCO (Bytes)	TIEMPO ENTRE BYTES (Microsegun.)
5¼	SIMPLE	128	18	2304	80.640 *	64
5¼	DOBLE	256	18	4608	161.280	32
5¼	SIMPLE	256	10	2560	89.600	64
8	SIMPLE	128	26	3328	256.256 **	32
8	DOBLE	256	26	6656	512.512	16
8	SIMPLE	256	15	3840	295.680	32

* Basado en 35 pistas por disco.

** Basado en 77 pistas por disco.

Se pueden utilizar otros formatos que a pesar de no ser compatibles con el formato IBM sean de mayor utilidad para el usuario, algunos ejemplos de ellos se pueden encontrar en los apéndices y bibliografía de esta tesis, allí se puede encontrar también información sobre los formatos utilizados en discos del tipo "hard sectored" que por su sencillez no han sido tratados aquí.

C A P I T U L O S E G U N D O

" CONTROLADOR E INTERFACE PARA DISCOS FLOPPY "

2. CONTROLADOR DE DISCOS FLOPPY E INTERFACE.

En el capítulo anterior se dio una descripción de los discos floppy, la forma de almacenar la información en ellos y las lectoras/escritoras de este tipo de discos, en este capítulo se estudiarán los circuitos que controlan el funcionamiento de las lectoras/escritoras así como los procesos de interface con el sistema principal.

Un diagrama de bloques general de un sistema acoplado a un periférico se muestra en la figura 12. La interface permite la interconexión de el sistema principal con un periférico, las señales de control y de datos son enviadas al periférico por medio de un "controlador de periférico" que convierte las señales de control y datos enviadas por la interface en aquellas señales específicas requeridas por el periférico. (14).

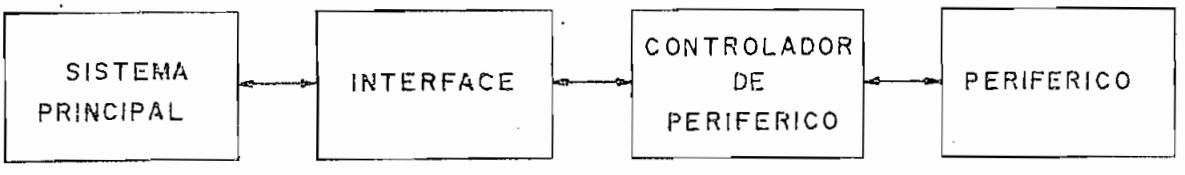


FIGURA 12

En el caso de discos floppy el controlador debe encar-

garse de las siguientes funciones principalmente:

- Recibir y transmitir datos en forma paralela desde o hacia el sistema principal. (En conjunto con la interface).
- Serializar los datos y codificarlos de acuerdo al sistema utilizado FM ó MFM.
- Realizar los procesos de precompensación si son utilizados.
- Indicar si se desea leer o escribir información en el disco.
- Impedir cualquier proceso si la lectora/escritora no está lista y en particular la escritura de un disco protegido contra escritura.
- Encontrar la pista número ØØ, para lo cual se deben mandar los pulsos necesarios por la señal de $\overline{\text{STEP}}$, actualizar la señal $\overline{\text{DIRECTION}}$ y esperar hasta que la señal $\overline{\text{TRACK ØØ}}$ se ponga en su estado activo.
- Luego de haber encontrado la pista ØØ, el controlador debe ser capaz de generar los pulsos necesarios en la línea $\overline{\text{STEP}}$ para posicionar la cabeza sobre cualquier pista del disco, en conjunto con la señal $\overline{\text{DIRECTION}}$, indicando además si la cabeza se encuentra en una pista mayor a la número 43.
- Producir demoras de tiempo suficientes como para que aquellos procesos en la lectura/escritora que producen movimientos mecánicos lleguen a un estado estable.

- Recibir la señal leida del disco, separar las señales de reloj y de datos e identificar si la información leída está formada por datos del usuario o si corresponde a alguna de las partes del formato.

Algunas de las funciones indicadas pueden ser realizadas utilizando programación por lo que antiguamente se las realizaba directamente en el sistema principal, ocupando tiempo y espacio de memoria en él pero ahorrando muchos circuitos en el controlador que aunque podrían realizar estas funciones, encarecerían al sistema completo. Actualmente, debido a la extensa utilización de discos floppy, se han creado circuitos integrados específicamente diseñados para actuar como controladores de las lectoras/escritoras de este tipo de discos, liberando de esta manera al sistema principal de muchas de las tareas que tenía a su cargo, haciendo que los discos floppy puedan incluso ser utilizados para sistemas pequeños además de haber disminuido el tamaño y costo del controlador para discos floppy.

2.1. METODOS DE SEPARACION DE SEÑALES DE DATOS Y DE RELOJ.

Una de las funciones más importantes que debe realizar el controlador es la de separar las señales de datos y de reloj de la señal leída del disco, por este motivo y por el hecho de que normalmente esta función no está implementada directamente en los circuitos integrados que actúan como controla-

dores, se van a estudiar aquí algunos de los métodos de separación especialmente aquellos novedosos que usualmente no son encontrados en la bibliografía relacionada con este tema.

Los circuitos de separación varían de acuerdo al sistema de grabación utilizado: FM o MFM. Dentro de cada sistema se pueden tener dos tipos de circuitos, los primeros son llamados circuitos de "separación completa", separan verdaderamente las señales de datos y de reloj de la señal enviada por la lectora/escritora. Otros circuitos separadores producen simplemente una señal de reloj sincronizada con la señal recibida, en ellos cada ciclo positivo o negativo de la señal producida contiene a un bit, ya sea de datos o de reloj (no importa la polaridad), su utilización es factible gracias a que la verdadera separación puede ser realizada por algunos circuitos integrados que sirven como controladores, de esta forma el circuito separador puede ser mucho más sencillo, especialmente en el caso del sistema MFM.

En la figura 13 se encuentra uno de los circuitos utilizados para la separación completa de las señales de datos y de reloj para el sistema FM; su funcionamiento es el siguiente: Los pulsos enviados por la lectora/escritora son acondicionados por medio del monoestable 10b, de esta manera pulsos negativos de duración constante son enviados a un demultiplexer 1 a 2 formado por las compuertas 3b, 4c y 4d, las dos salidas

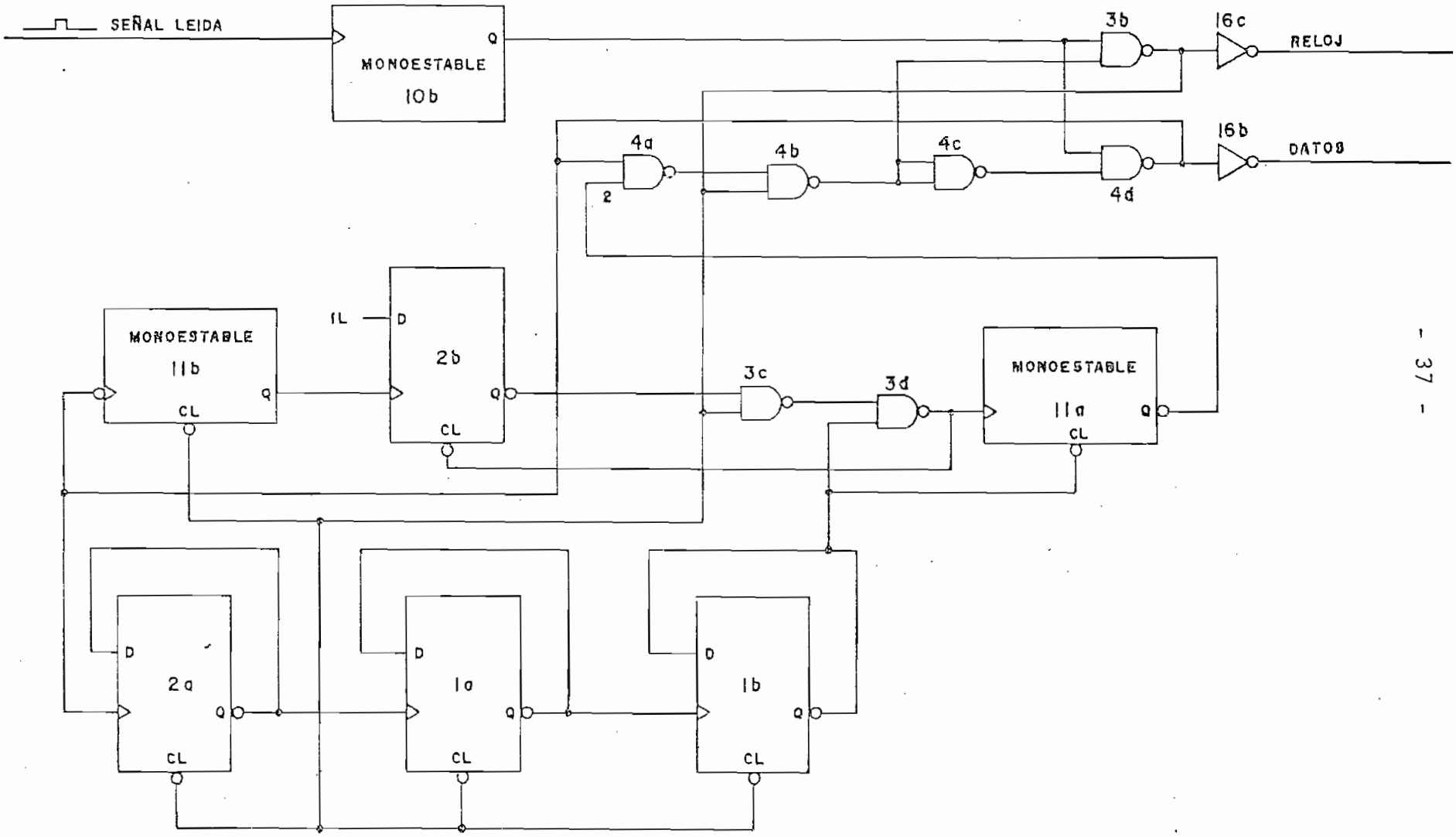


FIGURA 13

CIRCUITO SEPARADOR DE DATOS

del mismo son complementadas por medio de las compuertas 16c y 16b para obtener pulsos positivos correspondientes a la señal de reloj y a la de datos, la señal de selección está unida a una de las entradas de la compuerta 4a; cuando ésta se encuentra en cero lógico permite la salida de la señal producida en 10b por la línea de datos, mientras que si se encuentra en uno lógico, la señal de 10b pasa hacia la línea de reloj. Las compuertas 4a y 4b mantienen la selección mientras exista un pulso en las líneas de reloj o de datos; de esta forma, los pulsos pasan completos sin importar que mientras estén presentes se seleccione otra salida.

La señal de selección es producida por un monoestable que puede ser disparado de dos maneras diferentes, la primera emplea la salida de la compuerta 3b (pulsos negativos de reloj) enviando esta señal por medio de las compuertas 3c y 3d a la entrada de disparo con transición negativa de ella, la segunda forma emplea la salida de 4d (pulsos negativos de datos) para disparar al monoestable 11b que produce una demora suficiente como para que su desactivación ocurra ligeramente después del instante que corresponde a un nuevo pulso de reloj (es preferible demorar un poco más de forma que si existe una variación pequeña de la posición del pulso de reloj con respecto a su posición nominal, no se envíe al pulso a la salida de datos). Al desactivarse el monoestable 11b, un cero lógico se produce en la salida 2b, este cero lógico produce un pul

so en la salida de 3d debido a la realimentación a la entrada de borrado de 11b, el pulso producido se usa para pisparar al monoestable de selección 11a. De esta forma 11a es siempre disparado ya sea por medio de un pulso de reloj o uno de datos pues en el sistema FM no puede darse el caso que dos pulsos consecutivos uno de reloj y otro de datos no se encuentren presentes, cabe notar que al dispararse 11a se selecciona la salida de datos en espera del siguiente pulso de datos.

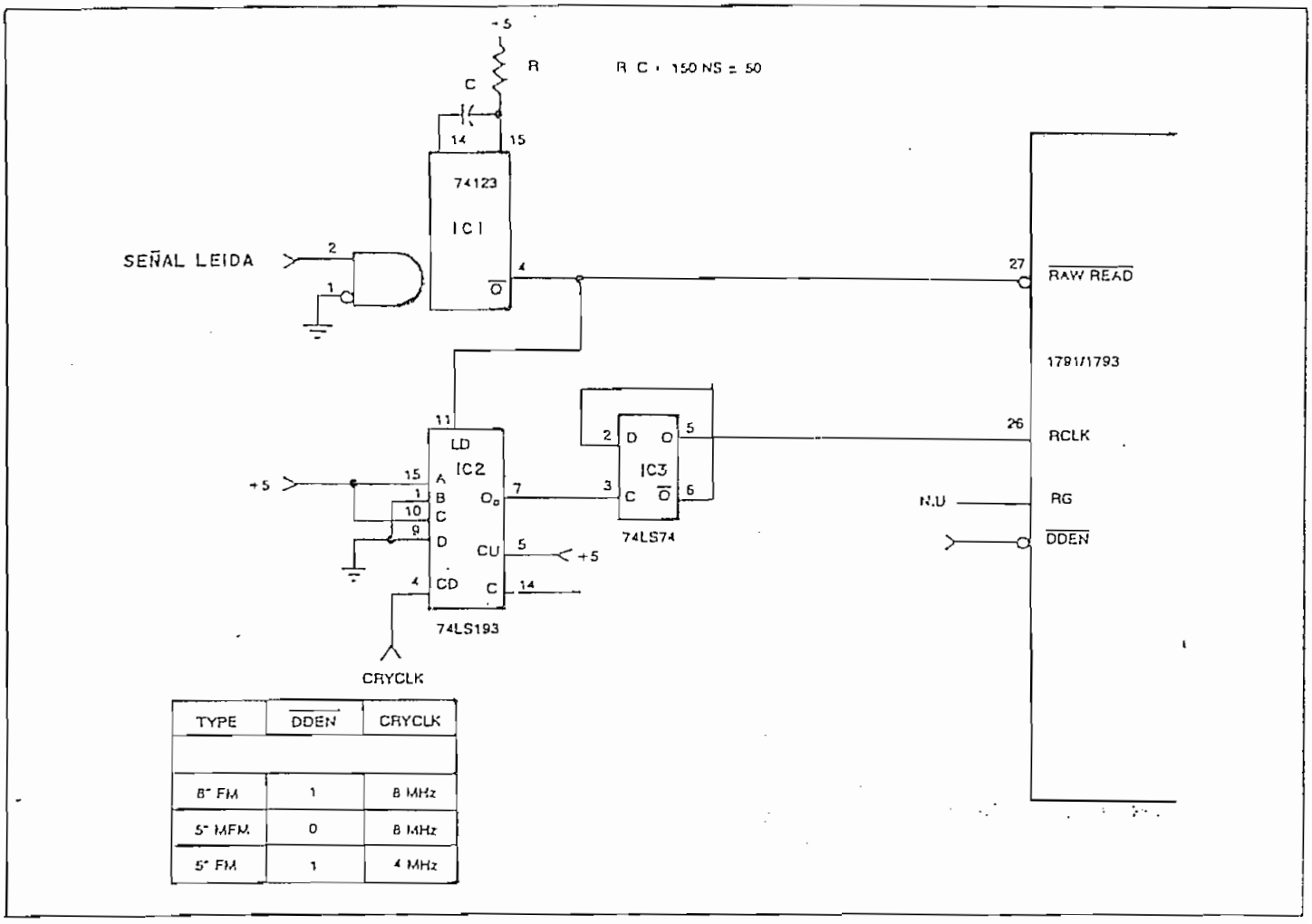
El circuito debe además determinar si se encuentra en la secuencia correcta, es decir, si la señal enviada a la salida de reloj corresponde realmente a la señal de reloj y no a la de datos, e igual cosa para la salida de datos. Esta función se realiza en base a tres flip-flops tipo D: 1a, 1b y 2a conectados en forma de un contador módulo 8 cuyas entradas de borrado se han conectado a la salida de la compuerta 3b (pulsos negativos de la señal de reloj), la entrada de reloj de este contador se la obtiene de la salida de la compuerta 4d (pulsos negativos de la señal de datos). Hay que recordar que la señal de reloj no puede tener más de tres ceros consecutivos (en las marcas de campos de datos o de identificación) y en estos casos los bits de datos que están entre ellos así como el que los precede y el que los antecede son son todos iguales a uno lógico. De esta forma si el contador no es borrado (por la señal de reloj) durante cuatro pulsos de datos, la salida de 1b se pone en cero lógico manteniendo por tanto la salida

del monoestable 11a en uno lógico, es decir, seleccionando la salida de reloj, si el siguiente bit leído por la lectora/escritora es un uno lógico (como sucede en una marca de campos de datos o de identificación), se borrará el contador a la vez que se producirá una transición negativa en la salida de la compuerta3d, disparándose el monoestable 11a para seleccionar al siguiente pulso de datos, en cambio si el siguiente bit es un cero lógico, el contador no será borrado por lo que se seguirá seleccionando la salida de reloj, hasta que aparezca en ella un pulso que borre el contador con lo que el circuito comenzará a funcionar normalmente y en la secuencia correcta. De esta manera el circuito admite hasta tres ceros lógicos por la señal de reloj, intercambiando las señales de datos y de reloj si existen más, consiguiendo de esta manera que el circuito separe las señales en la secuencia correcta.

En la figura 14 se indica otro de los circuitos utilizados para la separación de datos. En este caso la separación no es completa sino que el circuito produce sólamente una señal simétrica de reloj, cada ciclo de la misma contiene a un bit de datos o de reloj pero no importa la polaridad, es decir que los ciclos positivos pueden contener a bits de reloj y los negativos a bits de datos o viceversa.

La señal proveniente de la lectora/escritora es acondicionada por medio del monoestable 1, su salida es utilizada

para cargar en forma paralela los bits del contador 2. El contador es del tipo "UP/DOWN" módulo 16 y mientras no se produzca un pulso en la salida del monoestable 1, se encuentra contando libremente hacia abajo por medio de la señal de reloj



TYPE	$\overline{DDEN}$	CRYCLK
8" FM	1	8 MHz
5" MFM	0	8 MHz
5" FM	1	4 MHz

FIGURA 14

CONTADOR/SEPARADOR

"CRYCLK". La salida Q_D del contador es utilizada como señal de reloj para un flip-flop tipo D conectado como un contador módulo 2, de esta forma el flip-flop produce una señal simétrica que, como se verá a continuación, contiene a un bit de reloj o de datos por cada ciclo de la misma.

Al producirse un pulso en la salida del monoestable, el contador es inicializado con el número 5_{10}. Al retornar la salida del monoestable a uno lógico, el contador empezará a contar -hacia abajo- hasta la cuenta de 15_{10} en donde la salida del flip-flop "RCLK" será complementada, asegurando de esta manera que el pulso quede contenido dentro de un ciclo de la señal RCLK. En es sistema FM con discos de 8" el siguiente pulso podrá ocurrir luego de 2 µSeg, tiempo para el cual el contador se encontrará nuevamente con la cuenta de 5_{10}, es decir que el circuito queda sincronizado con la señal enviada por la lectora/escritora y de producirse un desplazamiento en la posición de los bits de datos o de reloj, el separador está en capacidad de alcanzar nuevamente el sincronismo. Casos similares se tendrán para discos de 5¼" y para el sistema MFM.

La posición relativa de un bit dentro de cada ciclo de la señal RCLK puede ser variada cambiando el número con que se inicializa el contador (en el ejemplo 5_{10}). El contador puede ser inicializado con números entre $Ø_{10}$ y 7_{10} pues al producirse la inicialización, no debe producirse una transición posi-

tiva en Q_D como podría ocurrir para números entre 8_{10} y 15_{10}. Cada bit quedará más centrado en el ciclo entre más se tienda a 7_{10} como número de inicialización, sin embargo, el número que se utilice para inicializar el contador debe ser aquel que produzca los mejores resultados en la práctica.

Al estudiar la precompensación en la escritura, se explicó que pueden ocurrir pequeños desplazamientos en la posición de los bits escritos en el disco, especialmente en el caso del sistema MFM. Estos desplazamientos son reducidos por medio de la precompensación pero no son completamente eliminados, por esta razón no es aconsejable la utilización del circuito separador de datos de la figura 14 en el caso de discos de 8" con el sistema MFM. De utilizarse este circuito, cada bit leído del disco sincronizaría al separador en forma diferente y no se alcanzaría un funcionamiento estable del mismo.

Las razones indicadas hacen que el circuito de separación de datos más utilizado para el sistema MFM con discos de 8" sea del tipo PLL (Phase-Lock-Loop), en este tipo de circuitos, el desplazamiento de un bit leído no produce una gran variación de la frecuencia de la señal de salida y sólamente desplazamientos sucesivos de varios bits la podrán variar significativamente. A pesar de que los circuitos PLL son los más apropiados para la separación de datos en el sistema MFM, se han desarrollado circuitos más sencillos, puramente digitales

ADDRESS	DATA
00	01
01	01
02	02
03	03
04	03
05	04
06	05
07	06
08	0B
09	0D
0A	0C
0B	0E
0C	0F
0D	0F
0E	00
0F	01
10	01
11	02
12	03
13	04
14	05
15	06
16	07
17	08
18	09
19	0A
1A	0B
1B	0C
1C	0D
1D	0E
1E	0F
1F	00

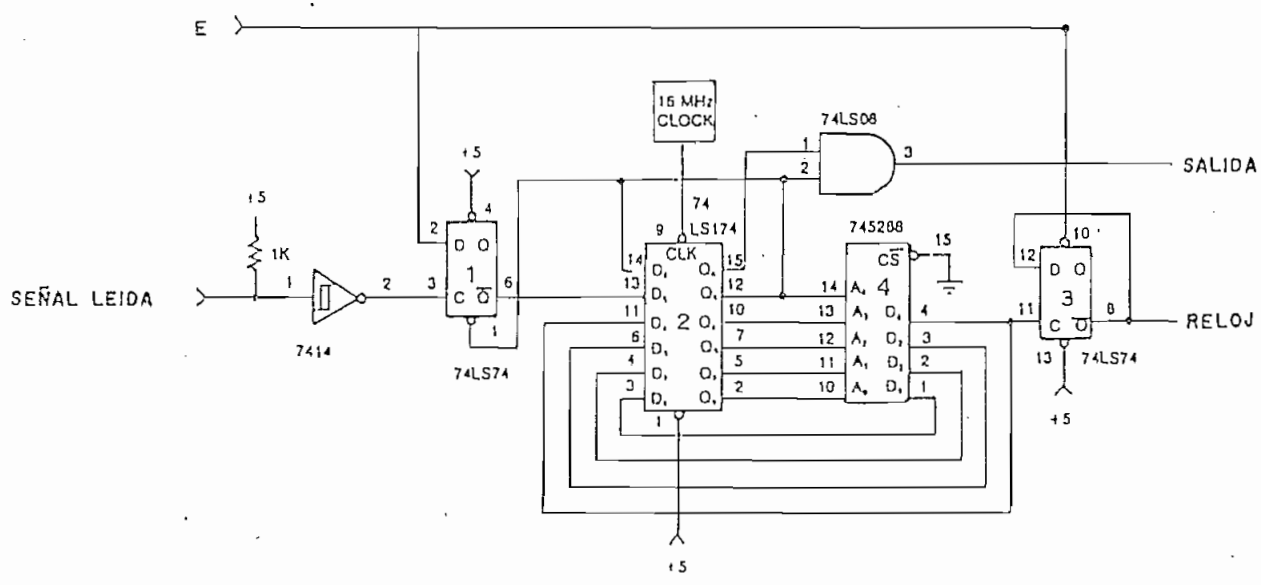


FIGURA 15

CIRCUITO SEPARADOR DE DATOS

que cumplen aproximadamente la misma función. Uno de estos circuitos se ilustra en la figura 15.

El funcionamiento del circuito es el siguiente: mientras la lectora/escritora no mande un bit de datos o de reloj, la salida $\overline{Q}$ del flip-flop IC1 se mantendrá en uno lógico; la salida Q5 de IC2 copiará este dato con la siguiente transición negativa del reloj de 16 MHz, seleccionando de esta forma las dieciseis últimas direcciones de la memoria PROM. Como puede verse en la tabla adjunta, la memoria PROM está programada de forma que si sus dieciseis últimas direcciones de memoria son seleccionadas, IC2 en conjunto con la memoria PROM actuarán como un contador módulo 16. El bit más significativo de este contador: D4, es utilizado como señal de reloj para IC3 que está conectado como un contador módulo 2 que además de producir una señal simétrica (que constituye la salida del circuito), impide la salida de señal si la línea de habilitación E está en cero lógico.

Si un bit de datos o de reloj es leído del disco y si E está en uno lógico, la salida $\overline{Q}$ de IC1 se pondrá en cero lógico; este dato es luego copiado en la salida Q5 haciendo que la salida $\overline{Q}$ de IC1 regrese a uno lógico, que la salida de la compuerta AND se ponga en cero lógico y se seleccionen las quince primeras direcciones de la memoria PROM. Con la siguiente transición negativa del reloj de 16 MHz, Q5 regresa a uno lógico, Q6 se pone en cero lógico (manteniendo la salida de la

compuerta AND en cero lógico) y el contador comienza a contar nuevamente. El conteo comienza con un número igual al que se tuvo al recibir el bit aumentado en uno, o con un número mayor o menor a este en una o dos unidades como se indica en la tabla de programación. Con la siguiente transición negativa en el reloj de 16 MHz, la salida de la compuerta AND regresará a uno lógico, es decir, que en esta salida se repetirán los pulsos que se tienen en la entrada del circuito, pero los pulsos producidos tendrán siempre una duración constante igual a 125 µSeg.

Puede notarse que el circuito actúa como un contador al que se lo adelanta o atrasa de forma que cuando un bit sea leído del disco, el contador se encuentre con la cuenta de cero, asegurando de esta forma que los bits queden centrados en cada ciclo de la señal de salida. Si el circuito se encuentra funcionando sin que se produzcan atrasos o adelantos, la señal de salida tendrá un período de 2 µSeg y puesto que los bits de datos o de reloj son leídos de un disco de 8" en el sistema MFM cada 2 µSeg, se tendrá que todos los ciclos positivos de la señal de salida contendrán a los bits de datos y los ciclos negativos a los de reloj o viceversa, pero ningún ciclo contendrá a más de un bit.

Cabe anotar que el mismo circuito puede ser utilizado como separador de datos para el sistema FM con discos de 8" y

C A P I T U L O T E R C E R O

"DISEÑO Y CONSTRUCCION DEL CONTROLADOR E INTERFACE PARA DISCOS FLOPPY"

los discos de 5¼" de un solo lado utilizando el sistema FM, son los más aconsejables para el desarrollo de la presente tesis.

En base a las consideraciones anteriores se decidió utilizar la lectora/escritora SA400 producida por la compañía Shugart Associates, diseñada para trabajar con discos de 5¼" de un solo lado y de densidad simple. Esta lectora/escritora puede trabajar con los dos tipos de discos indicados en el capítulo II, esto es: "hard sectored" y soft sectored"; la velocidad angular del disco es controlada electrónicamente, por lo que resultan menos vulnerables a variaciones de frecuencia o voltaje en la red, además su bajo costo y confiabilidad hacen que estas lectoras/escritoras sean muy utilizadas. (15).

A pesar de que las necesidades de almacenamiento no son mayores actualmente, el sistema completo deberá preveer expansiones futuras por lo que el controlador que se diseñe deberá funcionar también con discos de 8" tanto en el sistema FM como en el MFM; con discos de un solo lado o doble lado y deberá tener la capacidad de trabajar con varias lectoras/escritoras a la vez. Por último, el sistema completo deberá trabajar con discos del tipo "soft sectored", permitiendo de esta forma que el usuario defina la longitud de los sectores en el disco de acuerdo a la aplicación específica que se de al equipo.

3.1 DISEÑO DEL CONTROLADOR.

TABLA IV

CARACTERISTICA	INTEGRADO NUMERO:					
	MC 6843	FD 1771	FD 1791	FD 1793	FD 1795	FD 1797
Compatibilidad con circuitos TTL	S	S	S	S	S	S
Número de fuentes de voltaje	1	3	2	2	2	2
Capacidad de interface directa con el microprocesador 6800	B	R	R	R	R	R
Puede trabajar en densidad simple y doble (FM y MFM)	N	N	S	S	S	S
Precompensación en la escritura (interna)	N	N	S	S	S	S
Separación interna de datos en FM	S	S	S	S	S	S
Separación interna de datos en MFM	N	N	S	S	S	S
Capacidad de DMA	S	S	S	S	S	S
Compatibilidad con discos "soft sectored"	S	S	S	S	S	S
Bus de datos normal	S	N	N	S	N	S
Bus de datos invertidos	N	S	S	N	S	N
Puede trabajar con discos de doble lado (directamente)	N	N	N	N	S	S
Generación y comprobación interna de CRC	S	S	S	S	S	S
Generación interna de pulsos de "Step" de duración programable	S	S	S	S	S	S
Demora programable para bajado de la cabeza	S	N	N	N	N	N
Capacidad para trabajar con formatos diferentes al IBM	M	B	B	B	B	B
Capacidad para trabajar con diferentes longitudes de sector	M	R	B	B	B	B
Capacidad de crear el formato en el disco	R	B	B	B	B	B
Número de comandos (operaciones preprogramadas)	10	11	11	11	11	11

SIMBOLOS UTILIZADOS: S = Si B = Buena R = Regular
N = No M = Mala

Puesto que la duración del pulso producido por el monoestable es relativamente larga, se ha escogido el integrado 74LS123 para cumplir con esta función pues con él se tendrán las siguientes ventajas: bajo consumo de corriente, conexión directa de condensadores electrolíticos (sin utilizar diodos de protección) y redisparo.

En este tipo de monoestable el ancho del pulso puede ser calculado con la siguiente fórmula: (17)

$$t_w = 0{,}45 \times R_t \times C_{ext}$$

Para el cálculo de R_t y C_{ext} se tomará a t_w = 100 mSeg. para asegurar que en la práctica el ancho del pulso no llegue a ser menor a 75 mSeg. a pesar de las tolerancias de los elementos.

Escogiendo una resistencia dentro de los valores típicos para el monoestable y aplicando la fórmula se llegó a determinar que:

$$R_t = 47\ K\Omega$$

$$C_{ext} = 4{,}7\ \mu F$$

Por recomendación de la compañía productora de la lectora/escritora SA400, se ha decidido que el motor de la misma de

be encenderse sólamente mientras esté en uso y mantenerse encendido durante unos dos segundos después de la última operación realizada. De esta forma se disminuye el desgaste tanto del disco como de la lectora/escritora. (18).

Para cumplir con este requisito de la lectora/escritora se utilizará el integrado 555 como se indica en la figura 16.

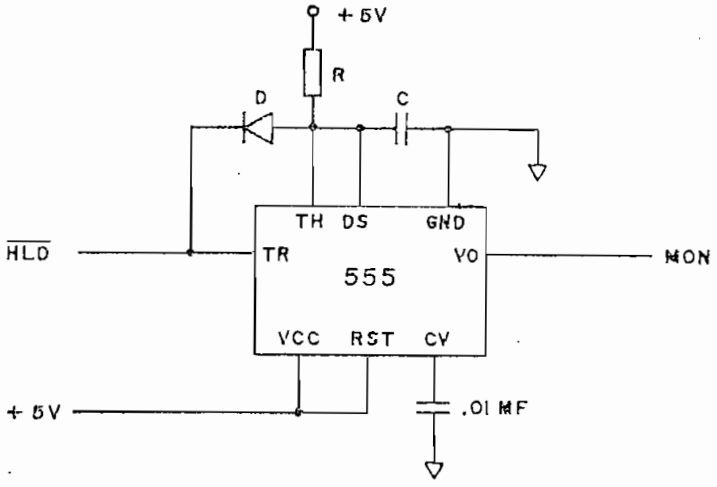


FIGURA 16

El funcionamiento del circuito es el siguiente:

Originalmente el condensador C se encuentra descargado (por medio de la línea DS) y la salida MON (Motor On) en cero lógico. Si la línea $\overline{HLD}$ se pone en cero lógico el integrado 555 se dispara, permitiendo que el condensador se cargue y que la salida MON cambie su estado a uno lógico, indicando de esta forma que el motor de la lectora/escritora debe encenderse. Por la presencia del diodo D, el condensador no podrá cargarse más allá de la suma del voltaje de conducción del diodo y el

voltaje en cero lógico de la línea $\overline{HLD}$.

Si la línea $\overline{HLD}$ regresa a uno lógico, el condensador C podrá cargarse por medio de R hasta el voltaje de transición "threshold voltage" del integrado 555. En ese instante el condensador será descargado por medio de la línea DS y la línea MON se pondrá nuevamente en cero lógico por lo que el motor de la lectora/escritora se apagará.

Para calcular los valores de R y C hay que recordar que el condensador se carga de acuerdo a la siguiente ecuación:

$$V_c = V_{cc} (1 - e^{-t/RC})$$

De donde:

$$t = RC \ln \left(\frac{V_{cc}}{V_{cc} - V_c} \right)$$

Tomando en cuenta que el condensador se carga desde un voltaje V_k igual a la suma del voltaje de conducción del diodo y el voltaje en cero lógico de la línea $\overline{HLD}$, hasta el voltaje de transición igual a $2/3V_{cc}$ se tiene que el tiempo que transcurre desde que $\overline{HLD}$ regresa a uno lógico y MON se pone en cero lógico será:

$$T = RC \ln \left(\frac{V_{cc} - V_k}{1/3\ V_{cc}} \right)$$

Asumiendo que el voltaje de conducción del diodo sea de 0,6 voltios y el voltaje en cero lógico de la línea $\overline{HLD}$ sea de 0,2 voltios se tiene:

$$T = 0{,}92\ RC$$

Si se desea que el motor de la lectora/escritora se mantenga encendido durante 2 Seg. después de que la línea $\overline{HLD}$ regresa a uno lógico, se pueden escoger los siguientes valores para R y C:

$$R = 220\ K\Omega\ ;\quad C = 10\ \mu F$$

El hecho de que la lectora/escritora no permanezca encendida en todo momento, hace necesario el utilizar otro circuito que impida la realización de cualquier operación en el disco hasta que se estabilice la velocidad del motor cada vez que se encienda. En el caso de la lectora/escritora SA400 el fabricante especifica que se debe dejar pasar por lo menos 1 Seg. para que dicha estabilización ocurra. (19).

El lapso de 1 Seg. puede ser medido en base a la duración de un pulso producido en el segundo de los monoestables del integrado 74LS123; el disparo del mismo debe producirse cada vez que se encienda el motor de la lectora/escritora es decir, con la señal MON.

Puesto que el integrado FD1797-02 no está provisto de una entrada adicional en la que se indique que la velocidad del motor se ha estabilizado, la única manera de impedir que éste integrado realice operaciones en el disco es la de utilizar nuevamente la línea HLT, haciendo que ésta se ponga en uno lógico sólamente cuando se cumplan dos condiciones: que la cabeza grabadora haya bajado y que la velocidad del motor sea estable. Para esto se hará que las salidas $\overline{Q}$ de los monoestables del integrado 74LS123 vayan a una compuerta AND 74LS08 cuya salida esté unida a la línea HLT del integrado FD1797-02.

Para asegurar que la duración del pulso producido por el monoestable no llegue a ser menor que 1 Seg., el cálculo de los valores de R_t y C_{ext} se lo hizo tomando el ancho del pulso como de 1,4 Seg. de acuerdo a la fórmula:

$$t_w = 0{,}45 \times R_t \times C_{ext}$$

De donde se llegó a determinar que:

$$R_t = 100\ K\Omega$$

$$C_{ext} = 33\ \mu F$$

Otro de los requerimientos del integrado FD1797-02 es el de una señal de reloj cuya frecuencia sea de 2 MHz para discos de 8" y de 1 MHz para los de 5¼". Esta señal se la obtendrá de uno de los osciladores controlados por voltaje (VCO) del integrado 74S124 y de dos flip-flops tipo D conectados como contadores módulo dos.

El integrado 74S124 tiene la ventaja de trabajar como un oscilador muy estable, conectando simplemente un cristal de la frecuencia deseada en lugar del condensador externo. Internamente contiene a dos osciladores controlados por voltaje por lo que se podrá utilizar uno de ellos para generar la señal de reloj y el restante para el circuito PLL que se utilizará en la separación de datos.

La frecuencia de oscilación deberá ser de 4 MHz. de forma que al ser dividida 2 ó 4 veces en los flip-flops tipo D se obtenga una señal simétrica de 2 ó 1 MHz. La frecuencia de salida podrá ser seleccionada por medio de dos "SPST DIP Switches" como se indica en la figura 17.

Por recomendación del fabricante (20), la línea de control de frecuencia deberá estar en cero lógico y la línea de rango en uno lógico, esto último será conseguido por medio de una resistencia de 1K conectada a Vcc.

El hecho de que se utilice un circuito del tipo Schottky y no uno de menor consumo de corriente, se debe primordialmente a las exigencias de linealidad en el control de frecuencia para el circuito PLL que se utiliza en la separación de datos. Este circuito ha sido tomado de las notas de aplicación editadas por la compañía Western Digital para su familia de integrados FD 179X-02.

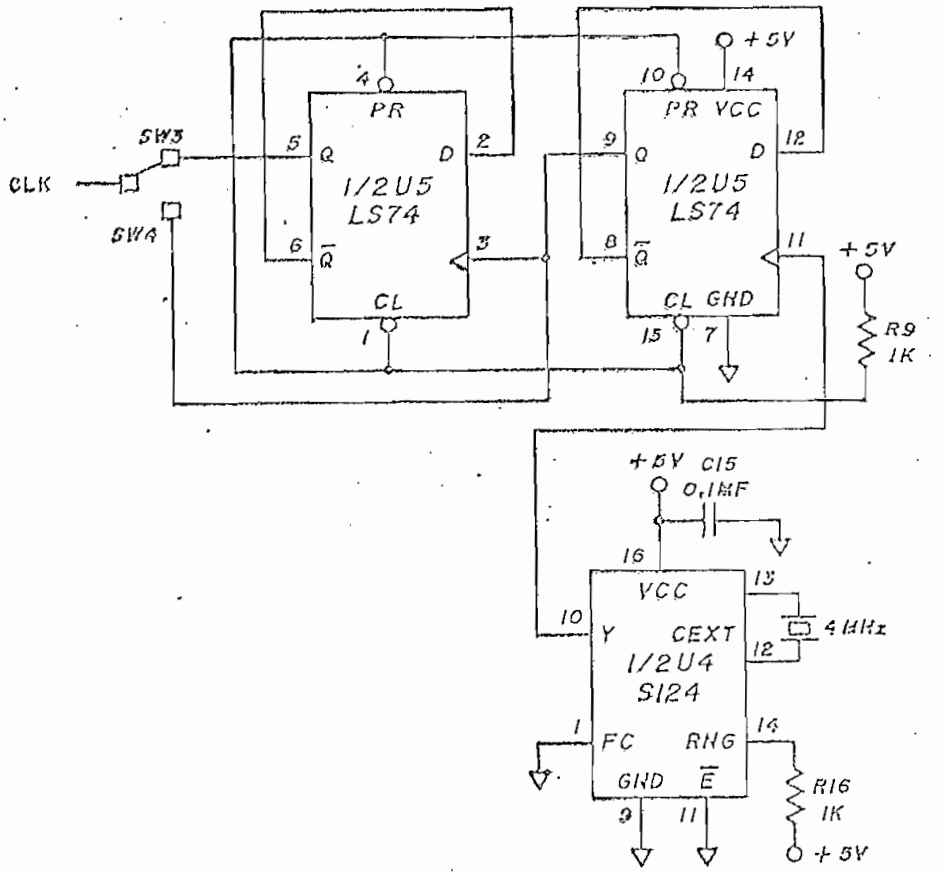
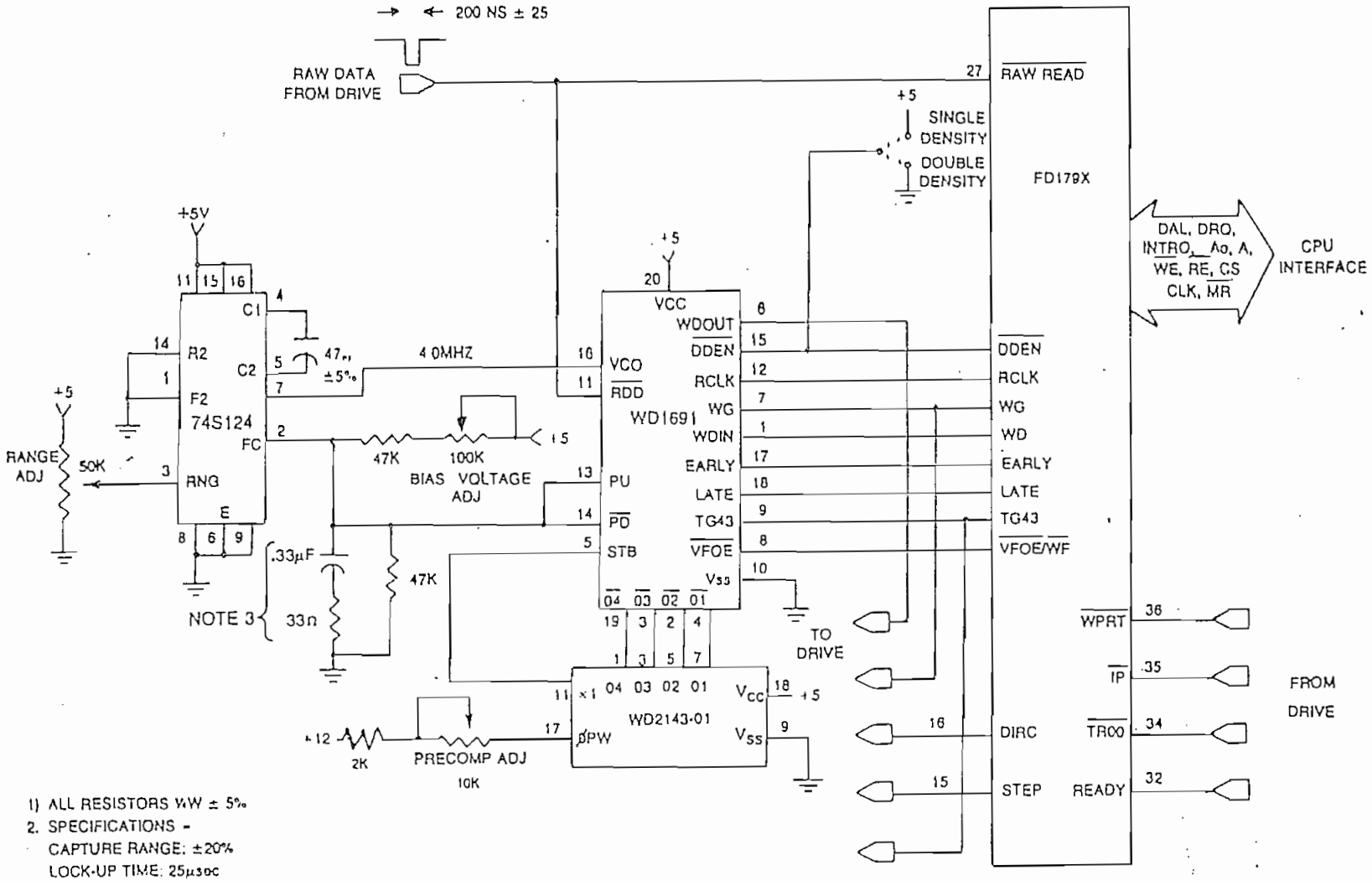


FIGURA 17

CIRCUITO GENERADOR DE LA SEÑAL DE RELOJ

Las ventajas de utilizar el circuito que se muestra en la figura 18 para la separación de datos son múltiples:

- Se utilizan sólamente tres integrados.
- Precompensación en la escritura encluida.
- Confiabilidad en la separación de datos por utilizar un



1) ALL RESISTORS ¼W ± 5%
2. SPECIFICATIONS -
CAPTURE RANGE: ±20%
LOCK-UP TIME: 25μsec
(ALL ONE'S PATTERN, MFM)

3) FOR 5 1/4" 8
.68μf .33μf
68Ω 33Ω

FIGURA 18

circuito PLL.

- Las exigencias en las tolerancias de las resistencias y condensadores externos no son mayores.
- La cantidad de precompensación deseada puede ser ajustada en forma sencilla.
- Trabaja en densidad simple o doble, con discos de 5¼" y de 8".
- Conexión directa con el integrado FD1797-02.

El integrado WD1691 consta de dos secciones:

1) Circuito de separación de datos.
2) Circuito de precompensación.

Este integrado está diseñado de forma que la precompensación en la escritura se realice sólamente en las pistas mayores a la número 43 siempre que se utilice densidad doble, es decir, cuando TG43 = 1L y $\overline{DDEN}$ = 0L. El funcionamiento de la sección de precompensación es muy similar al del circuito de la figura 6 que fuera explicado al estudiar los formatos de escritura.

La sección de separación de datos es habilitada solamente cuando el integrado FD1797-02 espera recibir datos confiables de la lectora/escritora es decir, cuando $\overline{VFOE/WF}$ = 0L y WRITE GATE = 0L. Si las condiciones anteriores se cumplen, las líneas PU y $\overline{PD}$ incrementarán o disminuirán la frecuencia de oscilación del integrado 74S124, de forma que los pulsos prove-

nientes de la lectora/escritora ocurran en el centro de cada ciclo de la señal de salida RCLK. Debe notarse que el circuito no produce una separación completa de los bits de datos y de reloj, sino que ésta se realiza en el integrado FD1797-02 a partir de la señal RCLK y aquella leída del disco.

Cabe anotar que el circuito original que consta en las notas de aplicación adolece de un error pues se considera que para trabajar con discos de 5¼" no hace falta cambiar la frecuencia del VCO, cuando uno de los requerimientos del integrado WD1691 es precisamente el de que la frecuencia central de oscilación del VCO sea de 2 MHz. cuando se trabaja con discos de 5¼". En el circuito de la figura 18 se ha corregido el error por medio de un condensador de 47 pF. que debe ser conectado -cuando se trabaje con discos de 5¼"- en paralelo con el condensador original externo del integrado 74S124. De esta forma se reduce la frecuencia de oscilación del VCO a la mitad, es decir, a 2 MHz.

La calibración del circuito de la figura 18 debe realizarse de la siguiente manera:

1) Desconectar el integrado WD1691.
2) Configurar el circuito para utilizarlo con discos de 8 ó 5¼ pulgadas, según se requiera.
3) Obtener un voltaje de 1.4 voltios en la línea de control de frecuencia (FC) variando el potenciómetro correspondiente.

4) Ajustar el potenciómetro de rango (RNG) hasta obtener una frecuencia de oscilación de 2 MHz. para discos de 5¼" o de 4 MHz. para discos de 8".

5) Conectar nuevamente el integrado WD1691.

Otro de los requerimientos del circuito de la figura 18 es el de que los pulsos provenientes de la lectora/escritora tengan una duración de 200 nSeg. ± 25 por lo que para asegurar compatibilidad con cualquier lectora/escritora se utilizará un monoestable que produzca pulsos de esta duración a partir de los pulsos recibidos.

Debido a la relativamente corta duración de los pulsos requeridos, se utilizará uno de los monoestables del integrado 74123. En este tipo de monoestables la duración del pulso puede ser calculada con la siguiente fórmula:

$$t_W = 0{,}28 \times R_t \times C_{ext} \times (1 + 0{,}7/R_t)$$

Utilizando la fórmula anterior para valores típicos de R_t se determinó que C_{ext} deberá ser de un valor menor a 1000 pF, en estos casos el fabricante recomienda escoger los valores de R_t y C_{ext} gráficamente (21). De acuerdo a las gráficas proporcionadas se llegó a determinar que el pulso de 200 nSeg. puede ser obtenido con un condensador de 39 pF y una resistencia de 10 KΩ Una vez obtenidos estos valores se decidió utilizar como resistencia externa a un potenciómetro de 10 KΩ en serie con una resistencia de 5,1 KΩ, de esta forma el ancho del pulso po

drá ser regulado con precisión, variando el valor de la resistencia externa dentro de amplios límites pero siempre dentro de los valores recomendados por el fabricante.

El segundo de los monoestables del integrado 74123 podrá ser utilizado para producir un pulso en la línea $\overline{\text{MASTER RESET}}$ del integrado FD1797-02 cada vez que la señal $\overline{\text{RESET}}$ del equipo MEK6800D2 se ponga en cero lógico. De esta forma el controlador de discos será inicializado conjuntamente con el equipo de microprocesador.

El fabricante especifica que la duración mínima del pulso en la línea $\overline{\text{MASTER RESET}}$ debe ser de 50 µSeg., pero teniendo en cuenta que al finalizar el pulso, el integrado FD1797-02 carga y ejecuta automáticamente un comando que hace regresar a la cabeza de grabación a la pista ØØ, se ha creido conveniente alargar la duración del pulso a unos 275 µSeg. de forma que la lectora/escritora pueda estar lista para aceptar el comando. La salida del monoestable podrá ser utilizada también para inicializar los circuitos de interface con el equipo de microprocesador.

Aplicando la fórmula -ya mencionada- para este tipo de monoestables, se llegó a determinar que un pulso de 275 µSeg. puede ser obtenido con los siguientes valores de resistencia y condensador externos:

$R_t = 12\ K\Omega$

$C_{ext} = 0{,}082\ \mu F$

3.2 DISEÑO DE LA INTERFACE.

En los procesos de interface entre el controlador y la lectora/escritora se seguirán las recomendaciones del fabricante de esta última, es decir, para cada salida del controlador se deberán utilizar compuertas de colector abierto, capaces de soportar una corriente de salida de 35 mA. o más en cero lógico, además, cada entrada deberá ser terminada por una resistencia de 150 Ω conectada a Vcc. y a una compuerta del tipo "Schmitt Trigger". (22).

Por las razones mencionadas se utilizarán compuertas 7438 en las salidas que necesiten inversión y 7417 en las que no. Para las entradas se utilizarán compuertas del tipo 74LS14 junto con resistencias de 150 Ω conectadas a Vcc.

En cuanto a la interface entre el controlador y el equipo MEK6800D2 se utilizará una PIA como principal elemento de la interface pues esta dotará de una mayor agilidad a ciertas secciones de los programas de soporte, además de que sus múltiples líneas periféricas podrán ser utilizadas para controlar los procesos de interface, transmitir datos desde o hacia el microprocesador y establecer las condiciones de funcionamiento del con

trolador.

El circuito de interface entre el controlador y la lectora/escritora se muestra en la figura 19. Podrá notarse -cuando se hable de los programas de soporte- que utilizando este circuito se consigue la mayor agilidad posible en los programas de transmisión y recepción de datos, permitiendo de esta manera que el microprocesador pueda trabajar a una frecuencia menor a la máxima para él permitida sin que se pierdan datos escritos o leídos del disco. Esta característica es de mucha importancia pues el equipo MEK6800D2 trabaja con una frecuencia de 614,4 KHz.

Debe notarse que se ha preferido utilizar un tipo de interface en la que la transmisión de datos se realiza "por programa" dejando de lado la utilización de DMA, pues su uso incrementaría el número de integrados utilizados en la interface en cambio, el circuito de la figura 19 hace uso de componentes incluidos en el equipo MEK6800D2, como son la PIA y los circuitos utilizados para el direccionamiento de la misma, además, la interface podrá ser fácilmente conectada al equipo ya que este dispone de salidas para el efecto.

A continuación se explica el funcionamiento del circuito:

La transmisión de datos entre el controlador y el equi

po MEK6800D2 se realiza por medio de las líneas PBØ - PB7 que deberán ser programadas como salidas o entradas, dependiendo de que se envíe un dato al controlador o se reciba un dato del mismo. Las líneas PAØ y PA1 tienen que ser programadas como

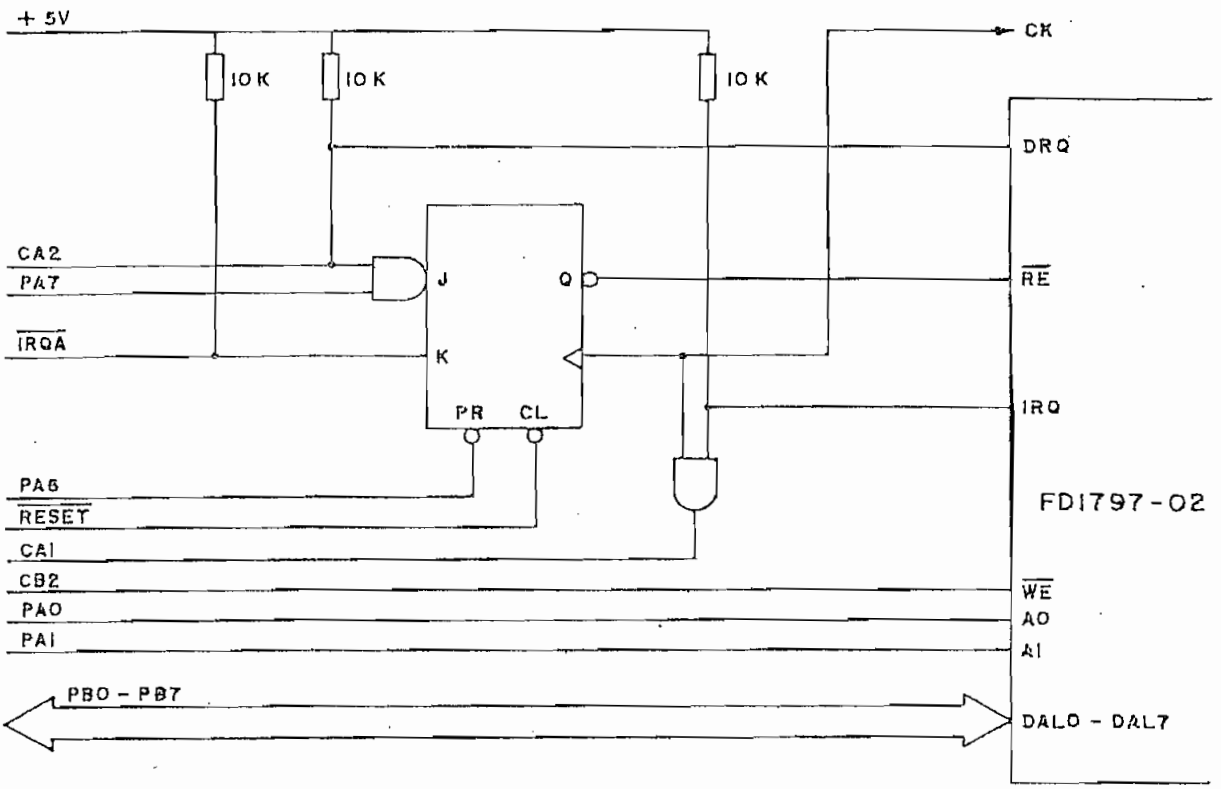


FIGURA 19

CIRCUITO DE INTERFACE

salidas y deberán apuntar al registro del integrado FD1797-02 que se requiera leer o escribir.

El control de escritura de datos en los registros del integrado FD1797-02 se lo realiza directamente por medio de la

línea de control CB2, mientras que para el control de lectura de datos desde el controlador hacia la PIA se utiliza un flip-flop JK en conjunto con algunas señales provenientes de la PIA y el controlador.

Para el control de lectura de datos hay que diferenciar dos casos: el primero en que se lee un registro del controlador en el momento en que se necesite de ello y el segundo -utilizando en la lectura del disco- en que el controlador ha recibido un byte de datos y por lo tanto requiere que el microprocesador lo lea antes de que el siguiente byte sea leído del disco. Esta última condición es indicada por el controlador por medio de un uno lógico en su salida DRQ.

De acuerdo a lo indicado, la línea $\overline{RE}$ deberá controlarse en forma tanto asincrónica como sincrónica con DRQ, lo que se ha conseguido con un flip-flop cuya salida puede ser puesta asincrónicamente en cero lógico bajo el control de la línea PA6 proveniente de la PIA, así pues, si se requieren leer datos en este modo de operación se deberá escribir un cero lógico en PA6, leer el dato y poner nuevamente PA6 en uno lógico. Si un cero lógico fue previamente escrito en PA7 -como requiere este modo de operación- las entradas J y K estarán en cero y uno lógico respectivamente por lo que $\overline{RE}$ regresará a uno lógico con la siguiente transición positiva de CK.

En la segunda forma de operación, un uno lógico debe

ser escrito en PA6 y PA7 y los bits 3-5 del registro de control A deben ser programados con 11Ø respectivamente, de esta forma cuando DRQ se ponga en uno lógico la salida del flip-flop se pondrá en cero lógico con la siguiente transición positiva de CK. Puesto que se han unido las líneas DRQ y CA2, al producirse la transición positiva de DRQ la salida $\overline{IRQA}$ se pondrá en cero lógico, manteniendo por tanto la salida del flip-flop en cero lógico. El microprocesador podrá reconocer que existe un dato listo, pues el bit 6 del registro de control A (CRA) se pondrá en uno lógico, deberá leerlo en el registro de datos B (DRB) e indicar que se lo ha leído por medio de una operación de lectura del registro de datos A (DRA) con lo que $\overline{IRQA}$ y CRA bit 6 regresarán a uno y cero lógico respectivamente.

Para el control de escritura, los bits 3-5 del registro de control B (CRB) deberán programarse con 1Ø1 respectivamente. De esta forma, cada vez que se realice un proceso de escritura en el registro de datos B (DRB), la PIA mandará un pulso por su salida CB2.

De manera similar a la lectura de datos, el controlador pondrá un uno lógico en su salida DRQ cada vez que esté listo para recibir un nuevo dato del microprocesador y escribirlo en el disco, por lo que será necesario deshabilitar el flip-flop de lectura escribiendo un cero y un uno lógico en PA7 y PA6 respectivamente. Un uno lógico en el bit 6 de CRA indicará al

microprocesador que el controlador está listo para recibir un nuevo dato, en ese momento se deberá escribir el dato en DRB y borrar la "bandera" leyendo el registro de datos A.

En la figura 19 se indica también la forma en que se han conectado las líneas IRQ del controlador y CA1 de la PIA. Recordando que la salida IRQ se pondrá en uno lógico al término de un comando o cuando se produce un error mientras se realiza el mismo, los bits 1 y 2 de CRA deberán ser programados como cero y uno lógico respectivamente, de forma que cuando IRQ se ponga en uno lógico, se informe de esta condición al microprocesador por medio de una bandera que en este caso es el bit 7 de CRA.

Cuando el microprocesador detecte un uno lógico en CRA bit 7, deberá borrar esta bandera leyendo DRA para luego leer el registro de "Status" del controlador y así determinar si se produjo un error o es que se ha terminado el comando.

3.3 EL CIRCUITO COMPLETO.

A continuación se presenta el diagrama de circuitos completo, en él se puede notar la presencia de algunos circuitos adicionales como es el caso del integrado 74LS138 que decodifica las señales de PA4 y PA5 para seleccionar la lectora/escritora deseada, de esta forma se podrá manejar hasta cuatro lectoras/escritoras con el mismo controlador. Puede observarse

también que la selección de una lectora/escritora puede ser mantenida en todo momento (SW2 cerrado) o sólamente cuando la línea $\overline{HLD}$ se ponga en cero lógico (SW1 cerrado), esta última forma de operación permite utilizar lectoras/escritoras -como la SA400- que bajan la cabeza de grabación al ser seleccionadas.

Por medio de PA2 se permite utilizar lectoras/escritoras que mantienen bajada la cabeza en todo momento y por lo tanto el controlador no necesita esperar a que se realice ese proceso. De utilizarse este tipo de lectoras/escritoras, un cero lógico deberá ser escrito en PA2, de forma que se inhabilite el disparo del monoestable que produce la demora para que se realice el proceso de bajado de la cabeza.

Algunas lectoras/escritoras no tienen la salida $\overline{READY}$, en esos casos se puede cerrar SW8 de forma que se considere que la lectora/escritora está lista cuando la señal $\overline{MON}$ del controlador se pone en cero lógico.

Lectoras/escritoras para discos de 5¼" o de 8" en densidad simple o doble pueden ser utilizadas como se indica en la tabla V.

El circuito ha sido armado, usando la técnica de "Wire Wrap", sobre una placa VECTOR 4609 diseñada para albergar cir-

TABLA V

TAMAÑO	DENSIDAD	SW3	SW4	SW5	SW6	SW7	C9
5¼"	Simple	C	D	D	C	D	C
	Doble	C	D	C	C	D	C
8"	Simple	D	C	D	D	C	D
	Doble	D	C	C	D	C	D

SIMBOLOS UTILIZADOS:

C = Conectado

D = Desconectado

cuitos de interface de microcomputadoras. Sus dos salidas para conectores planos, han sido utilizadas para unir al circuito con el equipo MEK6800D2 -por medio de un cable de 50 conductores- y con la lectora/escritora -por medio de un cable de 34 conductores-.

La placa tiene líneas impresas para V_{cc} y tierra que han sido utilizadas en conjunto con los enlaces VECTOR T112-1 para llevar los voltajes de polarización hasta cada integrado. Terminales de tipo VECTOR R32 han sido utilizados para el cristal de 4 MHz y C9 de forma que puedan ser facilmente removidos de la placa. Resistencias, diodos y condensadores, han sido soldados a postes del tipo VECTOR T44-1 de forma que puedan ser conectados mediante la técnica de "Wire Wrap".

Condensadores de desacoplamiento han sido conectados siguiendo las recomendaciones del fabricante de la placa (23). Estos suman una capacidad total de 30 μF en los terminales para la fuente de 5 V y de 0,1 μF en la de 12 V.

La poscición de los integrados ha sido estudiada para minimizar las distancias entre aquellos integrados con líneas comunes, tratando además de que los integrados con alto consumo de corriente o muy ruidosos, queden cerca de las entradas de voltaje. De esta forma se han eliminado, en lo posible, las capacidades parásitas y los ruidos, quedando la posición de los elementos en la placa, como se muestra en la figura 20.

Los voltajes de 5 y 12 voltios serán tomados de las fuentes de voltaje del equipo MEK6800D2, por lo que éstas deberán soportar un incremento mínimo de corriente de 1A para la fuente de 5 voltios y de 50 mA para la de 12 voltios. Operando normalmente, el circuito deberá tomar alrededor de 0,5 A y 15 mA de dichas fuentes.

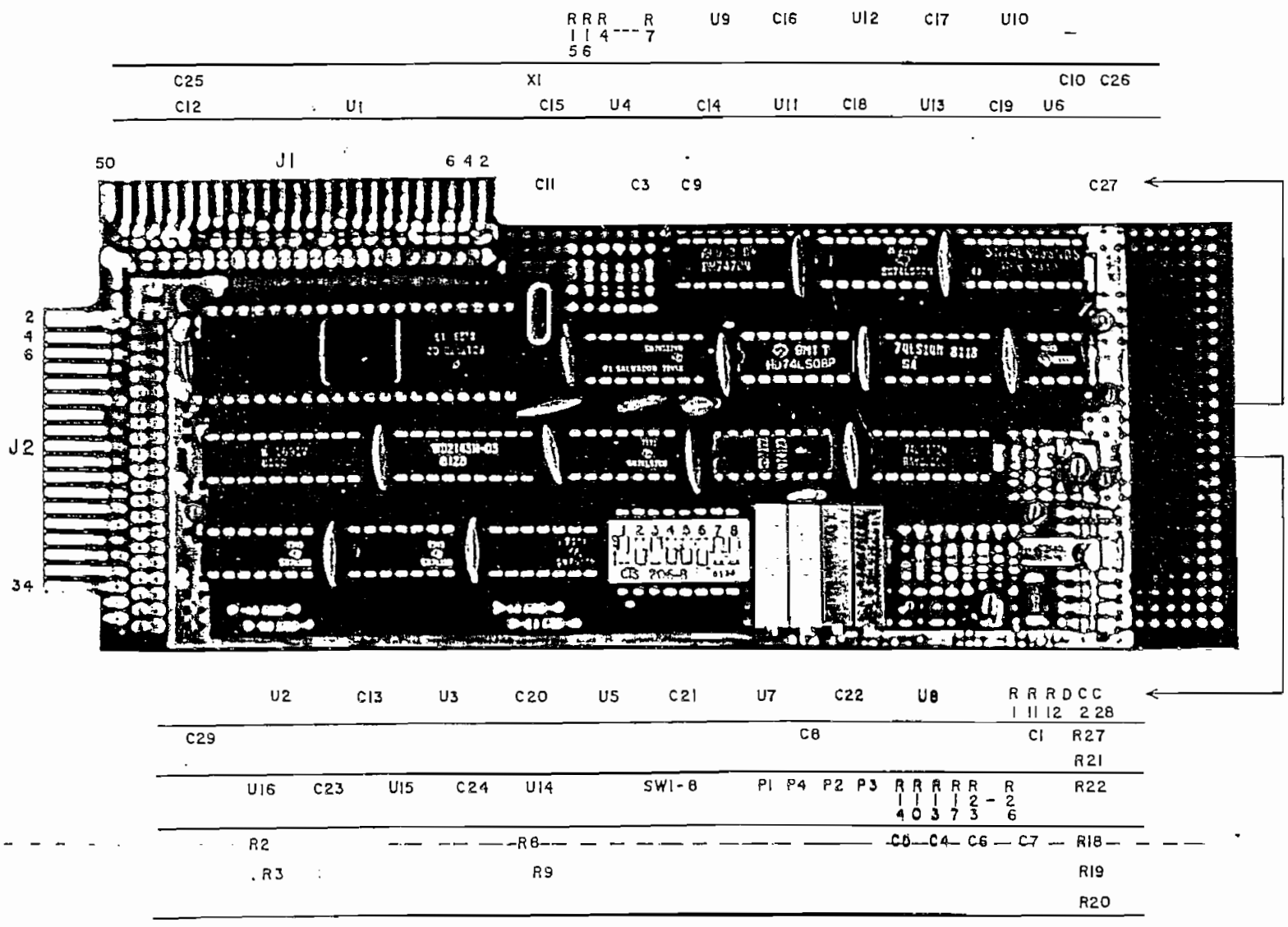


FIGURA 20

PLACA DE CIRCUITOS

C A P I T U L O C U A R T O

" PROGRAMAS DE SOPORTE "

4. LOS PROGRAMAS DE SOPORTE

Los programas de soporte complementan el funcionamiento del circuito, convirtiendo las instrucciones sencillas, utilizadas por el usuario, en una serie de procesos involucrados en la lectura o escritura de información en el disco. De esta forma los procesos de control, transmisión y almacenamiento de datos, resultan transparentes al usuario.

Para el desarrollo de los programas de soporte, se ha creído conveniente separarlos en tres categorías:

1.- Programas de control
2.- Programas operativos
3.- Programas de consola

El primer tipo de programas deben trasladar las instrucciones o comandos, propios del controlador, a subrutinas que además de ejecutar esos comandos, se encarguen de los procesos de interface. Por esta razón, cada una de estas subrutinas podrá ser utilizada como si se tratara de una instrucción que se envía al controlador.

Los programas operativos deberán hacer uso de las subrutinas de control para la grabación y lectura de los programas o datos del usuario. Deberán encargarse de la inicialización (creación del formato) de los discos que van a ser utilizados por primera vez; leer o escribir el número de sectores requeri

dos para almacenar la información del usuario y llevar un registro, tanto de las pistas y sectores ocupados por la información en el disco, como de las localidades de memoria a donde deberá ser leída dicha información.

Los programas de consola tendrán a su cargo el ingreso y salida de datos alfanuméricos que permitan establecer un diálogo entre el usuario y los programas operativos. Deberán, por tanto, mantener el control sobre los elementos de entrada/salida y estar en capacidad de "entender" las instrucciones impartidas por el usuario.

Al final de este capítulo se han incluído los programas terminados, tanto en código nemotécnico como en lenguaje de máquina, que serán utilizados -en conjunto con el controlador- para la lectura y escritura de discos floppy. Para el desarrollo de estos programas se ha partido siempre del diagrama de flujo correspondiente, por lo que, se ha creído conveniente presentar, en las siguientes páginas, a cada uno de los programas en diagrama de flujo junto con una pequeña explicación de su utilidad y de las condiciones de entrada/salida de los mismos.

El conjunto completo de programas han sido grabados en una memoria EPROM que será colocada en la posición U1Ø del equipo MEK6800D2, de forma que estos no ocupen el reducido espacio de memoria RAM de que se dispone actualmente, sin embargo, se han reservado los sectores 3-1Ø de la pista Ø para que en el

futuro se puedan almacenar allí la mayoría de los programas de soporte, dejando en la EPROM solamente aquellos indispensables para que el microprocesador pueda leer los programas restantes a memoria RAM. Con ello, se podrán almacenar en la EPROM todos aquellos programas que necesiten realmente estar siempre presentes en memoria.

4.1. PROGRAMAS DE CONTROL

A continuación se da una ligera explicación de los programas de control. El usuario podrá utilizar eventualmente los programas de control "principales" que han sido marcados con un asterisco (*). Estos programas se caracterizan porque al término de los mismos, los registros de la PIA permanecen sin cambios o son inicializados, de esta forma se los puede utilizar -uno tras otro- sin preocuparse por la condición de dichos registros.

4.1.1. INIC.- *

Esta subrutina inicializa los registros de la PIA y los registros asignados a las lectoras/escritoras. Por medio de ella, las líneas periféricas del lado A de la PIA son programadas como salidas y las del lado B como entradas; CRA y CRB son programados con $1E y $2C respectivamente (de acuerdo a lo explicado en el diseño de la interface) y, $44 es escrito en DRA con lo que se permite el disparo del monoestable de bajado de

la cabeza; se selecciona la lectora/escritora número cero; se apunta a "command register" del controlador y se inhabilita el flip-flop de lectura.

Los cinco registros asignados a las lectoras/escritoras son inicializados -con la subrutina INIC- a $ØØ. El primero de estos registros especifica el número de la lectora/escritora que se está utilizando y los cuatro restantes almacenan el número de pista sobre la cual se encuentra la cabeza de grabación de cada una de las lectoras/escritoras.

CONDICIONES DE ENTRADA**	CONDICIONES DE SALIDA**
Ninguna	A = $44 B = $2C X = DRA

**NOTA: Las condiciones de entrada/salida serán dadas solamente para los registros: A, B y X del microprocesador, incluyendo, cuando sea de importancia, el estado de uno o más bits del registro de código de condición (CCR) como se indica en los siguientes ejemplos:

A = 44	- acumulador A contiene el número 44 decimal (2C hexadecimal)-
B = $44	- acumulador B contiene el número 44 Hex.
A = (REGISTRO)	- acumulador A cargado con el <u>contenido</u> de un REGISTRO-
X = REGISTRO	- X contiene la <u>dirección</u> de un REGISTRO-

X = X + 256 - X contiene la suma de 256 Dec. más el valor de X de entrada.

3,X = $Ø3 - en la dirección 3 + (X) está almacenado el número 3 Hex.

CCR: Z = Ø - bit Z de CCR está en cero lógico-

C = 1 - bit C (Carry) de CCR está en uno lógico-

Debe entenderse además que las condiciones de entrada/salida se han incluído para ayudar al usuario que desea crear programas que utilicen las presentes subrutinas y por lo tanto deberá conocer la función de cada uno de los registros que se han creado en memoria y de los registros internos del controlador, luego de haber estudiado los programas terminados que se han incluído al final de este capítulo.

4.1.2. DISC.- *

Esta subrutina selecciona una lectora/escritora por medio de PA4 y PA5. El número de pista en la que se encuentra la lectora/escritora que se estuvo utilizando anteriormente se almacena en su registro correspondiente (de acuerdo a lo indicado en la subrutina INIC), a la vez que se recobra el número de pista en que se encuentra la lectora/escritora que se está seleccionando y se lo almacena en el registro de pista del controlador (Track Register). Por último se almacena el número de la lectora/escritora seleccionada en el registro reservado para el objeto.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
3,X = Número de lectora/escritora a seleccionarse	NORMALES: A = Pista en que se encuentra la L/E seleccionada B = Ø X = REGPIS + número de L/E seleccionada CCR: Z = 1; C = Ø DE ERROR: A = (3,X) B = $ØF X = X CCR: Z = Ø; C = 1

4.1.3. REGR.- *

La subrutina REGR posiciona la cabeza de grabación, de la lectora/escritora seleccionada, sobre la pista ØØ, utilizando para ello un comando "RESTORE" del controlador. REGR selecciona también una lectora/escritora por medio de la subrutina DISC, pues normalmente se necesitará que la cabeza de grabación de una lectora/escritora que se utiliza por primera vez, regrese a la pista ØØ. En caso de que no se requiera seleccionar una lectora/escritora, se puede utilizar la subrutina REGR desde la posición REGR1.

Al final de la subrutina se lee el registro de estado

del controlador ("Status Register") para averiguar si se produjo algún error en el proceso. El código de error es leído en el acumulador B del microprocesador, indicando que se ha producido un error por medio de un cero lógico en el bit Z de CCR. Esta forma de control de errores será utilizada -siempre que sea posible- en las subrutinas restantes, debiendo notar que el código de error será el mismo que el utilizado por el integrado FD1797-02 que puede encontrarse en los apéndices de la presente tesis.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
REGR: 3,X = Número de la L/E a seleccionarse	NORMALES: A = (DRA)
	B = (STATUS REGISTER)
REGR1: Ninguna.	X = X
	CCR: Z = 1
	DE ERROR: A = (DRA)
	B = (STATUS REGISTER)
	X = X
	CCR: Z = Ø

4.1.4. BUSQ.- *

Esta subrutina posicona la cabeza de grabación sobre una pista especificada. BUSQ será normalmente utilizada antes de leer o escribir un sector en el disco, por lo que se ha hecho que almacene también el número de sector en el registro correspondiente del controlador ("Sector Register").

Al estudiar el diagrama de flujo de la subrutina BUSQ, se encontrará que no se ha utilizado el comando de búsqueda ("SEEK") del controlador, sinó que se llega a la pista especificada "por programa". Esto se debe a que el comando SEEK mueve la cabeza de grabación "por pasos" cuya duración máxima es de 30 mSeg. mientras que la lectora/escritora SA400 -que se está utilizando- necesita como mínimo de 40 mSeg. (24).

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
	NORMALES:	A = Pista especificada
A = Pista especificada		B = Pista especificada
B = Sector especificado		X = X
	CCR:	Z = 1
	DE ERROR:	A = Pista especificada
		B = (STATUS REG.)
		X = X
	CCR:	Z = Ø

4.1.5. LEA.- *

LEA se utiliza para leer un sector del disco. Haciendo uso de la subrutina BUSQ, posiciona la cabeza de grabación sobre la pista especificada y lee el sector especificado a las localidades de memoria comprendidas entre aquella apuntada por el registro X del microprocesador y las 255 localidades siguientes.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	NORMALES: A = (DRA)
	B = (STATUS REG.)
A = Pista especificada	X = X + 256
B = Sector especificado	CCR: Z = 1
X = Comienzo del espacio	DE ERROR: A = (DRA)
de RAM a donde se	B = (STATUS REG.)
lee el sector.	X = Indeterminado
	CCR: Z = Ø

4.1.6. ESCRIB.- *

Esta subrutina es utilizada para escribir un sector en el disco. Al igual que la subrutina LEA, posiciona la cabeza de grabación sobre la pista especificada y luego escribe el sector especificado con el contenido de las 256 localidades consecutivas de memoria que comienzan desde aquella apuntada por el registro X del microprocesador.

A pesar de la optimización de que ha sido objeto la subrutina ESCRIB y de las ventajas que representa la utilización de la PIA como elemento de interface, ESCRIB es la subrutina que limita la frecuencia mínima de reloj que puede ser utilizada en el microprocesador. Desde luego, esto depende de la lectora/escritora utilizada y por lo tanto del tiempo entre bits de datos (como fuera indicando en la sección 1.3.2.).

Tomando en cuenta el número de ciclos de reloj que se

utilizan en la subrutina ESCRIB -en lenguaje de máquina- para almacenar un byte de datos en el disco y las características del controlador, se ha llegado a determinar que la frecuencia mínima utilizable es:

$$f_m = 25/t_{service\ (WRITE)}$$

De acuerdo a esta fórmula, la frecuencia mínima es el caso de discos de 5¼", en densidad simple, ($t_{service\ (WRITE)}$ = 47,Ø µSeg.) es de 532 KHz. (25). Puesto que el equipo MEK6800D2 trabaja a 614,4 KHz., no se necesitará de ningún cambio en la frecuencia de reloj mientras se utilice la lectora/escritora SA400.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
	NORMALES:	A = (DRA)
A = Pista especificada		B = (STATUS REG.)
B = Sector especificado		X = X + 256
X = Comienzo del espacio	CCR:	Z = 1
de RAM de donde se	DE ERROR:	A = (DRA)
lee el sector a es-		B = (STATUS REG.)
cribirse.		X = Indeterminado
	CCR:	Z = Ø

4.1.7. VERIF.- *

Esta subrutina se utiliza para verificar el último sector escrito, por lo que supone que la cabeza de grabación se

encuentra sobre la pista que se quiere verificar y que el controlador contiene el número de sector que es objeto de esta verificación.

VERIF busca sólamente errores de CRC (lo que cubre la mayoría de los errores posibles) efectuando para ello un proceso de lectura del sector pero sin almacenar los bytes leídos en memoria.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
Ninguna	NORMALES:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = 1
	DE ERROR:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = Ø

4.1.8. CHKRDY.- *

CKKRDY debe utilizarse para averiguar la condición de la lectora/escritora, es decir, si se encuentra lista para realizar algún proceso.

La subrutina considera que una lectora/escritora está lista si la línea $\overline{\text{READY}}$ está en ØL, si este no es el caso, mueve la cabeza de grabación un paso adelante y un paso atrás,

con lo que se consigue que la línea $\overline{HLD}$ del controlador se ponga en ØL y por lo tanto se tenga la seguridad de que una lectora/escritora está seleccionada (ver sección 3.3.), su cabeza bajada y su motor encendido ($\overline{MON}$ = ØL); luego, la subrutina busca nuevamente un ØL en la línea $\overline{READY}$, indicando que hay un error si esto no se produce.

El hecho de mover la cabeza de grabación un paso adelante y otro atrás ayuda también a un mejor posicionamiento de la cabeza en una pista del disco, por esta razón, la subrutina CHKRDY puede ser utilizada también luego de haberse producido un error en la lectura o escritura del disco.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
Ninguna	NORMALES:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = 1
	DE ERROR:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = Ø

4.1.9. ENTR.-

Por medio de esta subrutina se programan las líneas periféricas del lado B de la PIA como Entradas.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = $2C
	X = X

4.1.10. SALID.-

SALID realiza el proceso inverso a ENTR, programando el lado B de la PIA como salidas.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = $2C
	X = X

4.1.11. DTARG.-

Esta subrutina escribe un 1L en PAØ y PA1, seleccionando por tanto a "DATA REGISTER" del controlador para cualquier proceso de lectura o escritura de este registro.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = (DRA)
	X = X

4.1.12. CMRG.-

Por medio de CMRG se escribe un ØL en PAØ y PA1, selec

cionando por tanto a "STATUS REGISTER" del controlador para un proceso de lectura y a "COMMAND REGISTER" para la escritura (como se indica en los apéndices de la presente tesis.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = (DRA)
	X = X

4.1.13. LEARG.-

Esta subrutina se utiliza para leer el registro del controlador apuntado por PAØ y PA1.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = (DRA)
Ninguna	B = (REGISTRO APUNTADO POR PAØ, PA1)
	X = X

4.1.14. MCMD.-

MCMD escribe un comando o un dato en el registro apuntado por PAØ y PA1.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
A = Comando o dato.	B = $2C
	X = X

4.1.15. MCSP.-

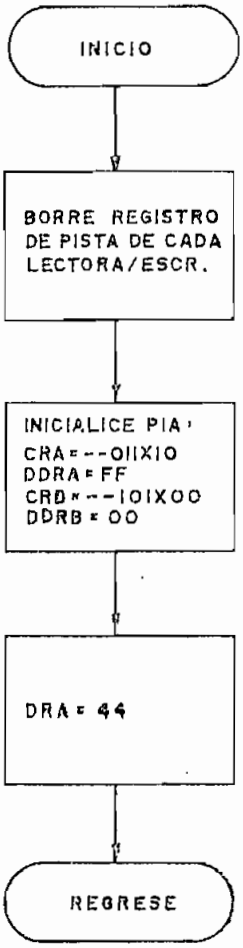
Esta subrutina manda un comando al controlador y espera a que se lo termine, luego lee el registro de estado del controlador ("STATUS REGISTER") para averiguar si se produjo algún error mientras se ejecutaba el comando.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = (DRA)
A = Comando	B = (STATUS REGISTER)
	X = X

4.1b. DIAGRAMAS DE FLUJO DE LOS PROGRAMAS DE CONTROL.

A continuación se presentarán los diagramss de flujo correspondientes a los programas de control.

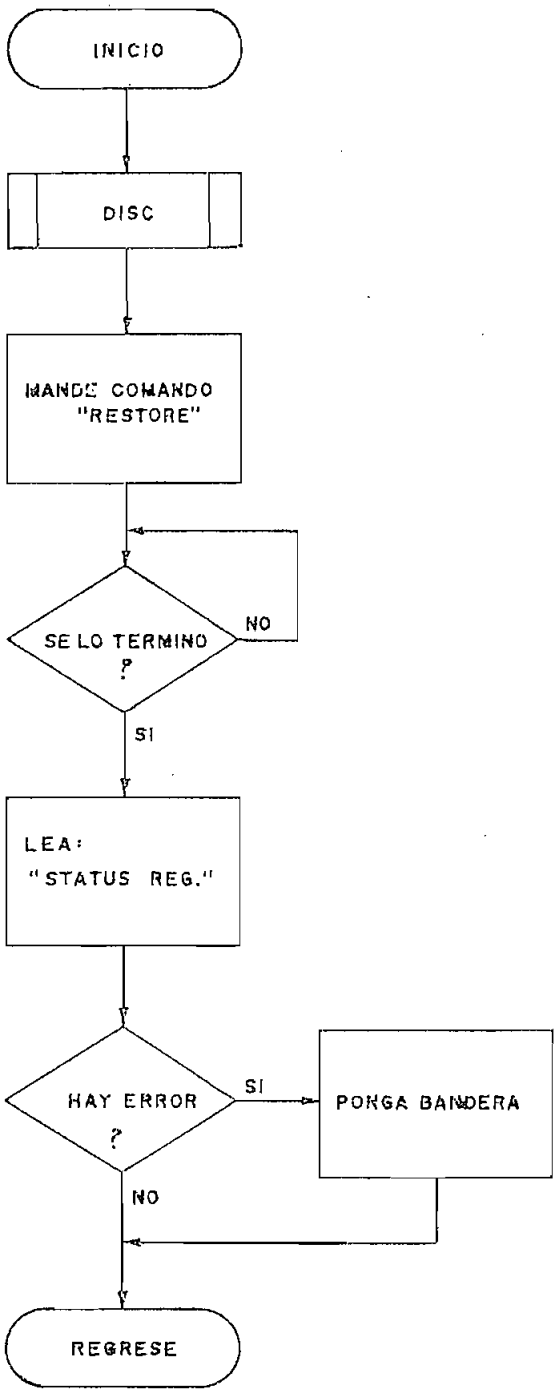
4.1b.1. INIC.


INICIO
BORRE REGISTRO DE PISTA DE CADA LECTORA/ESCR.
INICIALICE PIA:
CRA=--011X10
DDRA=FF
CRB=--101X00
DDRB=00
DRA=44
REGRESE

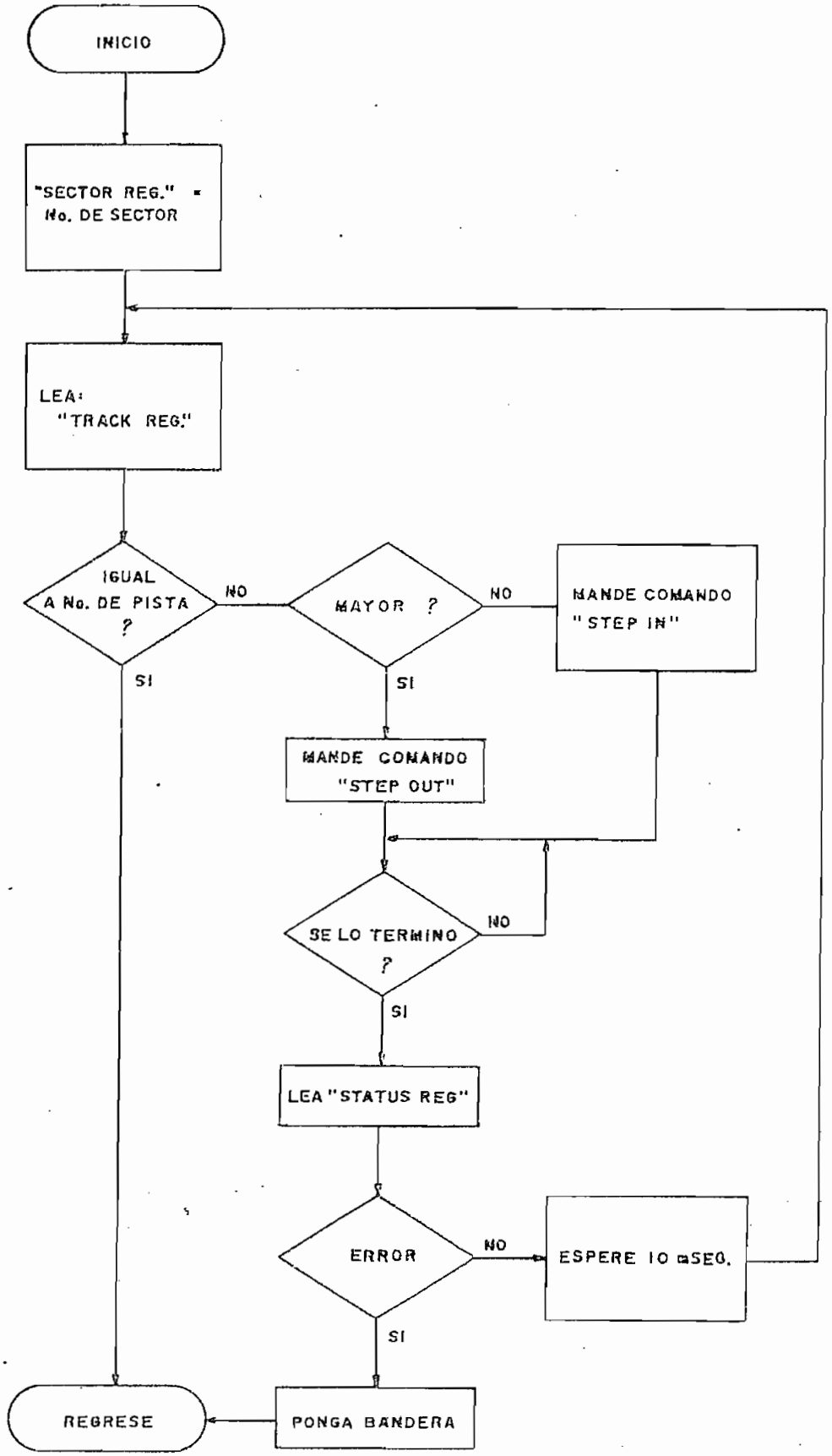
4.1b.2. DISC.


INICIO
ALMACENE No. DE PISTA DE LEC/ES. QUE SE UTILIZABA
SELECCIONE LA NUEVA LEC/ESC.
ALMACENE SU No. DE PISTA EN "TRACK REG."
REGRESE

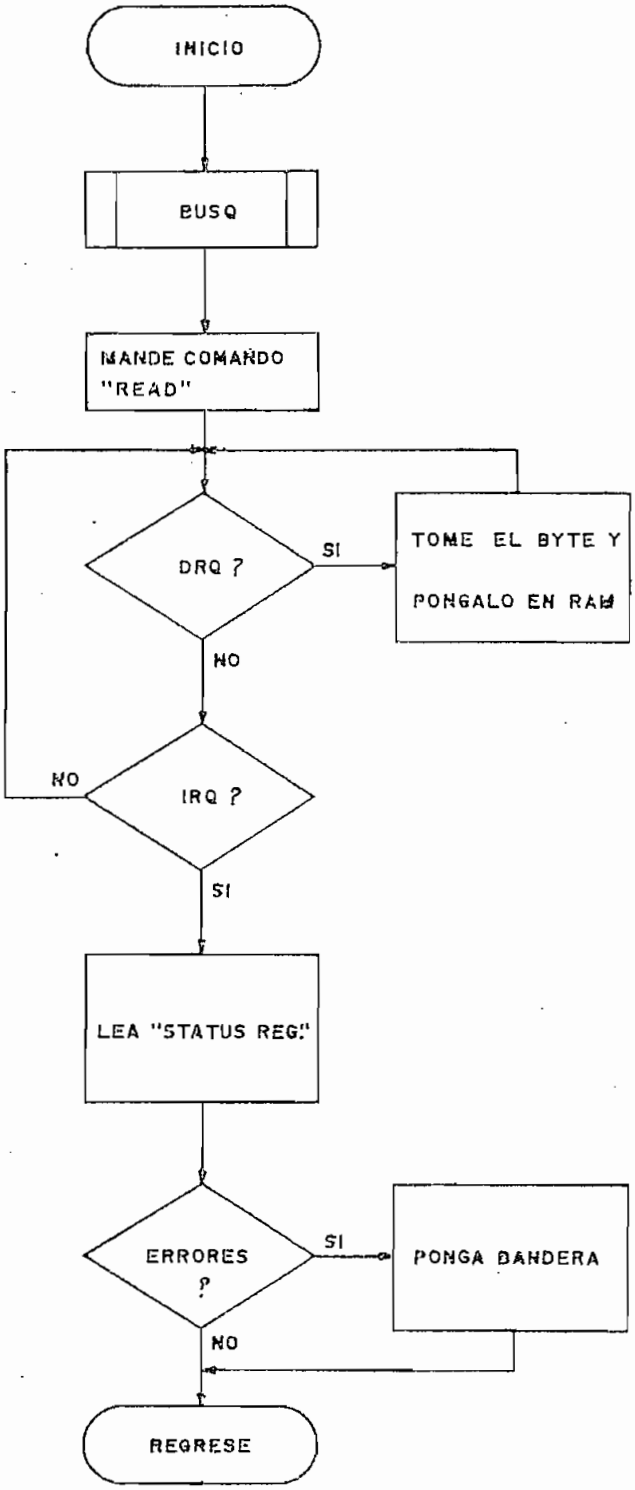
4.1b.3. REGR.


INICIO
DISC
MANDE COMANDO "RESTORE"
SE LO TERMINO ?
NO
SI
LEA: "STATUS REG."
HAY ERROR ?
SI
PONGA BANDERA
NO
REGRESE

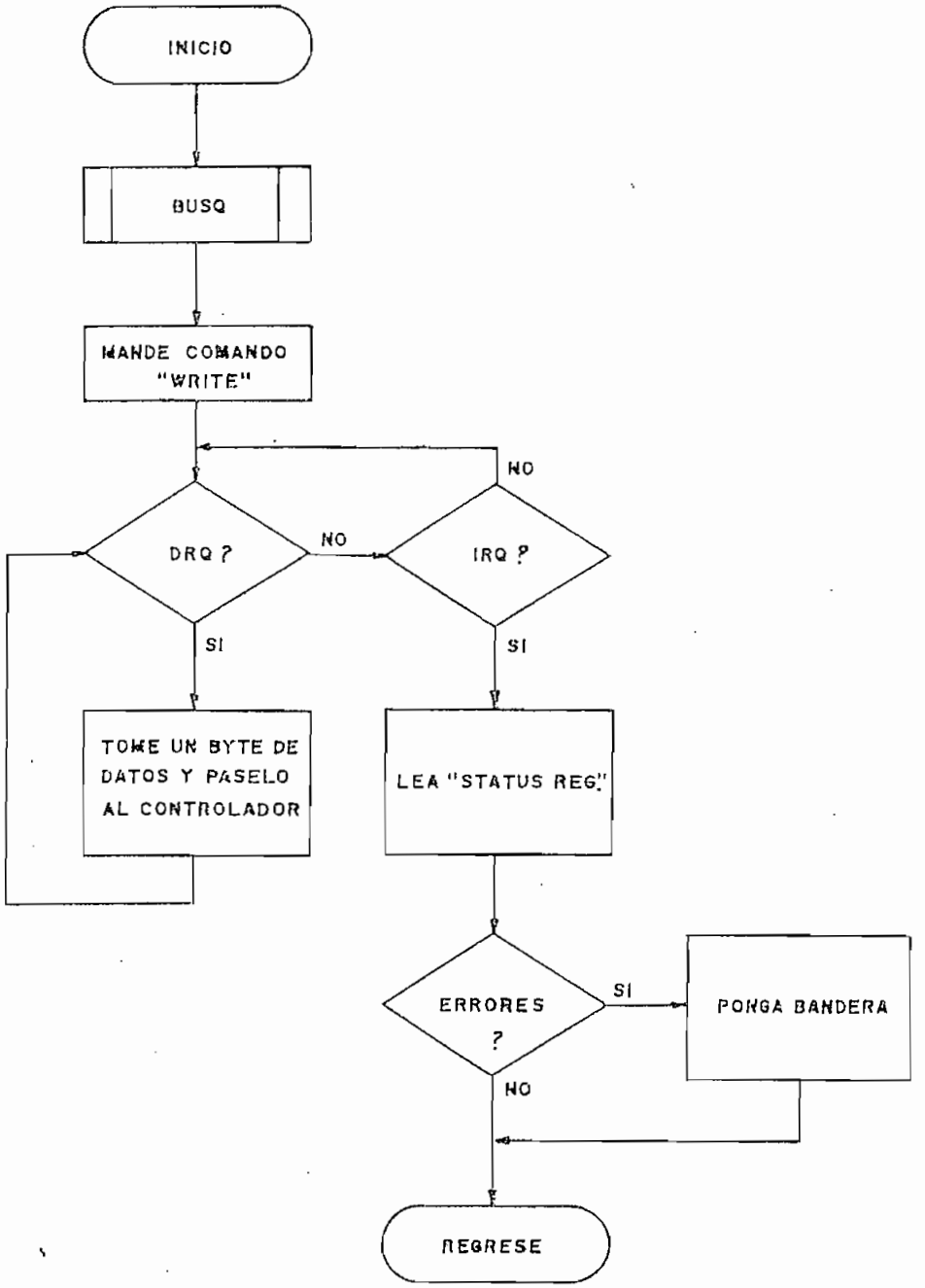
4.1b.4. BUSQ.


INICIO
"SECTOR REG." = No. DE SECTOR
LEA: "TRACK REG."
IGUAL A No. DE PISTA ?
NO
MAYOR ?
NO
MANDE COMANDO "STEP IN"
SI
SI
MANDE COMANDO "STEP OUT"
SE LO TERMINO ?
NO
SI
LEA "STATUS REG"
ERROR
NO
ESPERE 10 mSEG.
SI
REGRESE
PONGA BANDERA

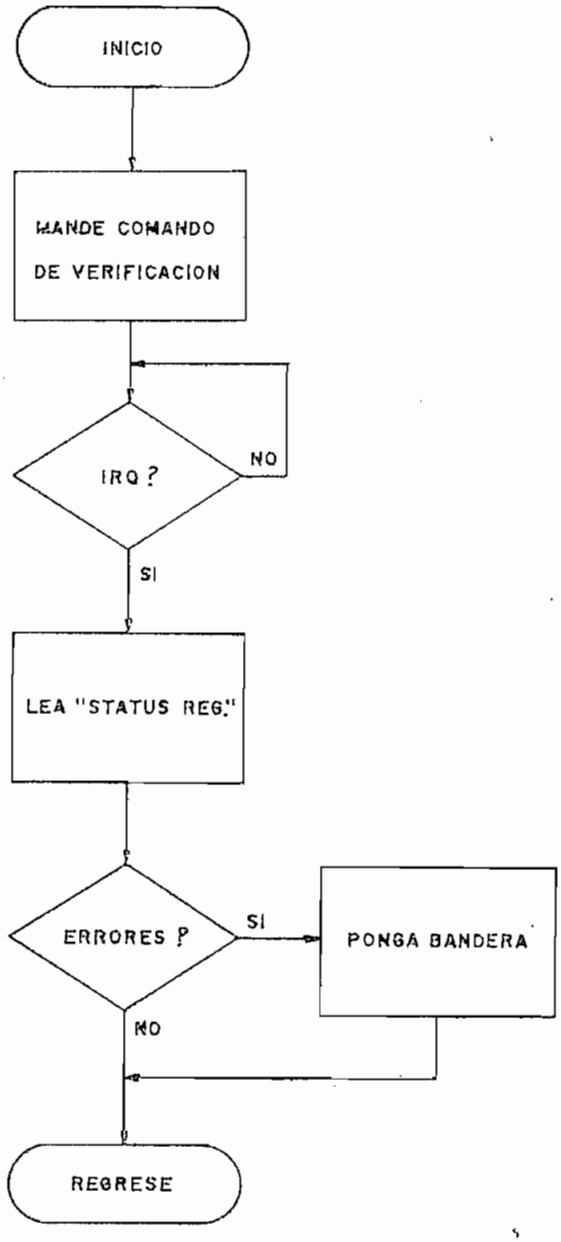
4.1b.5. LEA.


INICIO
BUSQ
MANDE COMANDO "READ"
DRQ ?
SI
TOME EL BYTE Y PONGALO EN RAM
NO
IRQ ?
NO
SI
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

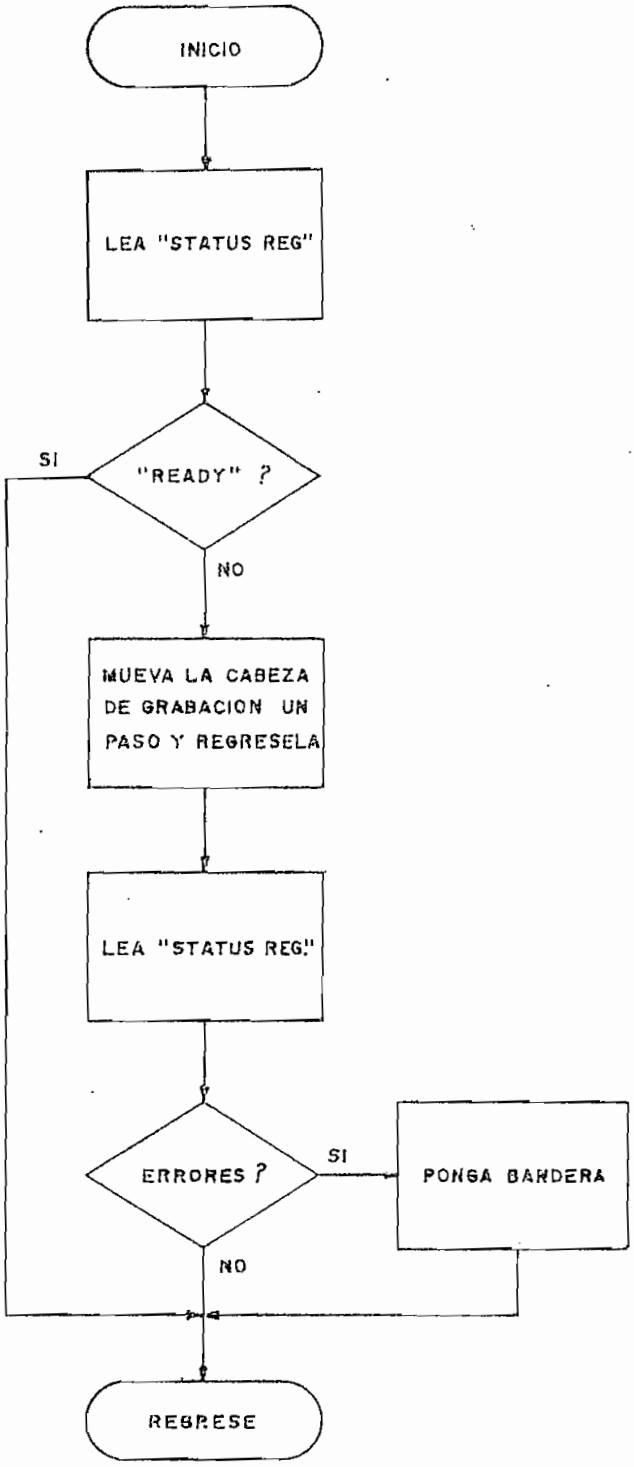
4.1b.6. ESCRIB.


INICIO
BUSQ
MANDE COMANDO "WRITE"
NO
DRQ ?
NO
IRQ ?
SI
SI
TOME UN BYTE DE DATOS Y PASELO AL CONTROLADOR
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

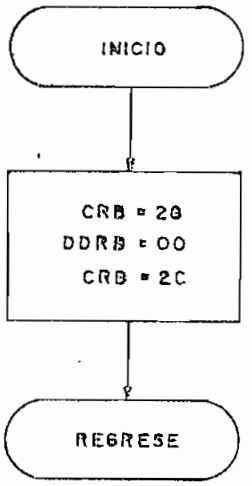
4.1b.7. VERIF.


INICIO
MANDE COMANDO DE VERIFICACION
IRQ ?
NO
SI
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

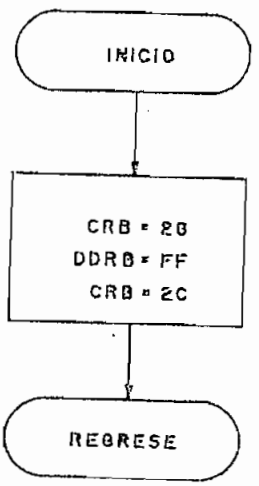
4.1b.8. CHKRDY.


INICIO
LEA "STATUS REG"
"READY" ?
SI
NO
MUEVA LA CABEZA DE GRABACION UN PASO Y REGRESELA
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

4.1b.9. ENTR.

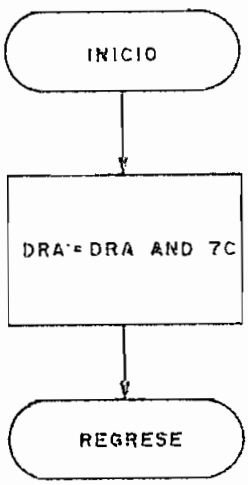


4.1b.10. SALID.



4.1b.11. DTARG.

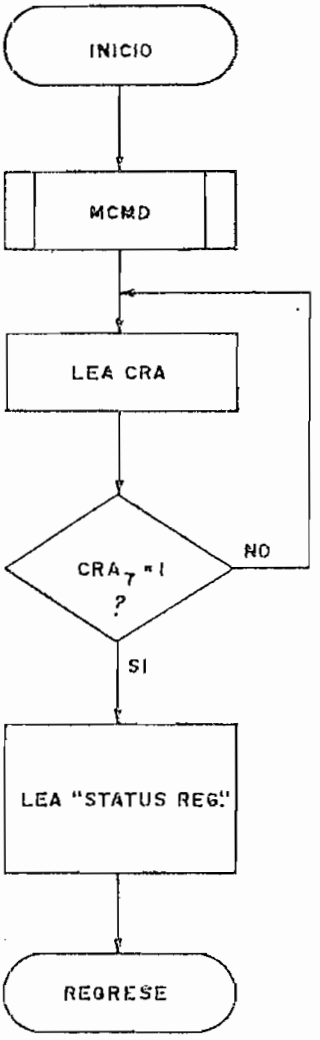
4.1b.12. CMRG.



4.1b.13. LEARG.

4.1b.14. MCMD.

4.1b.15. MCSP.



4.2. PROGRAMAS OPERATIVOS.

Seguidamente se da una ligera explicación de los programas operativos. Como en el caso de los programas de control, se darán las condiciones de entrada/salida para cada programa, incluyendo -cuando sea de importancia- el contenido de uno o más registros creados en memoria.

Los programas operativos podrán ser utilizados por el usuario siempre que se satisfagan las condiciones de entrada, la PIA se encuentre inicializada (por medio de la subrutina INIC), se haya seleccionado una lectora/escritora y el registro de pista del controlador contenga el número de pista sobre la cual se encuentra la cabeza de grabación de la lectora/escritora seleccionada (pudiendo utilizar para ello la subrutina REGR).

4.2.1. FORMAT.-

Por medio de esta subrutina se crea el formato en aquellos discos que van a ser utilizados por primera vez. El formato utilizado es de densidad simple y divide al disco en 35 pistas, subdivididas a su vez en 10 sectores de 256 bytes cada uno. La capacidad total de almacenamiento por disco será de 87,5 K bytes, de acuerdo a lo indicado en la sección 1.3.2.

En la sección 1.3.2. se indicó además la forma en que

se crea el formato en el disco y la función de cada uno de los bytes utilizados para ello. Este complejo proceso se reduce -utilizando el integrado FD1797-02- a escribir en cada pista los bytes indicados a continuación:

Nº DE BYTES	CONTENIDO	FUNCION
6	$ØØ	Fin de GAP IV (a partir del pulso índice)
1	$FC	Marca de pulso índice.
16	$FF	Comienzo de GAP I
* 6	$ØØ	Fin de GAPS I/III
1	$FE	Marca de bytes de identificación.
1	$XX	Número de pistas ($ØØ - $22)
1	$ØØ	Número de lado.
1	$ØX	Número de sector ($Ø1 - $ØA)
1	$Ø1	256 bytes/sector
1	$F7	Código para escritura de 2 bytes de CRC
11	$FF	Comienzo de GAP II
6	$ØØ	Fin de GAP II
1	$FB	Marca de campo de Datos
256	$E5	Datos
1	$F7	Código para escritura de 2 bytes de CRC.
14	$FF	Comienzo de GAP III/IV
72	$FF	Continuación de GAP IV

* NOTA: Los bytes indicados deberán ser escritos para cada sector en la pista (1Ø veces por pista).

Si bien, entre los programas de control no existe uno

para la escritura de una pista completa, se ha utilizado para ello a la subrutina ESCRIB desde la posición ESCRI2, almacenando previamente el código del comando "Write Track" en el acumulador A. La forma de utilización de este comando se puede encontrar en los apéndices de la presente tesis y su estudio ayudará a la mejor comprensión de la subrutina FORMAT.

Cabe anotar que en la creación del formato se ha utilizado la técnica de "sectores entrelazados", en ella la posición física de los sectores en una pista, no sigue una secuencia numérica, sinó que se intercalan sectores entre aquellos numéricamente consecutivos, con el objeto de proveer al sistema principal de un tiempo suficiente como para reorganizar los datos leídos y decidir si se debe leer el siguiente sector o no.

En el caso de discos de 5¼" la posición de los sectores en una pista sigue usualmente la siguiente secuencia: 1,3,5,7, 9,2,4,6,8,10 (secuencia Impar/Par), con ello, el tiempo requerido para leer dos o más sectores consecutivos puede ser reducido practicamente a la mitad del que se requeriría al no utilizar la técnica de sectores entrelazados, desde luego, lo anterior sucederá sólamente en el caso en que se tomen decisiones o se realice algún proceso cada vez que se termine la lectura de un sector.

Una vez creado el formato en el disco, FORMAT para directamente a la subrutina VRFDIS para verificar todos los sectores e inicializar el directorio del disco.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
Ninguna	NORMALES:	A = (DRA) B = (STATUS REG.) X = MEMPIS + 512 CCR: Z = 1
	DE ERROR1:	A = 3 B = (STATUS REG.) X = Indeterminado CCR: Z = Ø
	DE ERROR2:	A = (DRA) B = (STATUS REG.) X = Indeterminado CCR: Z = Ø

NOTA: ERROR1 = Error en la lectura de un sector.
ERROR2 = Error en la escritura del directorio.

4.2.2. VRFDIS.-

Esta subrutina puede ser considerada como parte de FORMAT y se la utiliza para verificar todos los sectores del disco e inicializar el directorio. Durante la verificación se realiza un proceso de lectura de todos los sectores pero, los datos leídos no son almacenados en RAM sinó que se buscan errores de CRC.

El directorio está constituído por los sectores 1 y 2 de la pista Ø, aunque solamente se utilizan los 350 primeros bytes de los 512 disponibles. Cada byte del directorio apunta a un sector en el disco y su contenido especifica el número de programa o grupo de datos al que pertenece dicho sector. Así por ejemplo, si en el byte número 12 del directorio se ha al

macenado el número 33 hexadecimal, significará que el sector número 2 de la pista 1, le corresponde al programa o grupo de datos número 33. Poniendo lo anterior en forma de una ecuación se tiene:

DIRECTORIO = Número de programa

Pista*10 + Sector

CONDICIONES DE ENTRADA		CONDICIONES DE SALIDA
	NORMALES:	A = (DRA) B = (STATUS REG.) X = MEMPIS + 512 CCR: Z = 1
Ninguna	DE ERROR1:	A = 3 B = (STATUS REG.) X = Indeterminado CCR: Z = Ø
	DE ERROR2:	A = (DRA) B = (STATUS REG.) X = Indeterminado CCR: Z = Ø

NOTA: ERROR1 = Error de lectura de un sector
ERROR2 = Error en la escritura del directorio.

4.2.3. GRBPR.-

Esta subrutina se utiliza para grabar un programa o un grupo de datos en el disco. El número de programa y las direcciones de memoria en donde comienza y termina el programa (o grupo de datos), son directamente preguntados al usuario por medio de las subrutinas de consola: PREGPD y PREGCF. Si el usuario no desea que estos datos sean ingresados desde el tecla

do sino que prefiere especificarlos dentro de alguno de sus programas, puede utilizar la subrutina GRBPR desde la posición GRBPRØ siempre que cumpla con las condiciones de entrada indicadas.

Las direcciones de memoria donde comienza y termina el programa, se adjuntan al comienzo del primer sector asignado para la grabación y son seguidas por los bytes que conforman el programa o grupo de datos. Los sectores asignados -cuyo número depende de la longitud del programa-, son siempre aquellos sectores libres (sin datos) que se encuentran lo más cerca posible al sector 1 de la pista Ø, es decir aquellos "más externos".

Si el número de programa que se está grabando ya ha sido utilizado, pero ocupa un menor número de sectores en el disco, GRBPR borra automáticamente los sectores sobrantes, de forma que puedan ser utilizados en grabaciones que se realicen posteriormente, con ello se logra una mejor utilización del espacio disponible en el disco.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
GRBPR: Ninguna	NORMALES: A = (DRA)
GRBPRØ: NPROG = Número de programa	B = (STATUS REG.)
COMADD = Dirección de memoria donde comienza el programa.	X = MEMPIS + 512
	FIN = 1
	CCR: Z = 1

FINADD = Dirección de memoria donde termina el programa.	DE ERROR1:	A = (DRA)
		B = (STATUS REG.)
		X = MEMPIS + 512
		FIN = Ø
		CCR: Z = 1
	DE ERROR2:	A = 3
		B = (STATUS REG.)
		X = Indeterminado
		CCR: Z = Ø ; C = 1
	DE ERROR3:	A = 3
		B = (STATUS REG.)
		X = Indeterminado
		CCR: Z = Ø ; C = Ø
	DE ERROR4:	A = (DRA)
		B = (STATUS REG.)
		X = Indeterminado
		CCR: Z = Ø ; C = Ø

NOTA: ERROR1 = Error de disco lleno.

ERROR2 = Error de lectura del directorio.

ERROR3 = Error de verificación del directorio o sector.

ERROR4 = Error de escritura del directorio o sector.

4.2.4. LEAPRG.-

La subrutina LEAPRG permite leer un programa del disco. El número de programa o grupo de datos es directamente preguntado al usuario, por medio de la subrutina de consola PREGPD,

pero si el usuario desea definir este número dentro de alguno de sus programas, puede utilizar la subrutina LEAPRG desde la posición LEAPRØ, para ello sólo tendrá que cumplir con las condiciones de entrada que se indican.

Como se indicó en la sección 4.2.3., los cuatro primeros bytes del primer sector asignado al programa contiene las localidades de memoria en las que éste comienza y termina, por lo que LEAPRG las utiliza, durante la lectura, para que el programa o grupo de datos sea repuesto a su posición original (aquella que ocupaba cuando fue grabado).

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
	NORMALES:	A = (DRA)
		B = (STATUS REG.)
		X = (FINADD)
		CCR: Z = 1 ; C = Ø
	DE ERROR1:	A = 3
		B = (STATUS REG.)
LEAPRG: Ninguna		X = Indeterminado
LEAPRØ: NPROG = Número de programa		CCR: Z = Ø ; C = 1
	DE ERROR2:	A = Nº de programa
		B = 35
		X = MEMPIS + 350
		CCR: Z = Ø ; C = Ø
	DE ERROR3:	A = 3
		B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = Ø

NOTA: ERROR1 = Error en la lectura del directorio.

ERROR2 = Programa no existe o está mal grabado.

ERROR3 = Error en la lectura de un sector del programa.

4.2.5. BORPRG.-

Utilizando esta subrutina se pueden borrar programas del disco, de igual forma que las subrutinas GRBPR y LEAPRG el usuario puede ingresar el número de programa a borrarse desde alguno de sus programas -utilizando para ello la subrutina BORPRG desde la posición BORPRØ- ó desde el teclado -utilizando la subrutina BORPRG desde le comienzo.

Cabe anotar que el programa no es propiamente borrado del disco, sino que se lo elimina del directorio, de esta forma los sectores que estuvieron ocupados por el programa podrán ser utilizados en alguna grabación posterior.

CONDICIONES DE ENTRADA

BORPRG: Ninguna

BORPRØ: NPROG = Número de programa

CONDICIONES DE SALIDA

NORMALES: A = (DRA)

B = (STATUS REG.)

X = MEMPIS + 512

CCR: Z = 1 ; C = Ø

DE ERROR1: A = Nº de programa

B = 35

X = MEMPIS + 35Ø

CCR: Z = Ø ; C = Ø

DE ERROR2: A = 3

B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = 1

DE ERROR3: A = 3

B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = Ø

DE ERROR4: A = (DRA)

B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = Ø

NOTA: ERROR1 = Programa no existe en el disco.

ERROR2 = Error de lectura del directorio.

ERROR3 = Error de verificación del directorio.

ERROR4 = Error de escritura del directorio.

4.2.6. LEADIR.-

Esta subrutina se utiliza para leer el directorio del disco a memoria. Debe recordarse que el directorio está constituído por los sectores 1 y 2 de la pista Ø.

CONDICIONES DE ENTRADA CONDICIONES DE SALIDA

NORMALES: A = (DRA)

	B = (STATUS REG.)
	X = MEMPIS + 512
	CCR: Z = 1
Ninguna	DE ERROR: A = (DRA)
	B = (STATUS REG.)
	X = Indeterminado
	CCR: Z = Ø

4.2.7. DIRINI.-

Esta subrutina lee el directorio a memoria, utilizando para ello la subrutina LEADIR. En caso de que se produzca un error de lectura, intenta leerlo dos veces adicionales y una vez que se tiene el directorio en memoria, pasa directamente a la subrutina ENCPRG.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	NORMALES: A = (NPROG)
	B = Indeterminado
	X = MEMPIS + 1Ø*(PISTA)
	+ (SECTOR) - 1
Ninguna	CCR: Z = 1
	DE ERROR1: A = (NPROG)
	B = $23
	X = MEMPIS + 35Ø
	CCR: Z = Ø
	DE ERROR2: A = 3

B = (STATUS REG.)

X = Indeterminado

CCR: C = 1

NOTA: ERROR1 = Error de disco lleno.

ERROR2 = Error de lectura del directorio.

4.2.8. ENCPRG.-

Por medio de la subrutina ENCPRG se encuentran los números de pista y de sector que han sido asignados a un programa. Una vez encontrado el primer sector asignado, se podrán encontrar los siguientes, utilizando esta subrutina desde la posición ENCPR1.

Si en algún momento se pide a la subrutina que encuentre un programa que no se encuentre en el directorio, o que se trate de buscarlo por sobre la pista 34 sector 1Ø, la subrutina pondrá el código de error de disco lleno o programa inexistente: $23.

Debe notarse que ENCPRG supone que el directorio ha sido leído y que los registros de pista y de sector se han inicializado con Ø1 y Ø1 respectivamente.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
ENCPRG: A = Nº de programa a encontrarse.	NORMALES: A = A
X = MEMPIS + 1Ø	B = Indeterminado

ENCPR1: A = Nº de programa a encontrarse

X = MEMPIS + 1Ø*(PISTA)

+ (SECTOR) - 1

X = MEMPIS + 1Ø*(PISTA) +(SECTOR) - 1

CCR: Z = 1

DE ERROR: A = A

B = $23

X = MEMPIS + 35Ø

CCR: Z = Ø

4.2.9. ALMAC.-

Esta subrutina se utiliza para escribir en memoria un número determinado de bytes con un valor especificado por el usuario.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
A = Valor especificado	A = A
B = Nº de bytes a escribir	B = 1
X = Dirección de memoria desde donde se empieza.	X = X+B

4.2.1Ø. INIVAR.-

Esta subrutina se utiliza para inicializar los registros de pista y de sector con $Ø1, haciendo además que el registro X apunte al byte número once del directorio.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
Ninguna	A = 1
	B = B

X ≐ MEMPIS + 1Ø

CCR: N = Ø ; Z = Ø ; V = Ø

4.2.11. INTEN.-

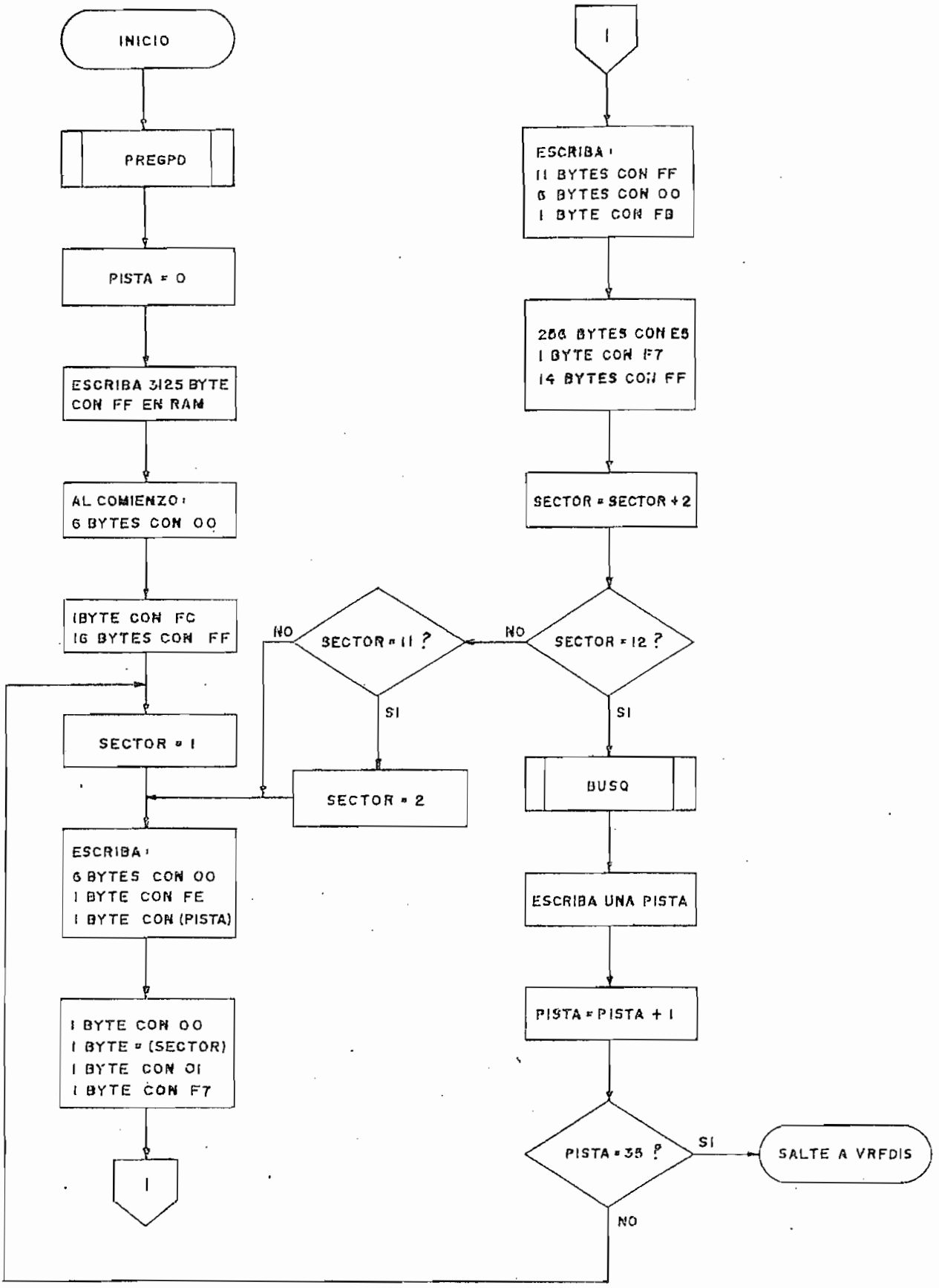
INTEN es un contador de intentos de lectura, además pondrá un indicador en el momento en que se hayan cumplido tres intentos.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = (INTENT)
	B = B
	X = X
	CCR: Z = 1 (3 Intentos)
	X = Ø (≠ 3 Intentos)

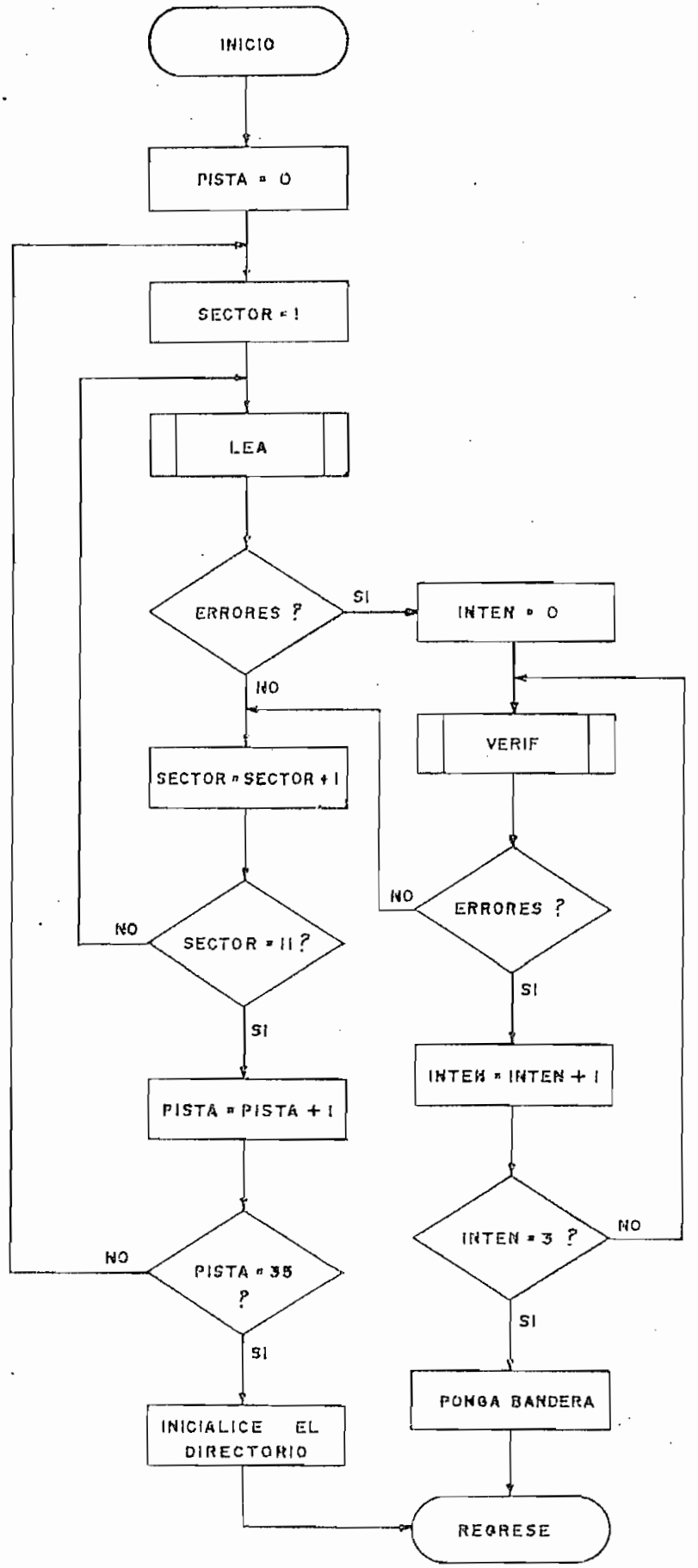
4.2b. DIAGRAMAS DE FLUJO DE LOS PROGRAMAS OPERATIVOS.

A continuación se presentan los diagramas de flujo correspondientes a los programas operativos.

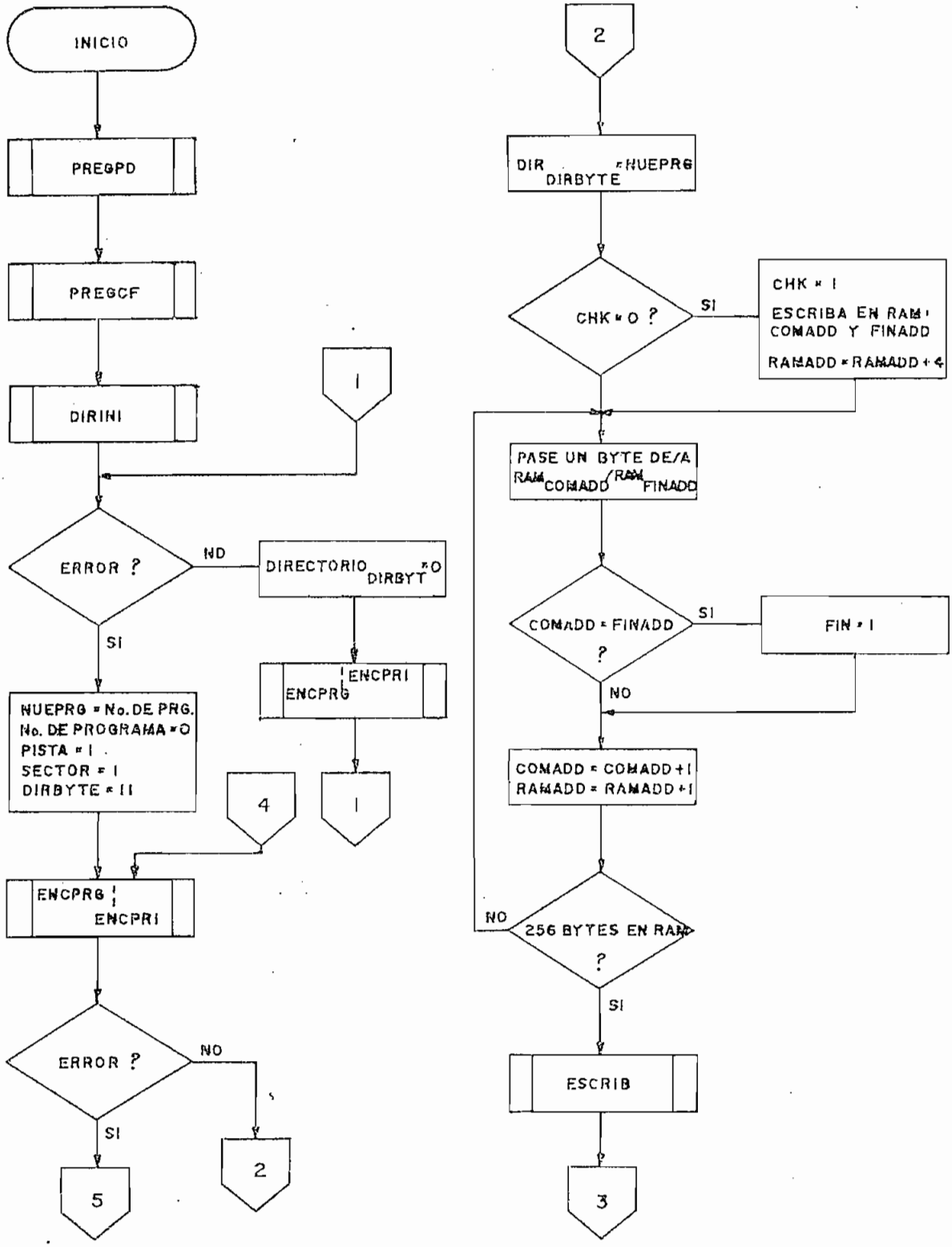
4.2b.1. FORMAT.


INICIO
PREGPD
PISTA = 0
ESCRIBA 3125 BYTE CON FF EN RAM
AL COMIENZO: 6 BYTES CON 00
1BYTE CON FC 16 BYTES CON FF
SECTOR = 1
ESCRIBA: 6 BYTES CON 00 1 BYTE CON FE 1 BYTE CON (PISTA)
1 BYTE CON 00 1 BYTE = (SECTOR) 1 BYTE CON 01 1 BYTE CON F7
1
SECTOR = 11 ?
NO
SI
SECTOR = 2
1
ESCRIBA: 11 BYTES CON FF 6 BYTES CON 00 1 BYTE CON FB
256 BYTES CON E5 1 BYTE CON F7 14 BYTES CON FF
SECTOR = SECTOR + 2
SECTOR = 12 ?
NO
SI
BUSQ
ESCRIBA UNA PISTA
PISTA = PISTA + 1
PISTA = 35 ?
SI
SALTE A VRFDIS
NO

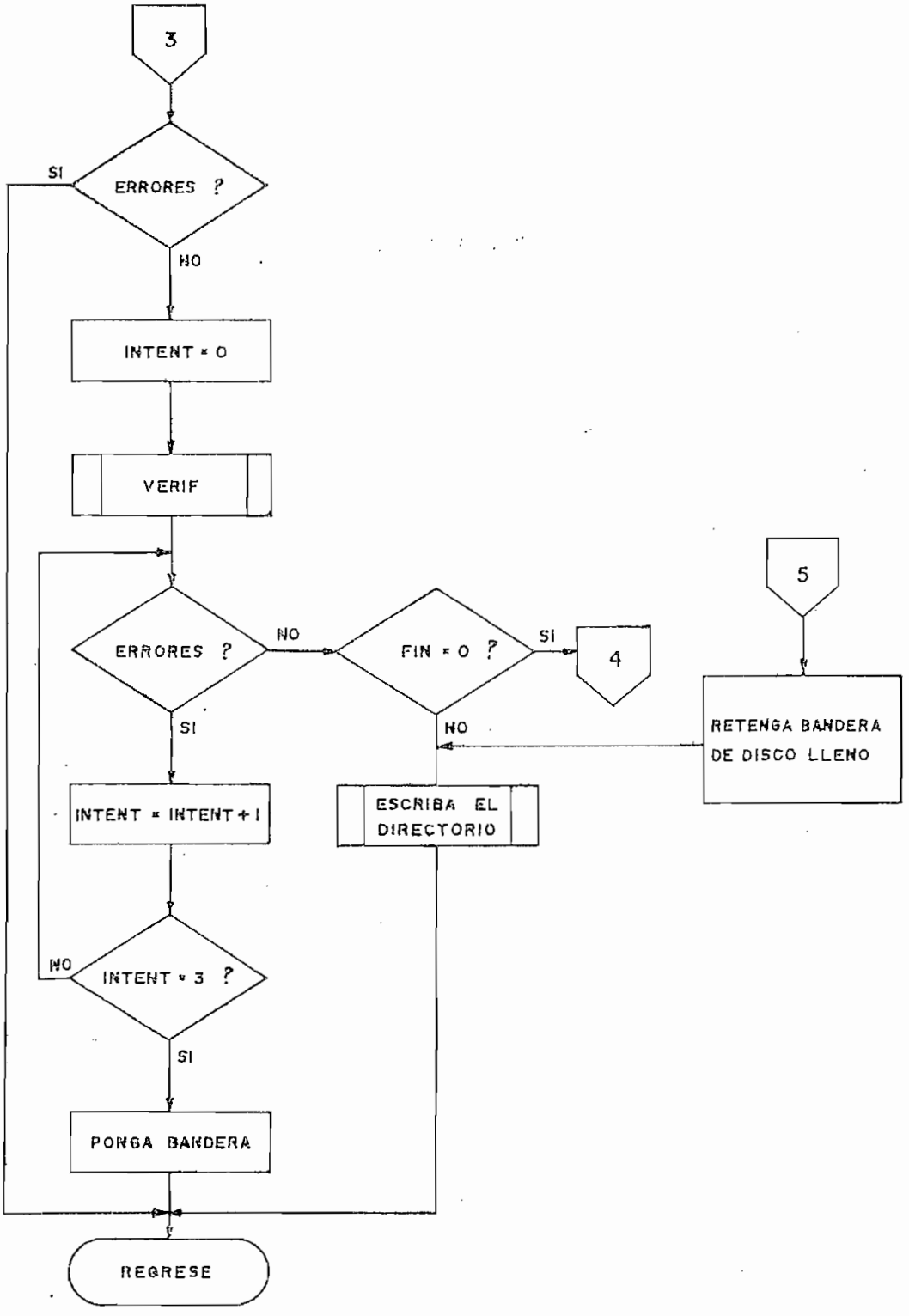
4.2b.2. VRFDIS.


INICIO
PISTA = 0
SECTOR = 1
LEA
ERRORES ?
SI
NO
INTEN = 0
VERIF
SECTOR = SECTOR + 1
NO
ERRORES ?
NO
SECTOR = 11 ?
SI
INTEN = INTEN + 1
PISTA = PISTA + 1
INTEN = 3 ?
NO
NO
PISTA = 35 ?
SI
SI
PONGA BANDERA
INICIALICE EL DIRECTORIO
REGRESE

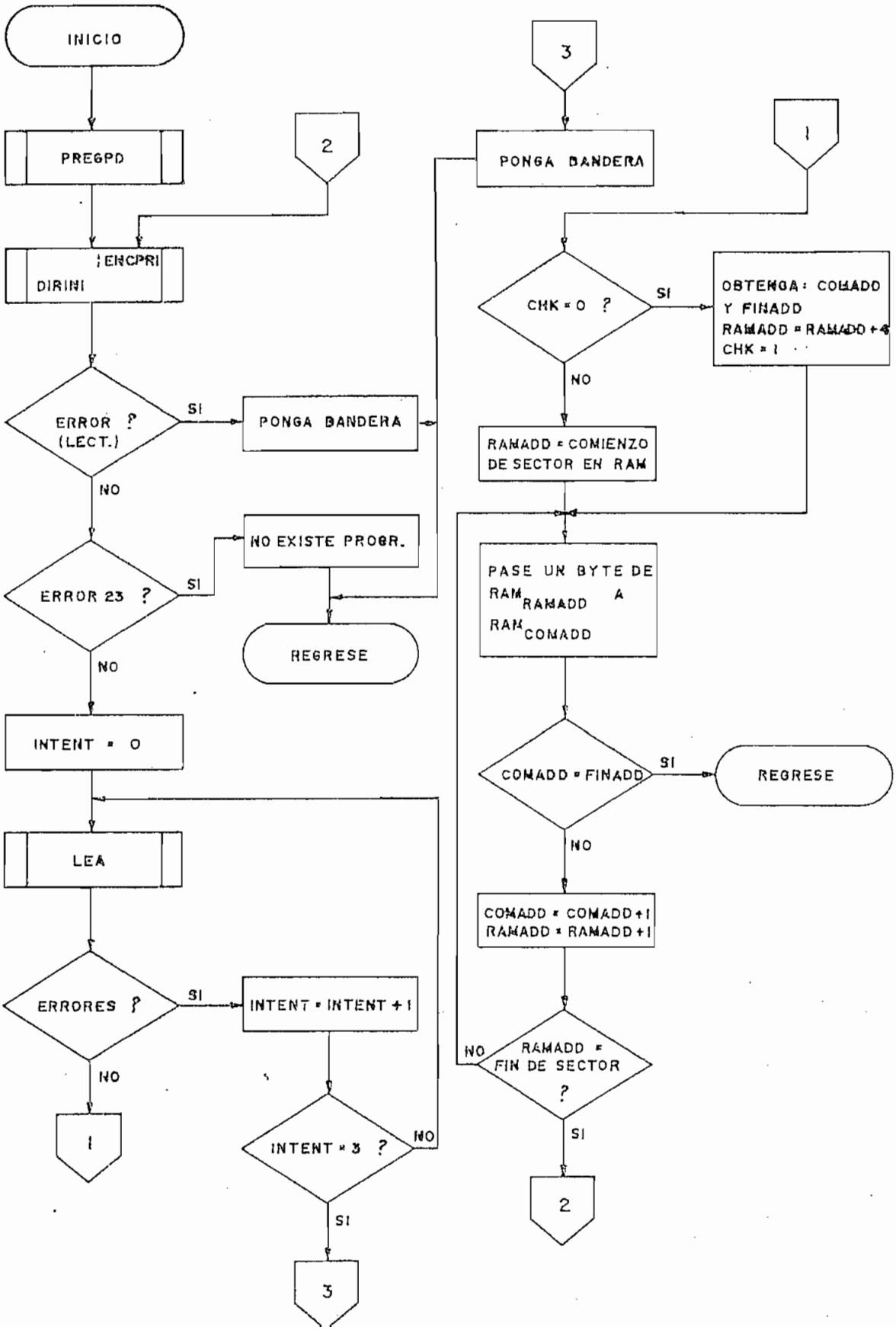
4.2b.3. GRBPRG.


INICIO
PREGPD
PREGCF
DIRINI
1
ERROR ?
NO
DIRECTORIO DIRBYT =0
ENCPRI
ENCPRG
1
SI
NUEPRG = No. DE PRG.
No. DE PROGRAMA = 0
PISTA = 1
SECTOR = 1
DIRBYTE = 11
4
ENCPRG
ENCPRI
ERROR ?
NO
SI
5
2
2
DIR DIRBYTE = NUEPRG
CHK = 0 ?
SI
CHK = 1
ESCRIBA EN RAM COMADD Y FINADD
RAMADD = RAMADD + 4
PASE UN BYTE DE/A RAM COMADD / RAM FINADD
COMADD = FINADD ?
SI
FIN = 1
NO
COMADD = COMADD + 1
RAMADD = RAMADD + 1
NO
256 BYTES EN RAM ?
SI
ESCRIB
3

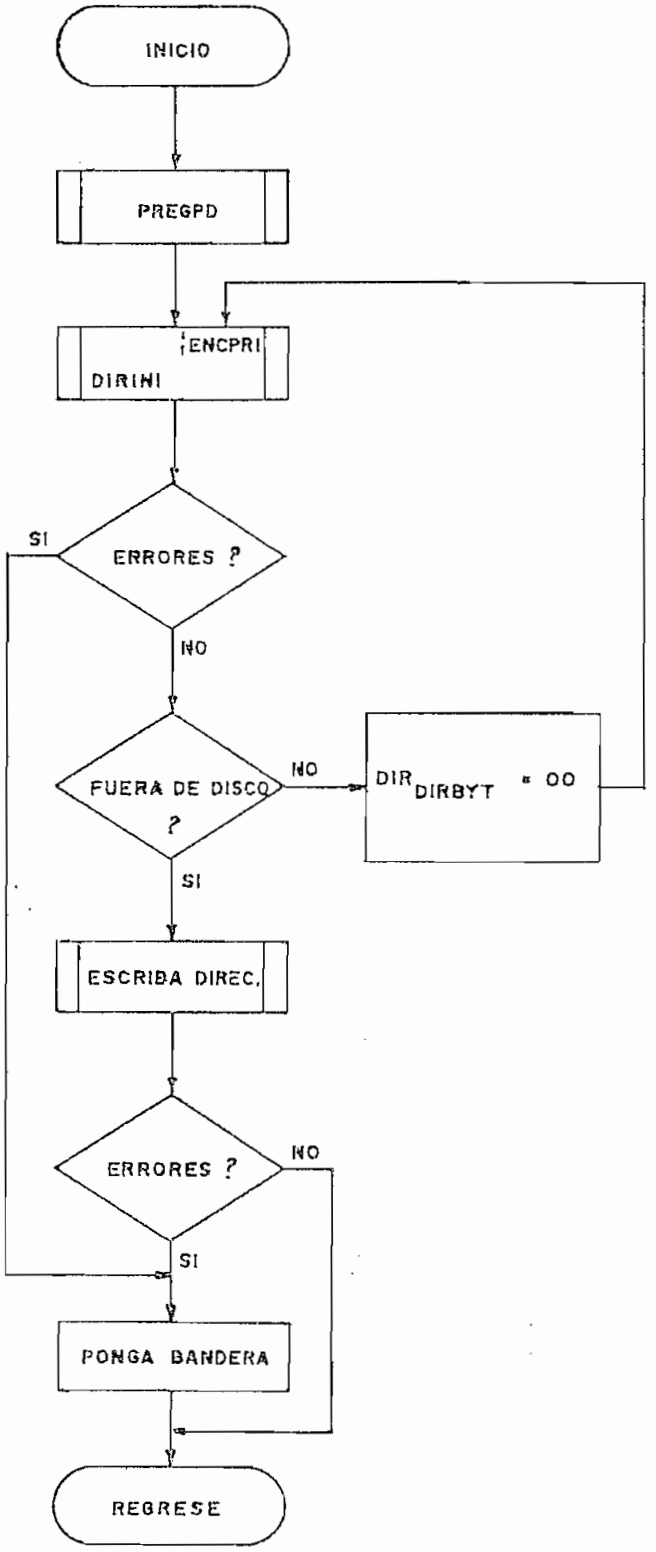
4.2b.3. GRBPRG.(Cont.)



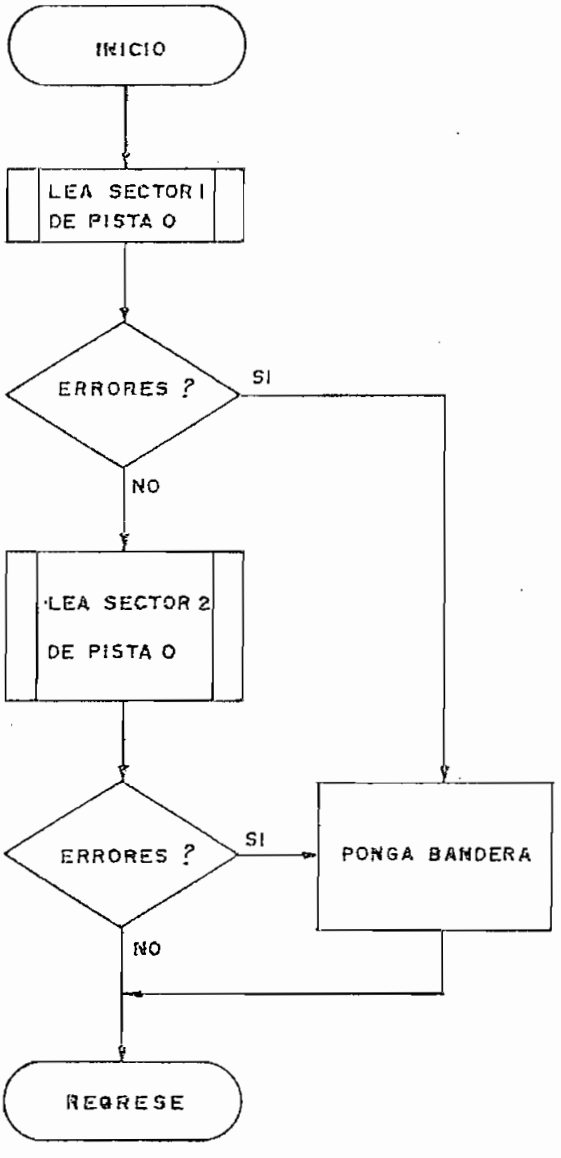
4.2b.4. LEAPRG.


INICIO
PREGPD
2
DIRINI
ENCPRI
ERROR ? (LECT.)
SI
PONGA BANDERA
NO
ERROR 23 ?
SI
NO EXISTE PROGR.
REGRESE
NO
INTENT = 0
LEA
ERRORES ?
SI
INTENT = INTENT + 1
NO
1
INTENT = 3 ?
NO
SI
3
3
PONGA BANDERA
1
CHK = 0 ?
SI
OBTENGA: COMADD Y FINADD
RAMADD = RAMADD + 4
CHK = 1
NO
RAMADD = COMIENZO DE SECTOR EN RAM
PASE UN BYTE DE RAM RAMADD A RAM COMADD
COMADD = FINADD
SI
REGRESE
NO
COMADD = COMADD + 1
RAMADD = RAMADD + 1
RAMADD = FIN DE SECTOR ?
NO
SI
2

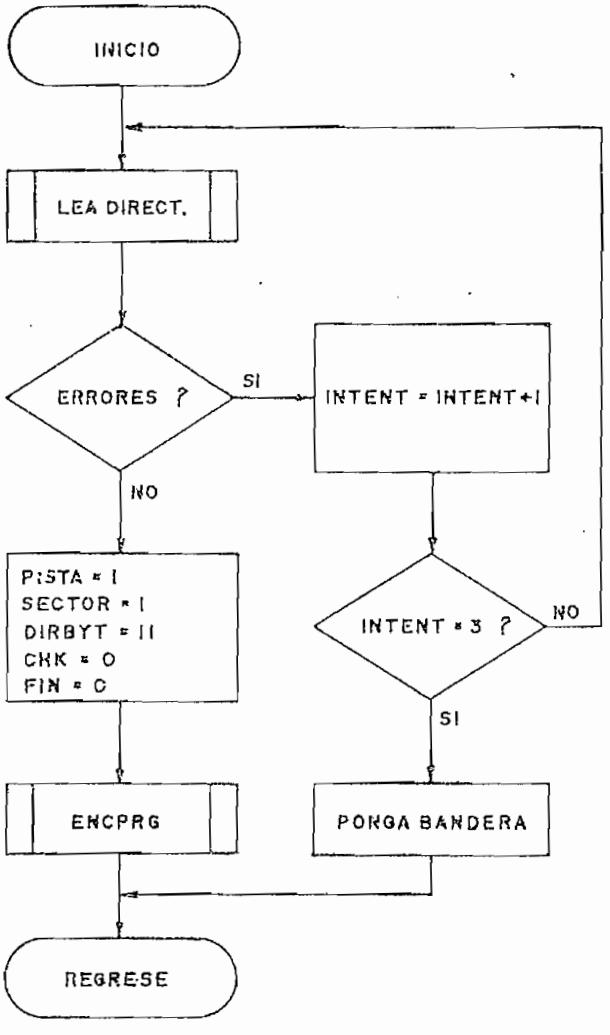
4.2b.5. BORPRG.


INICIO
PREGPD
ENCPRI
DIRINI
ERRORES ?
SI
NO
FUERA DE DISCO ?
NO
DIR DIRBYT = 00
SI
ESCRIBA DIREC.
ERRORES ?
NO
SI
PONGA BANDERA
REGRESE

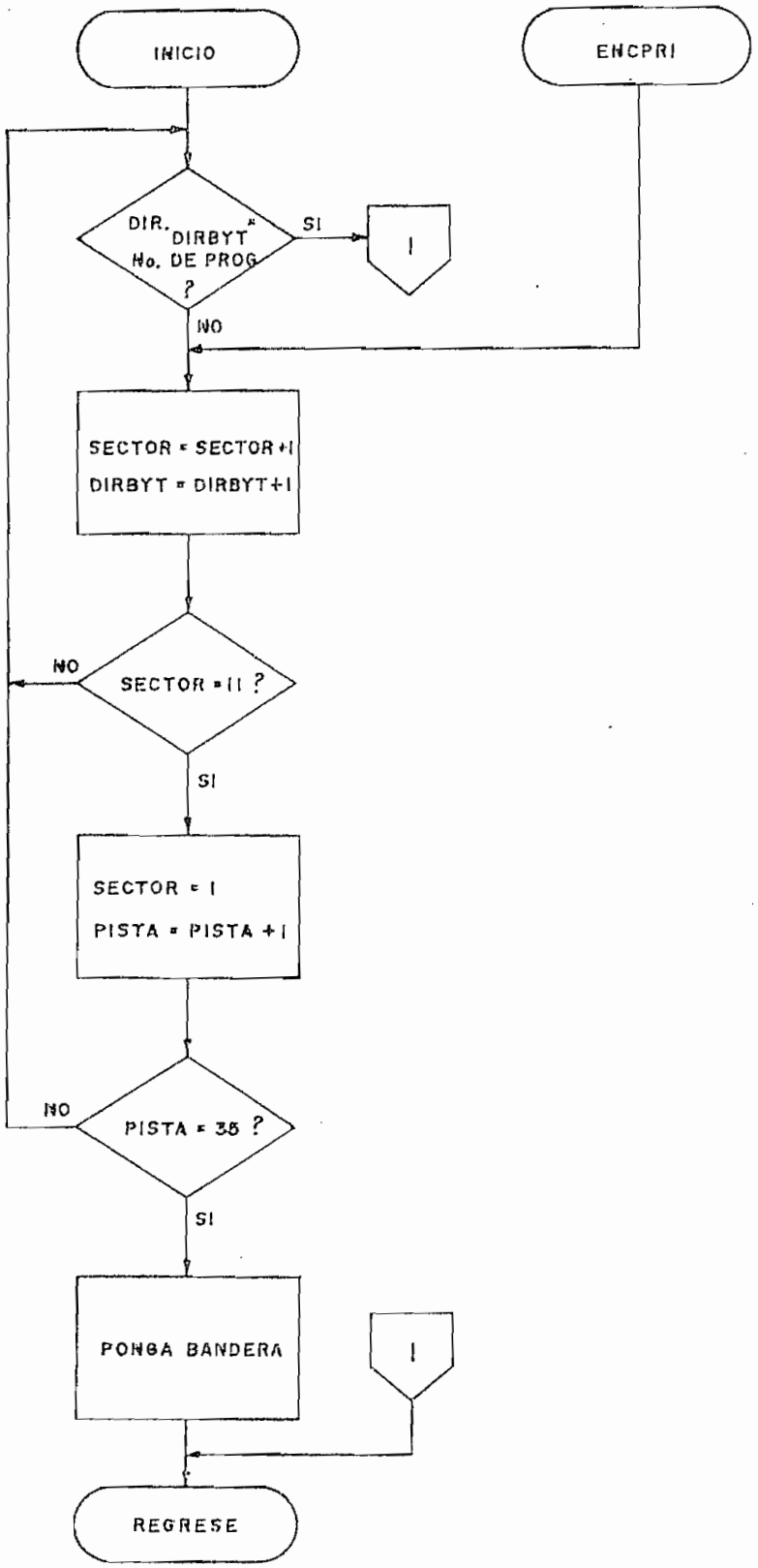
4.2b.6. LEADIR.


INICIO
LEA SECTOR 1
DE PISTA 0
ERRORES ?
SI
NO
LEA SECTOR 2
DE PISTA 0
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

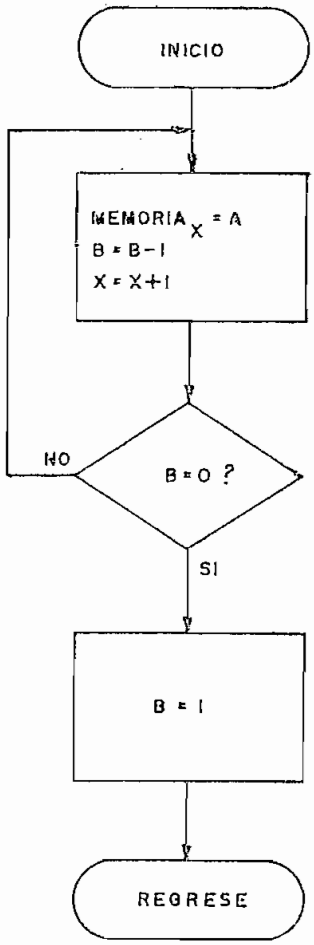
4.2b.7. DIRINI.


INICIO
LEA DIRECT.
ERRORES ?
SI
INTENT = INTENT+1
NO
PISTA = 1
SECTOR = 1
DIRBYT = 11
CHK = 0
FIN = 0
INTENT = 3 ?
NO
SI
ENCPRG
PONGA BANDERA
REGRESE

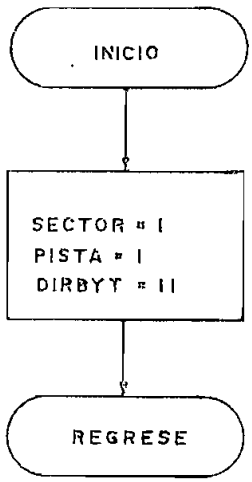
4.2b.8. ENCPRG.


INICIO
ENCPRI
DIR. DIRBYT* No. DE PROG ?
SI
1
NO
SECTOR = SECTOR+1
DIRBYT = DIRBYT+1
NO
SECTOR = 11 ?
SI
SECTOR = 1
PISTA = PISTA +1
NO
PISTA = 35 ?
SI
PONGA BANDERA
1
REGRESE

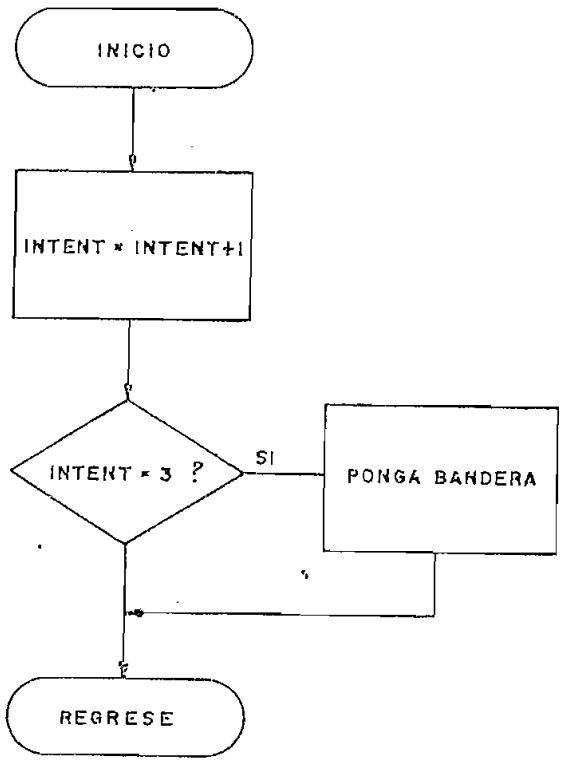
4.2b.9. ALMAC.



4.2b.10. INIVAR.



4.2b.11. INTEN.



4.3. PROGRAMAS DE CONSOLA

Los programas de consola se encargan de los procesos de comunicación con el usuario, en el presente caso controlan el teclado y la pantalla para el ingreso y salida de datos, por esta razón los programas de consola deberán ser modificados cada vez que se realice un cambio en esos elementos de entrada/salida.

Como en los otros tipos de programas se darán -siempre que sea posible- las condiciones de entrada/salida. Deberá notarse que algunos de los programas de consola se manejan directamente desde el teclado y que una vez finalizados regresan a él, por lo que en ellos no se podrá hablar de condiciones de entrada/salida.

4.3.1. PREGPD.-

Esta subrutina se utiliza para preguntar al usuario el número de lectora/escritora a utilizarse y el número de programa que se requiere para algún proceso. La subrutina pondrá una "D" en la pantalla para preguntar el número de lectora/escritora; una vez ingresado un dígito correcto (Ø ≤ dígito ≤ 3) verá si esa lectora/escritora está ya seleccionada, y de no ser así, la seleccionará por medio de la subrutina DISC. Una vez realizado el proceso anterior PREGPD pondrá una "A" en la pantalla para preguntar a qué número de programa se hace referencia, en este caso esperará el ingreso de dos dígitos (ØØ - FF) y almacenará este número en el registro NPROG.

Cada vez que se seleccione una nueva lectora/escritora, PREGPD averiguará en los registros de pista, si la cabeza de esa lectora/escritora se encuentra sobre la pista ØØ, si este es el caso mandará un comando de regreso a la pista ØØ, evitando de esta forma que se produzcan errores debido a que en la subrutina INIC se inicializaron todos los registros de pista con $ØØ. Si se produce un error mientras se busca la pista ØØ, la subrutina PREGPD regresará un nivel de subrutina más alto que aquel desde el cual fue llamada.

Mientras PREGPD esté pidiendo datos, el usuario tendrá la opción de "escapar" de la subrutina, aplastando la tecla E, en este caso se hará un salto a la subrutina PPRIN en su posición ESCAPE.

CONDICIONES DE ENTRADA		CONDICIONES DE SALIDA
	NORMALES:	A = Nº de programa B = 4 últimos bits de A X = (BPADR)

Ninguna	DE ERROR:	A = (DRA)
		B = (STATUS REG.)
		X = REGPIS + # de lectora/escritora seleccionada
		SP = SP + 4
		CCR: Z = Ø

4.3.2. PREGCF.-

Por medio de esta subrutina se preguntan al usuario las direcciones de memoria en donde comienza un programa y donde termina el mismo, estas direcciones son convertidas a números

hexadecimales de dos bytes para su almacenamiento en los registros COMADD y FINADD creados en memoria.

PREGCF pondrá una "C" en la pantalla para preguntar la dirección en donde comienza un programa y una "F" al preguntar la dirección en donde finaliza. Cuatro dígitos deberán ser ingresados para cada una de estas direcciones y como sucede en la subrutina PREPD, el usuario tiene la opción de "escapar" de la subrutina por medio de la tecla E.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
Ninguna	A = Byte menos significativo de (FINADD) B = 4 últimos bits de A. X = (FINADD)

4.3.3. INGRES.-

Esta subrutina se utiliza para el ingreso de datos desde el teclado. El número de dígitos que ingresen no podrá ser mayor que seis o menor a uno y serán almacenados en los registros creados en memoria: DSBUF - DSBUF+5.

INGRES escribe también una letra en la pantalla: A - F (códigos ØA - ØF), de esta forma el usuario podrá reconocer qué tipo de dato se le está pidiendo. Si se requiere que en la pantalla aparezca "-" ó un espacio en blanco, se podrán utilizar los códigos: 1Ø y 11 respectivamente.

Los cuatro primeros datos ingresados podrán ser convertidos a un número hexadecimal de dos bytes, para ello deberá

hacerse uso de la subrutina de monitor BLDX. Esta subrutina toma los datos almacenados en los registros: DSBUF - DSBUF+3, los convierte a dos bytes hexadecimales y guarda el byte más significativo en el registro BPADR y el menos significativo en el registro BPADR+1, por último el número completo es también almacenado en el registro X del microprocesador.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
A = Número de dígitos a ingresar	A = Indeterminado
B = Código para la pantalla	B = 4 últimos bits de último dato
	X = (XDBCHK)
	CCR: Z = 1

4.3.4. TECDIS.-

TECDIS tiene funciones múltiples, puede actuar como subrutina para el ingreso de datos (en cuyo caso es llamada desde la subrutina INGRES) o como programa que controla las opciones que tiene el usuario para el manejo de la información en sus discos, en este caso la función de cada una de las teclas es diferente a la que tenían bajo el control de los programas de monitor.

Mientras está actuando como subrutina (CHK≠Ø), se permite unicamente el ingreso de las teclas numéricas (Ø - F) y de la tecla de control: E, esta tecla hará que la subrutina regrese al monitor si interrupciones del tipo NMI están habilitadas, si este no es el caso, la subrutina regresará al programaPPRIN

en su posición ESCAPE. Debe notarse que las interrupciones NMI son inhabilitadas por medio de la subrutina de monitor DISNMI.

Mientras TECDIS esté actuando como programa de control de las opciones del teclado (CHK=Ø), sólamente permitirá el ingreso de las teclas de control: L,G,V,N,M. La función de cada una de estas teclas se explica a continuación:

- Tecla L: Lectura de un programa desde el disco a memoria. (Transfiere el control al programa LEER).
- Tecla G: Grabación de un programa en el disco. (Programa GRABAR).
- Tecla V: Borrado de un programa en el disco. (Programa BORRAR).
- Tecla N: Nuevo disco, creación del formato en él. (Programa NUEDIS).
- Tecla M: Monitor, escape al monitor. (Programa EREST del monitor).

Para TECDIS no se darán las condiciones de entrada/salida, pues cuando se lo utiliza como subrutina se la deberá llamar desde INGRES y las condiciones de salida serán iguales que para esa subrutina.

4.3.5. LEER.-

LEER hace uso de LEAPRG para leer un programa del disco,

de producirse un error, lo indicará poniendo el código del mismo en la pantalla (el código ØØ indicará que la lectura se efectuó sin problemas). Una vez terminado el proceso, LEER transferirá el control al programa TECDIS.

4.3.6. GRABAR.-

Este programa se utiliza para la grabación de datos ó programas en el disco, de igual forma que en el programa LEER al final del proceso se pondrá el código de error en la pantalla y se transferirá el control a TECDIS.

4.3.7. BORRAR.-

Por medio de BORRAR se pueden eliminar programas del directorio del disco. Para su funcionamiento utiliza la subrutina BORPRG poniendo el código de error en la pantalla y transfiriendo el control a TECDIS una vez terminada esta subrutina.

4.3.8. NUEDIS.-

NUEDIS debe utilizarse sólamente para aquellos discos que no contengan información o en los que ésta no tenga importancia. Puesto que NUEDIS utiliza a la subrutina FORMAT para su funcionamiento, el usuario tendrá la posibilidad de "escapar" de NUEDIS mientras se estén pidiendo los números de lectora/escritora y de volúmen, aplastando para ello la tecla E.

Si se produce un error, el usuario deberá intentar nue

vamente la creación del formato y, de mantenerse el error, desechar el disco que se esté utilizando.

4.2.9. PPRIN.-

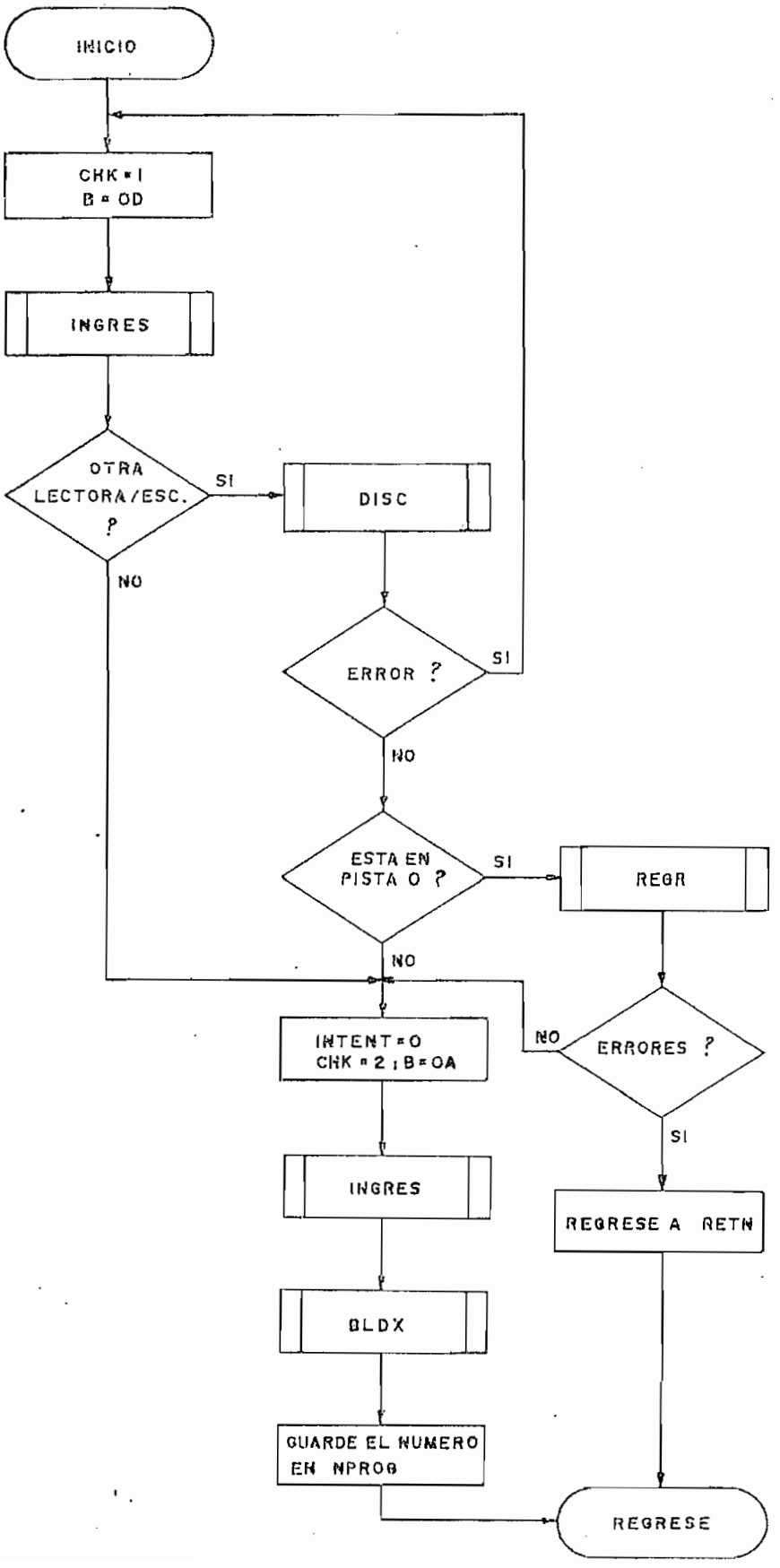
Este programa se encarga de los procesos de inicialización para el funcionamiento de la interface y el controlador de discos floppy y constituye por tanto la única entrada hacia el control y utilización de las lectoras/escritoras que se utilicen con la presente tesis.

En su posición ESCAPE se inicializa el "STACK"; se inhabilita interrupciones NMI; se borra la pantalla y se transfiere el control a TECDIS.

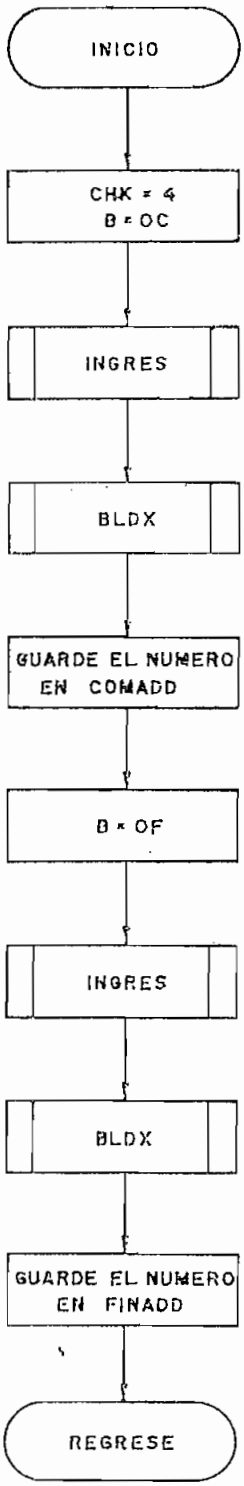
4.3b. DIAGRAMAS DE FLUJO DE LOS PROGRAMAS DE CONSOLA.

A continuación se presentan los diagramas de flujo correspondientes a los programas de consola.

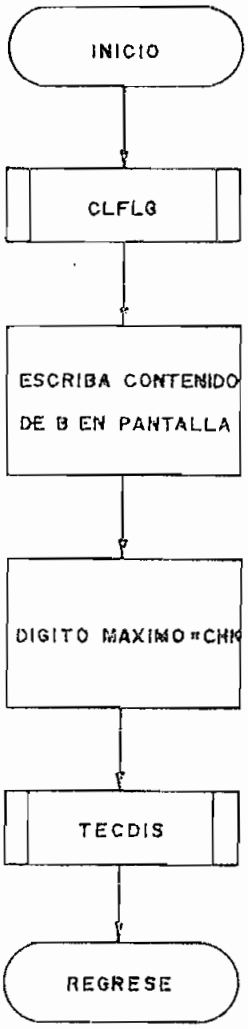
4.3b.1. PREGPD.


INICIO
CHK = I
B = OD
INGRES
OTRA LECTORA/ESC. ?
SI
DISC
NO
ERROR ?
SI
NO
ESTA EN PISTA O ?
SI
REGR
NO
INTENT = O
CHK = 2 ; B = OA
NO
ERRORES ?
SI
INGRES
REGRESE A RETN
BLDX
GUARDE EL NUMERO EN NPROG
REGRESE

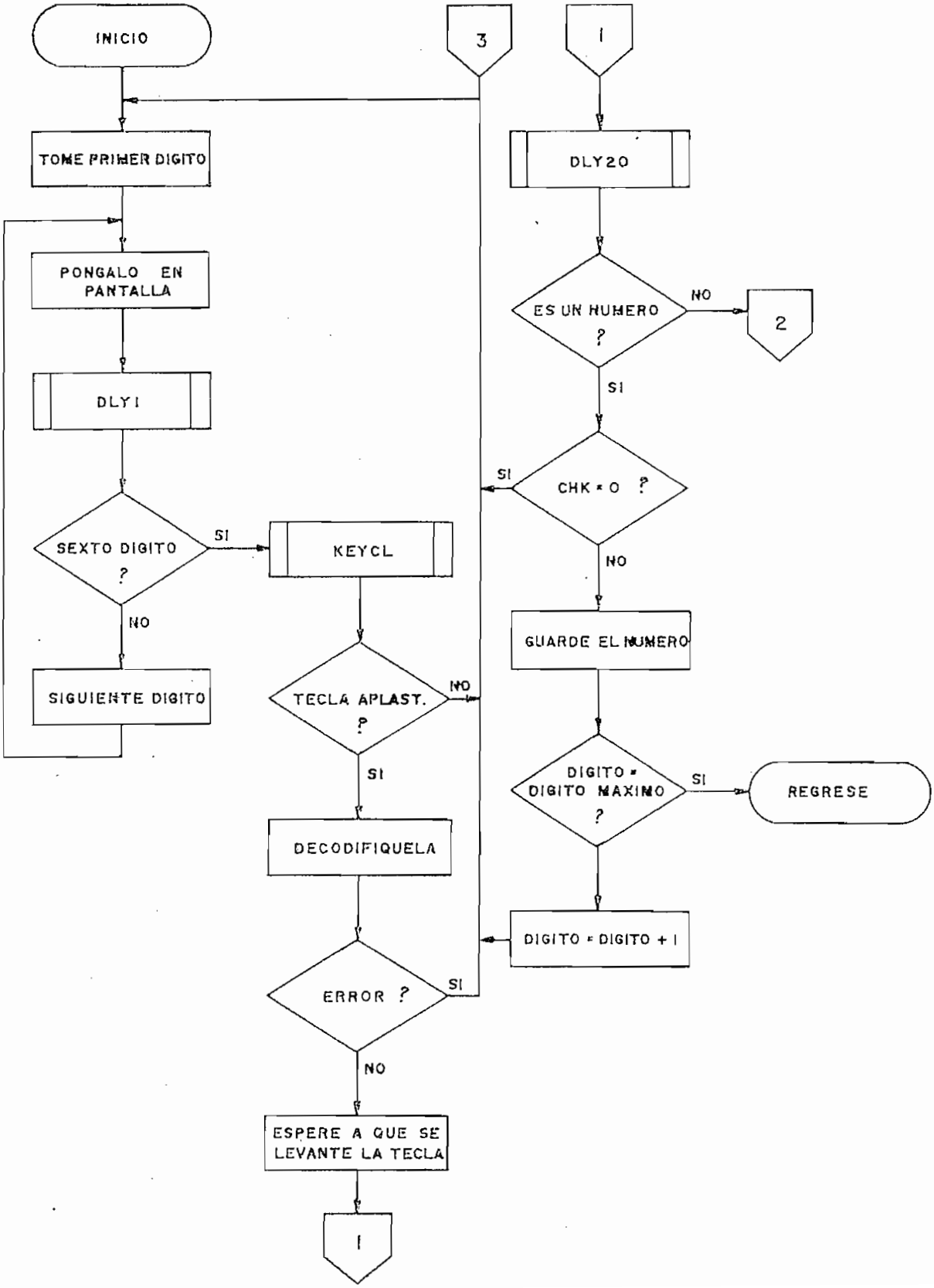
4.3b.2. PREGCF.


INICIO
CHK = 4
B = OC
INGRES
BLDX
GUARDE EL NUMERO
EN COMADD
B = OF
INGRES
BLDX
GUARDE EL NUMERO
EN FINADD
REGRESE

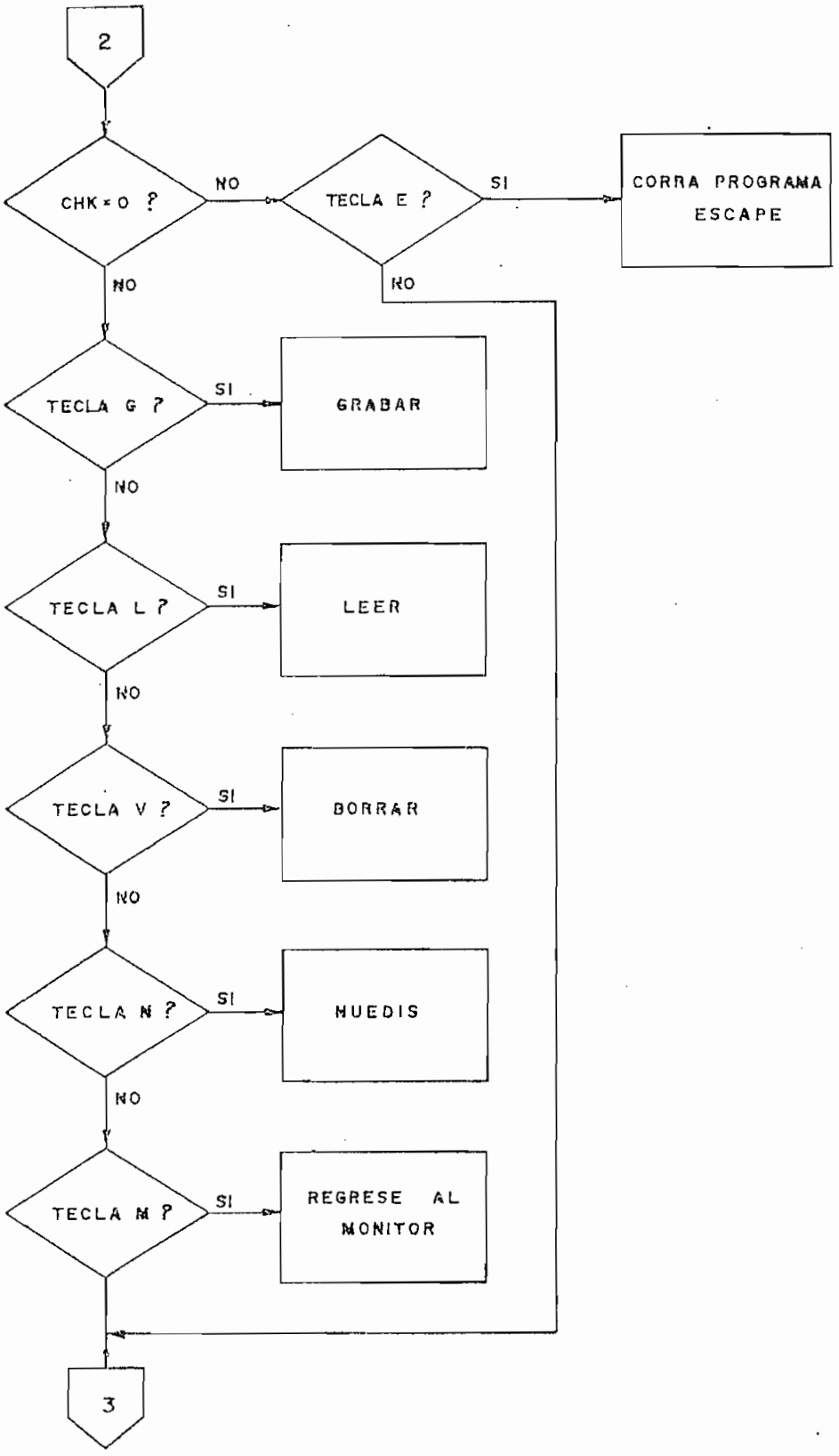
4.3b.3. INGRES.


INICIO
CLFLG
ESCRIBA CONTENIDO
DE B EN PANTALLA
DIGITO MAXIMO=CHK
TECDIS
REGRESE

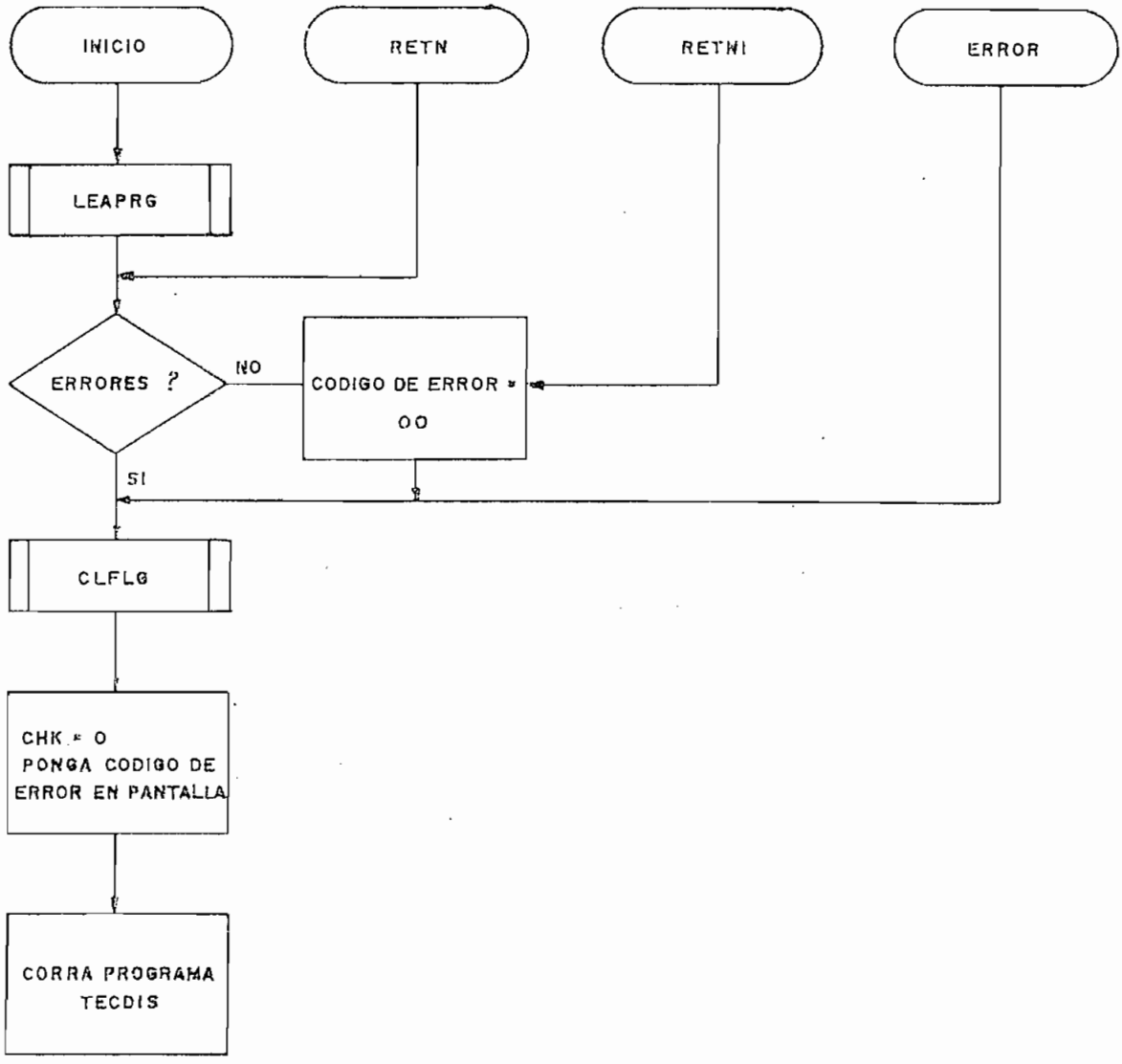
4.3b.4. TECDIS.


INICIO
3
1
TOME PRIMER DIGITO
DLY20
PONGALO EN PANTALLA
ES UN NUMERO ?
NO
2
SI
DLY1
SI
CHK = 0 ?
SEXTO DIGITO ?
SI
KEYCL
NO
NO
GUARDE EL NUMERO
SIGUIENTE DIGITO
TECLA APLAST. ?
NO
SI
DIGITO = DIGITO MAXIMO ?
SI
REGRESE
DECODIFIQUELA
DIGITO = DIGITO + 1
ERROR ?
SI
NO
ESPERE A QUE SE LEVANTE LA TECLA
1

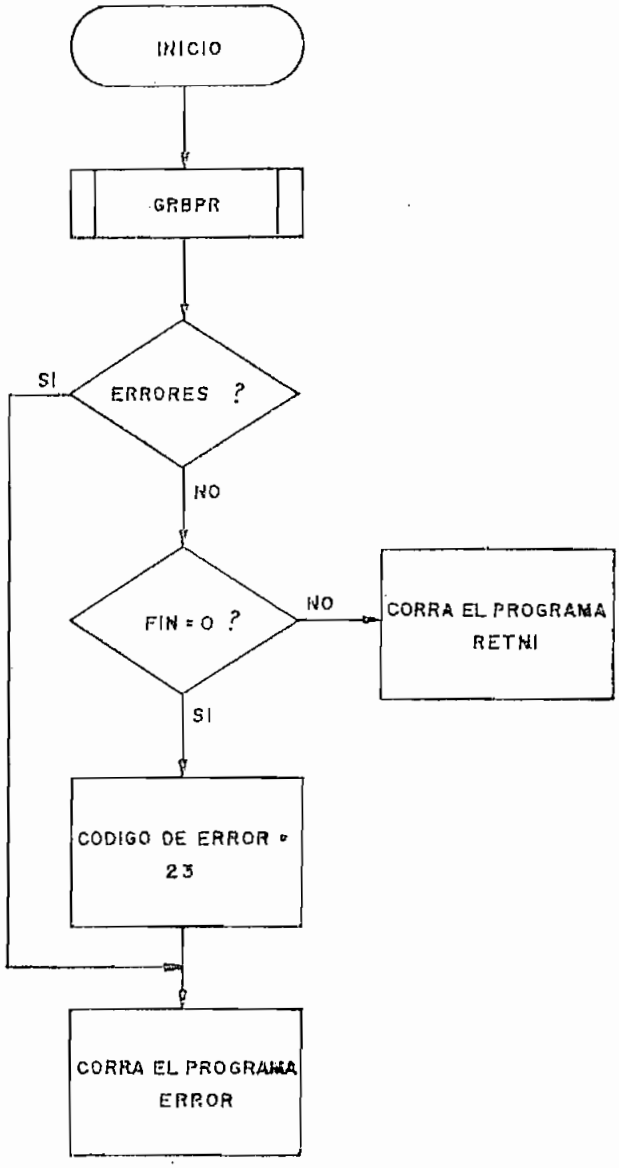
4.3b.4. TECDIS.(Cont.)



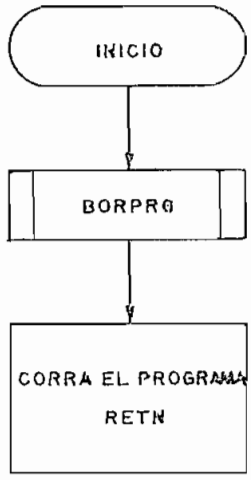
4.3b.5. LEER.


INICIO
RETN
RETNI
ERROR
LEAPRG
ERRORES ?
NO
CODIGO DE ERROR =
00
SI
CLFLG
CHK = 0
PONGA CODIGO DE
ERROR EN PANTALLA
CORRA PROGRAMA
TECDIS

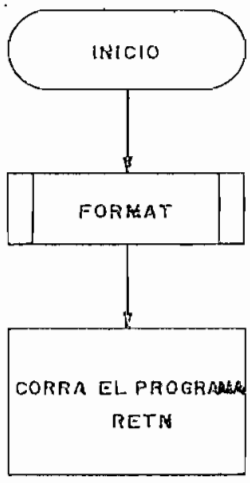
4.3b.6. GRABAR.


INICIO
GRBPR
ERRORES ?
SI
NO
FIN = 0 ?
NO
CORRA EL PROGRAMA RETNI
SI
CODIGO DE ERROR = 23
CORRA EL PROGRAMA ERROR

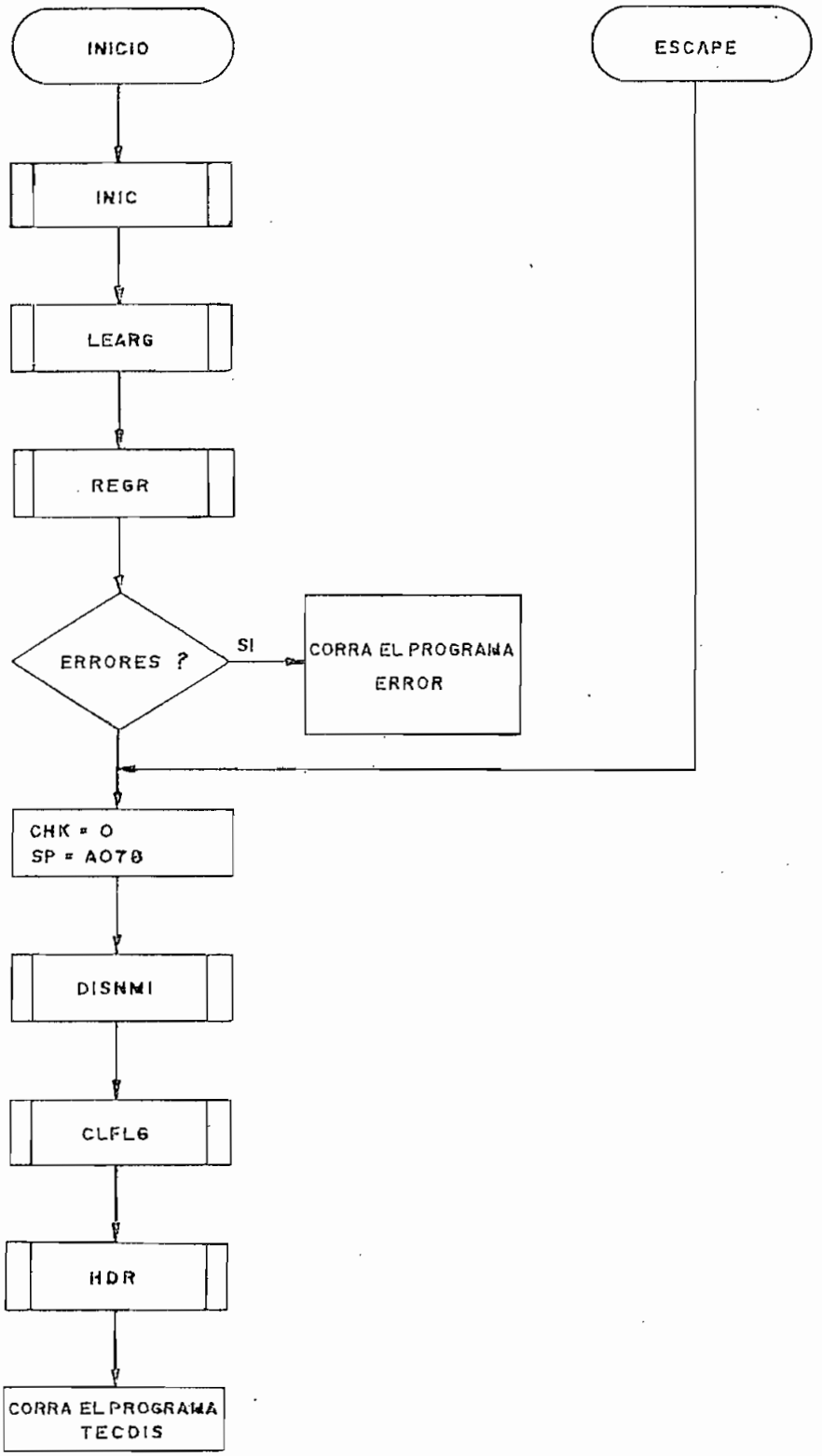
4.3b.7. BORRAR.



4.3b.8. NUEDIS.



4.3b.9. PPRIN.


INICIO
ESCAPE
INIC
LEARG
REGR
ERRORES ?
SI
CORRA EL PROGRAMA ERROR
CHK = 0
SP = A078
DISNMI
CLFLG
HDR
CORRA EL PROGRAMA TECDIS

4.4. CONJUNTO COMPLETO DE PROGRAMAS DE SOPORTE.

Antes de presentar el conjunto completo de programas de soporte -en código mnemotécnico y en lenguaje de máquina- será conveniente dar una ligera explicación de cada uno de los registros que han sido creados en memoria para el funcionamiento de los programas de soporte:

- DISACT = Número de la lectora/escritora actualmente seleccionada.
- REGPIS = Cuatro registros que contienen el número de pista en la que se encuentra cada una de las cuatro lectoras/escritoras.
- PISTA = Número de pista en el que se está trabajando.
- SECTOR = Número de sector que se utiliza.
- NPROG = Número de programas sobre el que se realiza un proceso.
- CHK = En el programa TECDIS: CHK=ØØ permite el ingreso de las teclas de control.
  CHK=n permite el ingreso de n teclas numéricas
  En LEAPRG y GRBPR: CHK=ØØ indica que el primer sector asignado a un programa se lee o escribe por primera vez.
  CHK=Ø1 En otros casos.

- FIN = Indicador de que el último byte de un programa, ha sido pasado para su grabación en el disco.
- INTENT = Contador de intentos de lectura.
- COMADD = Dirección de memoria en donde comienza un programa.
- FINADD = Dirección de memoria en donde termina un programa.
- RAMADD = Dirección de memoria a/de la cual se pasa un byte del/al programa.
- XDRBYT = Dirección de memoria que apunta a un byte del directorio.
- XDBCHK = Dirección de memoria en donde se almacenará el último dígito que ingrese desde la subrutina INGRES.
- DISBUF = Ocho registros que contienen los dígitos que se muestran en la pantalla.
- XKEYBF = Dirección de memoria en donde se almacenará el siguiente dígito que ingrese desde el teclado.
- SCNCNT = Contiene un bit en 1L que va desplazándose ala derecha para encender el siguiente dígito en la pantalla.
- BPADR = Registro de 16 bits en donde se almacenan los dos bytes que se crean a partir de los cuatro primeros dígitos almacenados en DISBUF.
- XDSBUF = Dirección de memoria de un dígito en DISBUF que se saca a la pantalla.

```
A00C                        ORG     $A00C
A00C 00          DISBUF     FCB     0,0,0,0,0,0,0,0
A01A                        ORG     $A01A
A01A 00 00       XKEYBF     FDB     0
A01C 00          SCNCNT     FCB     0
A01D 00          VFLAG      FCB     0
A01E 00 00       BPADR      FDB     0
A020 00 00       XDSBUF     FDB     0
A031                        ORG     $A031
A031             RAM        EQU     *
A031 00          DISACT     FCB     0
A032 00 00       REGPIS     FDB     0,0
A036 00          PISTA      FCB     0
A037 00          SECTOR     FCB     0
A038 00          NPROG      FCB     0
A039 00          CHK        FCB     0
A03A 00          FIN        FCB     0
A03B 00          INTENT     FCB     0
A03C 00 00       COMADD     FDB     0
A03E 00 00       FINADD     FDB     0
A040 00 00       RAMADD     FDB     0
A042 00 00       XDRBYT     FDB     0
A044 00 00       XDBCHK     FDB     0
                 *
2200                        ORG     $2200
2200 00 00       RAMSEC     FDB     0
                 *
                 * EQUATES
                 *
22FF             FINSEC     EQU     $22FF
E3CA             DIGTBL     EQU     $E3CA
E3DC             KEYTBL     EQU     $E3DC
8020             DISREG     EQU     $8020
8022             SCNREG     EQU     $8022
E0E0             DLY1       EQU     $E0E0
E0DD             DLY20      EQU     $E0DD
E12F             KEYCL      EQU     $E12F
E13A             KEYCL1     EQU     $E13A
E08D             RESTAR     EQU     $E08D
E084             DISNMI     EQU     $E084
E0B2             CLFLG      EQU     $E0B2
E0D7             HDR        EQU     $E0D7
E0E4             BLDX       EQU     $E0E4
E31C             REGSTS     EQU     $E31C
2000             MEMPIS     EQU     $2000
2C37             FINPIS     EQU     $2C37
2017             MEMSEC     EQU     $2017
                 *
00F4             WTKMND     EQU     $F4
008C             RDCMND     EQU     $8C
00AC             WTCMND     EQU     $AC
001B             SKCMND     EQU     $1B
000B             RSCMND     EQU     $0B
```

```
007B                SOCMNI   EQU     $7B
005B                SICMNI   EQU     $5B
009C                MSCLCT   EQU     $9C
0010                MSCBSQ   EQU     $10
00DC                MSCESC   EQU     $DC
0098                MSCVER   EQU     $98
0088                MSCRST   EQU     $88
0080                MSCRDY   EQU     $80
000C                SECMAX   EQU     $0C
000B                SECMXI   EQU     $0B
0023                PISMAX   EQU     $23
                    *
8004                DRA      EQU     $8004
8005                CRA      EQU     $8005
8006                DDRB     EQU     $8006
8007                CRB      EQU     $8007
8006                DRB      EQU     $8006
8004                DDRA     EQU     $8004
                    *
                    * JUMP TABLE (TABLA DE SALTOS)
                    *
C000                         ORG     $C000
C000 7E C0 4A       DLEA     JMP     LEA
C003 7E C0 C1       DESCR    JMP     ESCRIB
C006 7E C1 46       DVERIF   JMP     VERIF
C009 7E C1 4F       DREGR    JMP     REGR
C00C 7E C1 51       DREGR1   JMP     REGR1
C00F 7E C1 13       DDISC    JMP     DISC
C012 7E C1 58       DCHK     JMP     CHKRDY
C015 7E C0 2B       DINIC    JMP     INIC
C018 7E C0 36       DWARM    JMP     INIC3
C01B 7E C0 76       DBUSQ    JMP     BUSQ
C01E 7E C0 C5       DESCR2   JMP     ESCRI2
                    *
                    * DEMORA
                    *
                    * Demora de 10 milisegundos para BUSQ (614.4 KHz)
                    *
C021 86 02          DEMORA   LDA A   #$02        INICIALICE EL CONTADOR
C023 4A             DEMOR2   DEC A
C024 26 FD                   BNE     DEMOR2
C026 5A                      DEC B
C027 26 F8                   BNE     DEMORA      SE HA PRODUCIDO LA DEMORA?
C029 20 54                   BRA     BUSQ2       SI, REGRESE

                    *
                    * INIC
                    *
                    * INICIALICE PIA Y REGISTROS PARA DISCOS
                    *
C02B CE A0 31       INIC     LDX     #DISACT     APUNTE A LAS VARIABLES
C02E C6 05                   LDA B   #$5         NO. DE VARIABLES A BORRAR
C030 6F 00          INIC2    CLR     0,X         BORRE LA VARIABLE
C032 08                      INX                 APUNTE A LA SIGIENTE VARIABLE
```

```
C033 5A                  DEC B
C034 26 FA               BNE     INIC2      CONTINUE HASTA TERMINAR
C036 CE 80 04   INIC3    LDX     #DRA       APUNTE A LA PIA
C039 6F 01               CLR     1,X        OBTENGA ACCESO A DRA
C03B 86 FF               LDA A   #$FF       PROGRAME PA0-PA7 COMO SALIDAS
C03D A7 00               STA A   0,X
C03F 86 1E               LDA A   #$1E       OBTENGA ACCESO A DRA
C041 A7 01               STA A   1,X
C043 86 44               LDA A   #$44       APUNTE A COMREG,DISCO 0, INHABILITE
C045 A7 00               STA A   0,X        F-F DE LECTURA Y HABILITE HLT
C047 8D 6A               BSR     ENTR       PROGRAME PB0-PB7 COMO ENTRADAS
C049 39                  RTS
                *
                * LEA
                *
                * LEA UN SECTOR DEL DISCO
                *
C04A 8D 2A      LEA      BSR     BUSQ       PONGA EL DISCO EN LA PISTA CORECTA
C04C 86 8C               LDA A   #ROCMND    CARGUE UN COMANDO DE LECTURA
C04E 0F         LEA2     SEI                INHABILITE INTERRUPCIONES
C04F 8D 5D               BSR     MCMD       MANDE EL COMANDO
C051 8D 4B               BSR     DTARG      APUNTE A "DATA REGISTER" DE WD
C053 CA 80               ORA B   #$80       HABILITE FLIP-FLOP DE LECTURA
C055 F7 80 04            STA B   DRA
C058 86 C0               LDA A   #$C0       MASCARA PARA IRQ O DRQ DE WD
C05A B5 80 05   LEA3     BIT A   CRA        EXISTE UN DRQ O IRQ DE WD?
C05D 27 FB               BEQ     LEA3       NO, TRATE NUEVAMENTE
C05F 2B 0B               BMI     LEA4       ES UN IRQ?
C061 F6 80 06            LDA B   DRB        ENTONCES ES UN DRQ, LEA EL DATO
C064 E7 00               STA B   0,X        GUARDE EL DATO EN MEMORIA
C066 F6 80 04            LDA B   DRA        BORRE LA BANDERA
C069 08                  INX                APUNTE AL SIGUIENTE BYTE DE MEMORIA
C06A 20 EE               BRA     LEA3       CONTINUE HASTA TERMINAR
C06C 8D 39      LEA4     BSR     CMRG       INHABILITE F-F Y APUNTE A STATUS RE
C06E 8D 04               BSR     ELEARG     LEA STATUS REG DE WD Y BORRE BANDER
C070 C5 9C               BIT B   #MSCLCT    ENCUENTRE ERRORES DE LECTURA
C072 0E                  CLI                HABILITE INTERRUPCIONES
C073 39                  RTS
                *
                * ESTE ES UN ENLACE A LEARG
                *
C074 20 7F      ELEARG   BRA     LEARG
                *
                * BUSQ
                *
                * Busque la pista especificada
                *
C076 36         BUSQ     PSH A              GUARDE LA PISTA ESPECIFICADA
C077 17                  TBA                ACC. A CON EL SECTOR ESPECIFICADO
C078 8D 24               BSR     DTARG      APUNTE A "DATA REGISTER" DE WD
C07A 7A 80 04            DEC     DRA        *APUNTE A "SECTOR REGISTER"
C07D 8D 2F               BSR     MCMD       GUARDE EL NUMERO DE SECTOR EN WD
C07F 8D 26      BUSQ2    BSR     CMRG       APUNTE A "COMMAND REGISTER" DE WD
C081 7C 80 04            INC     DRA        *APUNTE A "TRACK REGISTER"
```

```
C084 8D 6F                BSR    LEARG      LEA SU CONTENIDO
C086 7A 80 04             DEC    DRA        *APUNTE A "COMMAND REGISTER" DE WD
C089 32                   PUL A             RECOBRE NO. DE PISTA
C08A 11                   CBA               COMPARELA CON LA DE WD
C08B 27 10                BEQ    BUSQ5      IGUALES
C08D 36                   PSH A
C08E 2E 04                BGT    BUSQ3      PISTA ESPECIFICADA ES MAYOR
C090 86 7B                LDA A  #$OCMND    CARGUE UN COMANDO "STEP OUT"
C092 20 02                BRA    BUSQ4      ENVIELO A WD
C094 86 5B      BUSQ3     LDA A  #$ICMND    CARGUE UN COMANDO "STEP IN"
C096 8D 56      BUSQ4     BSR    MCSP       MANDE EL COMANDO
C098 C4 10                AND B  #MSCERSQ   ENCUENTRE ERRORES
C09A 27 85                BEQ    DEMORA     ESPERE 10 mSEG SI NO HAY ERROR
C09C 32                   PUL A             DEJE EL 'STACK' SIN CAMBIOS
C09D 39         BUSQ5     RTS
                *
                * DTARG
                *
                * APUNTE CON PA0 Y PA1 A "DATA REGISTER" DE WD
                *
C09E F6 80 04   DTARG     LDA B  DRA        LEA EL CONTENIDO ANTERIOR
C0A1 CA 03                ORA B  #$03       PONGA PA0 Y PA1 EN 1L
C0A3 F7 80 04   DTARG2    STA B  DRA        *ALMACENE EL RESULTADO
C0A6 39                   RTS
                *
                * CMRG
                *
                * APUNTE A "STATUS REGISTER" Y "COMMAND REGISTER DE WD
                * E INHABILITE FLIP-FLOP DE LECTURA
                *
C0A7 F6 80 04   CMRG      LDA B  DRA        LEA EL CONTENIDO ANTERIOR
C0AA C4 7C                AND B  #$7C       PONGA PA0,PA1,Y PA7 EN OL
C0AC 20 F5                BRA    DTARG2     ALMACENE EL RESULTADO
                *
                * MCMD
                *
                * MANDE COMANDO O DATO A WD
                *
C0AE 8D 59      MCMD      BSR    SALID      PROGRAME LADO B DE PIA COMO SALIDAS
C0B0 B7 80 06             STA A  DRB        MANDE EL COMANDO O DATO
                *
                * ENTR
                *
                * PROGRAME PB0-PB7 COMO ENTRADAS
                *
C0B3 C6 28      ENTR      LDA B  #$28       OBTENGA ACESO A DDRB
C0B5 F7 80 07             STA B  CRB
C0B8 7F 80 06             CLR    DDRB       PROGRAME LADO B COMO ENTRADAS
C0BB C6 2C      ENTR2     LDA B  #$2C       OBTENGA ACESO A DRB
C0BD F7 80 07             STA B  CRB
C0C0 39                   RTS
                *
                * ESCRIB
                *
```

```
                    * ESCRIBA UN SECTOR
                    *
 C0C1 8D B3         ESCRIB  BSR     BUSQ        BUSQUE LA PISTA ESPECIFICADA
 C0C3 86 AC                 LDA A   #WTCMND     CARGUE UN COMANDO DE ESCRITURA
 C0C5 0F            ESCRI2  SEI                 INHABILITE INTERUPCIONES
 C0C6 8D E6                 BSR     MCMD        MANDE EL COMANDO
 C0C8 8D D4                 BSR     DTARG       APUNTE A "DATA REGISTER"
 C0CA 8D 3D                 BSR     SALID       PROGRAME LADO B COMO SALIDAS
 C0CC 86 C0                 LDA A   #$C0        MASCARA PARA IRQ O DRQ
 C0CE E6 00                 LDA B   0,X         TENGA LISTO EL PRIMER DATO
 C0D0 B5 80 05      ESCRI3  BIT A   CRA         DRQ O IRQ DE WD?
 C0D3 27 FB                 BEQ     ESCRI3      NO, TRATE NUEVAMENTE
 C0D5 2B 0B                 BMI     ESCRI4      IRQ?
 C0D7 F7 80 06              STA B   DRB         ES DRQ, ESCRIBA UN DATO
 C0DA F6 80 04              LDA B   DRA         BORRE LA BANDERA
 C0DD 08                    INX                 APUNTE AL SIGUIENTE DATO
 C0DE E6 00                 LDA B   0,X         CARGUELO
 C0E0 20 EE                 BRA     ESCRI3      CONTINUE HASTA TERMINAR
 C0E2 8D CF         ESCRI4  BSR     ENTR        PROGRAME LADO B COMO ENTRADAS
 C0E4 8D C1                 BSR     CMRG        APUNTE A "STATUS REG"
 C0E6 8D 0D                 BSR     LEARG       LEA ESE REGISTRO
 C0E8 C5 DC                 BIT B   #MSCESC     ENCUENTRE ERRORES
 C0EA 0E                    CLI                 HABILITE INTERUPCIONES
 C0EB 39                    RTS
                    *
                    * ESTE ES UN ENLACE A MCMD
                    *
 C0EC 20 C0         EMCMD   BRA     MCMD
                    *
                    * MCSP
                    *
                    * MANDE UN COMANDO Y ESPERE QUE SE LO TERMINE
                    *
 C0EE 8D BE         MCSP    BSR     MCMD        MANDE EL COMANDO
 C0F0 B6 80 05      MCSP2   LDA A   CRA         IRQ DE WD?
 C0F3 2C FB                 BGE     MCSP2       PRUEBE NUEVAMENTE
                    *
                    * LEARG
                    *
                    * LEA UN REGISTRO DE WD
                    *
 C0F5 B6 80 04      LEARG   LDA A   DRA         LEA EL CONTENIDO ANTERIOR
 C0F8 84 BF                 AND A   #$BF        PONGA PA6 EN 0
 C0FA B7 80 04              STA A   DRA
 C0FD F6 80 06              LDA B   DRB         LEA EL REGISTRO
 C100 8A 40                 ORA A   #$40        PONGA PA6 EN 1
 C102 B7 80 04              STA A   DRA
 C105 B6 80 04              LDA A   DRA         BORRE BANDERA IRQ SI EXISTE
 C108 39                    RTS
                    *
                    * SALID
                    *
                    * PROGRAME PB0-PB7 COMO SALIDAS
                    *
```

```
C109 C6 28      SALID   LDA B  #$28      OBTENGA ACESO A DDRB
C10B F7 80 07           STA B  CRB
C10E 7A 80 06           DEC    DDRB     PROGRAME LADO B COMO SALIDAS
C111 20 A8              BRA    ENTR2    OBTENGA ACESO A DRB
                *
                * DISC
                *
                * SELECCIONE EL DISCO ESPECIFICADO
                *
C113 A6 03      DISC    LDA A  3,X      ENCUENTRE EL NO DE DISCO
C115 36                 PSH A           GUARDELO
C116 81 03              CMP A  #$3      ASEGURESE DE QUE ES MENOR QUE 4
C118 23 05              BLS    DISC2    OK
C11A C6 0F              LDA B  #$0F     PONGA CODIGO DE ERROR
C11C 32                 PUL A           DEJE EL 'STACK' SIN CAMBIOS
C11D 0D                 SEC             PONGA BANDERA
C11E 39                 RTS
C11F 8D 48      DISC2   BSR    ENCPIS   X APUNTA A REG DE PISTA DISCO ANT.
C121 7C 80 04           INC    DRA      *APUNTE A "TRACK REG"
C124 8D CF              BSR    LEARG    LEA LA PISTA DEL DISCO ANTERIOR
C126 E7 00              STA B  0,X      GUARDE EL NUMERO
C128 84 CF              AND A  #$CF     SELECCIONE EL DISCO 0
C12A B7 80 04           STA A  DRA
C12D 32                 PUL A           RECOBRE NUMERO DE DISCO DESEADO
C12E B7 A0 31           STA A  DISACT   SELECCIONE NUEVO DISCO
C131 48                 ASL A           POSICIONE LOS BITS PARA PIA
C132 48                 ASL A
C133 48                 ASL A
C134 48                 ASL A
C135 BA 80 04           ORA A  DRA      SELECCIONE EL DISCO
C138 B7 80 04           STA A  DRA
C13B 8D 2C              BSR    ENCPIS   APUNTE X A REG DE PISTA DISCO ACTUAL
C13D A6 00              LDA A  0,X      ENCUENTRE EN QUE PISTA ESTA
C13F 8D AB              BSR    EMCMD    PONGLA EN "TRACK REG" DE WD
C141 7A 80 04           DEC    DRA      *APUNTE A "STATUS REG"
C144 5F                 CLR B           INDIQUE QUE NO HAY ERRORES
C145 39                 RTS
                *
                * VERIF
                *
                * VERIFIQUE EL ULTIMO SECTOR ESCRITO
                *
C146 86 8C      VERIF   LDA A  #RDCMND  CARGUE UN COMANDO DE LECTURA
C148 0F         VERIF2  SEI             INHABLITE INTERRUPCIONES
C149 8D A3              BSR    MCSP     MANDE EL COMANDO
C14B C5 98              BIT B  #MSCVER  ERRORES?
C14D 0E                 CLI             HABILITE INTERRUPCIONES
C14E 39                 RTS
                *
                * REGR
                *
                * REGRESE LA CABEZA A LA PISTA 00
                *
C14F 8D C2      REGR    BSR    DISC     CAMBIE A DISCO ESPECIFICADO
```

```
C151 86 0B      REGR1   LDA A  #RSCMND   CARGUE EL COMANDO
C153 8D 99              BSR    MCSP      MANDELO
C155 C5 88              BIT B  #MSCRST   ERRORES?
C157 39                 RTS
                *
                * CHKRDY
                *
                * CHEQUEE QUE EL DISCO ESTE LISTO
                *
C158 8D 9B      CHKRDY  BSR    LEARG     LEA "STATUS REG"
C15A C5 80              BIT B  #$80      LISTO?
C15C 27 0A              BEQ    CHKRY2
C15E 86 78              LDA A  #SOCMND   CARGUE UN COMANDO "STEP OUT"
C160 8D 8C              BSR    MCSP      MANDELO
C162 86 58              LDA A  #SICMND   CARGUE UN COMANDO "STEP IN"
C164 8D 88              BSR    MCSP
C166 C5 80              BIT B  #:SCRDY   ERRORES?
C168 39         CHKRY2  RTS
                *
                * ENCPIS
                *
                * APUNTE X AL REGISKO DE PISTA DE DISCO SELECCIONADO
                *
C169 CE A0 32   ENCPIS  LDX    #REGPIS   APUNTE X A REGISTROS DE PISTA
C16C F6 A0 31           LDA B  DISACT    ENCUNTRE NO. DE DISCO ACTUAL
C16F 27 04      ENCPS2  BEQ    ENCPS3    X APUNTE REG DE DISCO ACTUAL
C171 08                 INX              NO, INCREMENTE X
C172 5A                 DEC B
C173 20 FA              BRA    ENCPS2    CONTINUE HASTA ENCONTRARLO
C175 39         ENCPS3  RTS
                *
                * FORMAT
                *
                * CREACION DEL FORMATO
                *
C176 BD C3 DF   FORMAT  JSR    PREGPD    PREGUNTE No. DE DISCO Y DE VOLUMEN
C179 0F                 SEI              INHABILITE INTERRUPCIONES
C17A 7F A0 36           CLR    PISTA     COMIENCE CON LA PISTA 00
C17D CE 20 00           LDX    #MEMPIS   APUNTE AL ESPACIO DE RAM DE TRABAJO
C180 C6 FF              LDA B  #$FF      INICIALICE TODO ESTE ESPACIO CON FF
C182 E7 00      FORM1   STA B  0,X
C184 08                 INX              APUNTE AL SIGUIENTE BYTE
C185 8C 2C 37           CPX    #FINPIS   INICIALIZACION COMPLETA?
C188 26 F8              BNE    FORM1     NO
C18A CE 20 00           LDX    #MEMPIS   REGRESE
C18D 4F                 CLR A            PONGA 6 CEROS
C18E C6 06              LDA B  #$06
C190 8D 77              BSR    ALMAC
C192 86 FC              LDA A  #$FC      CARGUE LA MARCA DE PULSO INDICE
C194 A7 00              STA A  0,X       ALMACENE
C196 CE 20 17   FORM2   LDX    #MEMSEC   APUNTE AL COMIENZO DEL PRIMER SECTO
C199 86 01              LDA A  #$01      COMIENCE DESDE EL SECTOR 01
C19B B7 A0 37   FORM4   STA A  SECTOR
C19E 4F         FORM3   CLR A            ALMACENE 6 CEROS
```

```
C19F C6 06                LDA B  #$6
C1A1 8D 66                BSR    ALMAC
C1A3 86 FE                LDA A  #$FE       MARCA DE BYTES DE IDENTIFICACION
C1A5 8D 62                BSR    ALMAC      ALMACENELA Y APUNTE A SIGUIENTE BY
C1A7 B6 A0 36             LDA A  PISTA      CARGUE EL NUMERO DE PISTA ACTUAL
C1AA 8D 5D                BSR    ALMAC      ALMACENELA
C1AC 6F 00                CLR    0,X        LADO=0
C1AE 08                   INX               APUNTE AL SIGUIENTE BYTE
C1AF B6 A0 37             LDA A  SECTOR     CARGUE EL NUMERO DEL SECTOR ACTUAL
C1B2 8D 55                BSR    ALMAC
C1B4 86 01                LDA A  #1         INDICADOR DE LONGITUD DE SECTOR
C1B6 8D 51                BSR    ALMAC
C1B8 86 F7                LDA A  #$F7       INDICADOR DE ESCRITURA DE 2 CRC'S
C1BA 8D 4D                BSR    ALMAC
C1BC 86 FF                LDA A  #$FF       ESCRIBA 11 BYTES CON FF
C1BE C6 0B                LDA B  #11
C1C0 8D 47                BSR    ALMAC
C1C2 4F                   CLR A
C1C3 C6 06                LDA B  #6         ESCRIBA 6 CEROS
C1C5 8D 42                BSR    ALMAC
C1C7 86 FB                LDA A  #$FB       CARGUE LA MARCA DE DATOS
C1C9 8D 3E                BSR    ALMAC
C1CB 86 E5                LDA A  #$E5       ESCRIBA 256 DATOS CON EL VALOR E5
C1CD 5F                   CLR B
C1CE 8D 39                BSR    ALMAC
C1D0 86 F7                LDA A  #$F7       INDICADOR DE ESCRITURA DE 2 CRC'S
C1D2 8D 35                BSR    ALMAC
C1D4 86 FF                LDA A  #$FF       ESCRIBA 14 BYTES CON FF
C1D6 C6 0E                LDA B  #14
C1D8 8D 2F                BSR    ALMAC
C1DA 7C A0 37             INC    SECTOR     CONTINUE CON EL SIGUIENTE SECTOR
C1DD 7C A0 37             INC    SECTOR
C1E0 B6 A0 37             LDA A  SECTOR     ENCUENTRE EN QUE SECTOR ESTA
C1E3 81 0C                CMP A  #SECMAX    SE HAN ESCRITO TODOS LOS SECTORES?
C1E5 27 08                BEQ    FORM5      SI, ESCRIBA UNA PISTA EN EL DISCO
C1E7 81 0B                CMP A  #SECMXI    TERMINO CON LOS SECTORES IMPARES?
C1E9 26 B3                BNE    FORM3
C1EB 86 02                LDA A  #2         COMIENCE CON LOS SECTORES PARES
C1ED 20 AC                BRA    FORM4
C1EF B6 A0 36   FORM5     LDA A  PISTA      CARGUE EL NUMERO DE PISTA ACTUAL
C1F2 BD C0 76             JSR    BUSQ       BUSQUE ESTA PISTA EN EL DISCO
C1F5 CE 20 00             LDX    #MEMPIS    APUNTE AL ESPACIO DE RAM A GRABARS
C1F8 86 F4                LDA A  #WTKMND    CARGUE COMANDO DE ESCRITURA DE PIS
C1FA BD C0 C5             JSR    ESCRI2     ESCRIBA TODA UNA PISTA
C1FD 7C A0 36             INC    PISTA      CONTINUE CON LA SIGUIENTE PISTA
C200 B6 A0 36             LDA A  PISTA      ENCUENTRE EN PSITA ESTA
C203 81 23                CMP A  #PISMAX    SE HAN ESCRITO TODAS LAS PISTAS?
C205 26 8F                BNE    FORM2      NO
C207 20 08                BRA    VRFDIS     VERIFIQUE TODOS LOS SECTORES
                 *
                 * ALMAC
                 *
                 * ALMACENE B BYTES CON EL VALOR DE A, COMENZANDO DESDE X
                 *
```

```
C209 A7 00      ALMAC   STA A  0,X        ALMACENE EL VALOR DE A
C20B 08                 INX               APUNTE AL SIGUIENTE BYTE
C20C 5A                 DEC B             B = NUMERO DE BYTES POR ALMACENAR
C20D 26 FA              BNE    ALMAC      CONTINUE HASTA TERMINAR
C20F 5C                 INC B             B=1 LISTO PARA ALMACENAR UN BYTE
C210 39                 RTS
                *
                * VRFDIS
                *
                * VERIFIQUE EL FORMATO EN EL DISCO E INICIALICE DIRECTORIO
                *
C211 7F A0 36   VRFDIS  CLR    PISTA      COMIENCE DESDE LA PISTA 00
C214 C6 01      VRFDS1  LDA B  #1         Y DESDE EL SECTOR NUMERO 1
C216 F7 A0 37   VRFDS2  STA B  SECTOR
C219 B6 A0 36           LDA A  PISTA      ENCUENTRE EN QUE PISTA SE ENCUENTRA
C21C CE 20 00           LDX    #MEMPIS    APUNTE AL ESPACIO DE RAM DE TRABAJO
C21F BD C0 00           JSR    DLEA       LEA EL SECTOR DE DISCO
C222 26 4A              BNE    VRFDS9     INTENTE LEER 3 VECES MAS
C224 F6 A0 37   VRFD11  LDA B  SECTOR     CONTINUE CON EL SIGUIENTE SECTOR
C227 5C                 INC B
C228 C1 0B              CMP B  #11        SE HAN LEIDO LOS 10 SECTORES/PISTA?
C22A 26 EA              BNE    VRFDS2     NO, CONTINUE CON EL SIGUIENTE SECTOR
C22C 7C A0 36           INC    PISTA      CONTINUE CON LA SIGUIENTE PISTA
C22F B6 A0 36           LDA A  PISTA      ENCUENTRE LA PISTA ACTUAL
C232 81 23              CMP A  #35        SE HAN LEIDO LAS 35 PISTAS?
C234 26 DE              BNE    VRFDS1     NO, ENTONCES LEA ESTA PISTA
C236 4F                 CLR A             INICIALICE EL DIRECTORIO EN PISTA 00
C237 5F                 CLR B             ALMACENE 512 BYTES CON 00 EN RAM
C238 CE 20 00           LDX    #MEMPIS
C23B BD CC              BSR    ALMAC
C23D 5F                 CLR B
C23E BD C9              BSR    ALMAC
C240 CE 20 00   VRFDS3  LDX    #MEMPIS    APUNTE AL DIRECTORIO EN RAM
C243 BD C0 03           JSR    DESCR      ESCRIBA 256 BYTES EN SECTOR 1
C246 26 25              BNE    VRFDS8     REGRESE SI HAY ERROR DE ESCRITURA
C248 7F A0 3B           CLR    INTENT     INICIALICE EL CONTADOR DE INTENTOS
C24B BD C0 06   VRFDS4  JSR    DVERIF     VERIFIQUE EL SECTOR ESCRITO
C24E 27 07              BEQ    VRFDS6     NO HAY ERROR
C250 BD C3 4B           JSR    INTEN      INTENTE HASTA TRES VECES
C253 26 F6              BNE    VRFDS4
C255 5D         VRFDS5  TST B             PONGA BANDERA
C256 39                 RTS
C257 4F         VRFDS6  CLR A             PISTA 0
C258 C6 02              LDA B  #2         SECTOR 2
C25A BD C0 03           JSR    DESCR      256 BYTES A PISTA 0 SECTOR 2
C25D 26 0E              BNE    VRFDS8     REGRESE SI HAY ERROR DE ESCRITURA
C25F 7F A0 3B           CLR    INTENT     INICIALICE EL CONTADOR DE INTENTOS
C262 BD C0 06   VRFDS7  JSR    DVERIF     VERIFIQUE EL SECTOR ESCRITO
C265 27 06              BEQ    VRFDS8     REGRESE SI NO HAY ERROR
C267 BD C3 4B           JSR    INTEN      INTENTE HASTA TRES VECES
C26A 26 F6              BNE    VRFDS7
C26C 5D                 TST B             PONGA BANDERA
C26D 39         VRFDS8  RTS
C26E 7F A0 3B   VRFDS9  CLR    INTENT     INICIALICE EL CONTADOR DE INTENTOS
```

```
C271 BD C0 06  VRFD10  JSR    DVERIF    VERIFIQUE EL SECTOR
C274 27 AE             BEQ    VRFD11    CONTINUE SI NO HAY ERROR
C276 BD C3 4B          JSR    INTEN     INTENTE HASTA TRES VECES
C279 26 F6             BNE    VRFD10
C27B 5D                TST B            PONGA BANDERA
C27C 39                RTS
                 *
                 * GRBPR
                 *
                 * SUBRUTINA PARA GRABAR UN PROGRAMA EN EL DISCO
                 *
C27D BD C3 DF  GRBPR   JSR    PREGPD    PREGUNTE POR No. DE PROGR. Y DISCO
C280 BD C4 12          JSR    PREGCF    PREGUNTE POR COMIENZO Y FIN DE PROG
C283 7F A0 3A  GRBPR0  CLR    FIN       INICIALICE LA VARIABLE
C286 BD C3 0C          JSR    DIRINI    LEA DIRECTORIO Y BUSQUE EL PROGRAMA
C289 25 6D             BCS    GRBPR9    ERROR DE LECTURA
C28B 26 07             BNE    GRBPR3    PROGRAMA AUN NO EXISTE
C28D 6F 00     GRBPR2  CLR    0,X       BORRE EL PROGRAMA DEL DIRECTORIO
C28F BD C3 23          JSR    ENCPR1    BUSQUE EL SIGUIENTE SECTOR ASIGNADO
C292 27 F9             BEQ    GRBPR2    CONTINUE HASTA HALLAR EL ULTIMO
C294 BD C3 3F  GRBPR3  JSR    INIVAR    INICIALICE VARIABLES
C297 4F                CLR A            ENCUENTRE UN SECTOR NO UTILIZADO
C298 BD C3 1E          JSR    ENCPRG
C29B 26 63     GRBPR4  BNE    GRBP12    DISCO LLENO
C29D B6 A0 38  GRBPR5  LDA A  NPROG     CARGUE EL NUMERO DE PROGRAMA
C2A0 A7 00             STA A  0,X       PONGALO EN EL DIRECTORIO (RAM)
C2A2 FF A0 42          STX    XDRBYT    GUARDE EL APUNTADOR DE DIRECTORIO
C2A5 CE 22 00          LDX    #RAMSEC   APUNTE AL SECTOR EN RAM
C2A8 7D A0 39          TST    CHK       PRIMER SECTOR A GRABARSE?
C2AB 26 12             BNE    GRBPR6    NO
C2AD 7C A0 39          INC    CHK
C2B0 FE A0 3C          LDX    COMADD    4 PRIMEROS BYTES DEL SECTOR INDICAN
C2B3 FF 22 00          STX    RAMSEC    EL COMIENZO Y FIN DEL PROGRAMA
C2B6 FE A0 3E          LDX    FINADD
C2B9 FF 22 02          STX    RAMSEC+2
C2BC CE 22 04          LDX    #RAMSEC+4 INDIQUE QUE YA HAY 4 BYTES EN RAM
C2BF FF A0 40  GRBPR6  STX    RAMADD
C2C2 FE A0 3C          LDX    COMADD    APUNTE A UN BYTE DE PROGRAMA
C2C5 A6 00             LDA A  0,X       CARGUELO
C2C7 BC A0 3E          CPX    FINADD    ULTIMO BYTE?
C2CA 26 03             BNE    GRBPR7    NO
C2CC 7C A0 3A          INC    FIN
C2CF 08        GRBPR7  INX              APUNTE AL SIGUIENTE BYTE DE PROGRAM
C2D0 FF A0 3C          STX    COMADD    GUARDE EL APUNTADOR
C2D3 FE A0 40          LDX    RAMADD    APUNTE A UN BYTE DE RAM
C2D6 A7 00             STA A  0,X       ALMACENE ALLI EL VALOR LEIDO
C2D8 08                INX              SIGUIENTE BYTE DE RAM
C2D9 BC 23 00          CPX    #FINSEC+1 FIN DE SECTOR EN RAM?
C2DC 26 E1             BNE    GRBPR6    NO, CONTINUE HASTA TERMINAR
C2DE CE 22 00          LDX    #RAMSEC   INICIALICE VARIABLES PARA GRABACION
C2E1 B6 A0 36          LDA A  PISTA
C2E4 F6 A0 37          LDA B  SECTOR
C2E7 BD C0 03          JSR    DESCR     ESCRIBA UN SECTOR EN EL DISCO
C2EA 26 0D             BNE    GRBP10    ERROR DE ESCRITURA
```

```
C2EC 7F A0 3B            CLR     INTENT     INICIALICE CONTADOR DE INTENTOS
C2EF BD C0 06  GRBPR8    JSR     DVERIF     VERIFIQUE EL ULTIMO SECTOR ESCRITO
C2F2 27 06               BEQ     GRBP11     NO HAY ERROR
C2F4 8D 55               BSR     INTEN      INTENTE HASTA TRES VECES
C2F6 26 F7               BNE     GRBPR8
C2F8 5D        GRBPR9    TST B              PONGA BANDERA
C2F9 39        GRBP10    RTS
C2FA 4F        GRBP11    CLR A
C2FB 7D A0 3A            TST     FIN        SE HA GRABADO TODO EL PROGRAMA?
C2FE 27 05               BEQ     GRBP13     NO, CONTIUE HASTA TERMINAR
C300 C6 01     GRBP12    LDA B   #1         ESCRIBA EL DIRECTORIO EN EL DISCO
C302 7E C2 40            JMP     VRFIS3
C305 FE A0 42  GRBP13    LDX     XURBYT     RECOBRE EL APUNTADOR DE DIRECTORIO
C308 8D 19               BSR     ENCPR1     ENCUENTRE UN NUEVO SECTOR VACIO
C30A 20 8F               BRA     GRBPR4     REPITA EL PROCESO
               *
               * DIRINI
               *
               * LEA EL DIRECTORIO, INICIALICE VARIABLES Y VAYA A ENCPR
               *
C30C 8D 46     DIRINI    BSR     LEADIR     LEA EL DIRECTORIO
C30E 27 06               BEQ     DIRIN2     NO HAY ERROR
C310 8D 39               BSR     INTEN      INTENTE HASTA TRES VECES
C312 26 F8               BNE     DIRINI
C314 0D                  SEC                PONGA BANDERA
C315 39                  RTS
C316 8D 27     DIRIN2    BSR     INIVAR     INICIALICE VARIABLES
C318 7F A0 39            CLR     CHK
C31B B6 A0 38            LDA A   NPROG      CARGUE EL NUMERO DE PROGRAMA
               *
               * ENCPRG
               *
               * ENCUENTRE NUMEROS DE PISTA Y SECTOR ASIGNADOS AL PROGRAMA
               *
C31E A1 00     ENCPRG    CMP A   0,X        ES No. DE PROGRAMA IGUAL A DIR(X)?
C320 26 01               BNE     ENCPR1     NO, CONTINUE HASTA ENCONTRARLO
C322 39                  RTS
C323 7C A0 37  ENCPR1    INC     SECTOR     APUNTE AL SIGUIENTE SECTOR
C326 08                  INX                SIGUIENTE BYTE DE DIRECTORIO
C327 F6 A0 37            LDA B   SECTOR     CARGUE EL NUMERO DEL SECTOR
C32A C1 0B               CMP B   #$0B       ES MAYOR QUE 10?
C32C 26 F0               BNE     ENCPRG     NO, VEA SI ESTA ASIGNADO
C32E C6 01               LDA B   #1         REGRESE AL SECTOR No. 1
C330 F7 A0 37            STA B   SECTOR
C333 7C A0 36            INC     PISTA      APUNTE A LA SIGUIENTE PISTA
C336 F6 A0 36            LDA B   PISTA      CARGUE EL NUMERO DE PISTA
C339 C1 23               CMP B   #35        ES MAYOR A 34?
C33B 26 E1               BNE     ENCPRG     NO
C33D 5D                  TST B              PONGA BANDERA
C33E 39                  RTS
               *
               * INIVAR
               *
               * INICIALICE VARIABLES
```

```
                    *
C33F 86 01          INIVAR  LDA A  #1
C341 B7 A0 37               STA A  SECTOR     COMIENCE CON EL SECTOR #1
C344 B7 A0 36               STA A  PISTA      COMIENCE CON LA PISTA #1
C347 CE 20 0A               LDX    #MEMPIS+10 APUNTE A BYTE 11 DE DIRECTORIO
C34A 39                     RTS
                    *
                    * INTEN
                    *
                    * SUBRUTINA PARA PERMITIR HASTA 3 INTENTOS
                    *
C34B 7C A0 3B       INTEN   INC    INTENT     INCREMENTE EL CONTADOR
C34E B6 A0 3B               LDA A  INTENT     CARGUE EL NUMERO DE INTENTOS
C351 81 03                  CMP A  #3         COMPARELO CON 3
C353 39                     RTS
                    *
                    * LEADIR
                    *
                    * LEA EL DIRECTORIO DEL DISCO
                    *
C354 CE 20 00       LEADIR  LDX    #MEMPIS    COMIENZO DEL ESPACIO DE RAM
C357 4F                     CLR A             PISTA 0
C358 C6 01                  LDA B  #1         SECTOR #1
C35A BD C0 00               JSR    DLEA       LEA ESTE SECTOR A RAM
C35D 26 06                  BNE    LEADR1     ERROR DE LECTURA
C35F 4F                     CLR A             PISTA 0
C360 C6 02                  LDA B  #2         SECTOR No. 2
C362 BD C0 00               JSR    DLEA       LEA ESTE SECTOR A CONTINUACION
C365 39             LEADR1  RTS
                    *
                    * BORPRG
                    *
                    * BORRE UN PROGRAMA DEL DISCO (REALMENTE SOLO DEL DIRECTORIO
                    *
C366 8D 77          BORPRG  BSR    PREGPD     PREGUNTE POR No. DE DISCO Y PROGRAM
C368 8D A2          BORPRO  BSR    DIRINI     LEA DIRECTORIO Y BUSQUE EL PROGRAMA
C36A 25 8C                  BCS    GRBPR9     ERROR DE LECTURA
C36C 26 09                  BNE    BORPR2     PROGRAMA NO EXISTE AUN
C36E 6F 00          BORPR1  CLR    0,X        BORRE EL BYTE DEL DIRECTORIO
C370 8D B1                  BSR    ENCPR1     ENCUENTRE EL SIGUIENTE BYTE A BORRA
C372 27 FA                  BEQ    BORPR1     CONTINUE HASTA TERMINAR
C374 4F                     CLR A             GRABE EL NUEVO DIRECTORIO
C375 20 89                  BRA    GRBP12
C377 39             BORPR2  RTS
                    *
C378 7E C2 F8       EGRBP9  JMP    GRBPR9
                    *
                    * LEAPRG
                    *
                    * SUBRUTINA PARA LEER UN PROGRAMA DEL DISCO
                    *
C37B 8D 62          LEAPRG  BSR    PREGPD     PREGUNTE POR No. DE DISCO Y PROGRAM
C37D 8D 80          LEAPRO  BSR    DIRINI     LEA DIRECTORIO Y BUSQUE EL PROGRAMA
C37F 25 F7                  BCS    EGRBP9     ERROR DE LECTURA
```

```
C381 26 5B      LEAPR1  BNE     LEAPR5      PROGRAMA NO EXISTE O FUE MAL GRABADO
C383 7F A0 38           CLR     INTENT      INICIALICE EL CONTADOR DE INTENTOS
C386 FF A0 42           STX     XIRBYT      GUARDE EL APUNTADOR DE DIRECTORIO
C389 CE 22 00   LEAPR2  LDX     #RAMSEC     APUNTE AL SECTOR EN RAM
C38C B6 A0 36           LDA A   PISTA       LEA EL SECTOR DEL DISCO
C38F F6 A0 37           LDA B   SECTOR
C392 BD C0 00           JSR     DLEA
C395 27 06              BEQ     LEAPR3      NO HAY ERROR DE LECTURA
C397 8D B2              BSR     INTEN       INTENTE HASTA 3 VECES
C399 26 EE              BNE     LEAPR2
C39B 5D                 TST B
C39C 39                 RTS
C39D CE 22 00   LEAPR3  LDX     #RAMSEC     APUNTE AL SECTOR EN RAM
C3A0 7D A0 39           TST     CHK         PRIMER SECTOR LEIDO?
C3A3 26 12              BNE     LEAPR4      NO
C3A5 7C A0 39           INC     CHK
C3A8 FE 22 00           LDX     RAMSEC      4 PRIMEROS BYTES DEL SECTOR INDICAN
C3AB FF A0 3C           STX     COMADD      EL COMIENZO Y FIN DEL PROGRAMA
C3AE FE 22 02           LDX     RAMSEC+2
C3B1 FF A0 3E           STX     FINADD
C3B4 CE 22 04           LDX     #RAMSEC+4   SE HAN LEIDO YA 4 BYTES DE RAM
C3B7 FF A0 40   LEAPR4  STX     RAMADD
C3BA A6 00              LDA A   0,X         CARGUE UN BYTE DE RAM
C3BC FE A0 3C           LDX     COMADD      APUNTE AL BYTE DE PROGRAMA
C3BF A7 00              STA A   0,X         ALMACENE ALLI EL BYTE CARGADO
C3C1 BC A0 3E           CPX     FINADD      ULTIMO BYTE?
C3C4 27 18              BEQ     LEAPR5      SI
C3C6 08                 INX                 APUNTE AL SIGUIENTE BYTE DE PROGRAMA
C3C7 FF A0 3C           STX     COMADD      GUARDE EL APUNTADOR
C3CA FE A0 40           LDX     RAMADD      CARGUE EL APUNTADOR DE RAM
C3CD 08                 INX                 APUNTE AL SIGUIENTE BYTE
C3CE 8C 23 00           CPX     #FINSEC+1   FINAL DEL SECTOR EN RAM?
C3D1 26 E4              BNE     LEAPR4      NO, CONTINUE HASTA TERMINAR
C3D3 B6 A0 38           LDA A   NPROG       RECOBRE EL No. DE PROGRAMA
C3D6 FE A0 42           LDX     XIRBYT      RECOBRE EL APUNTADOR DE DIRECTORIO
C3D9 BD C3 23           JSR     ENCPR1      ENCUENTRE OTRO SECTOR CON PROGRAMA
C3DC 20 A3              BRA     LEAPR1      REPITA EL PROCESO
C3DE 39         LEAPR5  RTS
                *
                * PREGPD
                *
                * PREGUNTE POR NUMERO DE PROGRAMA Y DISCO
                *
C3DF 86 01      PREGPD  LDA A   #1          PERMITA EL INGRESO DE UN DATO
C3E1 C6 0D              LDA B   #$0D        PONGA "D" EN LA PANTALLA
C3E3 8D 46              BSR     INGRES      SUBRUTINA DE INGRESO DE DATOS
C3E5 F1 A0 31           CMP B   DISACT      ES # DE DISCO IGUAL AL ACTUAL ?
C3E8 27 15              BEQ     PREGP1      SI
C3EA CE A0 09           LDX     #DISBUF-3   APUNTE AL DATO LEIDO PARA DISC
C3ED BD C0 0F           JSR     DDISC       SELECCIONE EL NUEVO DISCO
C3F0 26 ED              BNE     PREGPD      DISCO SELECCIONADO MAYOR QUE 3
C3F2 4D                 TST A               SE ENCUENTRA EN LA PISTA 00 ?
C3F3 26 0A              BNE     PREGP1      NO, ENTONCES YA FUE UTILIZADO
C3F5 BD C0 0C           JSR     DREGR1      REGRESE A LA PISTA 00
```

```
C3F8 27 05                BEQ     PREGP1     CONTINUE SI NO HAY ERROR
C3FA 31                   INS                REGRESE AL PROGRAMA RETN
C3FB 31                   INS
C3FC 31                   INS
C3FD 31                   INS
C3FE 39                   RTS
C3FF 86 02       PREGP1   LDA A   #2         PERMITA EL INGRESO DE DOS DATOS
C401 7F A0 38             CLR     INTENT     INICIALICE LA VARIABLE
C404 C6 0A                LDA B   #$0A       PONGA "A" EN LA PANTALLA
C406 8D 23                BSR     INGRES     SUBRUTINA DE INGRESO DE DATOS
C408 BD E0 E4             JSR     BLDX       ARME UN NUMERO DE 8 BITS DE DATOS
C40B B6 A0 1E             LDA A   BPADR      CARGUELO
C40E B7 A0 38             STA A   NPROG      ALMACENELO EN NUMERO DE PROGRAMA
C411 39                   RTS
                 *
                 * PREGCF
                 *
                 * PREGUNTE POR COMIENZO Y FIN DE PROGRAMA
                 *
C412 86 04       PREGCF   LDA A   #4         PERMITA EL INGRESO DE CUATRO DATOS
C414 C6 0C                LDA B   #$0C       PONGA "C" EN LA PANTALLA
C416 8D 13                BSR     INGRES     SUBRUTINA DE INGRESO DE DATOS
C418 BD E0 E4             JSR     BLDX       ARME UN NUMERO DE 16 BITS DE DATOS
C41B FF A0 3C             STX     COMADD     GUARDELO COMO COMIENZO DE PROGRAMA
C41E 86 04                LDA A   #4         PERMITA EL INGRESO DE CUATRO DATOS
C420 C6 0F                LDA B   #$0F       PONGA "F" EN LA PANTALLA
C422 8D 07                BSR     INGRES     SUBRUTINA DE INGRESO DE DATOS
C424 BD E0 E4             JSR     BLDX       ARME UN NUMERO DE 16 BITS DE DATOS
C427 FF A0 3E             STX     FINADD     GUARDELO COMO FINAL DE PROGRAMA
C42A 39                   RTS
                 *
                 * INGRES
                 *
                 * SUBRUTINA DE INGRESO DE DATOS
                 * ESCRIBE TAMBIEN UN DATO EN PANTALLA
                 *
C42B B7 A0 39    INGRES   STA A   CHK        INDIQUE QUE SE VAN A INGRESAR DATOS
C42E BD E0 B2             JSR     CLFLG      BORRE LA PANTALLA
C431 CE A0 0B             LDX     #DISBUF-1  APUNTE AL REGISTRO DE PANTALLA -1
C434 E7 01                STA B   1,X        ESCRIBA LA LETRA EN PANTALLA
C436 F6 A0 39             LDA B   CHK        CALCULE LA POSICION DE DIGITO MAXIMO
C439 08          INGRE1   INX
C43A 5A                   DEC B
C43B 26 FC                BNE     INGRE1     CONTINUE HASTA TERMINAR
C43D FF A0 44             STX     XDBCHK     ALMACENE EL RESULTADO
                 *
                 * TECDIS
                 *
                 * SUBRUTINA DE MANEJO DE TECLADO Y DISPLAY
                 *
C440 CE A0 0C    TECDIS   LDX     #DISBUF    APUNTE AL REGISTRO DE PANTALLA
C443 A6 00       TECDS1   LDA A   0,X        TOME EL PRIMER DIGITO
C445 4C                   INC A              ALISTELO PARA DECODIFICARLO
C446 08                   INX                APUNTE AL SIGUIENTE DIGITO
```

```
C447 FF A0 20              STX     XDSBUF      GUARDE EL APUNTADOR
C44A CE E3 C9              LDX     #DIGTBL-1   APUNTE A LA TABLA DE DECODIFICACION
C44D 08            TECDS2  INX
C44E 4A                    DEC A               ENCUENTRE EL CODIGO PARA PANTALLA
C44F 26 FC                 BNE     TECDS2      CONTINUE HASTA ENCONTRARLO
C451 7F 80 22              CLR     SCNREG      BORRE LA PANTALLA
C454 A6 00                 LDA A   0,X         CARGUE EL CODIGO
C456 B7 80 20              STA A   DISREG      MANDELO A LA PANTALLA
C459 B6 A0 1C              LDA A   SCNCNT      CARGUE EL NUMERO DE DIGITO
C45C B7 80 22              STA A   SCNREG      SELECCIONE EL DIGITO
C45F CE 00 4D              LDX     #$40        PREPARESE PARA ESPERAR 1 MS.
C462 BD E0 E0              JSR     DLY1        ESPERE 1 MS.
C465 FE A0 20              LDX     XDSBUF      RECOBRE EL APUNTADOR
C468 8C A0 12              CPX     #DISBUF+6   ULTIMO DIGITO ?
C46B 27 05                 BEQ     TECDS3      SI, VAYA AL TECLADO
C46D 74 A0 1C              LSR     SCNCNT      SELECCIONE EL SIGUIENTE DIGITO
C470 20 D1                 BRA     TECDS1      CONTINUE HASTA EL ULTIMO DIGITO
C472 86 20         TECDS3  LDA A   #$20        INICIALICE EL APUNTADOR DE DIGITO
C474 B7 A0 1C              STA A   SCNCNT
C477 BD E1 2F      TECDS4  JSR     KEYCL       BUSQUE UNA TECLA APLASTADA
C47A 27 C4                 BEQ     TECDIS      NO SE HA APLASTADO UNA TECLA
C47C BD E0 DD              JSR     DLY20       ESPERE 20 MS.
C47F CE 80 20              LDX     #DISREG     APUNTE AL REGISTRO DE PANTALLA/TECL
C482 86 01                 LDA A   #$01        PRIMERA FILA
C484 A7 02                 STA A   2,X
C486 BD E1 3A      TECDS5  JSR     KEYCL1      BUSQUE LA TECLA APLASTADA
C489 26 0A                 BNE     TECDS6      TECLA ENCONTRADA
C48B A6 02                 LDA A   2,X         BORRE BANDERA NMI
C48D 81 20                 CMP A   #$20        ULTIMA FILA ?
C48F 27 AF                 BEQ     TECDIS      SI, VAYA A LA PANTALLA
C491 68 02                 ASL     2,X         SIGUIENTE FILA
C493 20 F1                 BRA     TECDS5      CONTINUE HASTA TERMINAR
C495 5F            TECDS6  CLR B               INICIALICE EL CONTADOR
C496 CE E3 DC              LDX     #KEYTBL     APUNTE A LA TABLA DE TECLAS
C499 A1 00         TECDS7  CMP A   0,X         ENCONTRO LA TECLA EN LA TABLA ?
C49B 27 09                 BEQ     TECDS8      SI, VEA SI ES UN NUMERO O NO
C49D 8C E3 F4              CPX     #KEYTBL+24  FIN DE LA TABLA ?
C4A0 27 9E                 BEQ     TECDIS      SI, VAYA A LA PANTALLA
C4A2 08                    INX                 AVANCE EN LA PANTALLA
C4A3 5C                    INC B
C4A4 20 F3                 BRA     TECDS7      CONTINUE HASTA TERMINAR
C4A6 BD E1 2F      TECDS8  JSR     KEYCL       SE HA LEVANTADO LA TECLA ?
C4A9 26 FB                 BNE     TECDS8      NO
C4AB BD E0 DD              JSR     DLY20       ESPERE 20 MS.
C4AE C1 0F                 CMP B   #$0F        ES UN NUMERO HEXADECIMAL ?
C4B0 2E 17                 BGT     TECD11      NO, DECODIFIQUE EL COMANDO
C4B2 7D A0 39              TST     CHK         SE ESTA PIDIENDO UN NUMERO ?
C4B5 27 89                 BEQ     TECDIS      NO, REGRESE A LA PANTALLA
C4B7 FE A0 1A              LDX     XKEYBF      RECOBRE EL APUNTADOR DE TECLADO
C4BA E7 00                 STA B   0,X         GUARDE EL NUMERO
C4BC BC A0 44              CPX     XDBCHK      ESTAN YA TODOS LOS DIGITOS PEDIDOS
C4BF 27 07                 BEQ     TECD10      SI, REGRESE
C4C1 08                    INX                 APUNTE AL SIGUIENTE DIGITO
C4C2 FF A0 1A              STX     XKEYBF      GUARDE EL APUNTADOR
```

```
C4C5 7E C4 40  TECDS9  JMP    TECD1S    ESPERE AL SIGUIENTE DIGITO
C4C8 39        TECD10  RTS
C4C9 7D A0 39  TECD11  TST    CHK       SE ESTA PIDIENDO UN NUMERO ?
C4CC 27 06             BEQ    TECD12    NO, ENTONCES ES OPCION DE TECLADO
C4CE C1 15             CMP B  #$15      ES LA TECLA E ?
C4D0 27 2A             BEQ    ESCAPE    SI, ESCAPE DE LA OPCION DE TECLADO
C4D2 20 F1             BRA    TECDS9    REGRESE A LA PANTALLA
C4D4 CE C4 BE  TECD12  LDX    #TECD15-32 ENCUENTRE LA OPCION EN LA TABLA
C4D7 08        TECD14  INX
C4D8 08                INX
C4D9 5A                DEC B
C4DA 26 FB             BNE    TECD14
C4DC 6E 00             JMP    0,X       VAYA A LA OPCION SELECCIONADA
C4DE 20 E5     TECD15  BRA    TECDS9    TECLA P
C4E0 20 2B             BRA    LEER      TECLA L
C4E2 20 4E             BRA    NUEDIS    TECLA N
C4E4 20 47             BRA    BORRAR    TECLA V
C4E6 20 06             BRA    EREST     TECLA M
C4E8 20 DB             BRA    TECDS9    TECLA E
C4EA 20 D9             BRA    TECDS9    TECLA R
C4EC 20 31             BRA    GRABAR    TECLA G
               *
C4EE 7E E0 8D  EREST   JMP    RESTAR    ESTE ES UN ENLACE A RESTAR
               *
               * PPRIN
               *
               * PROGRAMA PRINCIPAL PARA EL CONTROL DE DISCOS FLOPPY
               *
C4F1 BD C0 15  PPRIN   JSR    DINIC     INICIALICE PIA Y REGISTROS DE DISCO
C4F4 BD C0 F5          JSR    LEARG     BORRE BANDERA IRQ SI EXISTE
C4F7 BD C0 0C          JSR    DREGR1    PONGA LA CABEZA EN LA PISTA 00
C4FA 26 17             BNE    ERROR     INDIQUE EL ERROR
C4FC 7F A0 39  ESCAPE  CLR    CHK       COMIENCE CON OPCIONES DE TECLADO
C4FF 8E A0 78          LDS    #$A078    INICIALICE EL STACK
C502 BD E0 84          JSR    DISNMI    INHABILITE INTERRUPCIONES NMI
C505 BD E0 B2          JSR    CLFLG     BORRE LA PANTALLA
C508 BD E0 D7          JSR    HDR       ESCRIBA "-" EN LA PANTALLA
C50B 20 B8             BRA    TECDS9    CONTROLE EL TECLADO Y PANTALLA
               *
               * LEER
               *
               * PROGRAMA PARA LEER DATOS O PROGRAMAS DEL DISCO
               *
C50D BD C3 7B  LEER    JSR    LEAPRG    VAYA A LA SUBRUTINA DE LECTURA
C510 26 01     RETN    BNE    ERROR     INDIQUE EL ERROR
C512 5F        RETN1   CLR B            INDIQUE QUE NO HAY ERROR
C513 BD E0 B2  ERROR   JSR    CLFLG     BORRE LA PANTALLA
C516 7F A0 39          CLR    CHK       PREPARECE PARA OPCIONES DE TECLADO
C519 17                TBA              CODIGO (DE ERROR O NO) EN ACC. A
C51A BD E3 1C          JSR    REGST5    PONGA EL CODIGO EN PANTALLA
C51D 20 A6             BRA    TECDS9    CONTROLE EL TECLADO Y PANTALLA
               *
               * GRABAR
               *
```

```
                    * PROGRAMA PARA GRABAR DATOS O PROGRAMAS EN EL DISCO
                    *
C51F BD C2 7D       GRABAR  JSR     GRBPR     VAYA A LA SUBRUTINA DE GRABACION
C522 26 EF                  BNE     ERROR     INDIQUE EL ERROR
C524 C6 23                  LDA B   #$23      CODIGO DE DISCO LLENO
C526 7D A0 3A               TST     FIN       ESTA EL DISCO LLENO ?
C529 27 E8                  BEQ     ERROR     SI, INDIQUE EL ERROR
C52B 20 E5                  BRA     RETN1     INDIQUE QUE NO HAY ERROR
                    *
                    * BORRAR
                    *
                    * PROGRAMA PARA BORRAR UN PROGRAMA DEL DISCO
                    *
C52D BD C3 66       BORRAR  JSR     BORPRG    VAYA A LA SUBRUTINA DE BORRADO
C530 20 DE                  BRA     RETN      CHEQUEE SI HAY ERRORES E INDIQUELO
                    *
                    * NUEDIS
                    *
                    * PROGRAMA PARA DISCOS NUEVOS, CREA EL FORMATO EN ELLOS
                    *
C532 BD C1 76       NUEDIS  JSR     FORMAT    VAYA A LA SUBRUTINA DE FORMATO
C535 20 D9                  BRA     RETN      CHEQUEE SI HAY ERRORES E INDIQUELO
                    *
                            END

NO ERROR(S) DETECTED
```

SYMBOL TABLE:

ALMAC	C209	BLDX	E0E4	BORPRO	C368	BORPR1	C36E	BORPR2	C377
BORPRG	C366	BORRAR	C52D	BPADR	A01E	BUSQ	C076	BUSQ2	C07F
BUSQ3	C094	BUSQ4	C096	BUSQ5	C09D	CHK	A039	CHKRDY	C158
CHKRY2	C168	CLFLG	E0B2	CMRG	C0A7	COMADD	A03C	CRA	8005
CRB	8007	DBUSQ	C01B	DCHK	C012	DDISC	C00F	DDRA	8004
DDRB	8006	DEMOR2	C023	DEMORA	C021	DESCR	C003	DESCR2	C01E
DIGTBL	E3CA	DINIC	C015	DIRIN2	C316	DIRIN1	C30C	DISACT	A031
DISBUF	A00C	DISC	C113	DISC2	C11F	DISNMI	E084	DISREG	8020
DLEA	C000	DLY1	E0E0	DLY20	E0DD	DRA	8004	DRB	8006
DREGR	C009	DREGR1	C00C	DTARG	C09E	DTARG2	C0A3	DVERIF	C006
DWARM	C018	EGRBP9	C378	ELEARG	C074	EMCMD	C0EC	ENCPIS	C169
ENCPR1	C323	ENCPRG	C31E	ENCPS2	C16F	ENCPS3	C175	ENTR	C083
ENTR2	C0BB	EREST	C4EE	ERROR	C513	ESCAPE	C4FC	ESCRI2	C0C5
ESCRI3	C0D0	ESCRI4	C0E2	ESCRIB	C0C1	FIN	A03A	FINADD	A03E
FINPIS	2C37	FINSEC	22FF	FORM1	C182	FORM2	C196	FORM3	C19E
FORM4	C19B	FORM5	C1EF	FORMAT	C176	GRABAR	C51F	GRBP10	C2F9
GRBP11	C2FA	GRBP12	C300	GRBP13	C305	GRBPR	C27D	GRBPR0	C283
GRBPR2	C28D	GRBPR3	C294	GRBPR4	C29B	GRBPR5	C29D	GRBPR6	C2BF
GRBPR7	C2CF	GRBPR8	C2EF	GRBPR9	C2F8	HDR	E0D7	INGRE1	C439
INGRES	C42B	INIC	C02B	INIC2	C030	INIC3	C036	INIVAR	C33F
INTEN	C34B	INTENT	A03B	KEYCL	E12F	KEYCL1	E13A	KEYTBL	E3DC
LEA	C04A	LEA2	C04E	LEA3	C05A	LEA4	C06C	LEADIR	C354
LEADR1	C365	LEAPR0	C37D	LEAPR1	C381	LEAPR2	C389	LEAPR3	C39D
LEAPR4	C3B7	LEAPR5	C3DE	LEAPRG	C37B	LEARG	C0F5	LEER	C50D
MCMD	C0AE	MCSP	C0EE	MCSP2	C0F0	MEMPIS	2000	MLMSEC	2017
MSCDSQ	0010	MSCESC	00DC	MSCLCT	009C	MSCRDY	0080	MSCRST	0088
MSCVER	0098	NPROG	A038	NUEDIS	C532	PISMAX	0023	PISTA	A036
PPRIN	C4F1	PREGCF	C412	PREGP1	C3FF	PREGPD	C3DF	RAM	A031
RAMADD	A040	RAMSEC	2200	RDCMND	008C	REGPIS	A032	REGR	C14F
REGR1	C151	REGST5	E31C	RESTAR	E080	RETN	C510	RETN1	C512
RSCMND	000B	SALID	C109	SCNCNT	A01C	SCNREG	8022	SECMAX	000C
SECMXI	000B	SECTOR	A037	SICMND	0058	SKCMND	0018	SOCMND	0078
TECD10	C4C8	TECD11	C4C9	TECD12	C4D4	TECD14	C4D7	TECD15	C4DE
TECDIS	C440	TECDS1	C443	TECDS2	C44D	TECDS3	C472	TECDS4	C477
TECDS5	C486	TECDS6	C495	TECDS7	C499	TECDS8	C4A6	TECDS9	C4C5
VERIF	C146	VERIF2	C148	VFLAG	A010	VRFD10	C271	VRFD11	C224
VRFDIS	C211	VRFDS1	C214	VRFDS2	C216	VRFDS3	C240	VRFDS4	C24B
VRFDS5	C255	VRFDS6	C257	VRFDS7	C262	VRFDS8	C26D	VRFDS9	C26E
WTCMND	00AC	WTKMND	00F4	XDBCHK	A044	XDRBYT	A042	XDSBUF	A020
XKEYBF	A01A								

C A P I T U L O Q U I N T O

"UTILIZACION Y EJEMPLO DE APLICACION"

5. UTILIZACION Y EJEMPLO DE APLICACION.

5.1. UTILIZACION.

Antes de utilizar el circuito desarrollado en la presente tesis, se deberá verificar las conexiones que van a la placa del microprocesador y a la lectora/escritora. Para ello se ha seguido con la convención utilizada en el equipo MEK6800D2 es decir, si los componentes de una placa están en la parte superior de la misma, los cables de varios conductores, deberán salir hacia abajo de la placa. En caso de duda se deberá estudiar los diagramas de circuitos de las diversas placas.

Si se piensa utilizar una nueva lectora/escritora con el circuito, se deberá estudiar la sección 4.3. y seleccionar las diversas opciones de operación que concuerden con las características de la nueva lectora/escritora.

Una vez concluídos los procesos de verificación anteriormente indicados, se puede proceder a la utilización del equipo completo, para ello se deberán seguir los siguientes pasos:

1.- Encender el equipo MEK6800D2 y aplastar el botón de "RESET" del mismo.

2.- Esperar 10 segundos y encender la lectora/escritora.

3.- Insertar un disco y correr el programa que se en-

cuentra en C4F1 (PPRIN).

4.- Aplastar la tecla correspondiente al proceso que se desea realizar:

- Tecla M: Se la utilizará para regresar al control del monitor, el funcionamiento del mismo no ha sido modificado y por lo tanto se podrá trabajar con él, en la forma usual. Si el usuario desea realizar algún proceso en el disco, deberá regresar al paso 3.
- Tecla L: Lectura de un programa. El usuario será preguntado por el número de lectora/escritora (1 dígito) y por el número de programa (2 dígitos), el programa será leído y aparecerá el código de error o no, en la pantalla (ØØ indicará que no se ha producido un error).
- Tecla G: Grabación de un programa. Los números de lectora/escritora y de programa serán preguntados al usuario, seguidamente se preguntarán las direcciones (4 dígitos) donde comienza y termina el programa, se lo grabará en el disco y el código de error aparecerá en la pantalla.
- Tecla V: Borrado de un programa. Se preguntarán los números de lectora/escritora y de programa, el programa será borrado del directorio y se pondrá el código de error en la pantalla.
- Tecla N: Nuevo disco. El usuario será preguntado por los números de lectora/escritora y de vo-

lúmen del disco (2 dígitos), se creará el formato en el disco y se verificarán todos los sectores del mismo, por último se pondrá el código de error en la pantalla.

- Tecla E: Escape. Esta tecla sólamente será aceptada mientras se esté pidiendo un dato al usuario. Una vez aplastada se regresará al paso 4.

El número de programa puede ser cualquiera comprendido entre Ø1 y FF, el número ØØ ha sido reservado para recuperar el último programa borrado, esto es, si accidentalmente se borra un programa del disco, se lo puede recuperar leyendo el programa ØØ.

Un programa que se graba no deberá ocupar las direcciones de memoria comprendidas entre 2ØØØ - 2FFF pues este espacio es utilizado para el control de la información en el disco. Si no se va a crear el formato en un disco nuevo, se podrán utilizar también las direcciones de memoria 23ØØ - 2FFF.

Debe recordarse que los discos pueden ser protegidos contra escritura por lo que será conveniente tomar esta precaución en aquellos que contengan información importante.

5.2. EJEMPLO DE APLICACION.

El siguiente ejemplo se ha diseñado para programar

EPROMS del tipo 2758 y 2716. El circuito y los programas de soporte no son más que versiones modificadas de un programador de EPROMS 2708 que fuera explicado en una revista de electrónica (26).

5.2.1. EL CIRCUITO.

El circuito utiliza una PIA como elemento de interface que ocupa las direcciones de memoria BØØØ - BØØ3. En el equipo MEK6800D2 se la puede direccionar en estas posiciones de memoria haciendo las siguientes conexiones:

MEK6800D2		PIA
AØ	-	RSØ
A1	-	RS1
A12	-	CS1
$\overline{\text{STACK}}$	-	$\overline{\text{CS2}}$

Una vez conectada la PIA a la placa del microprocesador, se deberá armar el circuito de la figura 21 para completar el programador de EPROMS.

De la figura 21 se deduce que el lado "A" de la PIA se utiliza para la transmisión de datos, por lo que éstas líneas periféricas deberán ser programadas como entradas o salidas según se lea o escriba en la EPROM. El lado "B" se utiliza para sacar las direcciones de memoria a la EPROM en conjunto con los flip-flops tipo D del integrado 74LS75. Produciendo un pulso

positivo en CB2 se almacenan en los flip-flops los 3 bits más significativos de la dirección de memoria, para luego sacar los 8 bits restantes directamente del lado "B" de la PIA.

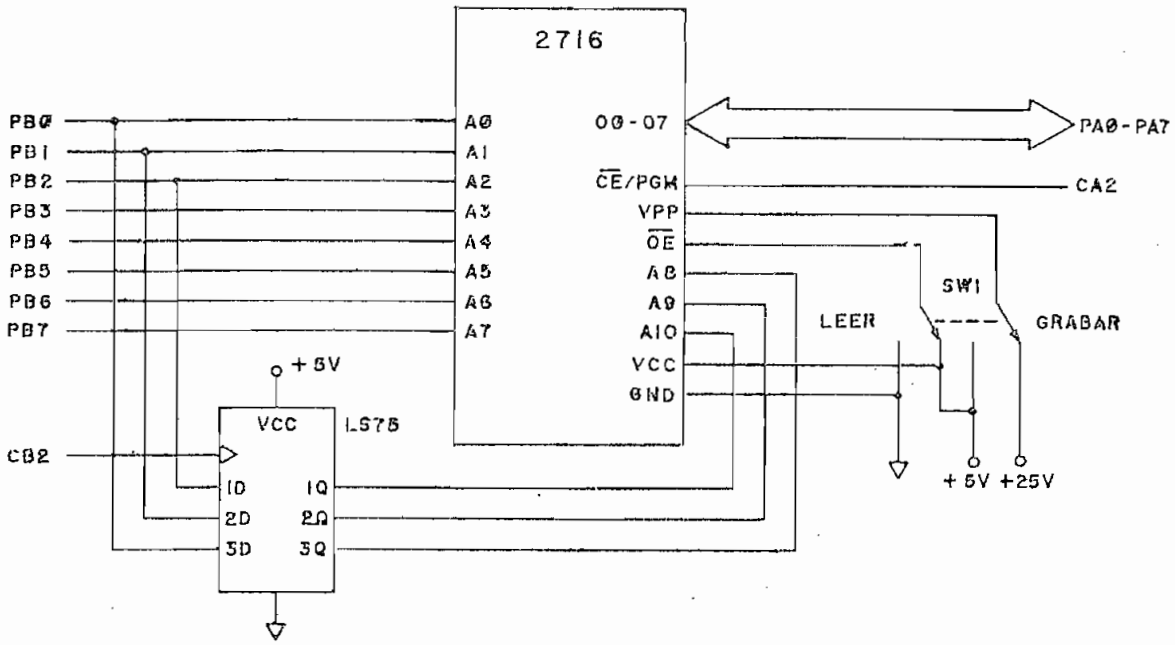


FIGURA 21

PROGRAMADOR DE EPROMS

Por medio de SW1 se enviarán 5 voltios a Vpp y un cero lógico a $\overline{CS}$ para procesos de lectura y 25 voltios a Vpp, y un uno lógico a $\overline{CS}$ para procesos de grabación.

CA2 deberá estar en cero lógico para leer la EPROM, mientras que un pulso positivo de 50 mSeg deberá enviarse por

esta línea para cada dato que se grabe.

5.2.2. LOS PROGRAMAS DE SOPORTE Y SU UTILIZACION.

Los programas de soporte sirven para leer, grabar y verificar el contenido de una EPROM, estos programas se han incluído al final de este capítulo y han sido grabados como programa Ø1 en el disco ØØ.

La función de cada una de las subrutinas utilizadas se podrá comprender leyendo los listados incluídos, por lo que será preferible estudiar la forma de utilización de las subrutinas principales.

5.2.2.1. VBLNK.- ($Ø1ØØ)

Esta subrutina se utiliza para verificar que una EPROM esté "limpia" es decir, que todas sus localidades de memoria contengan el número FF hexadecimal. Al igual que en la subrutina VRFY, al finalizar el proceso se hace un salto al monitor -si no se ha producido un error- o de producirse se regresa a él, por medio de una interrupción SWI, en este último caso se puede analizar la causa del error por medio de las siguientes condiciones de regreso:

X = Dirección del byte errado en la EPROM.

A = Byte leído de la EPROM.

B = Dato esperado.

5.2.2.2. LEA.- ($Ø1Ø2)

LEA se utiliza para transferir el contenido de la EPROM a memoria, antes de utilizar esta subrutina se deberán definir los valores de las siguientes variables: PRMAD ($Ø1FØ); BEGAD ($Ø1F2) y ENDAD ($Ø1F4), las mismas en las que se deberán definir las siguientes direcciones de memoria:

PRMAD = Dirección inicial en la EPROM (usualmente ØØØØ).
BEGAD = Dirección inicial en memoria.
ENDAD = Dirección final de memoria (ENDAD = BEGAD + número de bytes a leer, escribir o verificar).

Una vez efectuado el proceso de lectura, LEA hace un salto directamente al monitor.

5.2.2.3. ESCRIB.- ($Ø1Ø4)

Esta subrutina se utiliza para grabar la EPROM, antes de utilizarla se deberán definir las variables PRMAD, BEGAD y ENDAD como en la sección 5.2.2.2., los datos a escribirse en la EPROM deberán estar entre las direcciones de memoria BEGAD y ENDAD, por último se deberá colocar SW1 en la posición GRABAR y correr el programa.

Al finalizar el proceso, ESCRIB hace un salto directamente al monitor y el usuario deberá colocar SW1 en la posición LEER nuevamente.

5.2.2.4. VRFY.- ($Ø1Ø6)

VRFY se utiliza para verificar lo grabado en la EPROM, comparando los datos almacenados en memoria con aquellos contenidos por la EPROM. Si se utiliza después de las subrutinas LEA o ESCRIBA, no será necesario definir nuevamente las variables: PRMAD, BEGAD y ENDAD.

El usuario podrá determinar si se produjo un error o no, como en la sección 5.2.2.1.

```
                        *
                        * ESTE PROGRAMA PUEDE UTILIZARSE EN CUALQUIER DIRECCION DE
                        * MEMORIA SIN QUE SE REQUIERAN CAMBIOS.
                        *
01F0                            ORG     $01F0
01F0  00  00            PRMAD   FDB     0
01F2  00  00            BEGAD   FDB     0
01F4  00  00            ENDAD   FDB     0
01F6  00  00            PRMAT   FDB     0
01F8  00  00            MEMAD   FDB     0
                        *
                        *
B000                    PAD     EQU     $B000
B001                    PAC     EQU     $B001
B002                    PBD     EQU     $B002
B003                    PBC     EQU     $B003
E08D                    JBUG    EQU     $E08D
                        *
                        *
                        * TABLA DE SALTOS
                        *
0100                            ORG     $0100
0100  20  06            VBLNK   BRA     VBO         VERIFIQUE EPROM BORRADA
0102  20  11            LEA     BRA     RDO         LEA EPROM A MEMORIA
0104  20  67            ESCRIB  BRA     BRNO        PROGRAME LA EPROM
0106  20  52            VRFY    BRA     VRFYL       VERIFIQUE LO PROGRAMADO
                        *
                        * VBO
                        *
                        * VERIFIQUE QUE LA EPROM HAYA SIDO BORRADA
                        *
0108  8D  1F            VBO     BSR     INIT        INICIALICE PIA PARA LECTURA
010A  C6  FF                    LDA B   #$FF        TODA LA EPROM DEBE CONTENER $FF
010C  8D  38            VLP     BSR     VERI        LEA UN DATO DE EPROM Y COMPARELO
010E  8C  08  00                CPX     #$0800      SE HA VERIFICADO TODA LA EPROM ?
0111  26  F9                    BNE     VLP         NO, CONTINUE HASTA TERMINAR
0113  20  42                    BRA     JJBG        REGRESE AL MONITOR
                        *
                        * RDO
                        *
                        * LEA EL CONTENIDO DE LA EPROM A MEMORIA
                        *
0115  8D  12            RDO     BSR     INIT        INICIALICE PIA PARA LECTURA
0117  FF  01  F8        RD1     STX     MEMAD       ALMACENE EL APUNTADOR DE MEM.
011A  8D  35                    BSR     ENTR        TOME UN DATO DE LA EPROM
011C  FE  01  F8                LDX     MEMAD       APUNTE A MEMORIA
011F  A7  00                    STA A   0,X         GUARDE EL DATO EN MEMORIA
0121  8C  01  F4                CPX     ENDAD       ULTIMO DATO ?
0124  27  31                    BEQ     JJBG        SI, REGRESE AL MONITOR
0126  08                        INX                 INCREMENTE EL APUNTADOR DE MEMORIA
0127  20  EE                    BRA     RD1         CONTINUE HASTA TERMINAR
                        *
                        * INIT
```

```
                        *
                        * PULSO
                        * SUBRUTINA PARA GENERAR UN PULSO DE PROGRAMACION
                        *
015C 86 3C              PULSO   LDA A   #$3C        "PRENDA" EL PULSO
015E B7 B0 01                   STA A   PAC
0161 CE 0E FF                   LDX     #$0EFF      PREPARESE PARA MANTENERLO 50 MSes.
0164 09                 PULS1   DEX                 PRODUZCA LA DEMORA
0165 26 FD                      BNE     PULS1       CONTINUE HASTA TERMINAR
0167 86 34                      LDA A   #$34        "APAGUE" EL PULSO
0169 B7 B0 01                   STA A   PAC
016C 39                         RTS
                        *
                        * BRNO
                        *
                        * SUBRUTINA DE PROGRAMACION DE LA EPROM
                        *
016D 8D BA              BRNO    BSR     INIT        INICIALICE LA PIA
016F 86 30                      LDA A   #$30        PROGRAME LADO A DE PIA, SALIDAS
0171 B7 B0 01                   STA A   PAC
0174 7A B0 00                   DEC     PAD
0177 86 34                      LDA A   #$34
0179 B7 B0 01                   STA A   PAC
017C FF 01 F8           BRN1    STX     MEMAD       ALMACENE EL APUNTADOR DE MEMORIA
017F 8D 29                      BSR     ADROUT      PONGA LA DIRECCION PARA LA EPROM
0181 FE 01 F8                   LDX     MEMAD       RECOBRE EL APUNTADOR DE MEMORIA
0184 A6 00                      LDA A   0,X         CARGUE UN BYTE DE MEMORIA
0186 B7 B0 00                   STA A   PAD         MANDELO A LA EPROM
0189 8D D1                      BSR     PULSO       MANDE UN PULSO DE PROGRAMACION
018B FE 01 F8                   LDX     MEMAD       RECOBRE EL APUNTADOR DE MEMORIA
018E BC 01 F4                   CPX     ENDAD       SE HA PROGRAMADO TODO LO PEDIDO?
0191 27 C4                      BEQ     JJBG        SI, REGRESE AL MONITOR
0193 08                         INX                 APUNTE AL SIGUIENTE BYTE DE MEM.
0194 20 E6                      BRA     BRN1        CONTINUE HASTA TERMINAR
                        *
                        * VFYO
                        *
                        * SUBRUTINA PARA VERIFICAR QUE LO ESCRITO EN LA EPROM SEA
                        * IGUAL A LO QUE SE TIENE EN MEMORIA
                        *
0196 8D 91              VFYO    BSR     INIT        INICIALICE PIA PARA LECTURA
0198 FF 01 F8           VFY1    STX     MEMAD       ALMACENE EL APUNTADOR DE MEMORIA
019B E6 00                      LDA B   0,X         TOME UN BYTE DE MEMORIA
019D 8D AA                      BSR     VERI        COMPARELO CON UNO DE LA EPROM
019F FE 01 F8                   LDX     MEMAD       CARGUE EL APUNTADOR DE MEMORIA
01A2 BC 01 F4                   CPX     ENDAD       SE HA VERIFICADO TODO LO PEDIDO?
01A5 27 B0                      BEQ     JJBG        SI, REGRESE AL MONITOR
01A7 08                         INX                 CONTINUE HASTA TERMINAR
01A8 20 EE                      BRA     VFY1
                        *
                        * ADROUT
                        *
                        * SUBRUTINA PARA SACAR UNA DIRECCION A LA EPROM
                        *
```

```
                    * PULSO
                    * SUBRUTINA PARA GENERAR UN PULSO DE PROGRAMACION
                    *
015C 86 3C          PULSO   LDA A  #$3C      "PRENDA" EL PULSO
015E B7 B0 01               STA A  PAC
0161 CE 0E FF               LDX    #$0EFF    PREPARESE PARA MANTENERLO 50 MSes
0164 09             PULS1   DEX              PRODUZCA LA DEMORA
0165 26 FD                  BNE    PULS1     CONTINUE HASTA TERMINAR
0167 86 34                  LDA A  #$34      "APAGUE" EL PULSO
0169 B7 B0 01               STA A  PAC
016C 39                     RTS
                    *
                    * BRNO
                    *
                    * SUBRUTINA DE PROGRAMACION DE LA EPROM
                    *
016D 8D BA          BRNO    BSR    INIT      INICIALICE LA PIA
016F 86 30                  LDA A  #$30      PROGRAME LADO A DE PIA, SALIDAS
0171 B7 B0 01               STA A  PAC
0174 7A B0 00               DEC    PAD
0177 86 34                  LDA A  #$34
0179 B7 B0 01               STA A  PAC
017C FF 01 F8       BRN1    STX    MEMAD     ALMACENE EL APUNTADOR DE MEMORIA
017F 8D 29                  BSR    ADROUT    PONGA LA DIRECCION PARA LA EPROM
0181 FE 01 F8               LDX    MEMAD     RECUERDE EL APUNTADOR DE MEMORIA
0184 A6 00                  LDA A  0,X       CARGUE UN BYTE DE MEMORIA
0186 B7 B0 00               STA A  PAD       MANDELO A LA EPROM
0189 8D D1                  BSR    PULSO     MANDE UN PULSO DE PROGRAMACION
018B FE 01 F8               LDX    MEMAD     RECUERDE EL APUNTADOR DE MEMORIA
018E BC 01 F4               CPX    ENDAD     SE HA PROGRAMADO TODO LO PEDIDO?
0191 27 C4                  BEQ    JJBG      SI, REGRESE AL MONITOR
0193 08                     INX              APUNTE AL SIGUIENTE BYTE DE MEM.
0194 20 E6                  BRA    BRN1      CONTINUE HASTA TERMINAR
                    *
                    * VFYO
                    *
                    * SUBRUTINA PARA VERIFICAR QUE LO ESCRITO EN LA EPROM SEA
                    * IGUAL A LO QUE SE TIENE EN MEMORIA
                    *
0196 8D 91          VFYO    BSR    INIT      INICIALICE PIA PARA LECTURA
0198 FF 01 F8       VFY1    STX    MEMAD     ALMACENE EL APUNTADOR DE MEMORIA
019B E6 00                  LDA B  0,X       TOME UN BYTE DE MEMORIA
019D 8D AA                  BSR    VERI      COMPARELO CON UNO DE LA EPROM
019F FE 01 F8               LDX    MEMAD     CARGUE EL APUNTADOR DE MEMORIA
01A2 BC 01 F4               CPX    ENDAD     SE HA VERIFICADO TODO LO PEDIDO?
01A5 27 B0                  BEQ    JJBG      SI, REGRESE AL MONITOR
01A7 08                     INX              CONTINUE HASTA TERMINAR
01A8 20 EE                  BRA    VFY1
                    *
                    * ADROUT
                    *
                    * SUBRUTINA PARA SACAR UNA DIRECCION A LA EPROM
                    *
01AA CE B0 00       ADROUT  LDX    #PAD      APUNTE A LA PIA
```

```
01AD B6 01 F6        LDA A  PRMAT      MSB DE DIRECCION DE LA EPROM
01B0 A7 02           STA A  2,X        MANDELO AL "LATCH"
01B2 86 3C           LDA A  #$3C       MANDE UN PULSO DE ALMACENAMIENTO
01B4 A7 03           STA A  3,X
01B6 86 34           LDA A  #$34       "APAGUE" EL PULSO
01B8 A7 03           STA A  3,X
01BA B6 01 F7        LDA A  PRMAT+1    LSB DE DIRECCION DE LA EPROM
01BD A7 02           STA A  2,X        MANDELO A LA EPROM
01BF FE 01 F6        LDX    PRMAT      CARGUE LA DIRECCION COMPLETA
01C2 08              INX               INCREMENTELA
01C3 FF 01 F6        STX    PRMAT
01C6 39              RTS
                     END
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADROUT	01AA	BEGAD	01F2	BRNO	016D	BRN1	017C	ENDAD	01F4
ENTR	0151	ESCRIB	0104	INIT	0129	JBUG	E08D	JJBG	0157
LEA	0102	MEMAD	01F8	PAC	B001	PAD	B000	PBC	B003
PBD	B002	PRMAD	01F0	PRMAT	01F6	PULS0	015C	PULS1	0164
RDO	0115	RD1	0117	VB0	0108	VBLNK	0100	VER1	0149
VFY0	0196	VFY1	0198	VLP	010C	VRFY	0106	VRFYL	015A
VRT	0150								

COMENTARIOS Y CONCLUSIONES

Una vez armado el circuito desarrollado en la presente tesis, se pudo constatar que el integrado 74S124 (U4) producía mucho ruido, interfiriendo con la operación normal del circuito PLL separador de datos, esto hizo que se cambie C14 por un condensador de tantalio de 4.7 μF.

Otro error se encontró en las notas de aplicación del integrado FD1797-02 y es el de que los pulsos en la señal leída del disco deben tener una duración de 400 nSeg. para discos de 5¼" y no de 200 nSeg. como se indica en estas notas. El error se ha corregido ajustando el potenciómetro P4.

Originalmente la subrutina BUSQ fue diseñada para localizar una pista en el disco -por programa- con el objeto de producir una demora de 10 mSeg. al pasar de una pista a otra pero, al existir la posibilidad de utilizar el circuito con una lectora/escritora más rápida que al no necesitar de esta demora, pemita la utilización del comando "SEEK" propio del controlador para localizar una pista en el disco, se ha creído conveniente presentar aquí las modificaciones que se deberían hacer en dicho caso, consiguiendo con ello que la subrutina BUSQ sea más corta, rápida y confiable:

1.- Eliminar la subrutina DEMORA (Que forma parte de BUSQ)

2.- Modificar la subrutina BUSQ como se indica a continuación:

```
CØ8D   BD CØ 9E   JSR      DTARG    APUNTE A "DATA REGISTER"
CØ9Ø   BD CØ AE   JSR      MCMD     ALMACENE EL NUMERO DE PISTA
CØ93   BD CØ A7   JSR      CMRG     APUNTE A "COMMAND REGISTER"
CØ96   86 1B      LDA A    #SKCMND  CARGUE UN COMANDO "SEEK"
CØ98   BD CØ EE   JSR      MCSP     MANDELO Y ESPERE TERMINARLO
CØ9B   39         RTS
```

Puesto que el equipo MEK6800D2 no dispone más que de 512 bytes de memoria mientras que el controlador de discos floppy necesita de alrededor de 4 K bytes para la creación del formato en el disco, se ha ampliado la memoria del equipo a 32 K utilizando para ello una placa producida por la compañía DIGITAL RESEARCH COMPUTERS que deberá ser utilizada mientras se concluye una tesis que se está desarrollando para el objeto.

Por último cabe anotar que el sistema terminado ha sido probado extensamente, pudiendo comprobar sus excelentes características de velocidad, confiabilidad y facilidad de manejo, superando -en la mayoría de los casos- las metas propuestas para la presente tesis.

A P E N D I C E 1

"ESPECIFICACIONES DE LA FAMILIA DE CONTROLADORES FD179X-02"

FD 179X-02 Floppy Disk Formatter/Controller Family

MAY 1980

FEATURES

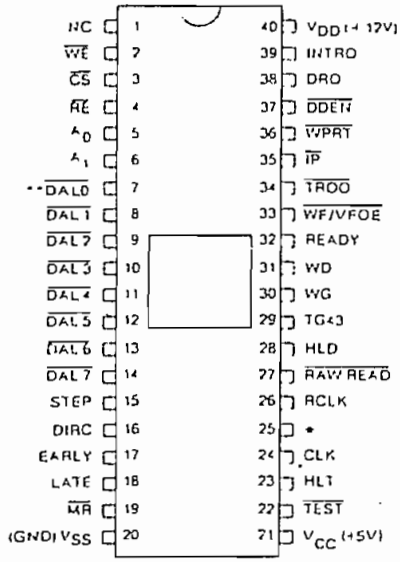
- TWO VFO CONTROL SIGNALS
- SOFT SECTOR FORMAT COMPATIBILITY
- AUTOMATIC TRACK SEEK WITH VERIFICATION
- ACCOMMODATES SINGLE AND DOUBLE DENSITY FORMATS
  IBM 3740 Single Density (FM)
  IBM System 34 Double Density (MFM)
- READ MODE
  Single/Multiple Sector Read with Automatic Search or Entire Track Read
  Selectable 128 Byte or Variable length Sector
- WRITE MODE
  Single/Multiple Sector Write with Automatic Sector Search
  Entire Track Write for Diskette Formatting
- SYSTEM COMPATIBILITY
  Double Buffering of Data 8 Bit Bi-Directional Bus for Data, Control and Status
  DMA or Programmed Data Transfers
  All Inputs and Outputs are TTL Compatible
  On-Chip Track and Sector Registers/Comprehensive Status Information
- PROGRAMMABLE CONTROLS
  Selectable Track to Track Stepping Time
  Side Select Compare
- WRITE PRECOMPENSATION
- WINDOW EXTENSION
- INCORPORATES ENCODING/DECODING AND ADDRESS MARK CIRCUITRY
- FD1792/4 IS SINGLE DENSITY ONLY
- FD1795/7 HAS A SIDE SELECT OUTPUT

179X-02 FAMILY CHARACTERISTICS

FEATURES	1791	1793	1795	1797
Single Density (FM)	X	X	X	X
Double Density (MFM)	X	X	X	X
True Data Bus		X		X
Inverted Data Bus	X		X	
Write Precomp	X	X	X	X
Side Selection Output			X	X

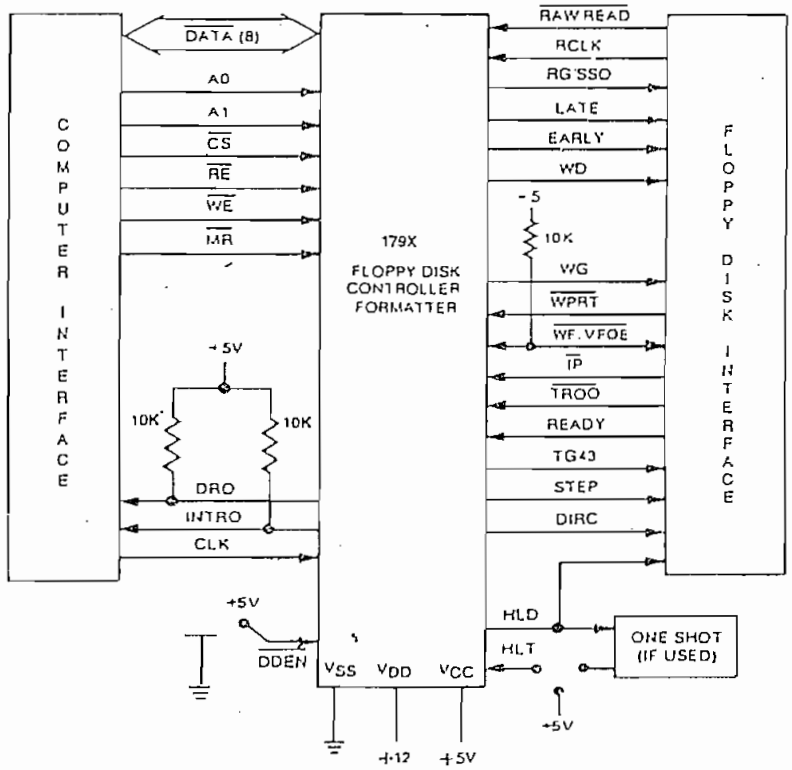
APPLICATIONS

FLOPPY DISK DRIVE INTERFACE
SINGLE OR MULTIPLE DRIVE CONTROLLER/FORMATTER
NEW MINI-FLOPPY CONTROLLER



*1791/3 = RG 1795/7 = SSO
**1793/7 TRUE BUS

PIN CONNECTIONS



FD179X SYSTEM BLOCK DIAGRAM

functions of a Floppy Disk Formatter/Controller in a single chip implementation. The FD179X, which can be considered the end result of both the FD1771 and FD1781 designs, is IBM 3740 compatible in single density mode (FM) and System 34 compatible in Double Density Mode (MFM). The FD179X contains all the features of its predecessor the FD1771, plus the added features necessary to read/write and format a double density diskette. These include address mark detection, FM and MFM encode and decode logic, window extension, and write precompensation. In order to maintain compatibility, the FD1771, FD1781, and FD179X designs were made as close as possible with the computer interface, instruction set, and I/O registers being identical. Also, head load signments vary by only a few pins from any one to another.

The processor interface consists of an 8-bit bidirectional bus for data, status, and control word transfers. The FD179X is set up to operate on a multiplexed bus with other bus-oriented devices.

The FD179X is fabricated in N-channel Silicon Gate MOS technology and is TTL compatible on all inputs and outputs. The 1793 is identical to the 1791 except the DAL lines are TRUE for systems that utilize true data busses.

The 1795/7 has a side select output for controlling double sided drives, and the 1792 and 1794 are "Single Density Only" versions of the 1791 and 1793. On these devices, DDEN must be left open.

PIN OUTS

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
1	NO CONNECTION	NC	Pin 1 is internally connected to a back bias generator and must be left open by the user.
19	$\overline{\text{MASTER RESET}}$	$\overline{\text{MR}}$	A logic low on this input resets the device and loads HEX 03 into the command register. The Not Ready (Status Bit 7) is reset during $\overline{\text{MR}}$ ACTIVE. When $\overline{\text{MR}}$ is brought to a logic high a RESTORE Command is executed, regardless of the state of the Ready signal from the drive. Also, HEX 01 is loaded into sector register.
20	POWER SUPPLIES	V_{SS}	Ground
21		V_{CC}	+5V ±5%
40		V_{DD}	+12V ±5%
COMPUTER INTERFACE:			
2	$\overline{\text{WRITE ENABLE}}$	$\overline{\text{WE}}$	A logic low on this input gates data on the DAL into the selected register when $\overline{\text{CS}}$ is low.
3	$\overline{\text{CHIP SELECT}}$	$\overline{\text{CS}}$	A logic low on this input selects the chip and enables computer communication with the device.
4	$\overline{\text{READ ENABLE}}$	$\overline{\text{RE}}$	A logic low on this input controls the placement of data from a selected register on the DAL when $\overline{\text{CS}}$ is low.
5,6	REGISTER SELECT LINES	A0, A1	These inputs select the register to receive/transfer data on the DAL lines under $\overline{\text{RE}}$ and $\overline{\text{WE}}$ control:

A1	A0	$\overline{\text{RE}}$	$\overline{\text{WE}}$
0	0	Status Reg	Command Reg
0	1	Track Reg	Track Reg
1	0	Sector Reg	Sector Reg
1	1	Data Reg	Data Reg

7-14	$\overline{\text{DATA ACCESS LINES}}$	$\overline{\text{DAL0}}$-$\overline{\text{DAL7}}$	Eight bit inverted Bidirectional bus used for transfer of data, control, and status. This bus is receiver enabled by $\overline{\text{WE}}$ or transmitter enabled by $\overline{\text{RE}}$.
24	CLOCK	CLK	This input requires a free-running square wave clock for internal timing reference, 2 MHz for 8" drives, 1 MHz for mini-drives.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
38	DATA REQUEST	DRQ	This open drain output indicates that the DR contains assembled data in Read operations, or the DR is empty in Write operations. This signal is reset when serviced by the computer through reading or loading the DR in Read or Write operations, respectively. Use 10K pull-up resistor to +5.
39	INTERRUPT REQUEST	INTRQ	This open drain output is set at the completion of any command and is reset when the STATUS register is read or the command register is written to. Use 10K pull-up resistor to +5.
FLOPPY DISK INTERFACE:			
15	STEP	STEP	The step output contains a pulse for each step.
16	DIRECTION	DIRC	Direction Output is active high when stepping in, active low when stepping out.
17	EARLY	EARLY	Indicates that the WRITE DATA pulse occurring while Early is active (high) should be shifted early for write precompensation.
18	LATE	LATE	Indicates that the write data pulse occurring while Late is active (high) should be shifted late for write precompensation.
22	$\overline{\text{TEST}}$	$\overline{\text{TEST}}$	This input is used for testing purposes only and should be tied to +5V or left open by the user unless interfacing to voice coil actuated motors.
23	HEAD LOAD TIMING	HLT	When a logic high is found on the HLT input the head is assumed to be engaged.
25	READ GATE (1791/3)	RG	A high level on this output indicates to the data separator circuitry that a field of zeros (or ones) has been encountered, and is used for synchronization.
25	SIDE SELECT OUTPUT (1795, 1797)	SSO	The logic level of the Side Select Output is directly controlled by the 'S' flag in Type II or III commands. When S = 1, SSO is set to a logic 1. When S = 0, SSO is set to a logic 0. The Side Select Output is only updated at the beginning of a Type II or III command. It is forced to a logic 0 upon a MASTER RESET condition.
26	READ CLOCK	RCLK	A nominal square-wave clock signal derived from the data stream must be provided to this input. Phasing (i.e. RCLK transitions) relative to RAW READ is important but polarity (RCLK high or low) is not.
27	$\overline{\text{RAW READ}}$	$\overline{\text{RAW READ}}$	The data input signal directly from the drive. This input shall be a negative pulse for each recorded flux transition.
28	HEAD LOAD	HLD	The HLD output controls the loading of the Read-Write head against the media.
29	TRACK GREATER THAN 43	TG43	This output informs the drive that the Read/Write head is positioned between tracks 44-76. This output is valid only during Read and Write Commands.
30	WRITE GATE	WG	This output is made valid before writing is to be performed on the diskette.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
31	WRITE DATA	WD	A 250 ns (MFM) or 500 ns (FM) pulse per flux transition. WD contains the unique Address marks as well as data and clock in both FM and MFM formats.
32	READY	READY	This input indicates disk readiness and is sampled for a logic high before Read or Write commands are performed. If Ready is low the Read or Write operation is not performed and an interrupt is generated. Type I operations are performed regardless of the state of Ready. The Ready input appears in inverted format as Status Register bit 7.
33	$\overline{\text{WRITE FAULT}}$ / VFO ENABLE	$\overline{\text{WF}}$/$\overline{\text{VFOE}}$	This is a bi-directional signal used to signify writing faults at the drive, and to enable the external PLO data separator. When WG = 1, Pin 33 functions as a WF input. If WF = 0, any write command will immediately be terminated. When WG = 0, Pin 33 functions as a VFOE output. VFOE will go low during a read operation after the head has loaded and settled (HLT = 1). On the 1795/7, it will remain low until the last bit of the second CRC byte in the ID field. VFOE will then go high until 8 bytes (MFM) or 4 bytes (FM) before the Address Mark. It will then go active until the last bit of the second CRC byte of the Data Field. On the 1791/3, VFOE will remain low until the end of the Data Field.
34	$\overline{\text{TRACK 00}}$	$\overline{\text{TR00}}$	This input informs the FD179X that the Read/Write head is positioned over Track 00.
35	$\overline{\text{INDEX PULSE}}$	$\overline{\text{IP}}$	This input informs the FD179X when the index hole is encountered on the diskette.
36	$\overline{\text{WRITE PROTECT}}$	$\overline{\text{WPRT}}$	This input is sampled whenever a Write Command is received. A logic low terminates the command and sets the Write Protect Status bit.
37	$\overline{\text{DOUBLE DENSITY}}$	$\overline{\text{DDEN}}$	This pin selects either single or double density operation. When $\overline{\text{DDEN}}$ = 0, double density is selected. When $\overline{\text{DDEN}}$ = 1, single density is selected. This line must be left open on the 1792/4

ORGANIZATION

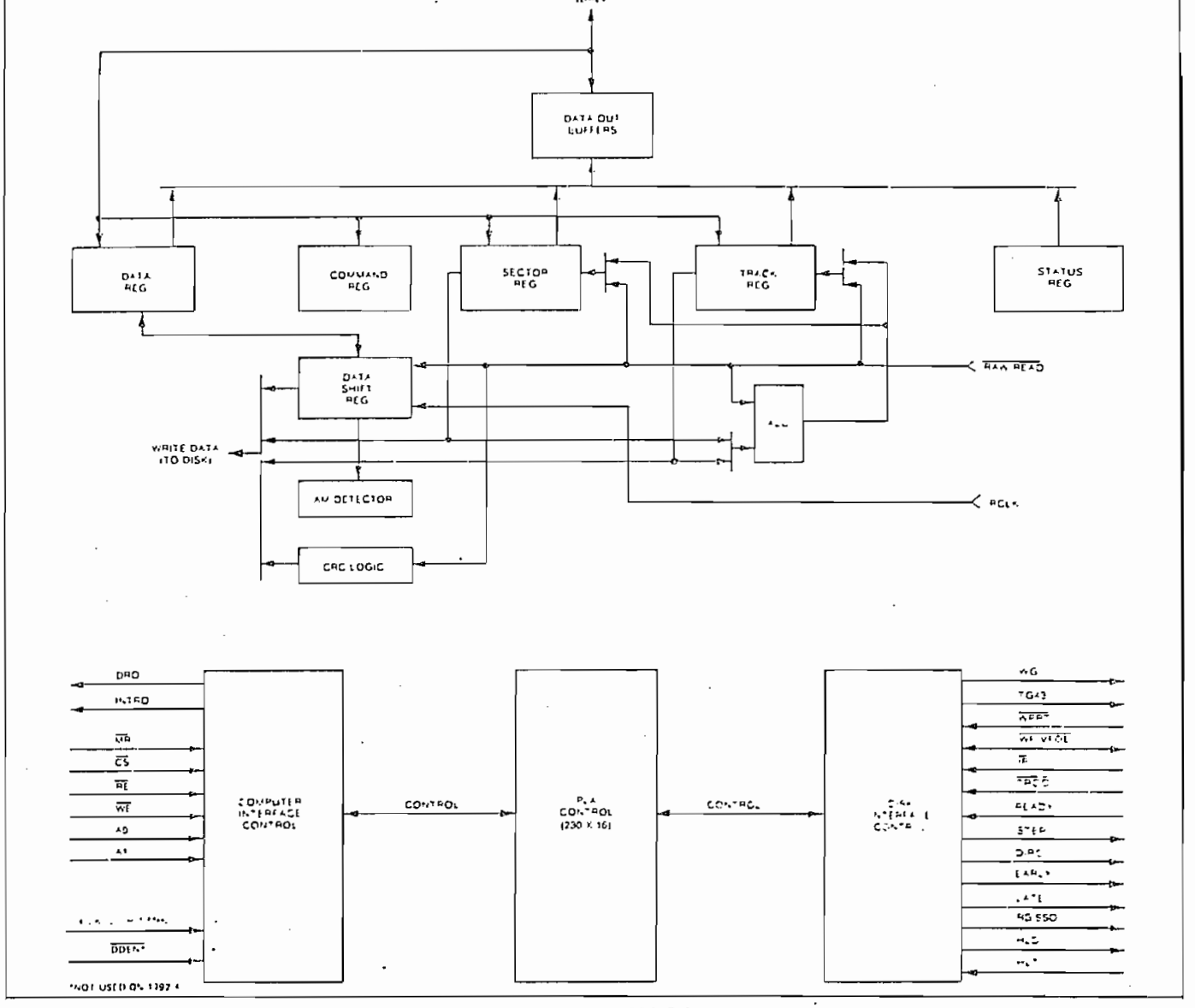
The Floppy Disk Formatter block diagram is illustrated on page 5. The primary sections include the parallel processor interface and the Floppy Disk interface.

Data Shift Register—This 8-bit register assembles serial data from the Read Data input ($\overline{\text{RAW READ}}$) during Read operations and transfers serial data to the Write Data output during Write operations.

Data Register—This 8-bit register is used as a holding register during Disk Read and Write operations. In Disk Read operations the assembled data byte is transferred in parallel to the Data Register from the Data Shift Register. In Disk Write operations information is transferred in parallel from the Data Register to the Data Shift Register.

When executing the Seek command the Data Register holds the address of the desired Track position. This register is loaded from the DAL and gated onto the DAL under processor control.

Track Register—This 8-bit register holds the track number of the current Read/Write head position. It is incremented by one every time the head is stepped in (towards track 76) and decremented by one when the head is stepped out (towards track 00). The contents of the register are compared with the recorded track number in the ID field during disk Read, Write, and Verify operations. The Track Register can be loaded from or transferred to the DAL. This Register should not be loaded when the device is busy.



FD179X BLOCK DIAGRAM

Sector Register (SR)—This 8-bit register holds the address of the desired sector position. The contents of the register are compared with the recorded sector number in the ID field during disk Read or Write operations. The Sector Register contents can be loaded from or transferred to the DAL. This register should not be loaded when the device is busy.

Command Register (CR)—This 8-bit register holds the command presently being executed. This register should not be loaded when the device is busy unless the new command is a force interrupt. The command register can be loaded from the DAL, but not read onto the DAL.

Status Register (STR)—This 8-bit register holds device Status information. The meaning of the Status bits is a function of the type of command previously executed. This register can be read onto the DAL, but not loaded from the DAL.

CRC Logic—This logic is used to check or to generate the 16-bit Cyclic Redundancy Check (CRC). The polynomial is: $G(x) = x^{16} + x^{12} + x^5 + 1$.

The CRC includes all information starting with the address mark and up to the CRC characters. The CRC register is preset to ones prior to data being shifted through the circuit.

Arithmetic/Logic Unit (ALU)—The ALU is a serial comparator, incrementer, and decrementer and is used for register modification and comparisons with the disk recorded ID field.

Timing and Control—All computer and Floppy Disk Interface controls are generated through this logic. The internal device timing is generated from an external crystal clock.

The FD1791/3 has two different modes of operation according to the state of $\overline{DDEN}$. When $\overline{DDEN} = 0$ double density (MFM) is assumed. When $\overline{DDEN} = 1$, single density (FM) is assumed.

AM Detector—The address mark detector detects ID, data and index address marks during read and write operations.

PROCESSOR INTERFACE

The interface to the processor is accomplished through the eight Data Access Lines ($\overline{DAL}$) and associated control signals. The $\overline{DAL}$ are used to transfer Data, Status, and Control words out of, or into the FD179X. The $\overline{DAL}$ are three state buffers that are enabled as output drivers when Chip Select (CS) and Read Enable ($\overline{RE}$) are active (low logic state) or act as input receivers when $\overline{CS}$ and Write Enable ($\overline{WE}$) are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and $\overline{CS}$ is made low. The address bits A1 and A0, combined with the signals $\overline{RE}$ during a Read operation or $\overline{WE}$ during a Write operation are interpreted as selecting the following registers:

A1-A0	READ ($\overline{RE}$)	WRITE ($\overline{WE}$)
0 0	Status Register	Command Register
0 1	Track Register	Track Register
1 0	Sector Register	Sector Register
1 1	Data Register	Data Register

During Direct Memory Access (DMA) types of data transfers between the Data Register of the FD179X and the processor, the Data Request (DRQ) output is used in Data Transfer control. This signal also appears as status bit 1 during Read and Write operations.

On Disk Read operations the Data Request is activated (set high) when an assembled serial input byte is transferred in parallel to the Data Register. This bit is cleared when the Data Register is read by the processor. If the Data Register is read after one or more characters are lost, by having new data transferred into the register prior to processor readout, the Lost Data bit is set in the Status Register. The Read operation continues until the end of sector is reached.

On Disk Write operations the data Request is activated when the Data Register transfers its contents to the Data Shift Register, and requires a new data byte. It is reset when the Data Register is loaded with new data by the processor. If new data is not loaded at the time the next serial byte is required by the Floppy Disk, a byte of zeroes is written on the diskette and the Lost Data bit is set in the Status Register.

At the completion of every command an INTRQ is generated. INTRQ is reset by either reading the status register or by loading the command register with a new command. In addition, INTRQ is generated if a Force Interrupt command condition is met.

FLOPPY DISK INTERFACE

The 179X has two modes of operation according to the state of $\overline{DDEN}$ (Pin 37). When $\overline{DDEN}$ = 1, single density is selected. In either case, the CLK input (Pin 24) is at 2 MHz. However, when interfacing with the mini-floppy, the CLK input is set at 1 MHz for both single density and double density. When the clock is at 2 MHz, the stepping rates of 3, 6, 10, and 15 ms are obtainable. When CLK equals 1 MHz these times are doubled.

HEAD POSITIONING

Five commands cause positioning of the Read-Write head (see Command Section). The period of each positioning step is specified by the r field in bits 1 and 0 of the command word. After the last directional step an additional 15 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. Note that this time doubles to 30 ms for a 1 MHz clock. If $\overline{TEST}$ = 0, there is zero settling time. There is also a 15 ms head settling time if the E flag is set in any Type II or III command.

The rates (shown in Table 1) can be applied to a Step-Direction Motor through the device interface.

Step—A 2 μs (MFM) or 4 μs (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output.

Direction (DIRC)—The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid 12 μs before the first stepping pulse is generated.

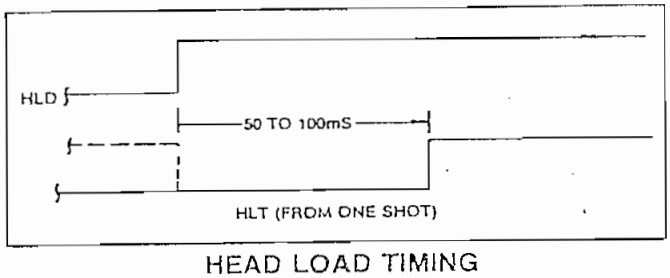
When a Seek, Step or Restore command is executed an optional verification of Read-Write head position can be performed by setting bit 2 (V = 1) in the command word to a logic 1. The verification operation begins at the end of the 15 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. The FD179X must find an ID field with correct track number and correct CRC within 5 revolutions of the media; otherwise the seek error is set and an INTRQ is generated.

Table 1. STEPPING RATES

CLK	2 MHz	2 MHz	1 MHz	1 MHz	2 MHz	1 MHz
$\overline{DDEN}$	0	1	0	1	x	x
R1 R0	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=0	$\overline{TEST}$=0
0 0	3 ms	3 ms	6 ms	6 ms	184μs	368μs
0 1	6 ms	6 ms	12 ms	12 ms	190μs	380μs
1 0	10 ms	10 ms	20 ms	20 ms	198μs	396μs
1 1	15 ms	15 ms	30 ms	30 ms	208μs	416μs

The Head Load (HLD) output controls the movement of the read/write head against the media. HLD is activated at the beginning of a Type I command if the h flag is set (h = 1), at the end of the Type I command if the verify flag (V = 1), or upon receipt of any Type II or III command. Once HLD is active it remains active until either a Type I command is received with (h = 0 and V = 0); or if the FD179X is in an idle state (non-busy) and 15 index pulses have occurred.

which is used for the head engage time. When HLT = 1, the FD179X assumes the head is completely engaged. The head engage time is typically 30 to 100 ms depending on drive. The low to high transition on HLD is typically used to fire a one shot. The output of the one shot is then used for HLT and supplied as an input to the FD179X.



HEAD LOAD TIMING

When both HLD and HLT are true, the FD179X will then read from or write to the media. The "and" of HLD and HLT appears as a status bit in Type I status.

In summary for the Type I commands: if h = 0 and V = 0, HLD is reset. If h = 1 and V = 0, HLD is set at the beginning of the command and HLT is not sampled nor is there an internal 15 ms delay. If h = 0 and V = 1, HLD is set near the end of the command, an internal 15 ms occurs, and the FD179X waits for HLT to be true. If h = 1 and V = 1, HLD is set at the beginning of the command. Near the end of the command, after all the steps have been issued, an internal 15 ms delay occurs and the FD179X then waits for HLT to occur.

For Type II and III commands with E flag off, HLD is made active and HLT is sampled until true. With E flag on, HLD is made active, an internal 15 ms delay occurs and then HLT is sampled until true.

DISK READ OPERATIONS

Sector lengths of 128, 256, 512 or 1024 are obtainable in either FM or MFM formats. For FM, $\overline{\text{DDEN}}$ should be placed to logical "1." For MFM formats, $\overline{\text{DDEN}}$ should be placed to a logical "0." Sector lengths are determined at format time by a special byte in the "ID" field. If this Sector length byte in the ID field is zero, then the sector length is 128 bytes. If 01 then 256 bytes. If 02, then 512 bytes. If 03, then the sector length is 1024 bytes. The number of sectors per track as far as the FD179X is concerned can be from 1 to 255 sectors. The number of tracks as far as the FD179X is concerned is from 0 to 255 tracks. For IBM 3740 compatibility, sector lengths are 128 bytes with 26 sectors per track. For System 34 compatibility (MFM), sector lengths are 256 bytes/sector with 26 sectors/track; or lengths of 1024 bytes/sector with 8 sectors/track. (See Sector Length Table.)

For read operations, the FD179X requires $\overline{\text{RAW READ}}$ Data (Pin 27) signal which is a 250 ns pulse per flux transition and a Read clock (RCLK) signal to indicate flux transition spacings. The RCLK (Pin 26) signal is *provided by some drives* but if not it may be counter techniques. In addition, a Read Gate Signal is provided as an output (Pin 25) which can be used to inform phase lock loops when to acquire synchronization. When reading from the media in FM. RG is made true when 2 bytes of zeroes are detected. The FD179X must find an address mark within the next 10 bytes; otherwise RG is reset and the search for 2 bytes of zeroes begins all over again. If an address mark is found within 10 bytes, RG remains true as long as the FD179X is deriving any useful information from the data stream. Similarly for MFM, RG is made active when 4 bytes of "00" or "FF" are detected. The FD179X must find an address mark within the next 16 bytes, otherwise RG is reset and search resumes.

During read operations (WG = 0), the $\overline{\text{VFOE}}$ (Pin 33) is provided for phase lock loop synchronization. $\overline{\text{VFOE}}$ will go active when:

a) Both HLT and HLD are True
b) Settling Time, if programmed, has expired
c) The 179X is inspecting data off the disk

If $\overline{\text{WF}}/\overline{\text{VFOE}}$ is not used, leave open or tie to a 10K resistor to +5.

DISK WRITE OPERATION

When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the FD179X before the Write Gate signal can be activated.

Writing is inhibited when the $\overline{\text{Write Protect}}$ input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set. The Write Fault input, when activated, signifies a writing fault condition detected in disk drive electronics such as failure to detect write current flow when the Write Gate is activated. On detection of this fault the FD179X terminates the current command, and sets the Write Fault bit (bit 5) in the Status Word. The $\overline{\text{Write Fault}}$ input should be made inactive when the Write Gate output becomes inactive.

For write operations, the FD179X provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of 500 ns pulses in FM ($\overline{\text{DDEN}}$ = 1) and 250 ns pulses in MFM ($\overline{\text{DDEN}}$ = 0). Write Data provides the unique address marks in both formats.

Also during write, two additional signals are provided for write precompensation. These are EARLY (Pin 17) and LATE (Pin 18). EARLY is active true when the WD pulse appearing on (Pin 30) is to be written early. LATE is active true when the WD pulse is to be written LATE. If both EARLY and LATE are low when the WD pulse is present, the WD pulse is to be written at nominal. Since write precompensation values vary from disk manufacturer to disk manufacturer, the actual value is determined by several one shots or delay lines which are located external to the FD179X. The write precompensation signals EARLY and LATE are valid for the duration of WD in both FM and MFM formats.

is received the FD179X samples the Ready input. If this input is logic low the command is not executed and an interrupt is generated. All Type I commands are performed regardless of the state of the Ready input. Also, whenever a Type II or III command is received, the TG43 signal output is updated.

COMMAND DESCRIPTION

The FD179X will accept eleven commands. Command words should only be loaded in the Command Register when the Busy status bit is off (Status bit 0). The one exception is the Force Interrrupt command. Whenever a command is being executed, the Busy status bit is set. When a command is completed, an interrupt is generated and the Busy status bit is reset. The Status Register indicates whether the completed command encountered an error or was fault free. For ease of discussion, commands are divided into four types. Commands and types are summarized in Table 2.

Table 2. COMMAND SUMMARY

		BITS							
TYPE	COMMAND	7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	u	h	V	r_1	r_0
I	Step In	0	1	0	u	h	V	r_1	r_0
I	Step Out	0	1	1	u	h	V	r_1	r_0
II	Read Sector	1	0	0	m	F_2	E	F_1	0
II	Write Sector	1	0	1	m	F_2	E	F_1	a_0
III	Read Address	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrrupt	1	1	0	1	I_3	I_2	I_1	I_0

Note: Bits shown in TRUE form.

Table 3. FLAG SUMMARY

TYPE I COMMANDS

h = Head Load Flag (Bit 3)

h = 1, Load head at beginning
h = 0, Unload head at beginning

V = Verify flag (Bit 2)

V = 1, Verify on destination track
V = 0, No verify

r_1r_0 = Stepping motor rate (Bits 1-0)

Refer to Table 1 for rate summary

u = Update flag (Bit 4)

u = 1, Update Track register
u = 0, No update

TYPE II & III COMMANDS

m = Multiple Record flag (Bit 4)

m = 0, Single Record
m = 1, Multiple Records

a_0 = Data Address Mark (Bit 0)

a_0 = 0, FB (Data Mark)
a_0 = 1, F8 (Deleted Data Mark)

E = 15 ms Delay (2MHz)

E = 1, 15 ms delay

E = 0, no 15 ms delay

(F_2) S = Side Select Flag (1791/3 only)

S = 0, Compare for Side 0
S = 1, Compare for Side 1

(F_1) C = Side Compare Flag (1791/3 only)

C = 0, disable side select compare
C = 1, enable side select compare

(F_1) S = Side Select Flag

(Bit 1, 1795/7 only)

S = 0 Update SSO to 0

S = 1 Update SSO to 1

(F_2) b = Sector Length Flag

(Bit 3, 1975/7 only)

	Sector Length Field			
	00	01	10	11
b = 0	256	512	1024	128
b = 1	128	256	512	1024

Table 5. FLAG SUMMARY

TYPE IV COMMAND

Ii = Interrupt Condition flags (Bits 3-0)

I0 = 1, Not-Ready to Ready Transition
I1 = 1, Ready to Not-Ready Transition
I2 = 1, Index Pulse
I3 = 1, Immediate Interrupt

I_3 -I_0 = 0, Terminate with no Interrupt

TYPE I COMMANDS

The Type I Commands include the Restore, Seek, Step, Step-In, and Step-Out commands. Each of the Type I Commands contains a rate field (r_0r_1), which determines the stepping motor rate as defined in Table 1.

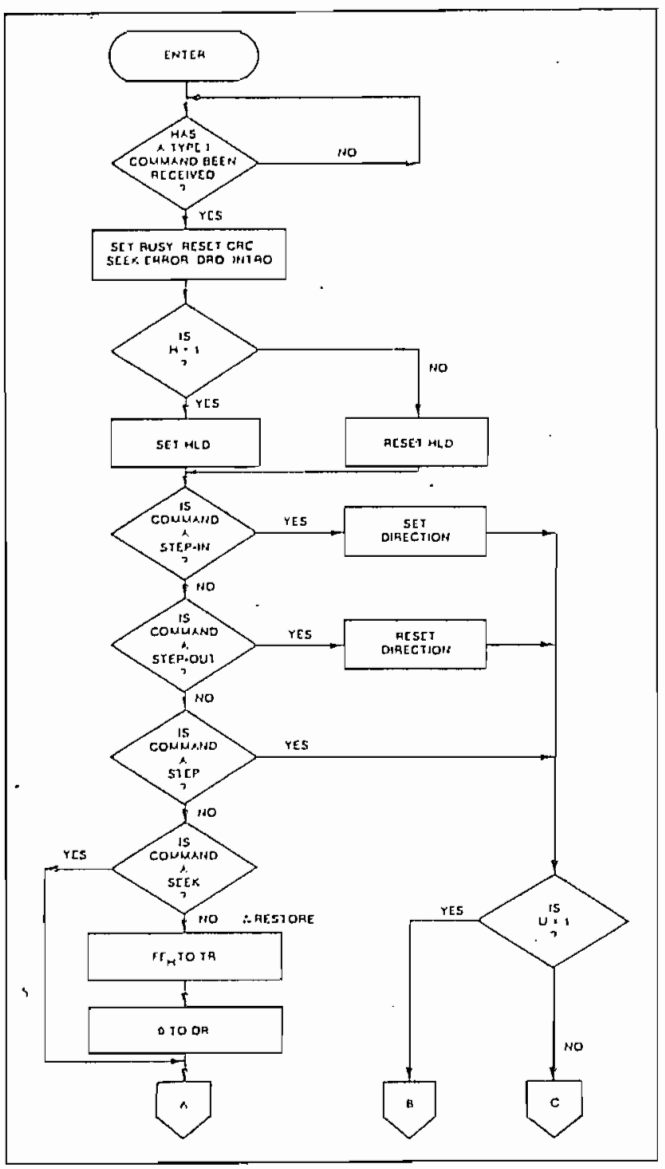
The Type I Commands contain a head load flag (h) which determines if the head is to be loaded at the beginning of the command. If h = 1, the head is loaded at the beginning of the command (HLD output is made active). If h = 0, HLD is deactivated. Once the head is loaded, the head will remain engaged until the FD179X receives a command that specifically disengages the head. If the FD179X is idle (busy = 0) for 15 revolutions of the disk, the head will be automatically disengaged (HLD made inactive).

The Type I Commands also contain a verification (V) flag which determines if a verification operation is to take place on the destination track. If V = 1, a verification is performed, if V = 0, no verification is performed.

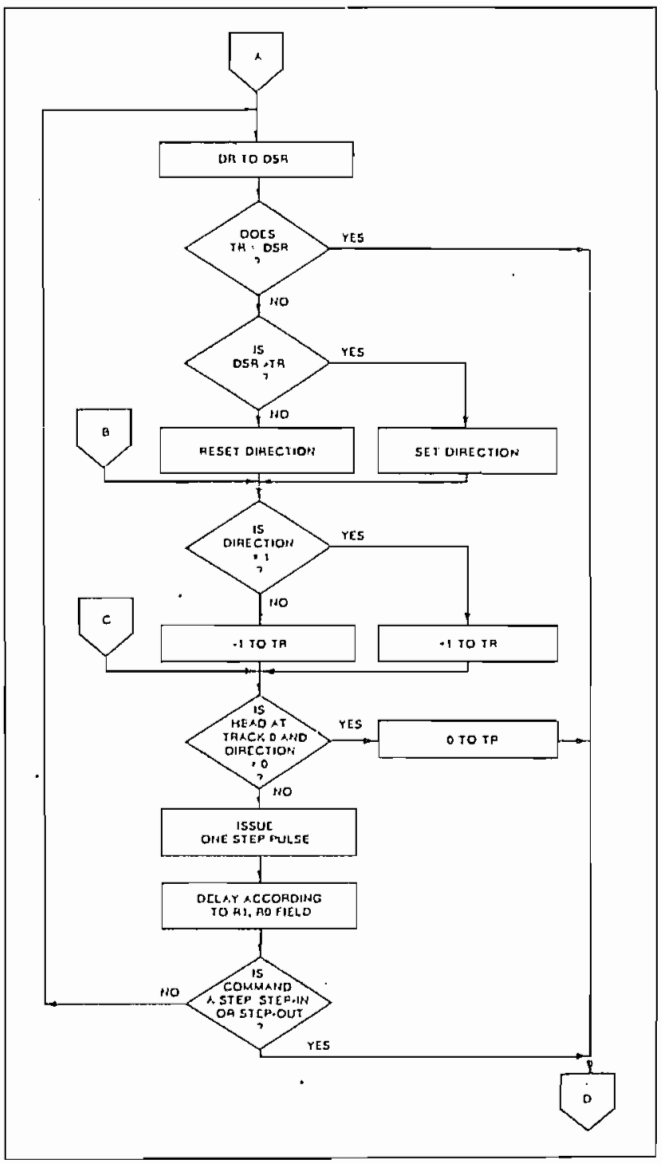
During verification, the head is loaded and after an internal 15 ms delay, the HLT input is sampled. When HLT is active (logic true), the first encountered ID field is read off the disk. The track address of the ID field is then compared to the Track Register. If there is a match and a valid ID CRC, the verification is complete, an interrupt is generated and the Busy status bit is reset. If there is not a match but there is valid ID CRC, an interrupt is generated, and Seek Error Status bit (Status bit 4) is set and the Busy status bit is reset. If there is a match but not a valid CRC, the CRC error status bit is set (Status bit 3), and the next encountered ID field is read from the disk for the verification operation. If an ID field with a valid CRC cannot be found after four revolutions of the disk, the FD179X terminates the operation and sends an interrupt, (INTRQ).

The Step, Step-In, and Step-Out commands contain an Update flag (U). When U = 1, the track register is updated by one for each step. When U = 0, the track register is not updated.

On the 1795/7 devices, the SSO output is not affected during Type 1 commands, and an internal side compare does not take place when the (V) Verify Flag is on.



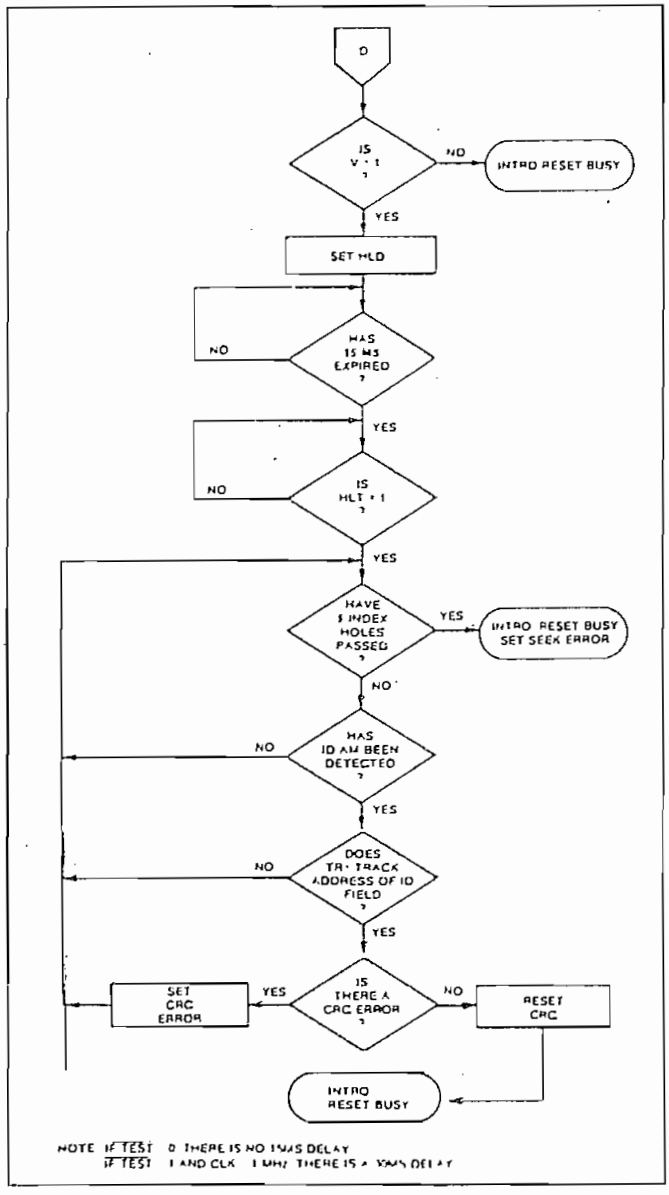
TYPE I COMMAND FLOW



TYPE I COMMAND FLOW

RESTORE (SEEK TRACK 0)

Upon receipt of this command the Track 00 ($\overline{TROO}$) input is sampled. If $\overline{TROO}$ is active low indicating the Read-Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TROO}$ is not active low, stepping pulses (pins 15 to 16) at a rate specified by the r_1r_0 field are issued until the $\overline{TROO}$ input is activated. At this time the Track Register is loaded with zeroes and an interrupt is generated. If the $\overline{TROO}$ input does not go active low after 255 stepping pulses, the FD179X terminates operation, interrupts, and sets the Seek error status bit. A verification operation takes place if the V flag is set. The h bit allows the head to be loaded at the start of command. Note that the Restore command is executed when $\overline{MR}$ goes from an active to an inactive state.



TYPE I COMMAND FLOW

SEEK

This command assumes that the Track Register contains the track number of the current position of the Read-Write head and the Data Register contains the desired track number. The FD179X will update the Track register and issue stepping pulses in the appropriate direction until the contents of the Track register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP

Upon receipt of this command, the FD179X issues one stepping pulse to the disk drive. The stepping motor direction is the same as in the previous step command. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. If the u flag is on, the Track Register is updated. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-IN

Upon receipt of this command, the FD179X issues one stepping pulse in the direction towards track 76. If the u flag is on, the Track Register is incremented by one. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-OUT

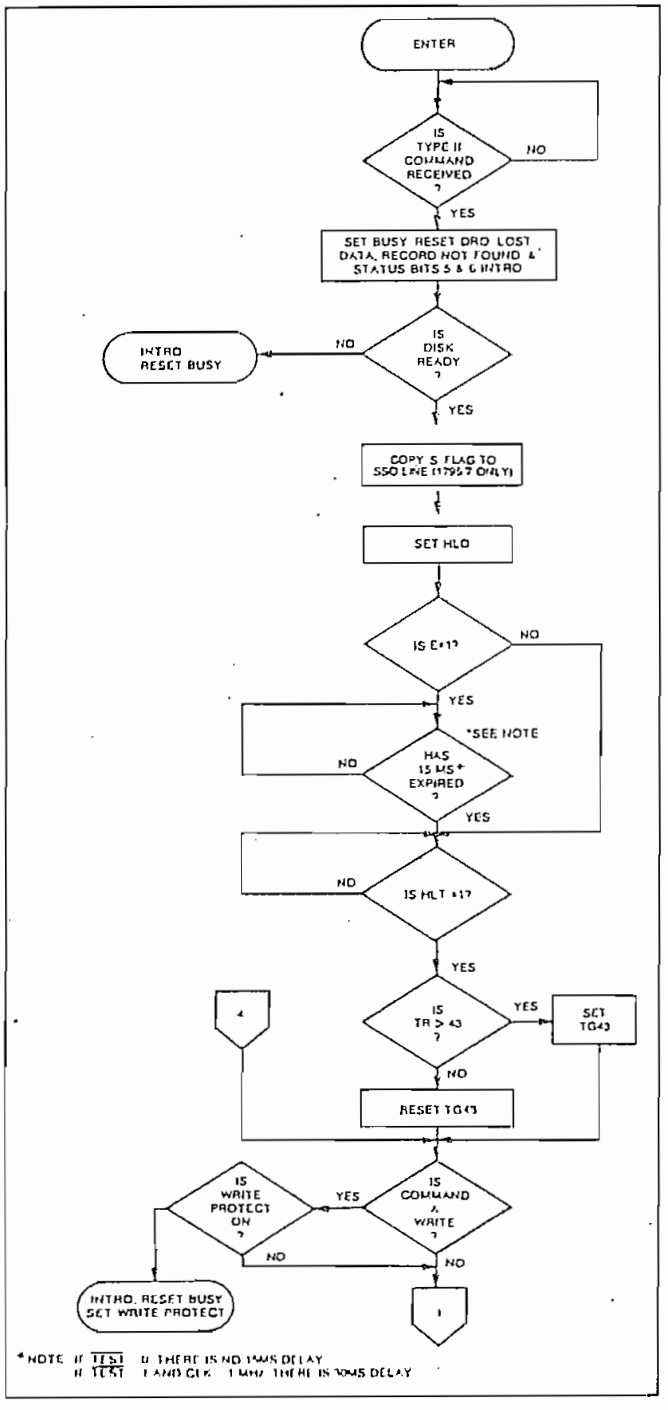
Upon receipt of this command, the FD179X issues one stepping pulse in the direction towards track 0. If the u flag is on, the Track Register is decremented by one. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

TYPE II COMMANDS

The Type II Commands are the Read Sector and Write Sector commands. Prior to loading the Type II Command into the Command Register, the computer must load the Sector Register with the desired sector number. Upon receipt of the Type II command, the busy status Bit is set. If the E flag = 1 (this is the normal case) HLD is made active and HLT is sampled after a 15 msec delay. If the E flag is 0, the head is loaded and HLT sampled with no 15 msec delay. The ID field and Data Field format are shown on page 13.

When an ID field is located on the disk, the FD179X compares the Track Number on the ID field with the Track Register. If there is not a match, the next en-

made. If there was a match, the Sector Number of the ID field is compared with the Sector Register. If there is not a Sector match, the next encountered ID field is read off the disk and comparisons again made. If the ID field CRC is correct, the data field is then located and will be either written into, or read from depending upon the command. The FD179X must find an ID field with a Track number, Sector number, side number, and CRC within four revolutions of the disk; otherwise, the Record not found status bit is set (Status bit 3) and the command is terminated with an interrupt.

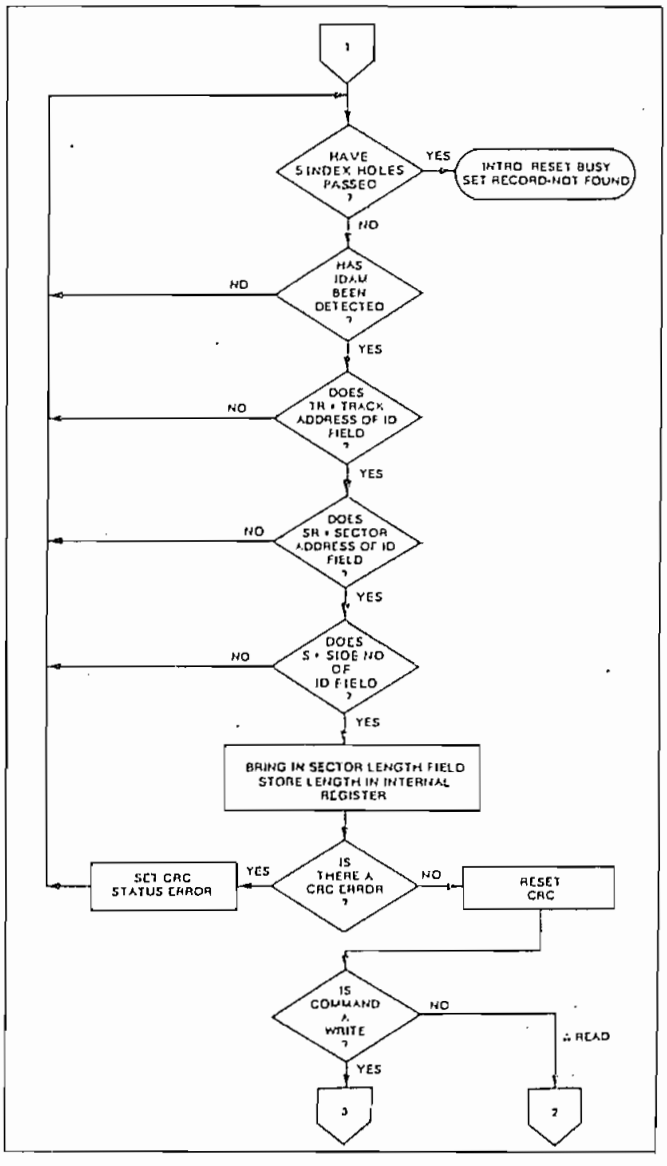


TYPE II COMMAND

Sector Length Table

Sector Length Field (hex)	Number of Bytes in Sector (decimal)
00	128
01	256
02	512
03	1024

Each of the Type II Commands contains an (m) flag which determines if multiple records (sectors) are to be read or written, depending upon the command. If m = 0, a single sector is read or written and an interrupt is generated at the completion of the command. If m = 1, multiple records are read or written with the sector register internally updated so that an address verification can occur on the next record. The FD179X will continue to read or write multiple records and update the sector register until the sector regis-



TYPE II COMMAND

ter exceeds the number of sectors on the track or until the Force Interrupt command is loaded into the Command Register, which terminates the command and generates an interrupt.

If the Sector Register exceeds the number of sectors on the track, the Record-Not-Found status bit will be set.

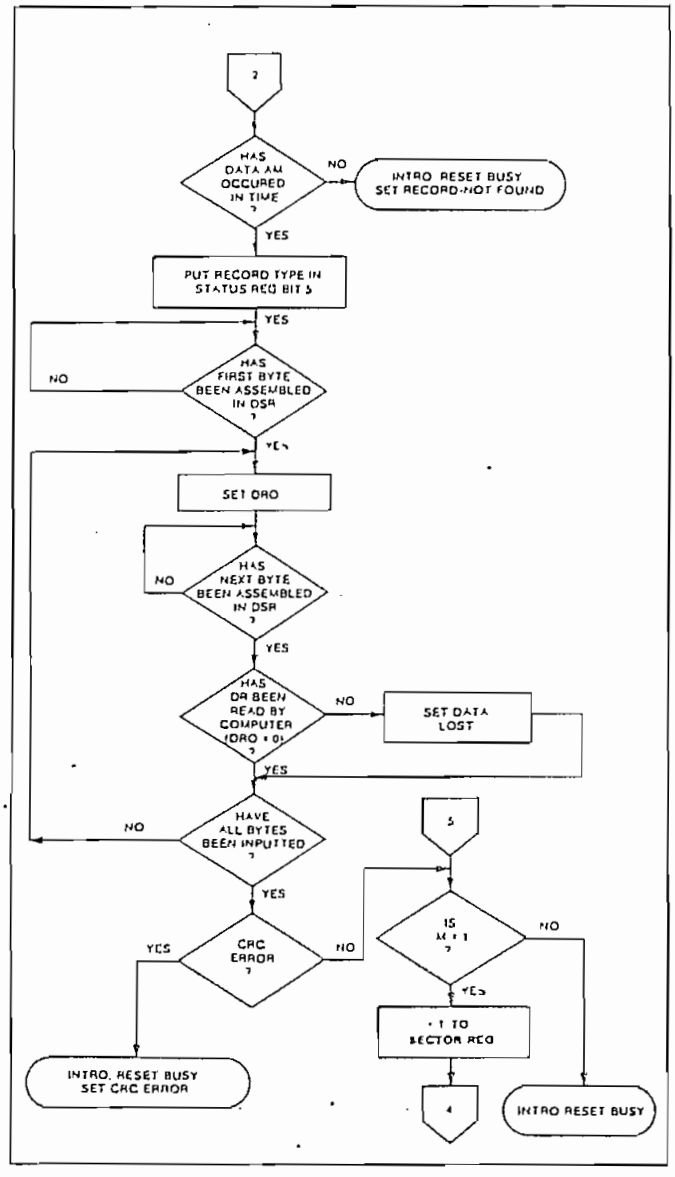
The Type II commands also contain side select compare flags. When C = 0, no side comparison is made. When C = 1, the LSB of the side number is read off the ID Field of the disk and compared with the contents of the (S) flag. If the S flag compares with the side number recorded in the ID field, the 179X continues with the ID search. If a comparison is not made within 5 index pulses, the interrupt line is made active and the Record-Not-Found status bit is set.

The 1795/7 READ SECTOR and WRITE SECTOR commands include a 'b' flag. The 'b' flag, in conjunction with the sector length byte of the ID Field, allows different byte lengths to be implemented in each sector. For IBM compatability, the 'b' flag should be set to a one. The S flag allows direct control over the [illegible] and is set or reset at the beginning of the command, dependent upon the value of this flag.

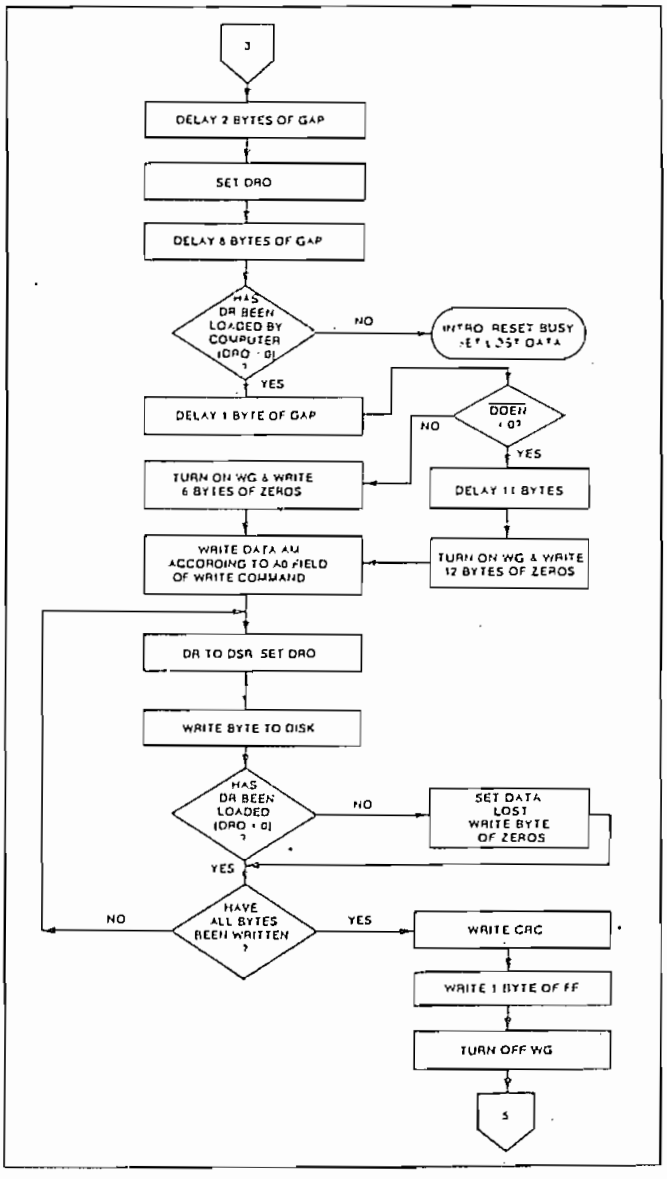
READ SECTOR

Upon receipt of the Read Sector command, the head is loaded, the Busy status bit set, and when an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, the data field is presented to the computer. The Data Address Mark of the data field must be found within 30 bytes in single density and 43 bytes in double density of the last ID field CRC byte; if not, the Record Not Found status bit is set and the operation is terminated.

When the first character or byte of the data field has been shifted through the DSR, it is transferred to the DR, and DRQ is generated. When the next byte is accumulated in the DSR, it is transferred to the DR and another DRQ is generated. If the Computer has not read the previous contents of the DR before a new character is transferred that character is lost and



TYPE II COMMAND



TYPE II COMMAND

tinues until the complete data field has been inputted to the computer. If there is a CRC error at the end of the data field, the CRC error status bit is set, and the command is terminated (even if it is a multiple record command).

At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Bit 5) as shown below:

STATUS BIT 5	
1	Deleted Data Mark
0	Data Mark

WRITE SECTOR

Upon receipt of the Write Sector command, the head is loaded (HLD active) and the Busy status bit is set. When an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, a DRQ is generated. The FD179X counts off 11 bytes in single density and 22 bytes in double density from the CRC field and the Write Gate (WG) output is made active if the DRQ is serviced (i.e., the DR has been loaded by the computer). If DRQ has not been serviced, the command is terminated and the Lost Data status bit is set. If the DRQ has been serviced, the WG is made active and six bytes of zeros in single density and 12 bytes in double density are then written on the disk. At this time the Data Address Mark is then written on the disk as determined by the a_0 field of the command as shown below:

a_0	Data Address Mark (Bit 0)
1	Deleted Data Mark
0	Data Mark

The FD179X then writes the data field and generates DRQ's to the computer. If the DRQ is not serviced in time for continuous writing the Lost Data Status Bit is set and a byte of zeros is written on the disk. The command is not terminated. After the last data byte has been written on the disk, the two-byte CRC is computed internally and written on the disk followed by one byte of logic ones in FM or in MFM. The WG output is then deactivated.

TYPE III COMMANDS

READ ADDRESS

Upon receipt of the Read Address command, the head is loaded and the Busy Status Bit is set. The disk, and the six data bytes of the ID field are assembled and transferred to the DR, and a DRQ is generated for each byte. The six bytes of the ID field are shown below:

TRACK ADDR	SIDE NUMBER	SECTOR ADDRESS	SECTOR LENGTH	CRC 1	CRC 2
1	2	3	4	5	6

Although the CRC characters are transferred to the computer, the FD179X checks for validity and the CRC error status bit is set if there is a CRC error. The Track Address of the ID field is written into the sector register. At the end of the operation an interrupt is generated and the Busy Status is reset.

READ TRACK

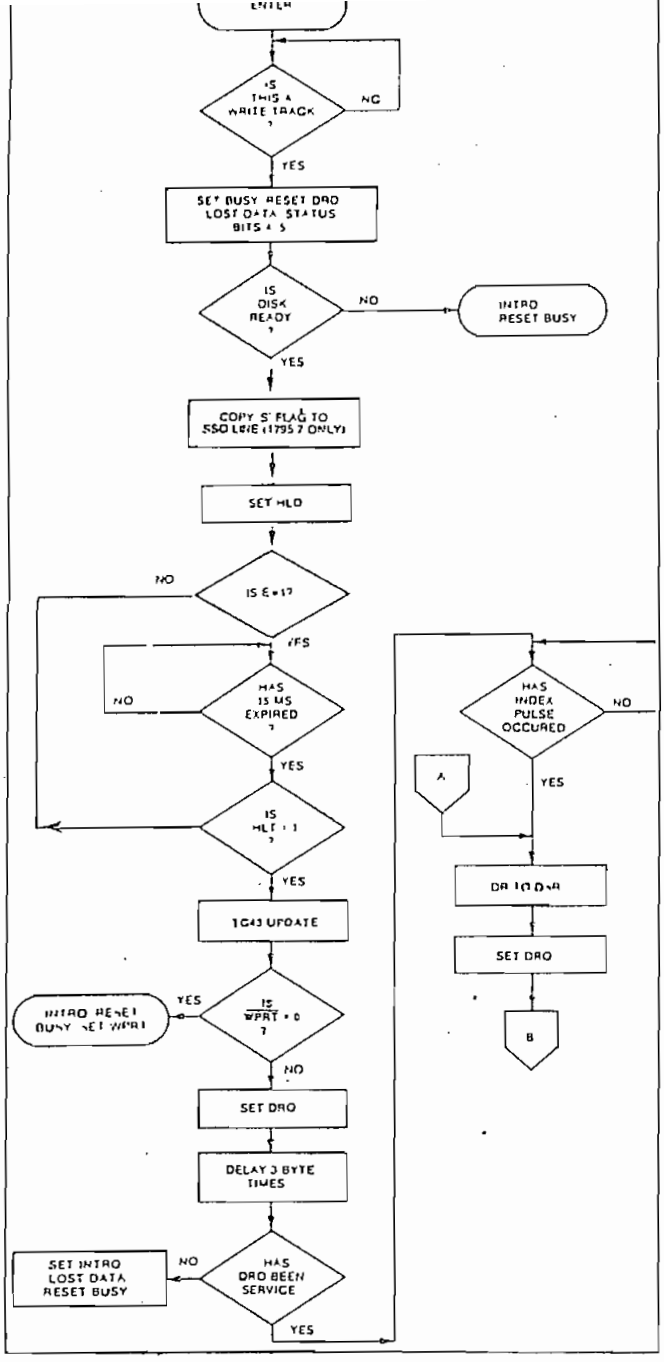
Upon receipt of the Read Track command, the head is loaded and the Busy Status bit is set. Reading starts with the leading edge of the first encountered index pulse and continues until the next index pulse. As each byte is assembled it is transferred to the Data Register and the Data Request is generated for each byte. No CRC checking is performed. Gaps are included in the input data stream. The accumulation of bytes is synchronized to each Address Mark encountered. Upon completion of the command, the interrupt is activated. RG is not activated during the Read Track Command. An internal side compare is not performed during a Read Track.

WRITE TRACK

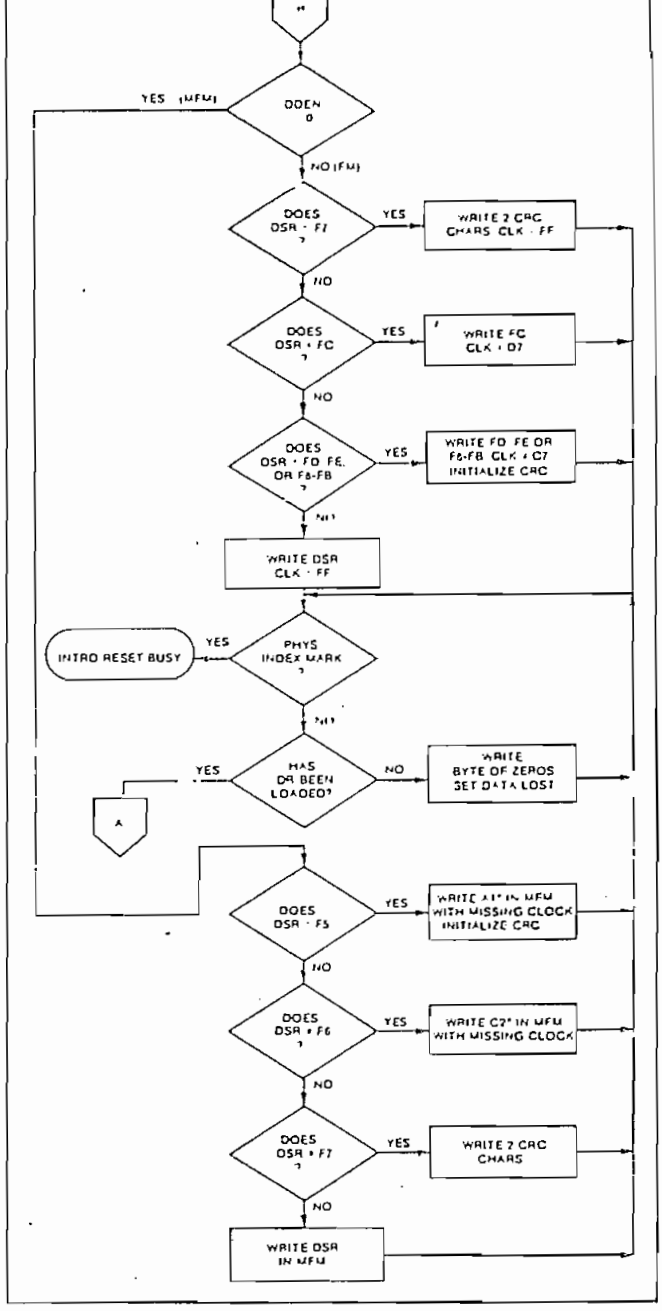
Upon receipt of the Write Track command, the head is loaded and the Busy Status bit is set. Writing starts with the leading edge of the first encountered index pulse and continues until the next index pulse, at which time the interrupt is activated. The Data Request is activated immediately upon receiving the command, but writing will not start until after the first byte has been loaded into the Data Register. If the DR has not been loaded by the time the index pulse is encountered the operation is terminated making the device Not Busy, the Lost Data Status Bit is set, and the Interrupt is activated. If a byte is not present in the DR when needed, a byte of zeros is substituted. Address Marks and CRC characters are written on the disk by detecting certain data byte patterns in the outgoing data stream as shown in the table below. The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR in FM or by receipt of F5 in MFM.

GAP III	ID AM	TRACK NUMBER	SIDE NUMBER	SECTOR NUMBER	SECTOR LENGTH	CRC 1	CRC 2	GAP II	DATA AM	DATA FIELD	CRC 1	CRC 2
ID FIELD										DATA FIELD		

In MFM only, IDAM and DATA AM are preceded by three bytes of A1 with clock transition between bits 4 and 5 missing.



TYPE III COMMAND WRITE TRACK



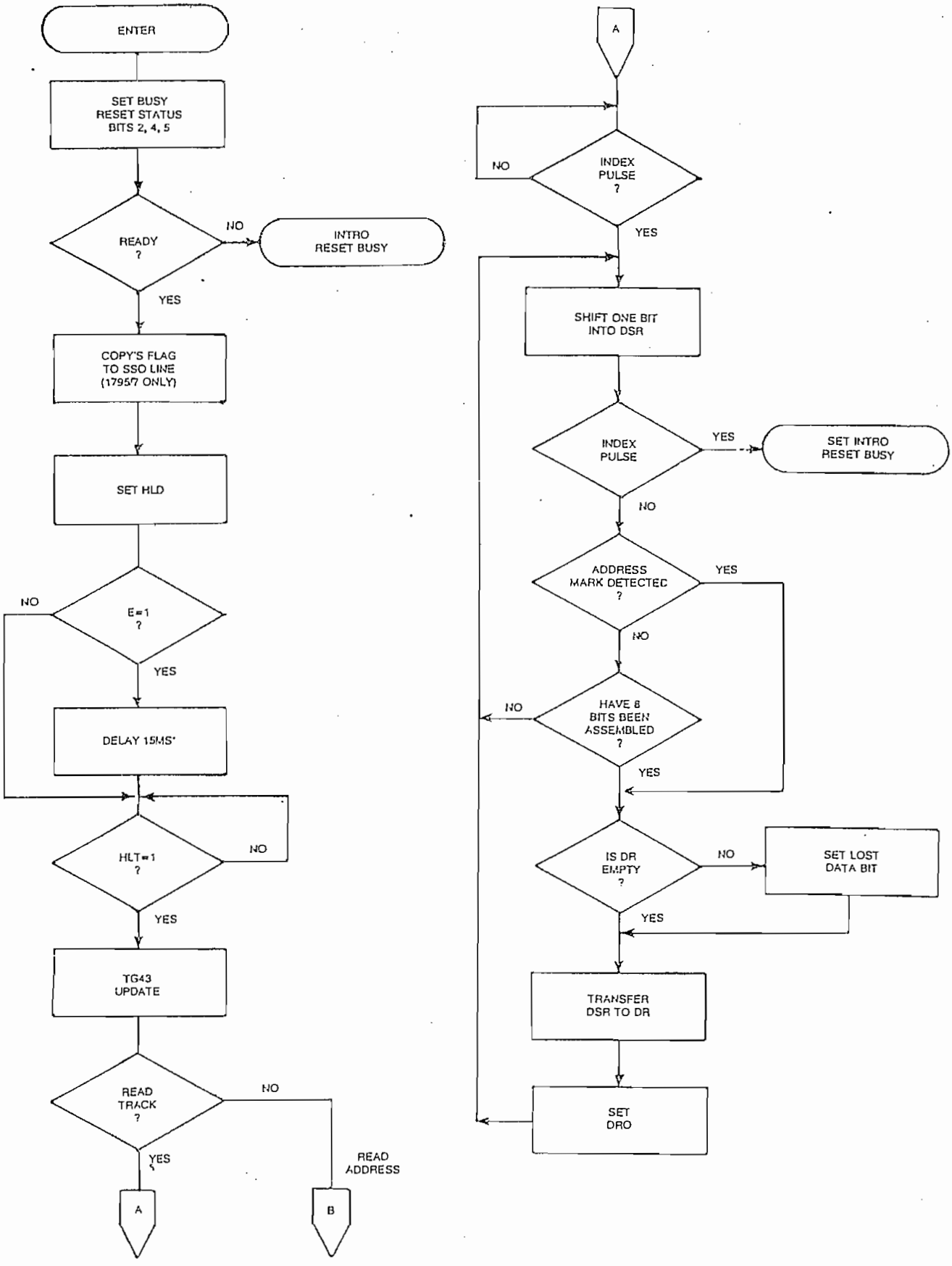
TYPE III COMMAND WRITE TRACK

CONTROL BYTES FOR INITIALIZATION

DATA PATTERN IN DR (HEX)	FD179X INTERPRETATION IN FM ($\overline{DDEN}$ = 1)	FD1791/3 INTERPRETATION IN MFM ($\overline{DDEN}$ = 0)
00 thru F4	Write 00 thru F4 with CLK = FF	Write 00 thru F4, in MFM
F5	Not Allowed	Write A1* in MFM, Preset CRC
F6	Not Allowed	Write C2** in MFM
F7	Generate 2 CRC bytes	Generate 2 CRC bytes
F8 thru FB	Write F8 thru FB, Clk = C7, Preset CRC	Write F8 thru FB, in MFM
FC	Write FC with Clk = D7	Write FC in MFM
FD	Write FD with Clk = FF	Write FD in MFM
FE	Write FE, Clk = C7, Preset CRC	Write FE in MFM
FF	Write FF with Clk = FF	Write FF in MFM

*Missing clock transition between bits 4 and 5

**Missing clock transition between bits 3 & 4

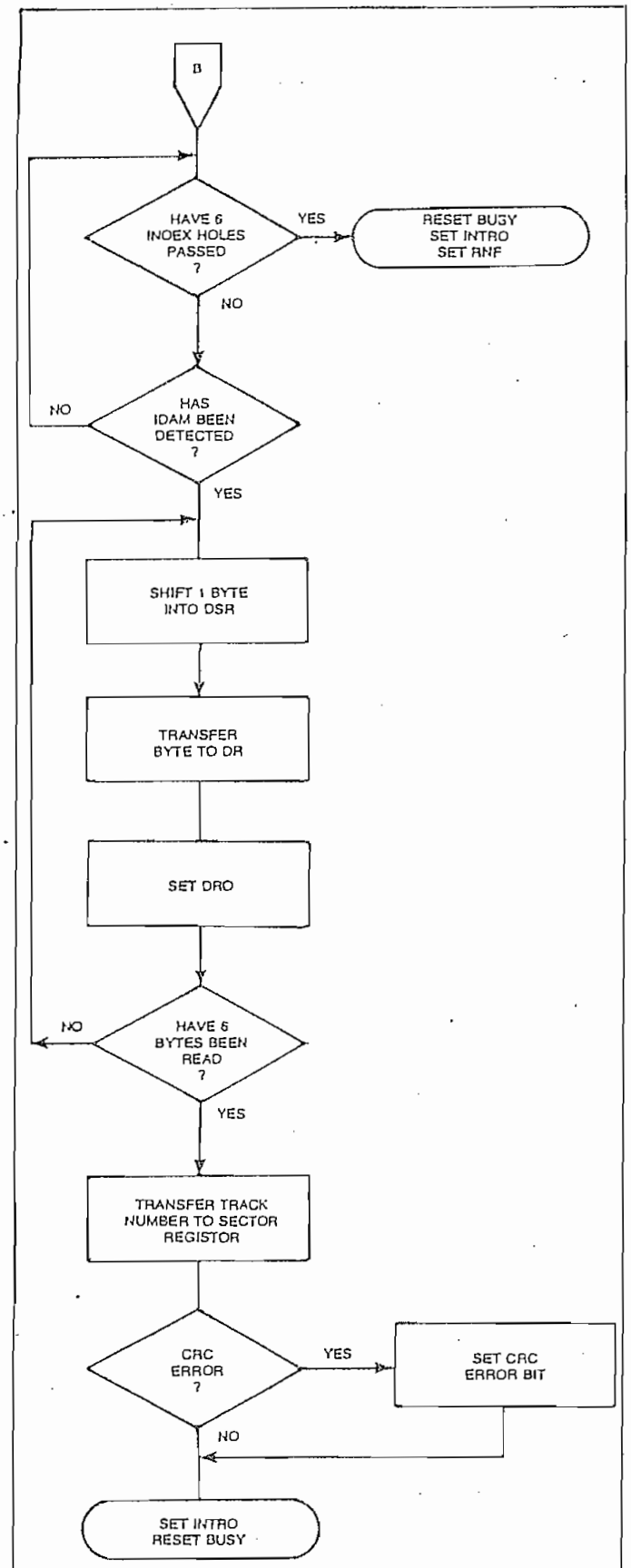


*If TEST = ϕ NO DELAY

If TEST = 1 and CLK = 1 MHZ, 30 MS DELAY

TYPE III COMMAND

Read Track/Address



TYPE III COMMAND

Read Track/Address

TYPE IV COMMAND

FORCE INTERRUPT

This command can be loaded into the command register at any time. If there is a current command under execution (Busy Status Bit set), the command will be terminated and an interrupt will be generated when the condition specified in the I_0 through I_3 field is detected. The interrupt conditions are shown below:

I_0 = Not-Ready-To-Ready Transition
I_1 = Ready-To-Not-Ready Transition
I_2 = Every Index Pulse
I_3 = Immediate Interrupt (requires reset, see Note)

NOTE: If $I_0 - I_3 = 0$, there is no interrupt generated but the current command is terminated and busy is reset. *This is the only command that will enable the immediate interrupt to clear on a subsequent Load Command Register or Read Status Register.*

STATUS DESCRIPTION

Upon receipt of any command, except the Force Interrupt command, the Busy Status bit is set and the rest of the status bits are updated or cleared for the new command. If the Force Interrupt Command is received when there is a current command under execution, the Busy status bit is reset, and the rest of the status bits are unchanged. If the Force Interrupt command is received when there is not a current command under execution, the Busy Status bit is reset and the rest of the status bits are updated or cleared. In this case, Status reflects the Type I commands.

The format of the Status Register is shown below:

(BITS)							
7	6	5	4	3	2	1	0
S7	S6	S5	S4	S3	S2	S1	S0

Status varies according to the type of command executed as shown in Table 6.

(Refer to section on Type III commands for flow diagrams.)

Formatting the disk is a relatively simple task when operating programmed I/O or when operating under Formatting the disk is accomplished by positioning the R/W head over the desired track number and issuing the Write Track command. Upon receipt of the Write Track command, the FD179X raises the Data Request signal. At this point in time, the user loads the data register with desired data to be written on the disk. For every byte of information to be written on the disk, a data request is generated. This sequence continues from one index mark to the next index mark. Normally, whatever data pattern appears in the data register is written on the disk with a normal clock pattern. However, if the FD179X detects a data pattern of F5 thru FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation. For instance, in FM an FE pattern will be interpreted as an ID address mark (DATA-FE, CLK-C7) and the CRC will be initialized. An F7 pattern will generate two CRC characters in FM or MFM. As a consequence, the patterns F5 thru FE must not appear in the gaps, data fields, or ID fields. Also, CRC's must be generated by an F7 pattern.

Disks may be formatted in IBM 3740 or System 34 formats with sector lengths of 128, 256, 512, or 1024 bytes.

IBM 3740 FORMAT - 128 BYTES/SECTOR

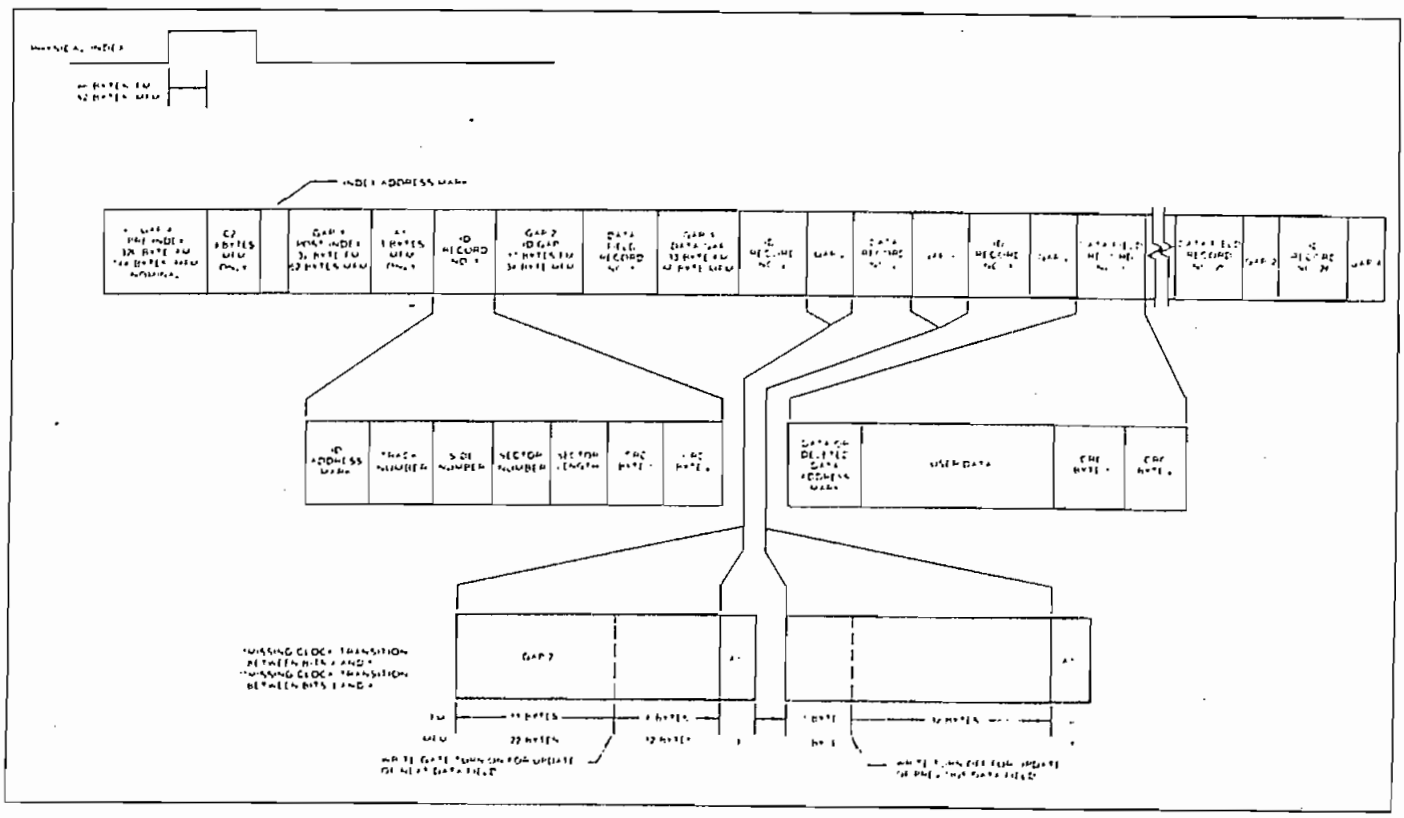
Shown below is the IBM single-density format with 128 bytes/sector. In order to format a diskette, the user must issue the Write Track command, and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
40	FF (or 00)[1]
6	00
1	FC (Index Mark)
26	FF (or 00)
6	00
1	FE (ID Address Mark)
1	Track Number
1	Side Number (00 or 01)
1	Sector Number (1 thru 1A)
1	00
1	F7 (2 CRC's written)
11	FF (or 00)
6	00
1	FB (Data Address Mark)
128	Data (IBM uses E5)
1	F7 (2 CRC's written)
27	FF (or 00)
247**	FF (or 00)

*Write bracketed field 26 times

**Continue writing until FD179X interrupts out. Approx. 247 bytes.

1-Optional '00' on 1795/7 only.



IBM TRACK FORMAT

256 BYTES/SECTOR

Shown below is the IBM dual-density format with 256 bytes/sector. In order to format a diskette the user must issue the Write Track command and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
80	4E
12	00
3	F6
1	FC (Index Mark)
50*	4E
12	00
3	F5
1	FE (ID Address Mark)
1	Track Number (0 thru 4C)
1	Side Number (0 or 1)
1	Sector Number (1 thru 1A)
1	01
1	F7 (2 CRCs written)
22	4E
12	00
3	F5
1	FB (Data Address Mark)
256	DATA
1	F7 (2 CRCs written)
54	4E
598**	4E

*Write bracketed field 26 times
**Continue writing until FD179X interrupts out. Approx. 598 bytes.

Variations in the IBM format are possible to a limited extent if the following requirements are met: sector size must be a choice of 128, 256, 512, or 1024 bytes; gap size must be according to the following table. Note that the Index Mark is not required by the 179X. The minimum gap sizes shown are that which is required by the 179X, with PLL lock-up time, motor speed variation, etc.; adding additional bytes.

	FM	MFM
Gap I	16 bytes FF	32 bytes 4E
Gap II	11 bytes FF	22 bytes 4E
*	6 bytes 00	12 bytes 00 3 bytes A1
Gap III	10 bytes FF	24 bytes 4E 3 bytes A1
**	4 bytes 00	8 bytes 00
Gap IV	16 bytes FF	16 bytes 4E

*Byte counts must be exact.
**Byte counts are minimum, except exactly 3 bytes of A1 must be written.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

V_{DD} With Respect to V_{SS} (Ground) =15 to −0.3V

Max. Voltage to Any Input With Respect to V_{SS} =15 to −0.3V

Operating Temperature 0°C to 70°C

Storage Temperature −55°C to +125°C

V_{DD} = ID ma Nominal V_{CC} = 35 ma Nominal

OPERATING CHARACTERISTICS (DC)

TA.= 0°C to 70°C, V_{DD} = + 12V ± .6V, V_{SS} = OV, V_{CC} = + 5V ± .25V

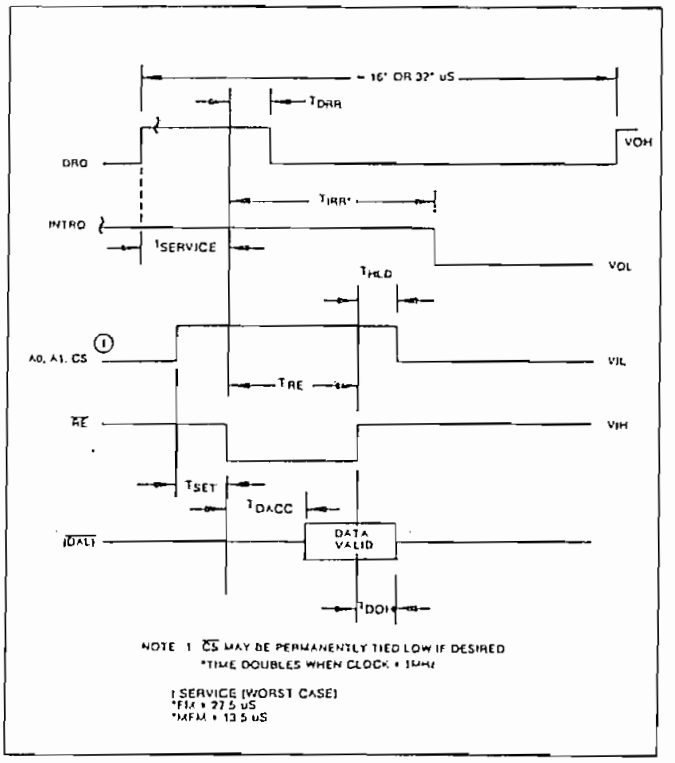
SYMBOL	CHARACTERISTIC	MIN.	MAX.	UNITS	CONDITIONS
I_{IL}	Input Leakage		10	μA	$V_{IN} = V_{DD}$
I_{OL}	Output Leakage		10	μA	$V_{OUT} = V_{DD}$
V_{IH}	Input High Voltage	2.6		V	
V_{IL}	Input Low Voltage		0.8	V	
V_{OH}	Output High Voltage	2.8		V	$I_O = -100\ \mu A$
V_{OL}	Output Low Voltage		0.45	V	$I_O = 1.6$ mA
P_D	Power Dissipation		0.5	W	

TIMING CHARACTERISTICS

T_A = 0°C to 70°C, V_{DD} = + 12V ± .6V, V_{SS} = 0V, V_{CC} = +5V ± .25V

READ ENABLE TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{RE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{RE}$	10			nsec	
TRE	$\overline{RE}$ Pulse Width	400			nsec	C_L = 50 pf
TDRR	DRQ Reset from $\overline{RE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{RE}$		500	3000	nsec	See Note 5
TDACC	Data Access from $\overline{RE}$			350	nsec	C_L = 50 pf
TDOH	Data Hold From $\overline{RE}$	50		150	nsec	C_L = 50 pf



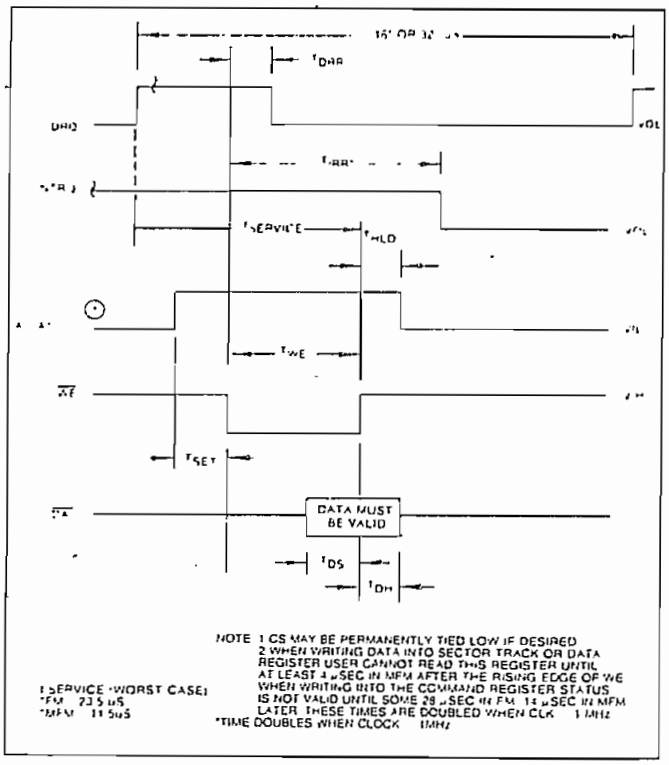
READ ENABLE TIMING

WRITE ENABLE TIMING

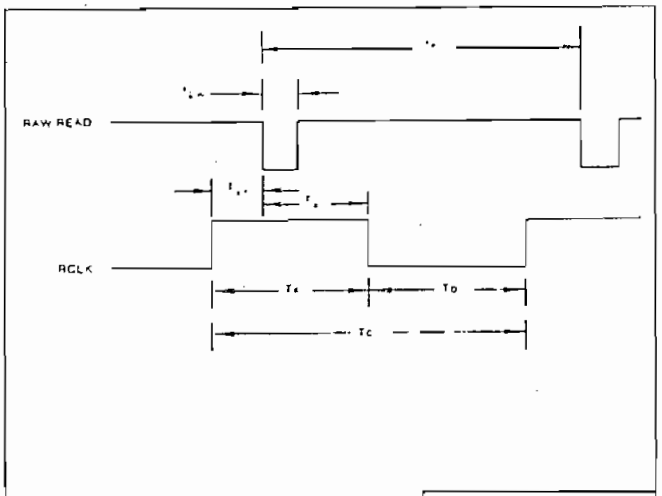
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{WE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{WE}$	10			nsec	
TWE	$\overline{WE}$ Pulse Width	350			nsec	
TDRR	DRQ Reset from $\overline{WE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{WE}$		500	3000	nsec	See Note 5
TDS	Data Setup to $\overline{WE}$	250			nsec	
TDH	Data Hold from $\overline{WE}$	70			nsec	

INPUT DATA TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Tpw	$\overline{Raw\ Read}$ Pulse Width	100	200		nsec	See Note 1
tbc	$\overline{Raw\ Read}$ Cycle Time		1500		nsec	1800 ns @ 70°C
Tc	RCLK Cycle Time		1500		nsec	1800 ns @ 70°C
Tx_1	RCLK hold to $\overline{Raw\ Read}$	40			nsec	See Note 1
Tx_2	$\overline{Raw\ Read}$ hold to RCLK	40			nsec	



WRITE ENABLE TIMING

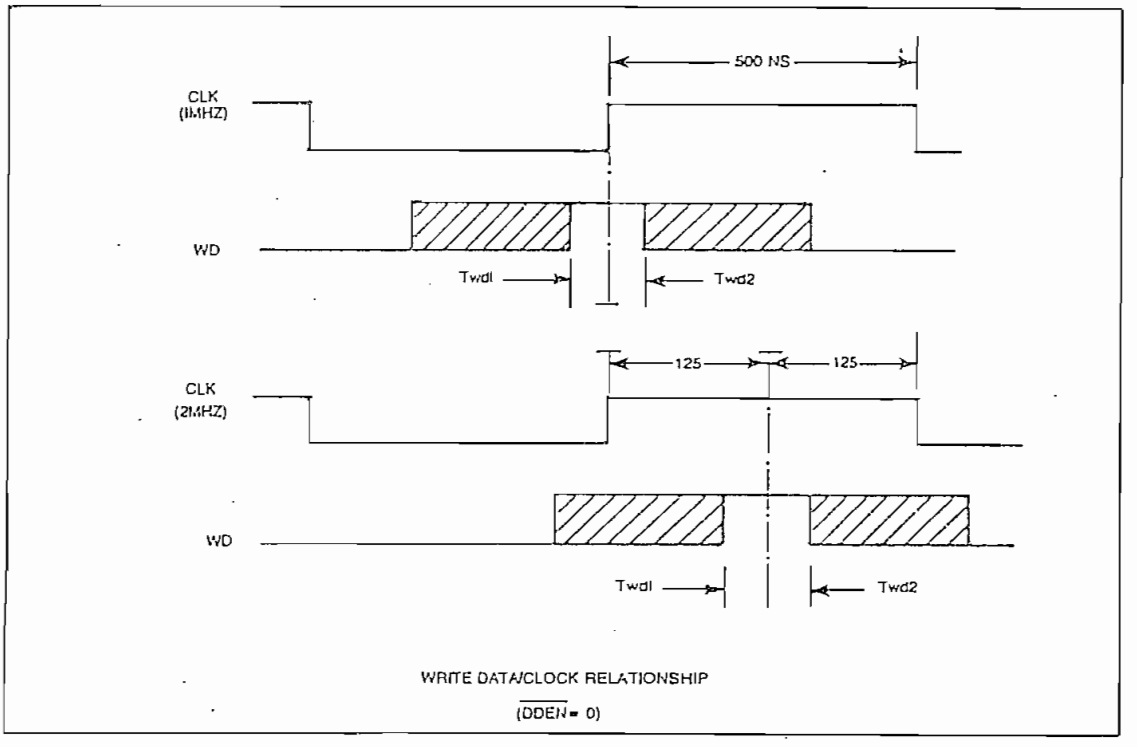


DISKETTE	MODE	$\overline{DDEN}$	CLK	NOMINAL T_1	NOMINAL T_h	NOMINAL T
8"	MFM	0	2 MHz	1 μs	1 μs	2 μs
8"	FM	1	2 MHz	2 μs	2μs	4 μs
5"	MFM	0	1 MHz	2 μs	2μs	4 μs
5"	FM	1	1 MHz	4 μs	4 μs	8 μs

INPUT DATA TIMING

WRITE DATA TIMING: (ALL TIMES DOUBLE WHEN CLK = 1 MHz)

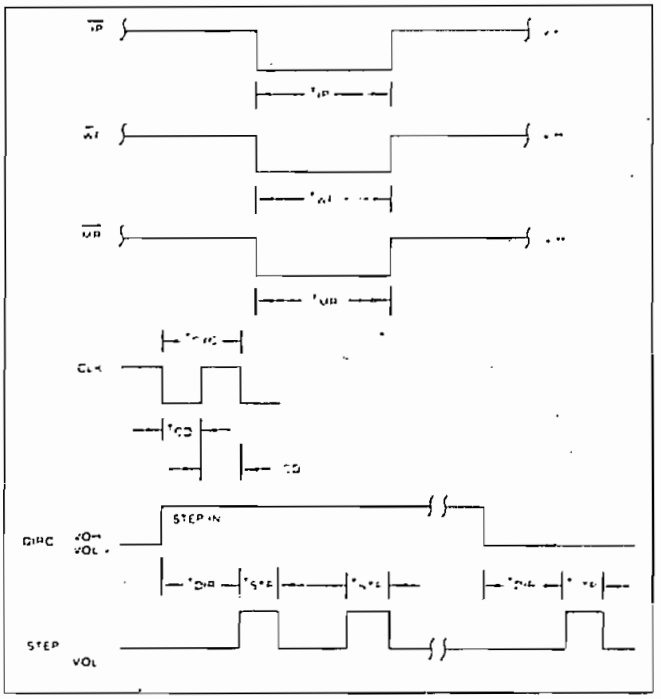
SYMBOL	CHARACTERISTICS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Twp	Write Data Pulse Width	450	500	550	nsec	FM
		150	200	250	nsec	MFM
Twg	Write Gate to Write Data		2		µsec	FM
			1		µsec	MFM
Tbc	Write data cycle Time		2,3, or 4		µsec	±CLK Error
Ts	Early (Late) to Write Data	125			nsec	MFM
Th	Early (Late) From Write Data	125			nsec	MFM
Twf	Write Gate off from WD		2		µsec	FM
			1		µsec	MFM
Twd1	WD Valid to Clk	100			nsec	CLK=1 MHZ
		50			nsec	CLK=2 MHZ
Twd2	WD Valid after CLK	100			nsec	CLK=1 MHZ
		30			nsec	CLK=2 MHZ



WRITE DATA TIMING

MISCELLANEOUS TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TCD_1	Clock Duty (low)	230	250	20000	nsec	
TCD_2	Clock Duty (high)	200	250	20000	nsec	
TSTP	Step Pulse Output	2 or 4			μsec	See Note 5
TDIR	Dir Setup to Step		12		μsec	± CLK ERROR
TMR	Master Reset Pulse Width	50			μsec	
TIP	Index Pulse Width	10			μsec	See Note 5
TWF	Write Fault Pulse Width	10			μsec	



MISCELLANEOUS TIMING

NOTES:

1. Pulse width on RAW READ (Pin 27) is normally 100-300 ns. However, pulse may be any width if pulse is entirely within window. If pulse occurs in both windows, then pulse width must be less than 300 ns for MFM at CLK = 2 MHz and 600 ns for FM at 2 MHz. Times double for 1 MHz.
2. A PPL Data Separator is recommended for 8" MFM.
3. tbc should be 2 μs, nominal in MFM and 4 μs nominal in FM. Times double when CLK = 1 MHz.
4. RCLK may be high or low during $\overline{\text{RAW READ}}$ (Polarity is unimportant).
5. Times double when clock = 1 MHz.

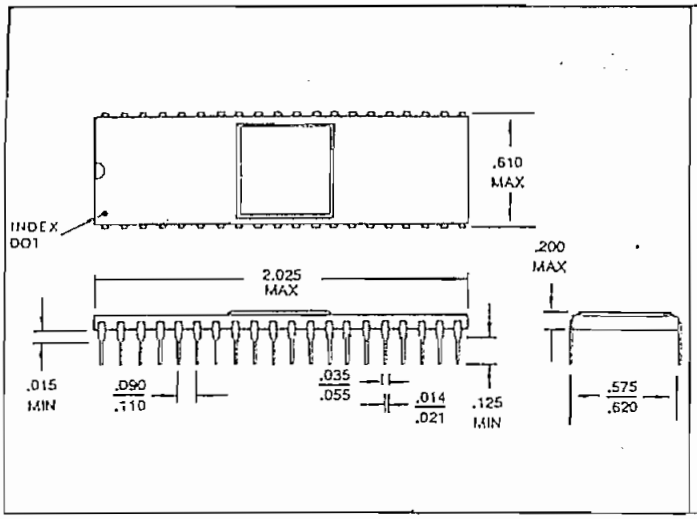
BIT	ALL TYPE I COMMANDS	READ ADDRESS	READ SECTOR	READ TRACK	WRITE SECTOR	WRITE TRACK
S7	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY
S6	WRITE PROTECT	0	0	0	WRITE PROTECT	WRITE PROTECT
S5	HEAD LOADED	0	RECORD TYPE	0	WRITE FAULT	WRITE FAULT
S4	SEEK ERROR	RNF	RNF	0	RNF	0
S3	CRC ERROR	CRC ERROR	CRC ERROR	0	CRC ERROR	0
S2	TRACK 0	LOST DATA	LOST DATA	LOST DATA	LOST DATA	LOST DATA
S1	INDEX	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

STATUS FOR TYPE I COMMANDS

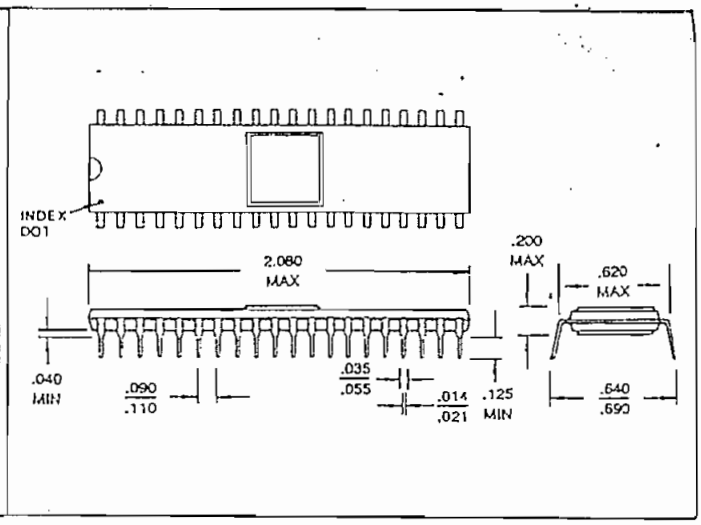
BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset it indicates that the drive is ready. This bit is an inverted copy of the Ready input and logically 'ored' with MR.
S6 PROTECTED	When set, indicates Write Protect is activated. This bit is an inverted copy of $\overline{WRPT}$ input.
S5 HEAD LOADED	When set, it indicates the head is loaded and engaged. This bit is a logical "and" of HLD and HLT signals.
S4 SEEK ERROR	When set, the desired track was not verified. This bit is reset to 0 when updated.
S3 CRC ERROR	CRC encountered in ID field.
S2 TRACK 00	When set, indicates Read/Write head is positioned to Track 0. This bit is an inverted copy of the $\overline{TR00}$ input.
S1 INDEX	When set, indicates index mark detected from drive. This bit is an inverted copy of the $\overline{IP}$ input.
S0 BUSY	When set command is in progress. When reset no command is in progress.

STATUS FOR TYPE II AND III COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset, it indicates that the drive is ready. This bit is an inverted copy of the Ready input and 'ored' with MR. The Type II and III Commands will not execute unless the drive is ready.
S6 WRITE PROTECT	On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated.
S5 RECORD TYPE/ WRITE FAULT	On Read Record: It indicates the record-type code from data field address mark. 1 = Deleted Data Mark. 0 = Data Mark. On any Write: It indicates a Write Fault. This bit is reset when updated.
S4 RECORD NOT FOUND (RNF)	When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated.
S3 CRC ERROR	If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated.
S2 LOST DATA	When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when updated.
S1 DATA REQUEST	This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated.
S0 BUSY	When set, command is under execution. When reset, no command is under execution.



FD179XA-02 CERAMIC PACKAGE



FD179XB-02 PLASTIC PACKAGE

This is a preliminary specification with tentative device parameters and may be subject to change after final product characterization is completed.

WESTERN DIGITAL CORPORATION
3128 REDHILL AVENUE, BOX 2180
NEWPORT BEACH, CA 92663 (714) 557-3550,TWX 910-595-1139

A P E N D I C E 2

" CITAS BIBLIOGRAFICAS "

CITAS BIBLIOGRAFICAS

(1) PEATMAN John B., MICROCOMPUTER-BASED DESIGN, Editorial Mc Graw-Hill, 1977, p. 180.

(2) PEATMAN John B., MICROCOMPUTER-BASED DESIGN, Editorial Mc Graw-Hill, 1977, p. 182.

(3) PEATMAN John B., MICROCOMPUTER-BASED DESIGN, Editorial Mc Graw-Hill, 1977, p. 183.

(4) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.2.

(5) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.1.

(6) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.4.

(7) TEXAS INSTRUMENTS, TMS 9900 FLOPPY DISK CONTROLLER, 1977, p.24.

(8) WESTERN DIGITAL, FD179X-02 APPLICATION NOTES, 1980, p.3.

(9) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.23.

(10) PEATMAN John B., MICROCOMPUTER NASED DESIGN, Editorial Mc Graw-Hill, 1977, p.187.

(11) RALSTON Anthony, ENCYCLOPEDIA OF COMPUTER SCIENCE, Editorial VAN NOSTRAND REINHOLD, 1976, p.383.

(12) KRUTZ Ronald L., MICROPROCESSORS AND LOGIC DESIGN, Editorial John Wiley & Sons, 1980, p.418.

(13) WAKERLY John, Error Detecting Codes, Self-Checking Circuits and Applications, Editorial Elsevier North-Holland, 1978, p.32.

(14) KRUTZ Ronald L., MICROPROCESSORS AND LOGIC DESIGN, Editorial John Wiley & Sons, 1980, p.365.

(15) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.1,2.

(16) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.8.

(17) TEXAS INSTRUMENTS, The TTL Data Book, 1976, p. 6-82.

(18) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.5.

(19) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.6,8,9.

(20) TEXAS INSTRUMENTS, The TTL Data Book, 1976, p.7-123.

(21) TEXAS INSTRUMENTS, The TTL Data Book, 1976, p.6 - 81.

(22) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.14.

(23) VECTOR ELECTRONIC COMPANY, 80-81 CATALOG, 1980, p.IV-34.

(24) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.9.

(25) WESTERN DIGITAL, 1981 Product Handbook, 1981, p.361.

(26) POPULAR ELECTRONICS, Agosto 1980, p.61.

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

"INTERFACE ENTRE MICROPROCESADOR Y DISCOS FLOPPY"

TESIS PREVIA A LA OBTENCION DEL TITULO
DE INGENIERO EN LA ESPECIALIZACION DE
ELECTRONICA Y TELECOMUNICACIONES

DIEGO MARCELO VALDEZ VITERI

JULIO 1982

Certifico que el presente trabajo ha sido elaborado en su totalidad por el Señor Marcelo Valdez Viteri.

ING. HERBERT JACOBSON
Director de Tesis

D E D I C A T O R I A

A MIS PADRES

A G R A D E C I M I E N T O

Al Ing. Herbert Jacobson por su acertada dirección y colaboración en el desarrollo y construcción de esta tesis.

I N D I C E

" INTERFACE ENTRE MICROPROCESADOR Y DISCOS FLOPPY "

INTRODUCCION

Dentro de los últimos años, el uso de discos "floppy" ha llegado a constituírse en uno de los medios más populares de almacenamiento -en línea- para sistemas pequeños de computación. Su tiempo de acceso rápido, gran confiabilidad y bajo costo por bit de información hacen que el disco "floppy" sea la solución para el almacenamiento no volátil de grandes cantidades de información en sistemas cuyo funcionamiento se basa en el uso de microprocesadores.

El hecho de que los discos pueden ser removidos del aparato que los lee o escribe y que la información almacenada en ellos no es volátil, permite la archivación de los discos para su uso futuro sin necesitar de un consumo de energía para retener la información en ellos escrita. Además, la facilidad de utilización y la versatilidad que presentan al poder ser escritos y leídos por máquinas diferentes, lo que permite el intercambio de datos o programas en forma sencilla y a un bajo costo, ha llevado a la rápida popularización de este sistema.

Debido a que este tipo de discos fueron creados por la Compañía IBM, las características de los mismos se convirtieron en un standard, por lo que la forma en que se conectan las lectoras/escritoras de este tipo de discos al sistema principal no varía mucho de un producto a otro, de forma que el usua

rio puede sustituír una lectora/escritora de un fabricante por la de otro, sin que esto represente grandes modificaciones en su sistema (1).

En la presente tesis se van a desarrollar los sistemas electrónicos necesarios para controlar el funcionamiento de lectoras/escritoras de este tipo de discos y para realizar los procesos de interface de estos sistemas con un equipo cuyo funcionamiento se basa en el uso de un microprocesador. Se desarrollarán además, los programas que sirvan de soporte para el correcto funcionamiento del sistema completo.

El sistema terminado tendrá todas las ventajas anteriormente indicadas y será sin duda, una gran ayuda para la creación y almacenamiento de programas que sirvan de base a trabajos futuros que se realicen en torno a este equipo, permitiendo además una mayor claridad en la enseñanza de la materia de microprocesadores. De esta manera se podrá hacer efectivo el uso de equipos que hasta ahora, a pesar de tenerlos disponibles practicamente no han sido utilizados debido a su dificultad de operación.

En el futuro se podrán adquirir discos previamente grabados, compatibles con el sistema que aquí se diseña, que contengan programas como editores, ensambladores, sistemas operativos y otras clases de programas que expandan la capacidad del equipo, faciliten su manejo y ayuden a la complementación de la enseñanza.

C A P I T U L O P R I M E R O

"INTRODUCCION A LOS DISCOS FLOPPY"

1. INTRODUCCION A LOS DISCOS FLOPPY.

1.1. LOS DISCOS FLOPPY.

El disco floppy o diskette es el medio que retiene la información, está recubierto por una funda plástica, delgada y flexible, que protege al disco de las partículas de suciedad.

La funda contiene internamente un material especialmente tratado para minimizar la fricción y las descargas de electricidad estática. Consta de tres aperturas principales, la primera permite que la cabeza de grabación pueda ponerse en contacto con la superficie del disco, la segunda asegura el acceso de eje que hace girar al disco dentro de su funda y la tercera permite la detección del agujero índice. En la mayoría de los casos se tiene otro agujero en el extremo de la funda, que posibilita la escritura de información en el disco dependiendo si está recubierto o no, pudiendo tener de esta forma discos "protegidos contra escritura".

El disco está formado por una película de Mylar de forma circular con su superficie recubierta por óxido magnético, la información es por tanto grabada magnéticamente en él. Su superficie es dividida en pistas concéntricas separadas una de otra, por aproximadamente 0,02 pulgadas (48 pistas por pulgada). Cada pista ocupa unas 0,012 pulgadas (2) y normalmente es subdividida en sectores de igual longitud.

El disco tiene un agujero central al cual se fija el eje que lo hace girar, cerca a éste se tiene otro llamado agujero índice que, al alinearse con la apertura de la funda, provee de un ángulo de referencia en el que se define el comienzo de una pista. Para la identificación del comienzo de cada sector se han adoptado dos maneras diferentes, ya sea por medio de la escritura de información adicional en cada pista o por medio de agujeros -uno por cada sector- localizados en la circunferencia que contiene al agujero índice. Estas dos formas de identificar el comienzo de cada sector han separado a los discos floppy en dos clases principales: "soft sectored" y "hard sectored" respectivamente.

El tamaño externo de los discos se ha estandarizado y por el momento existen dos tipos cuyas dimensiones son: 8 × 8 pulgadas y 5¼ × 5¼ pulgadas, teniendo cada uno de ellos las características anteriormente indicadas.

1.2. LAS LECTORAS/ESCRITORAS DE DISCOS FLOPPY

Las lectoras/escritoras de discos floppy contienen todas las partes mecánicas y eléctricas que permiten guardar o recibir información del disco. Aquí nos limitaremos a dar una breve descripción de las partes que no tengan que ver con las señales de interface a las cuales se les dará una mayor importancia debido a que su comprensión es indispensable para el desarrollo de la presente tesis. Mayor información se puede en

contrar en los apéndices y bibliografía al final de este trabajo.

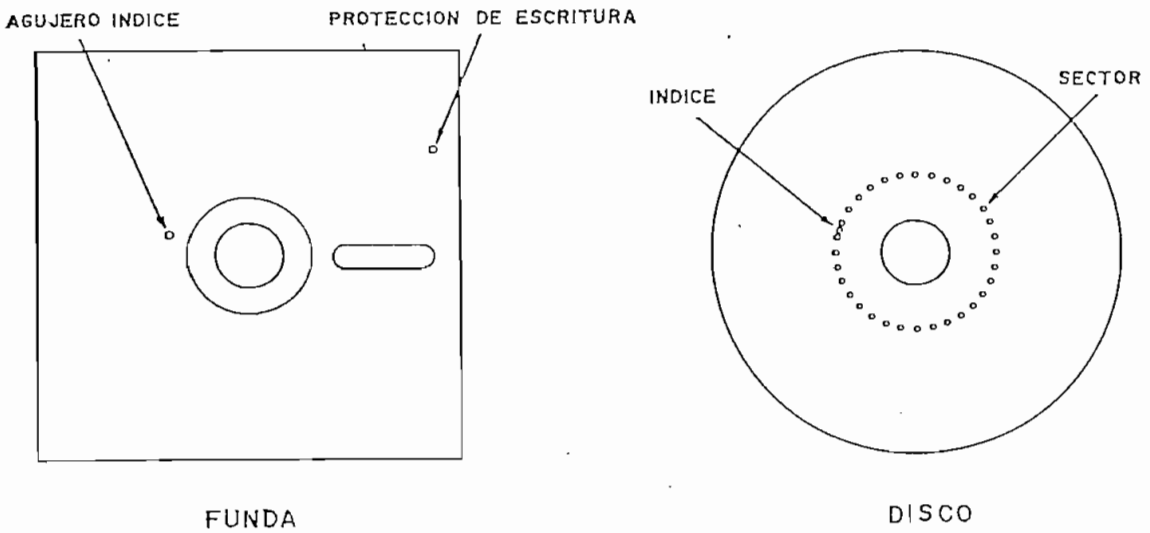


FIGURA 1

Las principales partes mecánicas tienen las siguientes funciones:

- Sujetar al disco en posición
- Girar al disco con una velocidad constante
- Mover la cabeza de lectura/escritura en pasos de igual longitud, ya sea hacia pistas internas (cercanas al centro del disco) o externas.
- Empujar a la parte expuesta del disco sobre la cabeza de lectura/escritura (este proceso debe entenderse como "bajar la cabeza")

El hecho de que se necesite bajar la cabeza para realizar un proceso de lectura o escritura prolonga la duración del

disco y de la cabeza de grabación pues no estarán en contacto a menos que esto sea requerido por el proceso que se esté desarrollando, esta característica extiende la vida útil de la cabeza a unos cinco años (Diez mil horas de contacto con el disco) y la del disco a unos dos años de mucho uso. (3),(4).

La cabeza de lectura/escritura no es más que una pequeña bobina con núcleos de ferrita usada para leer, grabar o borrar información del disco. La superficie de contacto con el disco está recubierta por una película de vidrio y ha sido diseñada para máxima transferencia de señal con un mínimo desgaste de la cabeza y el disco, contiene además elementos para borrar el espacio entre pistas de forma que el nivel de señal a ruido no se degrade al utilizar discos grabados en otra lectora/escritora. (6).

La rotación del disco se consigue por medio de un motor que lo hace girar a velocidad angular constante, este motor es usualmente de corriente alterna en discos de 8" mientras que en discos de 5¼" se usan preferentemente los de corriente continua, siendo estos últimos más estables a variaciones de voltaje de la línea ya que su velocidad es controlada electrónicamente por medio de un servomecanismo.(5).

Un segundo motor mueve la cabeza de lectura/escritura a las diferentes pistas en el disco, el movimiento se realiza "por pasos" de igual longitud, este tipo de motores son llama

dos "Stepper Motors".

La parte electrónica de las lectoras/escritoras se encarga de controlar el funcionamiento del sistema, así como de los procesos de interface con el sistema principal. Las funciones principales que se realizan en esta parte se describen a continuación:

- Detección del agujero índice, proceso que se realiza en base a un emisor de luz (usualmente un LED infrarrojo) y un detector luminoso, localizados en lados opuestos al disco de forma que cada vez que el agujero índice se alinea con el de la funda, exista una transmisión de luz desde el emisor al detector, éste último convierte esta señal luminosa en un pulso eléctrico llamado pulso índice.

- Generación de las diferentes fases eléctricas para hacer girar un cierto ángulo (siempre constante) al motor que mueve la cabeza en uno u otro sentido de acuerdo a una señal externa que indica la dirección deseada y otra que indica el momento en que debe producirse el giro.

- Comprobación de que el disco esté listo para cualquier proceso, es decir, que la puerta de acceso del disco esté cerrada, los niveles de voltaje y la velocidad angular del disco sean los correctos.

- Detección de que la cabeza se encuentra en la pista Ø ó

de que se trata de un disco protegido contra escritura, condiciones que son detectadas por medio de switches o sensores ópticos. Estas informaciones son enviadas al sistema principal en forma de señales eléctricas para su interpretación y uso.

- Activación del pulsador de bajado de la cabeza (usualmente por medio de un relé) de acuerdo a una señal externa.

- Especialmente en el caso de discos de 5¼", control electrónico de la velocidad del disco.

- Control de la cabeza de lectura/escritura de forma que se pueda guardar información o recobrarla del disco. Esta parte debe contener todos los amplificadores y filtros necesarios para este objeto, debe deshabilitar los circuitos de escritura cuando un disco está protegido y controlar los procesos de lectura o escritura de acuerdo a señales externas.

- En algunos casos, generación de voltajes regulados para uso de la lectora/escritora.

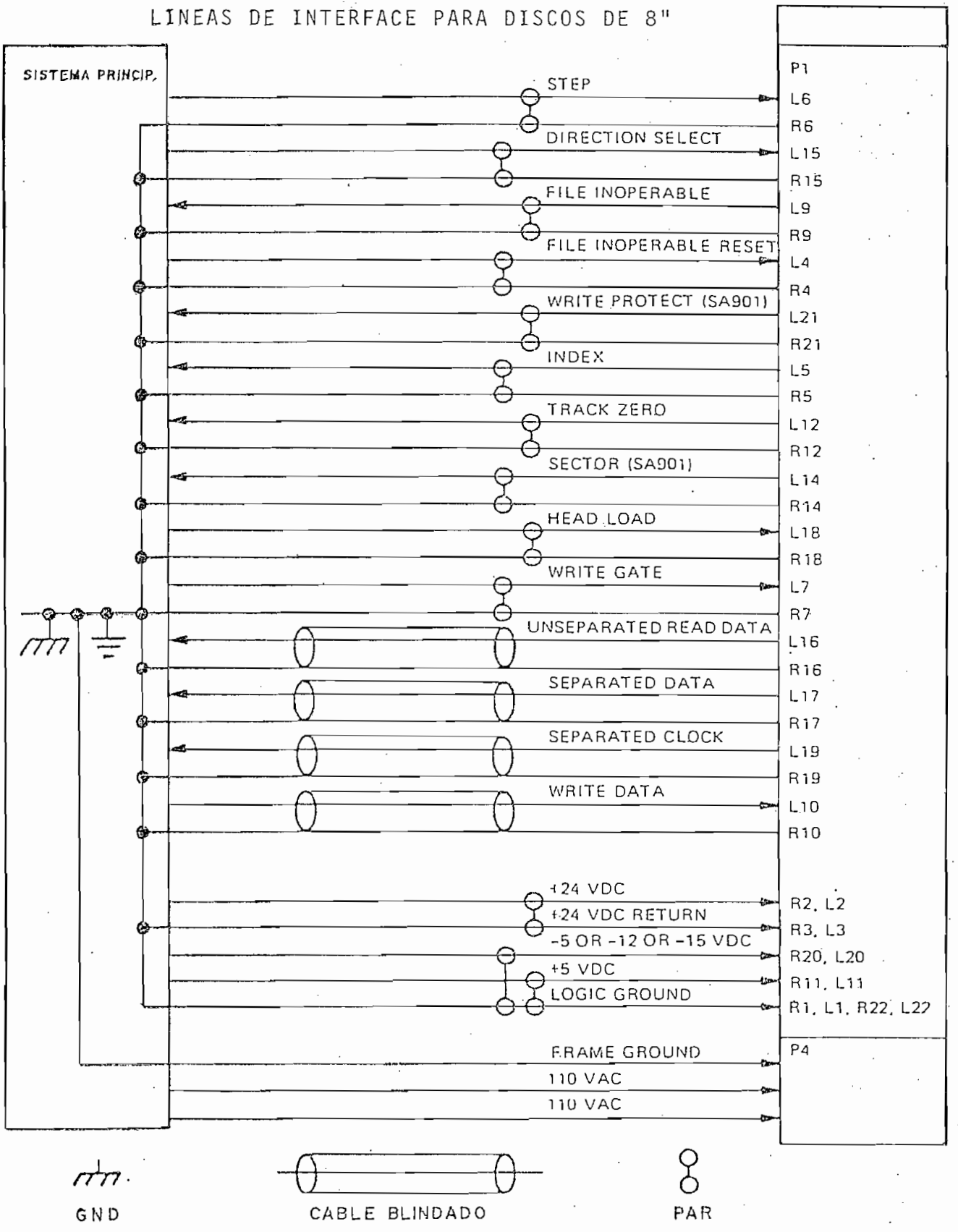
Las figuras 2 y 3 muestran sistemas típicos de interface para discos de 8" y 5¼" respectivamente, el significado de cada una de estas señales se explica a continuación.

1.2.1. SEÑALES DE INTERFACE ENVIADAS A LA LECTORA/ESCRITORA.

DIRECTION:

FIGURA 2

LINEAS DE INTERFACE PARA DISCOS DE 8"



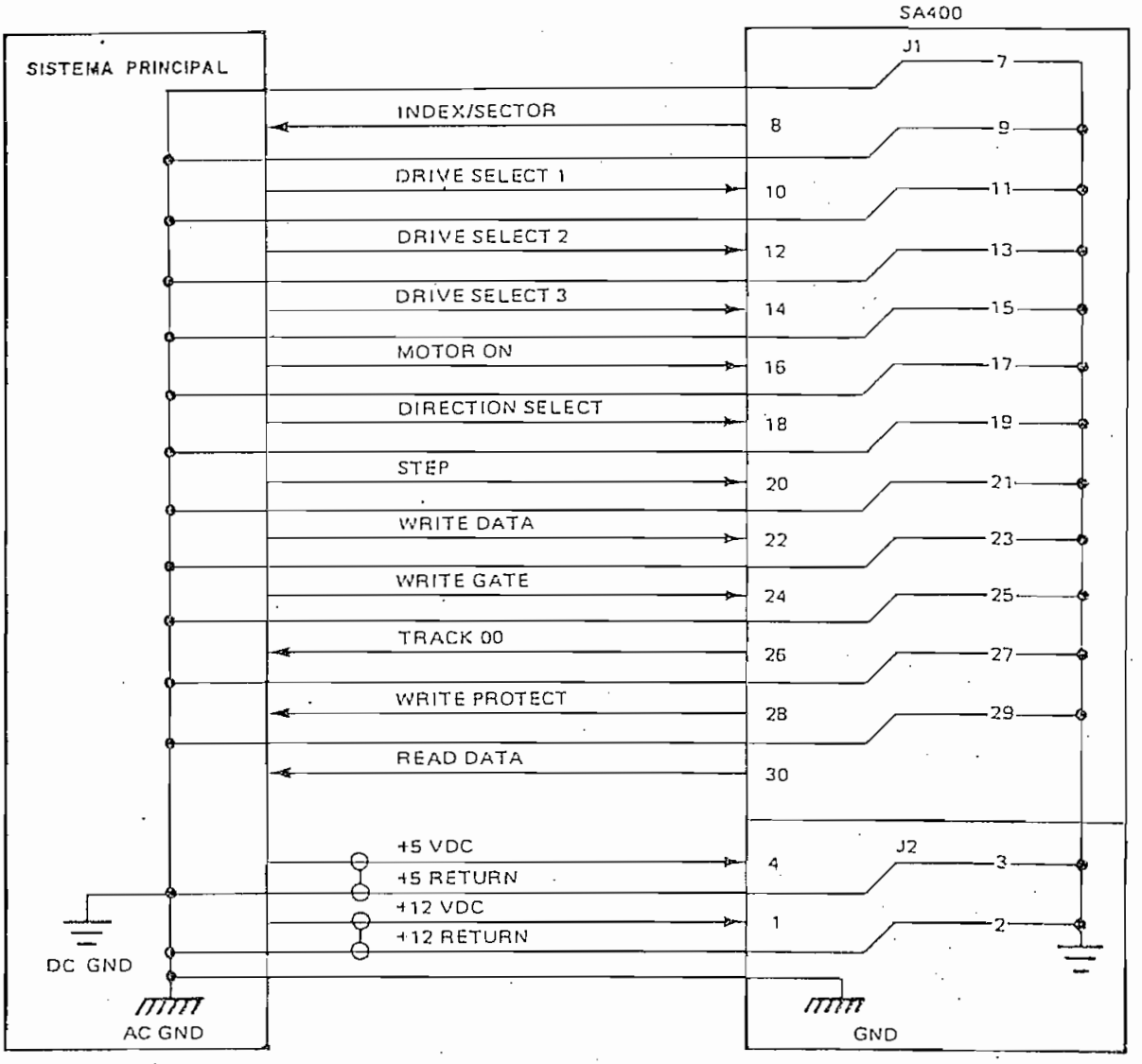


FIGURA 3

LINEAS DE INTERFACE PARA DISCOS DE 5¼"

Esta señal define la dirección de movimiento de la cabeza de lectura/escritura cuando se manda un pulso de "STEP". Un uno lógico define la dirección como "hacia afuera" y si un pulso es aplicado a la línea de STEP, la cabeza se moverá hacia pistas cada vez más distantes del centro del disco, lo contrario sucederá si el pulso es aplicado cuando la señal $\overline{\text{DIRECTION}}$ está en cero lógico. Hay que anotar que esta convención no siempre se mantiene y puede variar de un fabricante a otro.

$\overline{\text{STEP}}$

Esta señal de control produce el movimiento de la cabeza de lectura/escritura de acuerdo a la señal $\overline{\text{DIRECTION}}$. El movimiento de la cabeza empieza con el final del pulso aplicado a la línea de $\overline{\text{STEP}}$, es decir, con una transición positiva de esta señal. La duración de los pulsos aplicados, el tiempo mínimo entre pulsos y el tiempo mínimo que debe transcurrir entre la estabilización de la señal $\overline{\text{DIRECTION}}$ y la aplicación de los pulsos son condiciones que hay que tener en cuenta para el correcto funcionamiento del sistema, por esta razón están siempre especificados por el fabricante.

$\overline{\text{HEAD LOAD}}$

Esta señal controla el pulsador de bajado de la cabeza, un uno lógico en ella desactiva el pulsador mientras que un cero lógico lo activa, haciendo que la parte expuesta del disco se ponga en contacto con la cabeza, en algunos casos también se usa esta señal para desconectar parte de la alimentación de vol

taje al motor que mueve la cabeza de forma que se evite el calentamiento excesivo de este motor. En algunos sistemas de interface como el ilustrado en la figura 3, puede no existir la señal HEAD LOAD en cuyo caso el activador de bajado de la cabeza es activado cada vez que se encienda el motor que mueve al disco o cuando la lectora/escritora en cuestión sea seleccionada, el uso de una u otra condición para que se produzca el bajado de la cabeza es usualmente controlado por el usuario.

DRIVE SELECT

Usualmente son varias las líneas de DRIVE SELECT, estas son utilizadas especialmente en casos en que el sistema principal se conecta a varias lectoras/escritoras, cada una de las cuales tiene asignada una línea de DRIVE SELECT de forma que si el sistema principal manda un cero lógico por una de estas líneas será solamente la lectora/escritora asignada a esa línea que entre en funcionamiento aceptando y enviando señales de interface del o hacia el sistema principal, las demás lectoras/escritoras se desconectan del sistema es decir, no envían señales y hacen caso omiso a las que reciben. En algunos casos esta señal también indica que la lectora/escritora debe bajar la cabeza.

MOTOR ON

Esta señal es utilizada principalmente en discos de 5¼" su función es la de encender al motor que gira el disco, esto

tiene el objeto de extender la vida útil de este motor así como la de las partes asociadas a él, un cero lógico en esta señal de interface pone en movimiento este motor y aunque esto constituye una ventaja en lo que se refiere a la duración del motor, tiene también una desventaja pues se necesita esperar un cierto tiempo -especificado por el fabricante- para iniciar cualquier proceso en el disco, en discos de 8" en los que se prefiere tener un menor tiempo de acceso se mantiene normalmente encendido al motor en todo momento.

TRACK GRATER THAN 43

Esta señal informa a la lectora/escritora que su cabeza está posicionada en una pista mayor a la pista 43. Su función es controlar la corriente enviada a la cabeza de lectura/escritura en procesos de escritura, esto se debe a que en pistas internas hay una mayor agrupación de la información debido a que su circunferencia se ha reducido considerablemente en relación a la de pistas externas.

SIDE SELECT

Esta señal es utilizada en lectoras/escritoras que pueden leer o escribir discos por sus dos lados, un cero en esta línea indica que se ha seleccionado el lado Ø del disco (el único que puede ser utilizado en discos de un solo lado) mientras que un uno lógico selecciona el lado opuesto a este.

WRITE DATA

Por esta línea de interface el sistema principal manda una señal que debe ser grabada en el disco en forma de cambios de polarización magnética, la señal no es más que pulsos correspondientes a los datos serializados más señales de reloj usadas para sincronizar la recepción de los mismos. La forma de esta señal será mejor comprendida cuando se estudien los diferentes formatos de grabación.

WRITE GATE ó WRITE ENABLE

Esta línea de interface controla la escritura de información en el disco. Un uno lógico en esta línea impide la escritura de datos en el disco, mientras que un cero lógico a la vez que habilita la escritura, inhabilita a los circuitos que controlan el movimiento de la cabeza.

1.2.2. SEÑALES DE INTERFACE PROVENIENTES DE LA LECTORA/ESCRITORA.

TRACK Ø

Esta señal se pone en cero lógico cuando se ha llegado a la pista cero del disco a la vez que impide que el motor que mueve la cabeza continúe moviéndola hacia pistas más externas. Esta señal es de gran importancia pues provee de un punto de referencia para encontrar las demás pistas en el disco.

READY

Un cero lógico en esta línea de interface indica que el

disco ha sido introducido correctamente, que la puerta está cerrada y que los niveles de voltaje y la velocidad del disco son correctos, un uno lógico indicará que cualquiera de estas condiciones no se cumple.

WRITE PROTECT

Un cero lógico en esta señal indicará que un disco protegido contra escritura ha sido introducido en la lectora/escritora, de cumplirse esta condición no se podrá escribir información sobre el disco aún cuando el sistema principal la mande.

INDEX

Esta señal indica que se ha detectado el agujero índice en el disco, la indicación se la hace enviando un pulso de corta duración a través de esta línea, la misma que permanece en uno lógico hasta la siguiente detección del agujero índice. Esta señal provee al sistema principal de una referencia que indica el comienzo de una pista y como es lógico se tendrá un pulso por cada revolución del disco.

READ DATA ó UNSEPARATED READ DATA

Por esta línea de interface se envía la señal que se lee del disco, la misma que tendrá la misma forma que aquella que se envía al disco para su escritura por la línea de WRITE DATA es decir, que contendrá tanto señales de datos como señales de reloj como se explica en la parte de formato de escritura.

SEPARATED DATA

Esta señal no siempre es entregada por las lectoras/escritoras y solamente se la tiene disponible en aquellas que tienen un separador de datos y señal de reloj interno, de esta forma, la señal enviada por la línea de SEPARATED DATA no es más que la transmisión serial de los datos escritos en el disco. Debido a la gran variedad de circuitos separadores de datos y señal de reloj se ha asignado todo un capítulo al estudio de estos circuitos, allí se podrá encontrar mayor información sobre esta señal.

SEPARATED CLOCK

Por esta línea de interface se envía la señal de reloj proveniente del separador de datos y como sucede con la señal de SEPARATED DATA, no siempre es proveída por la lectora/escritora.

1.3. FORMATOS DE ESCRITURA

1.3.1 FORMAS DE CODIFICACION DE LA INFORMACION PARA SU ALMACENAMIENTO EN EL DISCO.

Antes de explicar los diversos formatos de escritura utilizados, se deben conocer las formas en que son codificados los datos para su escritura en el disco. Como se dijo anteriormente, la señal que se escribe en el disco está formada por dos señales distintas, una proveniente de la serialización de los

datos y otra que corresponde a una señal de sincronismo, el uso de esta última señal se debe a que la velocidad angular del disco puede variar en pequeños porcentajes por diversos factores, por lo que, de no existir esta señal se perdería el sincronismo en el circuito que recobra los datos, esto sucedería especialmente en el caso de leer largas cadenas de ceros las cuales no habrán producido cambios de polarización magnética en la superficie del disco cuando fueron escritos.

Debido a lo explicado anteriormente, el uso de una señal de reloj se hace imprescindible para recuperar la información escrita en el disco. Puesto que la forma más sencilla de lograr esto es la de enviar un tren de pulsos de reloj entre los cuales se intercalan los bits de información, fue esta la forma de codificación que se utilizó originalmente y es conocida como FM por ser, como se muestra en la figura 4, una forma

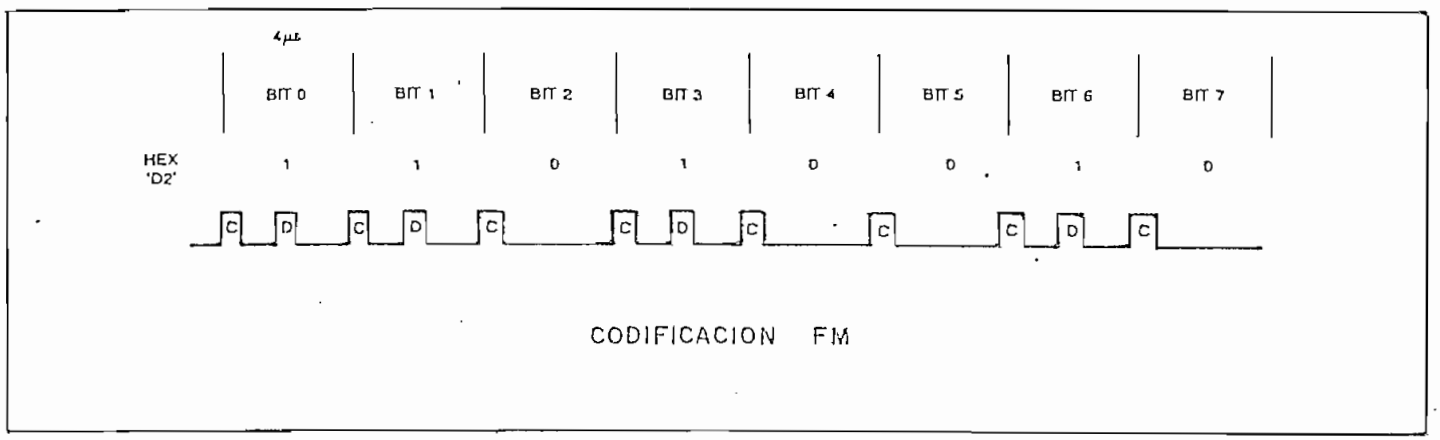


FIGURA 4

de modulación en frecuencia. De forma que no se pierda completamente el espacio del disco ocupado por estos pulsos de reloj, se manda también información por ellos, que sirve para facilitar la detección de ciertas áreas específicas de cada pista en el disco como se explicará al estudiar los formatos de grabación, este hecho convierte a los pulsos de reloj en "bits" de reloj".

El sistema FM ha sido utilizado con mucho éxito y lo sigue siendo, sin embargo, debido a que los bits de reloj ocupan tanto espacio en el disco como los de datos, se buscó un mejor sistema de codificación, al cual se lo ha denominado MFM (FM modificado), una señal típica en este sistema ha sido ilustrado en la figura 5, su característica es la de que los bits de reloj son escritos sólamente en el caso de que tanto el último bit de datos escrito, como el que será escrito a continuación, sean ceros, de esta forma se puede escribir el doble de información

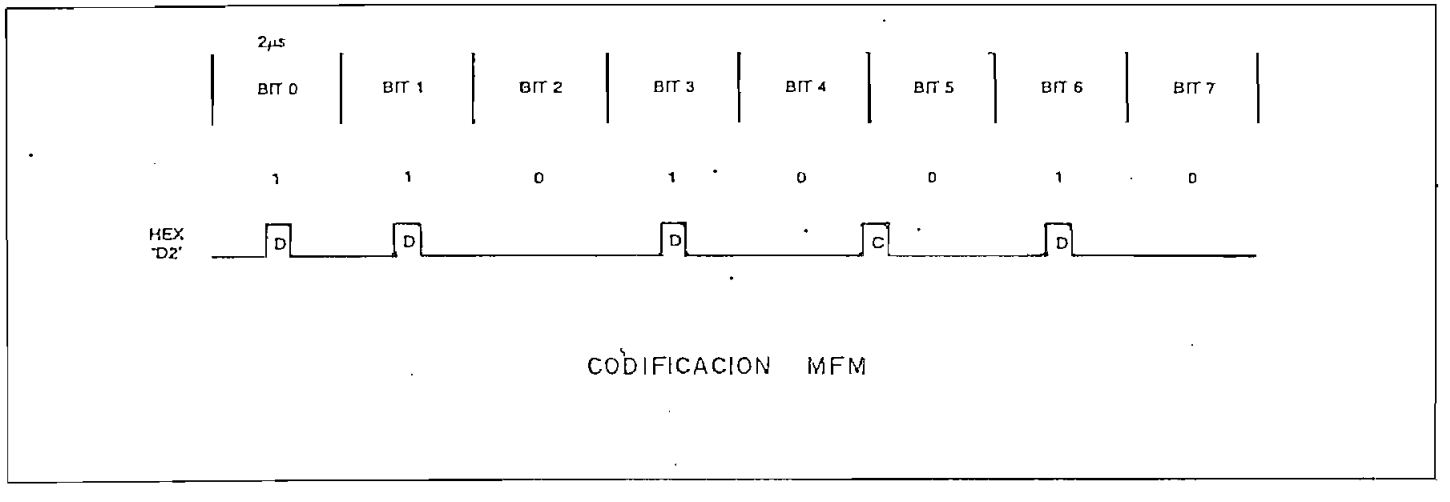


FIGURA 5

en el disco que utilizando el sistema FM sin que aumente el número máximo de cambios de polarización magnética por unidad de longitud en la superficie del disco. Este sistema es también llamado de "densidad doble" y, si bien permite almacenar mayor cantidad de información, dificulta la separación de las señales de datos y reloj de la señal leída del disco, para facilitar este proceso se utiliza la precompensación de escritura que será explicada a continuación.

El uso de precompensación en la escritura se debe a que al grabar señales como las de las figuras 4 ó 5 en el disco, se producen desplazamientos en la posición de aquellos bits que no se encuentran a igual distancia del bit que los precede y el que los sigue, el desplazamiento está dirigido hacia el punto medio entre estos dos bits (7), por esta razón la precompensación produce un desplazamiento en sentido opuesto al que producirá la grabación de forma que los bits se encuentren en su posición nominal en la lectura.

Usualmente no se utiliza precompensación en el sistema FM o de densidad simple, mientras que en densidad doble es usada especialmente para discos de 8". Puesto que la precompensación de escritura es una función de la lectora/escritora utilizada, la cantidad de precompensación será usualmente especificada por el fabricante y a su valor típico está en el rango de los 100 - 300 nanosegundos. (8).

La decisión de si un bit debe ser escrito "antes" ó "después" de su posición nominal debe ser tomada por el circuito que crea la señal $\overline{\text{WRITE DATA}}$ en base a los criterios ya explicados.

En la figura 6 se ilustra un circuito de precompensación cuyo funcionamiento se basa en un generador de cuatro fases como elemento de retraso. Cuando un pulso es mandado por la lí

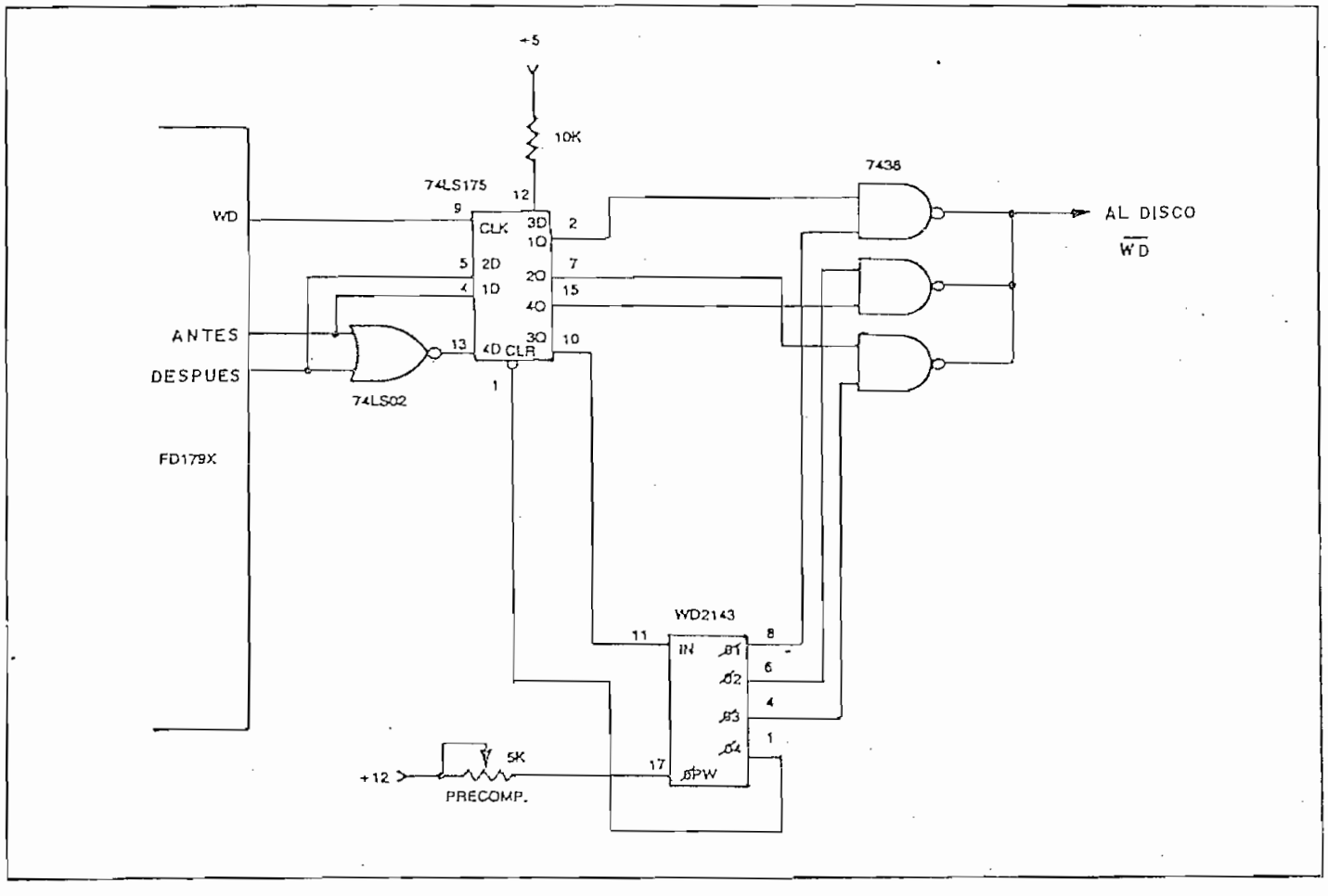


FIGURA 6

nea de WRITE DATA, las entradas de los flip-flops tipo D son copiadas en sus salidas, y ya que la entrada 3D está siempre en uno lógico, se producirá una transición positiva en 3Q que dará inicio a la generación de las fases como se indica en la figura 7, de esta forma si la señal "antes" estuvo en uno lógico será la fase número uno la que se escriba en el disco, si lo estuvo la señal "después" la fase tres será escrita, y si ninguna de estas señales estuvo en uno lógico entonces el bit es escrito en su posición nominal correspondiente a la fase dos. la fase

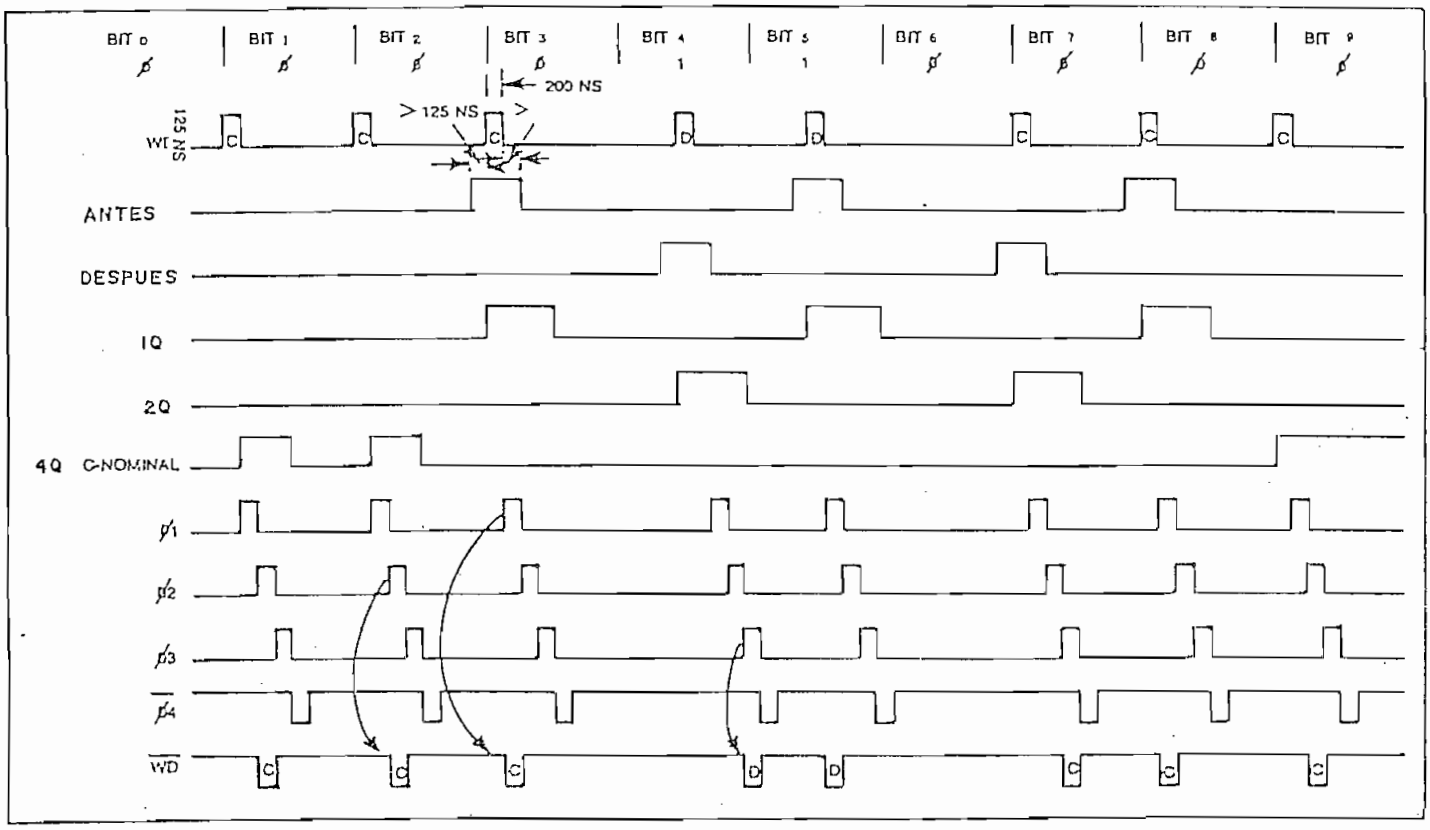


FIGURA 7

número cuatro es utilizada para poner en cero lógico todas las salidas Q de forma que se pueda producir una transición positiva en Q3 con el siguiente pulso en WRITE DATA así como impedir el envío de señales al disco. La ventaja de este método es la de que permite variar la cantidad de precompensación variando la duración de los pulsos de cada fase.

Otro método de precompensación es el digital, un circuito que utiliza este método se ilustra en la figura 8. La cantidad de precompensación en este caso no puede ser variada fácilmente y en el circuito indicado tiene un valor de 250 ns, u

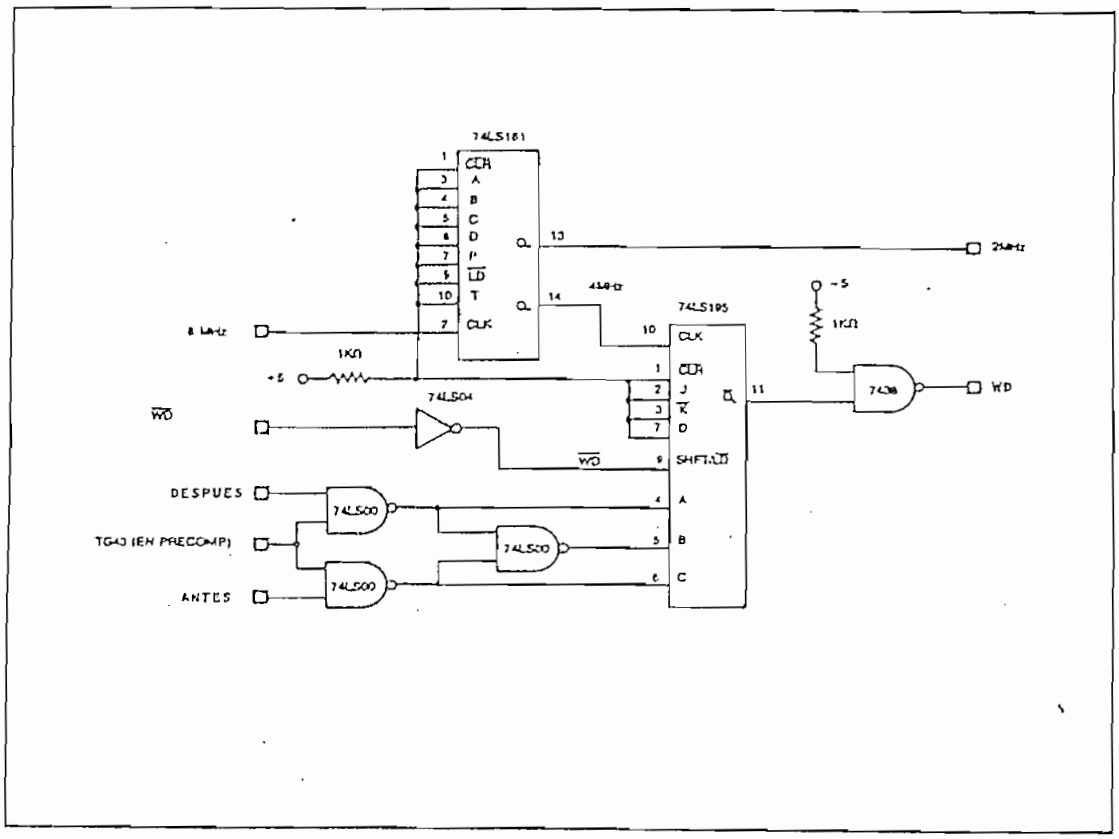


FIGURA 8

na señal de 4 MHz es utilizada por un registro de desplazamiento de cuatro bits como señal de reloj, mientras la señal $\overline{\text{WRITE DATA}}$ esté en uno lógico existirá solamente un desplazamiento de "unos" hacia la salida Q_D pues las entradas seriales J y $\overline{K}$ están en uno lógico también, al producirse un pulso en la señal $\overline{\text{WRITE DATA}}$ un cero lógico es cargado en una de las entradas paralelas A, B ó C de acuerdo a si el bit debe ser escrito antes de, en su posición nominal o después de ella, el cero lógico es luego desplazado por el registro hasta que alcanza la salida Q_D por lo que la demora introducida vendrá dada por el número de desplazamientos de que ha sido objeto. En el circuito indicado la precompensación se realiza solamente en pistas mayores a la 43 como se usa generalmente.

Habiendo estudiado ya la forma de la señal escrita en el disco, veamos ahora su capacidad de almacenamiento de datos. La capacidad de un disco depende de su tamaño, velocidad angular, número de pistas, tiempo entre un bit de datos y el siguiente y, el método de grabación utilizado: FM ó MFM. Como ya se indicó el sistema MFM permite almacenar el doble de información que el FM pues el tiempo entre bits de datos puede ser reducido a la mitad. La velocidad de rotación del disco varía también dependiendo del tamaño del disco, en los sistemas actuales se utilizan velocidades de 360 y 300 RPM para discos de 8 y 5¼ pulgadas respectivamente, el número de pistas por disco no es tampoco una constante, usualmente se tienen 77 pistas en discos de 8 pulgadas y 35 en los de 5¼ pulgadas. Todas las

características indicadas hacen que la capacidad de almacenamiento de datos varíe de la forma indicada en la Tabla I.

TABLA I

TAMAÑO (Pulgad.)	DENSIDAD	CAPACIDAD NOMINAL SIN FORMATO (Bytes) POR PISTA	POR DISCO	TIEMPO ENTRE BITS DE DATOS (Microsegundos)
5¼	SIMPLE	3.125	109.375	8
5¼	DOBLE	6.250	219.750	4
8	SIMPLE	5.208	401.016	4
8	DOBLE	10.416	802.032	2

NOTA: La capacidad nominal por disco ha sido calculada para 35 pistas en discos de 5¼ pulgadas y 77 para los de 8 pulgadas, en discos de doble lado la capacidad total por disco será el doble de la indicada.

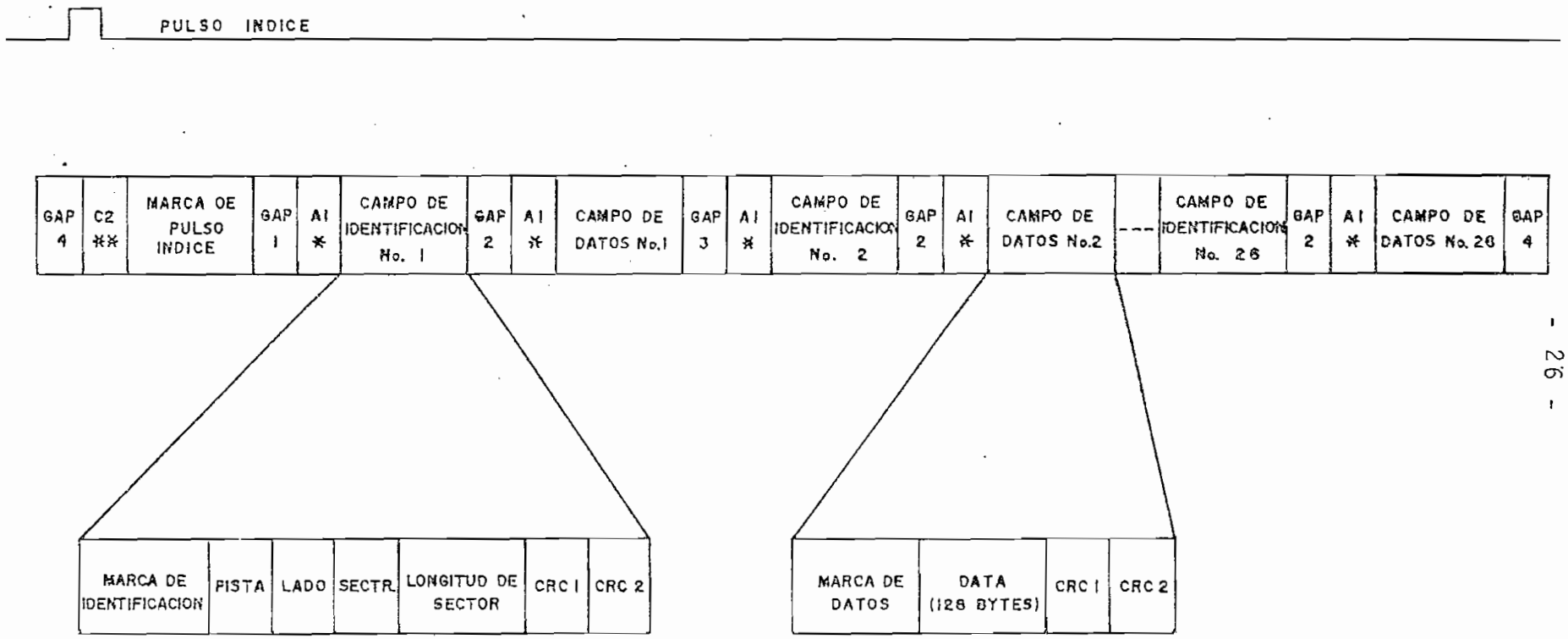
1.3.2 FORMATOS DE ESCRITURA.

Una vez comprendida la forma de grabación de la información en el disco y su capacidad de almacenamiento, podemos pasar a estudiar los formatos. La necesidad de utilizar formatos se debe a que la información que se almacena en el disco debe ser ordenada en cierta forma que permita su recuperación fácilmente. El primer paso a este ordenamiento es el de dividir a la superficie del disco en varias pistas como ya fue indicado pero esto no es suficiente pues la cantidad de información que puede ser escrita en una pista es mucho mayor que las necesida

des usuales, por esta razón se subdivide a cada pista en varios sectores, de esta forma el usuario puede identificar el lugar en que se almacena su información especificando la pista y el sector en que se encuentra. Para poder comprobar que la cabeza de grabación se encuentre sobre la pista y el sector especificados debe escribirse información adicional de forma que al ser leida se pueda identificar el inicio de una pista (en conjunto con el pulso índice) o de un sector, su número y especialmente si la información que se lee es de identificación o de datos. Esta información adicional en conjunto con ciertos espacios utilizados para sincronización de los circuitos de separación de datos y los espacios para los datos del usuario es lo que se conoce como formato.

En la figura 9 se indica la forma general del formato más utilizado en discos floppy de 8 pulgadas, se lo conoce como formato IBM por ser esta compañía la que lo utilizó para sus discos por primera vez. (1) Algo parecido se puede encontrar en la bibliografía adjunta, para discos de 5¼ pulgadas.

Los espacios de sincronización son también llamados "gaps" por su nombre en inglés, su función no es únicamente la de permitir la sincronización del circuito separador de datos sino que además producen una cierta demora para que el sistema que lee los datos pueda decidir cual será su siguiente paso, por último proveen de una separación entre sectores lo suficien



* PULSO DE RELOJ ELIMINADO ENTRE BITS 4 Y 5 (USADO SOLO EN MFM)

** PULSO DE RELOJ ELIMINADO ENTRE BITS 3 Y 4 (SOLO EN MFM)

FIGURA 9

FORMATO IBM

temente grande como para que pequeñas variaciones en la velocidad de rotación del disco no hagan que nuevos datos que se escriban puedan pasar hasta el inicio del siguiente sector.(10).

El comienzo de cada pista, campo de identificación ó campo de datos están precedidos por "marcas de identificación". En el sistema FM pueden ser facilmente detectadas pues, cada una tiene asignado un número hexadecimal, tanto en su señal de datos como en la de reloj, como se indica en la Tabla II.

La forma de la señal escrita en el disco para cada una de ellas puede ser entendida observando el ejemplo de la figura 10 en el que se representa la señal escrita para una marca de pulso índice; otros ejemplos pueden ser encontrados en los apéndices y bibliografía de la presente tesis.

En el sistema MFM no se puede enviar todo un byte por la señal de reloj pues algunos bits de esta señal no son escritos en el disco, por esta razón las marcas de identificación tienen asignado un número hexadecimal tan sólo para su señal de datos, su señal de reloj es escrita en la forma acostumbrada, además, antes de cada marca se escribe otro byte de datos en cuya correspondiente señal de reloj no se escribe uno de los pulsos que de acuerdo con el sistema MFM debería ser escrito, en la Tabla II pueden verse los códigos asignados a cada marca y a los bytes que las anteceden.

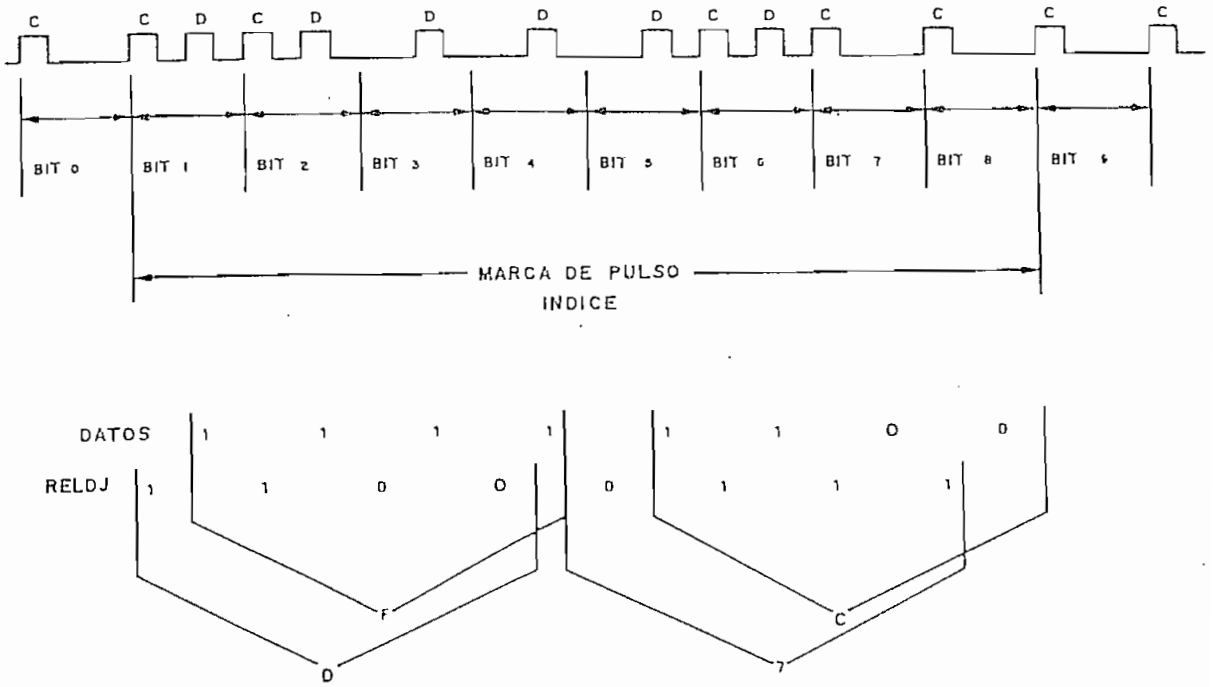


FIGURA 10

MARCA DE PULSO INDICE

TABLA II

FUNCION	CODIGO EN FM		CODIGO EN MFM	
	DATOS	RELOJ	DATOS	BYTE PRECEDENTE
Marca de pulso índice	FC	D7	FC	C2*
Marca de campo de identificación	FE	C7	FE	A1**
Marca de campo de datos	FB	C7	FB	A1**

* Pulso de reloj no escrito entre los bits 3 y 4 del byte de datos.

** Pulso de reloj no escrito entre los bits 4 y 5 del byte de datos.

En algunos casos especiales la marca del campo de datos puede ser cambiada por el código F8 para la señal de datos y C7 para la de reloj, a esta nueva marca se la denomina en inglés "Deleted Data Address Mark" e indica que el campo de datos escrito a continuación tiene alguna característica específica definida por el usuario.

Con el fin de detectar errores de lectura o escritura se escriben dos bytes adicionales al final de los càmpos de identificación y el de datos, estos bytes son conocidos como CRC (en inglés Cyclic Redundancy Check). Para generar los bytes de CRC se considera a la información como un polinomio, por ejemplo, si la información tiene la forma: 110010011 puede ser considerada como un polinomio: $M(X) = X^8 + X^7 + X^4 + X + 1$, si se desea incluir a continuación dos bytes de CRC podemos considerar que en conjunto con la información forman un solo mensaje es decir que el polinomio M(X) debe ser desplazado dieciséis posiciones a la izquierda para dar cabida a los dos bytes de CRC, lo que equivale a multiplicarlo por X^{16}. Si a este nuevo polinomio se lo divide -módulo dos- por otro llamado "polinomio generador" G(X) de orden 16, obtendremos un cuociente Q(X) y un residuo R(X). Puede demostrarse que $X^{16} M(X) + R(X)$ módulo dos, es exactamente divisible para G(X). (11). Cabe anotar que la división módulo dos se la realiza en base a restas módulo dos y que en este tipo de operaciones tanto la suma como la resta producen los mismos resultados. (12).

Para la grabación de información en discos floppy se utiliza el polinomio: $X^{16} + X^{12} + X^5 + 1$ como polinomio generador y ya que $X^{16} M(X)$ tiene sus últimos dos bytes iguales a cero, la suma módulo dos $X^{16} M(X) + R(X)$ tendrá sus últimos dos bytes iguales a $R(X)$, de esta forma la información puede ser escrita normalmente en el disco pero aumentando a continuación de ella los dos bytes del residuo $R(X)$ o bytes de CRC. Durante la lectura se puede hacer la división módulo dos de el conjunto, es decir, de la información y los bytes de CRC para el polinomio generador, en este caso el residuo debe ser igual a cero si no se ha producido algún error en la escritura o en la lectura.

Un circuito digital que realiza esta operación se ilustra en la figura 11, en este circuito se inicializa primero a todos los flip-flops con cero lógico, por medio de la señal de CLEAR, luego cada bit de datos en el campo, empezando desde el bit más significativo de la marca de identificación, son desplazados hacia la entrada del circuito enviando un pulso en la entrada de reloj por cada bit que se envía, para esto la señal de DATOS/CRC debe ser puesta en uno lógico, cuando todos los bits hayan sido pasados por el circuito los flip-flops quedarán con el valor de los bits de CRC correspondientes a la información enviada como se indica en la figura, si se está escribiendo en el disco se puede enviar a continuación cada bit de CRC poniendo la señal de DATOS/CRC en cero lógico y enviando cuatro pulsos a la señal de reloj. Si se está leyendo informa-

ción del disco, se puede realizar un proceso parecido esto es, poner todos los flip-flops en cero lógico por medio de la señal de CLEAR, y pasar a todos los bits de la información y los bits de CRC manteniendo la señal DATOS/CRC en uno lógico, en este caso el valor final almacenado en todos los flip-flops debe ser igual a cero si no se ha producido ningún error (lo que puede ser detectado por medio de una compuerta OR de cuatro entradas). (13). Cabe anotar que el mismo proceso puede ser realizado por medio de programación.

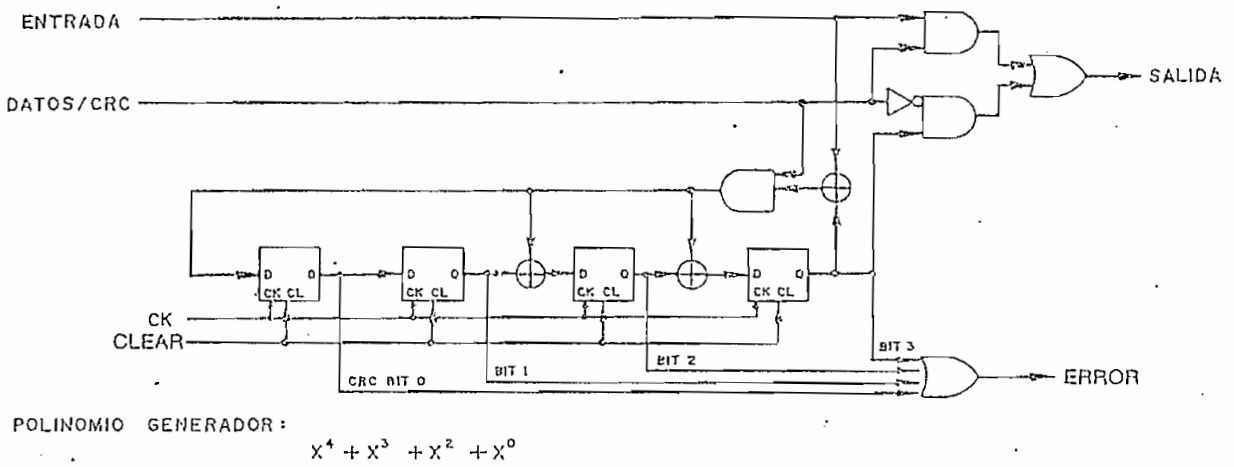


FIGURA 11

GENERADOR DE CRC.

El formato IBM no está limitado a 128 bytes por sector sino que sin perder compatibilidad se puede tener longitudes de 256, 512, 1024 bytes por sector, desde luego entre mayor sea la longitud del sector, menor será el número de sectores que

pueden entrar en una pista. Al tener 256 bytes por sector o más, se puede aprovechar en mejor forma la capacidad de almacenamiento del disco ya que con menos sectores por pista se eliminan muchos de los espacios entre sectores así como algunos bytes de identificación. La tabla III sumariza las longitudes de sector más utilizadas para cada tamaño de disco y la capacidad de almacenamiento de datos para cada caso, además se indica el tiempo que transcurre entre la escritura o lectura de dos bytes consecutivos. Cabe anotar que la capacidad de almacenamiento de datos puede ser duplicada utilizando discos de doble lado.

TABLA III

TAMAÑO (Pulg.)	DENSIDAD	BYTES POR SECTOR	NUMERO DE SECTORES	CAPACIDAD DE ALMACENAMIENTO POR PISTA (Bytes)	POR DISCO (Bytes)	TIEMPO ENTRE BYTES (Microsegun.)
5¼	SIMPLE	128	18	2304	80.640 *	64
5¼	DOBLE	256	18	4608	161.280	32
5¼	SIMPLE	256	10	2560	89.600	64
8	SIMPLE	128	26	3328	256.256 **	32
8	DOBLE	256	26	6656	512.512	16
8	SIMPLE	256	15	3840	295.680	32

* Basado en 35 pistas por disco.

** Basado en 77 pistas por disco.

Se pueden utilizar otros formatos que a pesar de no ser compatibles con el formato IBM sean de mayor utilidad para el usuario, algunos ejemplos de ellos se pueden encontrar en los apéndices y bibliografía de esta tesis, allí se puede encontrar también información sobre los formatos utilizados en discos del tipo "hard sectored" que por su sencillez no han sido tratados aquí.

C A P I T U L O S E G U N D O

" CONTROLADOR E INTERFACE PARA DISCOS FLOPPY "

2. CONTROLADOR DE DISCOS FLOPPY E INTERFACE.

En el capítulo anterior se dio una descripción de los discos floppy, la forma de almacenar la información en ellos y las lectoras/escritoras de este tipo de discos, en este capítulo se estudiarán los circuitos que controlan el funcionamiento de las lectoras/escritoras así como los procesos de interface con el sistema principal.

Un diagrama de bloques general de un sistema acoplado a un periférico se muestra en la figura 12. La interface permite la interconexión de el sistema principal con un periférico, las señales de control y de datos son enviadas al periférico por medio de un "controlador de periférico" que convierte las señales de control y datos enviadas por la interface en aquellas señales específicas requeridas por el periférico. (14).

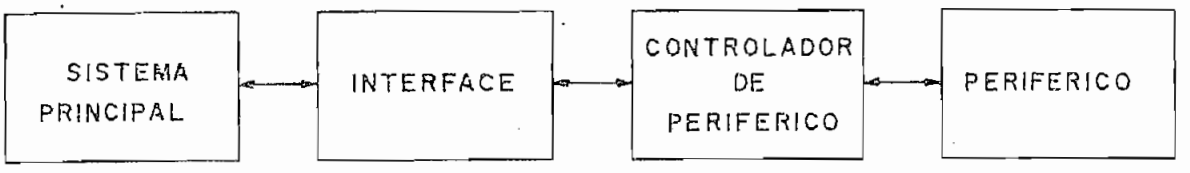


FIGURA 12

En el caso de discos floppy el controlador debe encar-

garse de las siguientes funciones principalmente:

- Recibir y transmitir datos en forma paralela desde o hacia el sistema principal. (En conjunto con la interface).
- Serializar los datos y codificarlos de acuerdo al sistema utilizado FM ó MFM.
- Realizar los procesos de precompensación si son utilizados.
- Indicar si se desea leer o escribir información en el disco.
- Impedir cualquier proceso si la lectora/escritora no está lista y en particular la escritura de un disco protegido contra escritura.
- Encontrar la pista número ØØ, para lo cual se deben mandar los pulsos necesarios por la señal de $\overline{\text{STEP}}$, actualizar la señal $\overline{\text{DIRECTION}}$ y esperar hasta que la señal $\overline{\text{TRACK ØØ}}$ se ponga en su estado activo.
- Luego de haber encontrado la pista ØØ, el controlador debe ser capaz de generar los pulsos necesarios en la línea $\overline{\text{STEP}}$ para posicionar la cabeza sobre cualquier pista del disco, en conjunto con la señal $\overline{\text{DIRECTION}}$, indicando además si la cabeza se encuentra en una pista mayor a la número 43.
- Producir demoras de tiempo suficientes como para que aquellos procesos en la lectura/escritora que producen movimientos mecánicos lleguen a un estado estable.

- Recibir la señal leida del disco, separar las señales de reloj y de datos e identificar si la información leída está formada por datos del usuario o si corresponde a alguna de las partes del formato.

Algunas de las funciones indicadas pueden ser realizadas utilizando programación por lo que antiguamente se las realizaba directamente en el sistema principal, ocupando tiempo y espacio de memoria en él pero ahorrando muchos circuitos en el controlador que aunque podrían realizar estas funciones, encarecerían al sistema completo. Actualmente, debido a la extensa utilización de discos floppy, se han creado circuitos integrados específicamente diseñados para actuar como controladores de las lectoras/escritoras de este tipo de discos, liberando de esta manera al sistema principal de muchas de las tareas que tenía a su cargo, haciendo que los discos floppy puedan incluso ser utilizados para sistemas pequeños además de haber disminuido el tamaño y costo del controlador para discos floppy.

2.1. METODOS DE SEPARACION DE SEÑALES DE DATOS Y DE RELOJ.

Una de las funciones más importantes que debe realizar el controlador es la de separar las señales de datos y de reloj de la señal leída del disco, por este motivo y por el hecho de que normalmente esta función no está implementada directamente en los circuitos integrados que actúan como controla-

dores, se van a estudiar aquí algunos de los métodos de separación especialmente aquellos novedosos que usualmente no son encontrados en la bibliografía relacionada con este tema.

Los circuitos de separación varían de acuerdo al sistema de grabación utilizado: FM o MFM. Dentro de cada sistema se pueden tener dos tipos de circuitos, los primeros son llamados circuitos de "separación completa", separan verdaderamente las señales de datos y de reloj de la señal enviada por la lectora/escritora. Otros circuitos separadores producen simplemente una señal de reloj sincronizada con la señal recibida, en ellos cada ciclo positivo o negativo de la señal producida contiene a un bit, ya sea de datos o de reloj (no importa la polaridad), su utilización es factible gracias a que la verdadera separación puede ser realizada por algunos circuitos integrados que sirven como controladores, de esta forma el circuito separador puede ser mucho más sencillo, especialmente en el caso del sistema MFM.

En la figura 13 se encuentra uno de los circuitos utilizados para la separación completa de las señales de datos y de reloj para el sistema FM; su funcionamiento es el siguiente: Los pulsos enviados por la lectora/escritora son acondicionados por medio del monoestable 10b, de esta manera pulsos negativos de duración constante son enviados a un demultiplexer 1 a 2 formado por las compuertas 3b, 4c y 4d, las dos salidas

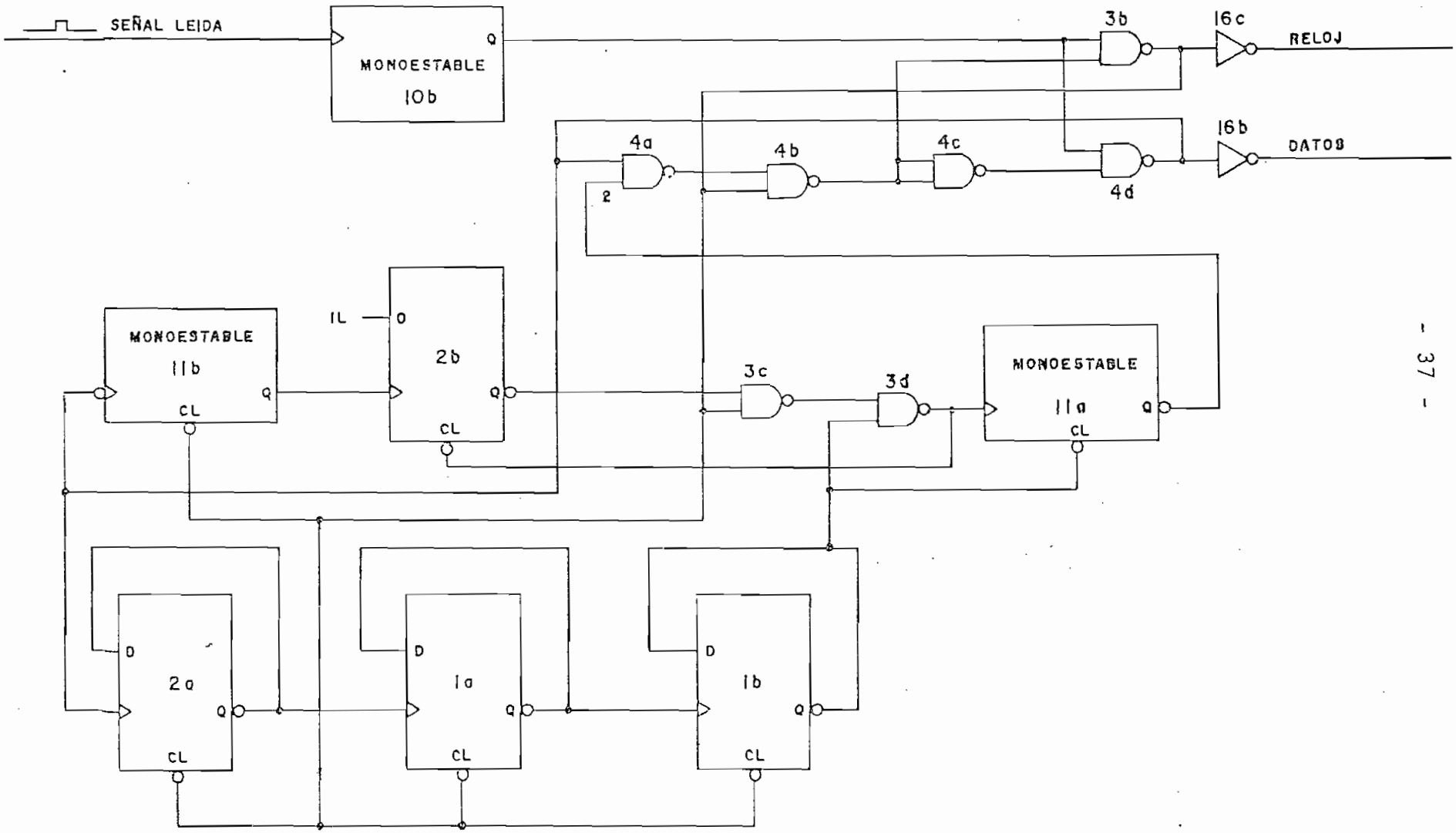


FIGURA 13

CIRCUITO SEPARADOR DE DATOS

del mismo son complementadas por medio de las compuertas 16c y 16b para obtener pulsos positivos correspondientes a la señal de reloj y a la de datos, la señal de selección está unida a una de las entradas de la compuerta 4a; cuando ésta se encuentra en cero lógico permite la salida de la señal producida en 10b por la línea de datos, mientras que si se encuentra en uno lógico, la señal de 10b pasa hacia la línea de reloj. Las compuertas 4a y 4b mantienen la selección mientras exista un pulso en las líneas de reloj o de datos; de esta forma, los pulsos pasan completos sin importar que mientras estén presentes se seleccione otra salida.

La señal de selección es producida por un monoestable que puede ser disparado de dos maneras diferentes, la primera emplea la salida de la compuerta 3b (pulsos negativos de reloj) enviando esta señal por medio de las compuertas 3c y 3d a la entrada de disparo con transición negativa de ella, la segunda forma emplea la salida de 4d (pulsos negativos de datos) para disparar al monoestable 11b que produce una demora suficiente como para que su desactivación ocurra ligeramente después del instante que corresponde a un nuevo pulso de reloj (es preferible demorar un poco más de forma que si existe una variación pequeña de la posición del pulso de reloj con respecto a su posición nominal, no se envíe al pulso a la salida de datos). Al desactivarse el monoestable 11b, un cero lógico se produce en la salida 2b, este cero lógico produce un pul

so en la salida de 3d debido a la realimentación a la entrada de borrado de 11b, el pulso producido se usa para pisparar al monoestable de selección 11a. De esta forma 11a es siempre disparado ya sea por medio de un pulso de reloj o uno de datos pues en el sistema FM no puede darse el caso que dos pulsos consecutivos uno de reloj y otro de datos no se encuentren presentes, cabe notar que al dispararse 11a se selecciona la salida de datos en espera del siguiente pulso de datos.

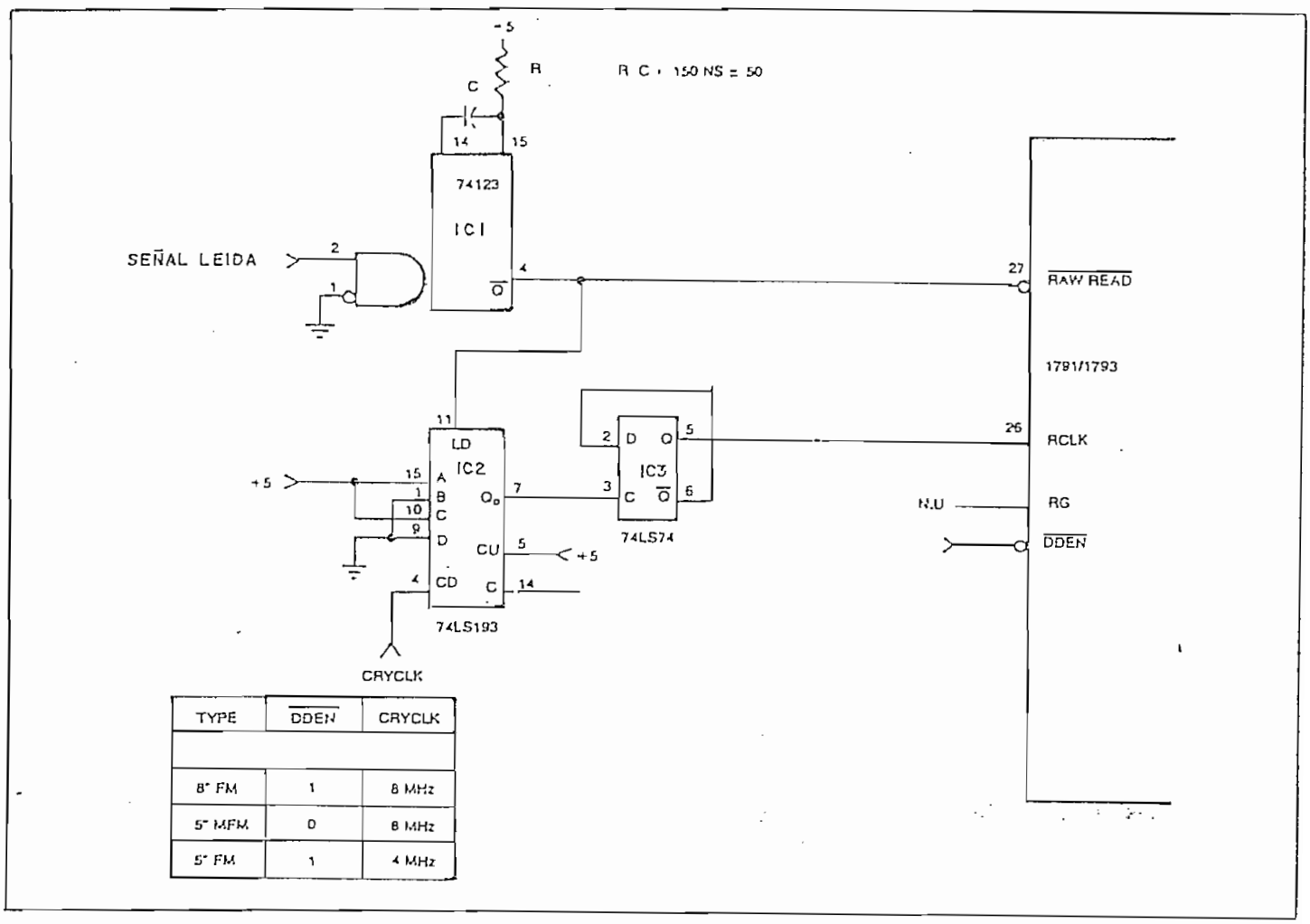
El circuito debe además determinar si se encuentra en la secuencia correcta, es decir, si la señal enviada a la salida de reloj corresponde realmente a la señal de reloj y no a la de datos, e igual cosa para la salida de datos. Esta función se realiza en base a tres flip-flops tipo D: 1a, 1b y 2a conectados en forma de un contador módulo 8 cuyas entradas de borrado se han conectado a la salida de la compuerta 3b (pulsos negativos de la señal de reloj), la entrada de reloj de este contador se la obtiene de la salida de la compuerta 4d (pulsos negativos de la señal de datos). Hay que recordar que la señal de reloj no puede tener más de tres ceros consecutivos (en las marcas de campos de datos o de identificación) y en estos casos los bits de datos que están entre ellos así como el que los precede y el que los antecede son son todos iguales a uno lógico. De esta forma si el contador no es borrado (por la señal de reloj) durante cuatro pulsos de datos, la salida de 1b se pone en cero lógico manteniendo por tanto la salida

del monoestable 11a en uno lógico, es decir, seleccionando la salida de reloj, si el siguiente bit leído por la lectora/escritora es un uno lógico (como sucede en una marca de campos de datos o de identificación), se borrará el contador a la vez que se producirá una transición negativa en la salida de la compuerta3d, disparándose el monoestable 11a para seleccionar al siguiente pulso de datos, en cambio si el siguiente bit es un cero lógico, el contador no será borrado por lo que se seguirá seleccionando la salida de reloj, hasta que aparezca en ella un pulso que borre el contador con lo que el circuito comenzará a funcionar normalmente y en la secuencia correcta. De esta manera el circuito admite hasta tres ceros lógicos por la señal de reloj, intercambiando las señales de datos y de reloj si existen más, consiguiendo de esta manera que el circuito separe las señales en la secuencia correcta.

En la figura 14 se indica otro de los circuitos utilizados para la separación de datos. En este caso la separación no es completa sino que el circuito produce sólamente una señal simétrica de reloj, cada ciclo de la misma contiene a un bit de datos o de reloj pero no importa la polaridad, es decir que los ciclos positivos pueden contener a bits de reloj y los negativos a bits de datos o viceversa.

La señal proveniente de la lectora/escritora es acondicionada por medio del monoestable 1, su salida es utilizada

para cargar en forma paralela los bits del contador 2. El contador es del tipo "UP/DOWN" módulo 16 y mientras no se produzca un pulso en la salida del monoestable 1, se encuentra contando libremente hacia abajo por medio de la señal de reloj.



TYPE	$\overline{DDEN}$	CRYCLK
8" FM	1	8 MHz
5" MFM	0	8 MHz
5" FM	1	4 MHz

FIGURA 14

CONTADOR/SEPARADOR

"CRYCLK". La salida Q_D del contador es utilizada como señal de reloj para un flip-flop tipo D conectado como un contador módulo 2, de esta forma el flip-flop produce una señal simétrica que, como se verá a continuación, contiene a un bit de reloj o de datos por cada ciclo de la misma.

Al producirse un pulso en la salida del monoestable, el contador es inicializado con el número 5_{10}. Al retornar la salida del monoestable a uno lógico, el contador empezará a contar -hacia abajo- hasta la cuenta de 15_{10} en donde la salida del flip-flop "RCLK" será complementada, asegurando de esta manera que el pulso quede contenido dentro de un ciclo de la señal RCLK. En es sistema FM con discos de 8" el siguiente pulso podrá ocurrir luego de 2 µSeg, tiempo para el cual el contador se encontrará nuevamente con la cuenta de 5_{10}, es decir que el circuito queda sincronizado con la señal enviada por la lectora/escritora y de producirse un desplazamiento en la posición de los bits de datos o de reloj, el separador está en capacidad de alcanzar nuevamente el sincronismo. Casos similares se tendrán para discos de 5¼" y para el sistema MFM.

La posición relativa de un bit dentro de cada ciclo de la señal RCLK puede ser variada cambiando el número con que se inicializa el contador (en el ejemplo 5_{10}). El contador puede ser inicializado con números entre $Ø_{10}$ y 7_{10} pues al producirse la inicialización, no debe producirse una transición posi-

tiva en Q_D como podría ocurrir para números entre 8_{10} y 15_{10}. Cada bit quedará más centrado en el ciclo entre más se tienda a 7_{10} como número de inicialización, sin embargo, el número que se utilice para inicializar el contador debe ser aquel que produzca los mejores resultados en la práctica.

Al estudiar la precompensación en la escritura, se explicó que pueden ocurrir pequeños desplazamientos en la posición de los bits escritos en el disco, especialmente en el caso del sistema MFM. Estos desplazamientos son reducidos por medio de la precompensación pero no son completamente eliminados, por esta razón no es aconsejable la utilización del circuito separador de datos de la figura 14 en el caso de discos de 8" con el sistema MFM. De utilizarse este circuito, cada bit leído del disco sincronizaría al separador en forma diferente y no se alcanzaría un funcionamiento estable del mismo.

Las razones indicadas hacen que el circuito de separación de datos más utilizado para el sistema MFM con discos de 8" sea del tipo PLL (Phase-Lock-Loop), en este tipo de circuitos, el desplazamiento de un bit leído no produce una gran variación de la frecuencia de la señal de salida y sólamente desplazamientos sucesivos de varios bits la podrán variar significativamente. A pesar de que los circuitos PLL son los más apropiados para la separación de datos en el sistema MFM, se han desarrollado circuitos más sencillos, puramente digitales

ADDRESS	DATA
00	01
01	01
02	02
03	03
04	03
05	04
06	05
07	06
08	08
09	0D
0A	0C
0B	0E
0C	0F
0D	0F
0E	00
0F	01
10	01
11	02
12	03
13	04
14	05
15	06
16	07
17	08
18	09
19	0A
1A	0B
1B	0C
1C	0D
1D	0E
1E	0F
1F	00

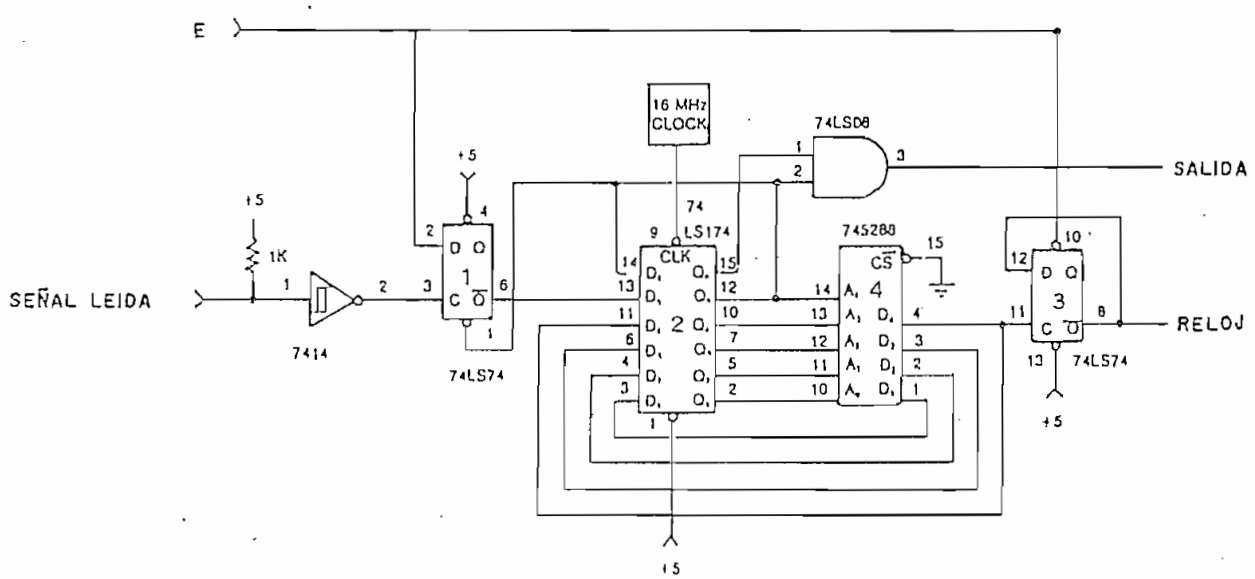


FIGURA 15

CIRCUITO SEPARADOR DE DATOS

que cumplen aproximadamente la misma función. Uno de estos circuitos se ilustra en la figura 15.

El funcionamiento del circuito es el siguiente: mientras la lectora/escritora no mande un bit de datos o de reloj, la salida $\overline{Q}$ del flip-flop IC1 se mantendrá en uno lógico; la salida Q5 de IC2 copiará este dato con la siguiente transición negativa del reloj de 16 MHz, seleccionando de esta forma las dieciseis últimas direcciones de la memoria PROM. Como puede verse en la tabla adjunta, la memoria PROM está programada de forma que si sus dieciseis últimas direcciones de memoria son seleccionadas, IC2 en conjunto con la memoria PROM actuarán como un contador módulo 16. El bit más significativo de este contador: D4, es utilizado como señal de reloj para IC3 que está conectado como un contador módulo 2 que además de producir una señal simétrica (que constituye la salida del circuito), impide la salida de señal si la línea de habilitación E está en cero lógico.

Si un bit de datos o de reloj es leído del disco y si E está en uno lógico, la salida $\overline{Q}$ de IC1 se pondrá en cero lógico; este dato es luego copiado en la salida Q5 haciendo que la salida $\overline{Q}$ de IC1 regrese a uno lógico, que la salida de la compuerta AND se ponga en cero lógico y se seleccionen las quince primeras direcciones de la memoria PROM. Con la siguiente transición negativa del reloj de 16 MHz, Q5 regresa a uno lógico, Q6 se pone en cero lógico (manteniendo la salida de la

compuerta AND en cero lógico) y el contador comienza a contar nuevamente. El conteo comienza con un número igual al que se tuvo al recibir el bit aumentado en uno, o con un número mayor o menor a este en una o dos unidades como se indica en la tabla de programación. Con la siguiente transición negativa en el reloj de 16 MHz, la salida de la compuerta AND regresará a uno lógico, es decir, que en esta salida se repetirán los pulsos que se tienen en la entrada del circuito, pero los pulsos producidos tendrán siempre una duración constante igual a 125 µSeg.

Puede notarse que el circuito actúa como un contador al que se lo adelanta o atrasa de forma que cuando un bit sea leído del disco, el contador se encuentre con la cuenta de cero, asegurando de esta forma que los bits queden centrados en cada ciclo de la señal de salida. Si el circuito se encuentra funcionando sin que se produzcan atrasos o adelantos, la señal de salida tendrá un período de 2 µSeg y puesto que los bits de datos o de reloj son leídos de un disco de 8" en el sistema MFM cada 2 µSeg, se tendrá que todos los ciclos positivos de la señal de salida contendrán a los bits de datos y los ciclos negativos a los de reloj o viceversa, pero ningún ciclo contendrá a más de un bit.

Cabe anotar que el mismo circuito puede ser utilizado como separador de datos para el sistema FM con discos de 8" y

para discos de 5¼" en los dos sistemas. Para ello simplemente se reduce la frecuencia de reloj a 8 MHz tanto para discos de 8" en FM como para discos de 5¼" en MFM y a una frecuencia de 4 MHz para discos de 5¼" en FM.

Algunos ejemplos adicionales de circuitos separadores de las señales de datos y de reloj pueden encontrarse en los apéndices y bibliografía de la presente tesis.

C A P I T U L O T E R C E R O

"DISEÑO Y CONSTRUCCION DEL CONTROLADOR E INTERFACE PARA DISCOS FLOPPY"

los discos de 5¼" de un solo lado utilizando el sistema FM, son los más aconsejables para el desarrollo de la presente tesis.

En base a las consideraciones anteriores se decidió utilizar la lectora/escritora SA400 producida por la compañía Shugart Associates, diseñada para trabajar con discos de 5¼" de un solo lado y de densidad simple. Esta lectora/escritora puede trabajar con los dos tipos de discos indicados en el capítulo II, esto es: "hard sectored" y soft sectored"; la velocidad angular del disco es controlada electrónicamente, por lo que resultan menos vulnerables a variaciones de frecuencia o voltaje en la red, además su bajo costo y confiabilidad hacen que estas lectoras/escritoras sean muy utilizadas. (15).

A pesar de que las necesidades de almacenamiento no son mayores actualmente, el sistema completo deberá preveer expansiones futuras por lo que el controlador que se diseñe deberá funcionar también con discos de 8" tanto en el sistema FM como en el MFM; con discos de un solo lado o doble lado y deberá tener la capacidad de trabajar con varias lectoras/escritoras a la vez. Por último, el sistema completo deberá trabajar con discos del tipo "soft sectored", permitiendo de esta forma que el usuario defina la longitud de los sectores en el disco de acuerdo a la aplicación específica que se de al equipo.

3.1 DISEÑO DEL CONTROLADOR.

TABLA IV

CARACTERISTICA	INTEGRADO NUMERO:					
	MC 6843	FD 1771	FD 1791	FD 1793	FD 1795	FD 1797
Compatibilidad con circuitos TTL	S	S	S	S	S	S
Número de fuentes de voltaje	1	3	2	2	2	2
Capacidad de interface directa con el microprocesador 6800	B	R	R	R	R	R
Puede trabajar en densidad simple y doble (FM y MFM)	N	N	S	S	S	S
Precompensación en la escritura (interna)	N	N	S	S	S	S
Separación interna de datos en FM	S	S	S	S	S	S
Separación interna de datos en MFM	N	N	S	S	S	S
Capacidad de DMA	S	S	S	S	S	S
Compatibilidad con discos "soft sectored"	S	S	S	S	S	S
Bus de datos normal	S	N	N	S	N	S
Bus de datos invertidos	N	S	S	N	S	N
Puede trabajar con discos de doble lado (directamente)	N	N	N	N	S	S
Generación y comprobación interna de CRC	S	S	S	S	S	S
Generación interna de pulsos de "Step" de duración programable	S	S	S	S	S	S
Demora programable para bajado de la cabeza	S	N	N	N	N	N
Capacidad para trabajar con formatos diferentes al IBM	M	B	B	B	B	B
Capacidad para trabajar con diferentes longitudes de sector	M	R	B	B	B	B
Capacidad de crear el formato en el disco	R	B	B	B	B	B
Número de comandos (operaciones preprogramadas)	10	11	11	11	11	11

SIMBOLOS UTILIZADOS: S = Si B = Buena R = Regular
N = No M = Mala

Puesto que la duración del pulso producido por el monoestable es relativamente larga, se ha escogido el integrado 74LS123 para cumplir con esta función pues con él se tendrán las siguientes ventajas: bajo consumo de corriente, conexión directa de condensadores electrolíticos (sin utilizar diodos de protección) y redisparo.

En este tipo de monoestable el ancho del pulso puede ser calculado con la siguiente fórmula: (17)

$$t_w = 0{,}45 \times R_t \times C_{ext}$$

Para el cálculo de R_t y C_{ext} se tomará a t_w = 100 mSeg. para asegurar que en la práctica el ancho del pulso no llegue a ser menor a 75 mSeg. a pesar de las tolerancias de los elementos.

Escogiendo una resistencia dentro de los valores típicos para el monoestable y aplicando la fórmula se llegó a determinar que:

$$R_t = 47\ K\Omega$$

$$C_{ext} = 4{,}7\ \mu F$$

Por recomendación de la compañía productora de la lectora/escritora SA400, se ha decidido que el motor de la misma de

be encenderse sólamente mientras esté en uso y mantenerse encendido durante unos dos segundos después de la última operación realizada. De esta forma se disminuye el desgaste tanto del disco como de la lectora/escritora. (18).

Para cumplir con este requisito de la lectora/escritora se utilizará el integrado 555 como se indica en la figura 16.

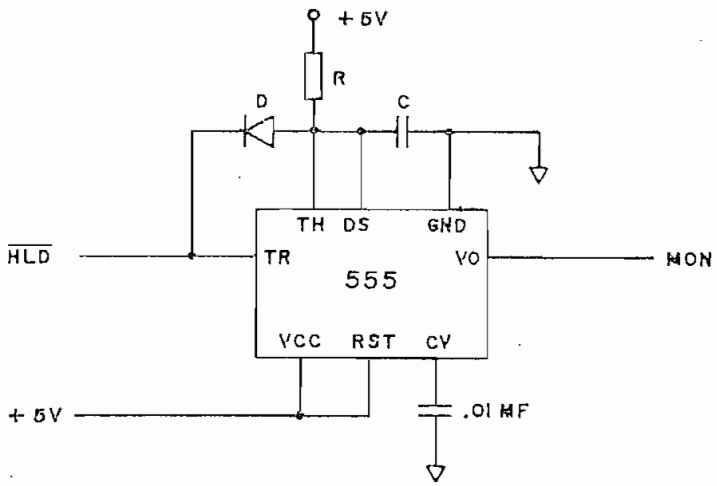


FIGURA 16

El funcionamiento del circuito es el siguiente:

Originalmente el condensador C se encuentra descargado (por medio de la línea DS) y la salida MON (Motor On) en cero lógico. Si la línea $\overline{HLD}$ se pone en cero lógico el integrado 555 se dispara, permitiendo que el condensador se cargue y que la salida MON cambie su estado a uno lógico, indicando de esta forma que el motor de la lectora/escritora debe encenderse. Por la presencia del diodo D, el condensador no podrá cargarse más allá de la suma del voltaje de conducción del diodo y el

voltaje en cero lógico de la línea $\overline{HLD}$.

Si la línea $\overline{HLD}$ regresa a uno lógico, el condensador C podrá cargarse por medio de R hasta el voltaje de transición "threshold voltage" del integrado 555. En ese instante el condensador será descargado por medio de la línea DS y la línea MON se pondrá nuevamente en cero lógico por lo que el motor de la lectora/escritora se apagará.

Para calcular los valores de R y C hay que recordar que el condensador se carga de acuerdo a la siguiente ecuación:

$$V_c = V_{cc} (1 - e^{-t/RC})$$

De donde:

$$t = RC \ln \left(\frac{V_{cc}}{V_{cc} - V_c} \right)$$

Tomando en cuenta que el condensador se carga desde un voltaje V_k igual a la suma del voltaje de conducción del diodo y el voltaje en cero lógico de la línea $\overline{HLD}$, hasta el voltaje de transición igual a $2/3V_{cc}$ se tiene que el tiempo que transcurre desde que $\overline{HLD}$ regresa a uno lógico y MON se pone en cero lógico será:

$$T = RC \ln \left(\frac{V_{cc} - V_k}{1/3\ V_{cc}} \right)$$

Asumiendo que el voltaje de conducción del diodo sea de 0,6 voltios y el voltaje en cero lógico de la línea $\overline{HLD}$ sea de 0,2 voltios se tiene:

$$T = 0{,}92\ RC$$

Si se desea que el motor de la lectora/escritora se mantenga encendido durante 2 Seg. después de que la línea $\overline{HLD}$ regresa a uno lógico, se pueden escoger los siguientes valores para R y C:

$$R = 220\ K\Omega\ ;\quad C = 10\ \mu F$$

El hecho de que la lectora/escritora no permanezca encendida en todo momento, hace necesario el utilizar otro circuito que impida la realización de cualquier operación en el disco hasta que se estabilice la velocidad del motor cada vez que se encienda. En el caso de la lectora/escritora SA400 el fabricante especifica que se debe dejar pasar por lo menos 1 Seg. para que dicha estabilización ocurra. (19).

El lapso de 1 Seg. puede ser medido en base a la duración de un pulso producido en el segundo de los monoestables del integrado 74LS123; el disparo del mismo debe producirse cada vez que se encienda el motor de la lectora/escritora es decir, con la señal MON.

Puesto que el integrado FD1797-02 no está provisto de una entrada adicional en la que se indique que la velocidad del motor se ha estabilizado, la única manera de impedir que éste integrado realice operaciones en el disco es la de utilizar nuevamente la línea HLT, haciendo que ésta se ponga en uno lógico sólamente cuando se cumplan dos condiciones: que la cabeza grabadora haya bajado y que la velocidad del motor sea estable. Para esto se hará que las salidas $\overline{Q}$ de los monoestables del integrado 74LS123 vayan a una compuerta AND 74LS08 cuya salida esté unida a la línea HLT del integrado FD1797-02.

Para asegurar que la duración del pulso producido por el monoestable no llegue a ser menor que 1 Seg., el cálculo de los valores de R_t y C_{ext} se lo hizo tomando el ancho del pulso como de 1,4 Seg. de acuerdo a la fórmula:

$$t_w = 0{,}45 \times R_t \times C_{ext}$$

De donde se llegó a determinar que:

$$R_t = 100\ K\Omega$$

$$C_{ext} = 33\ \mu F$$

Otro de los requerimientos del integrado FD1797-02 es el de una señal de reloj cuya frecuencia sea de 2 MHz para discos de 8" y de 1 MHz para los de 5¼". Esta señal se la obtendrá de uno de los osciladores controlados por voltaje (VCO) del integrado 74S124 y de dos flip-flops tipo D conectados como contadores módulo dos.

El integrado 74S124 tiene la ventaja de trabajar como un oscilador muy estable, conectando simplemente un cristal de la frecuencia deseada en lugar del condensador externo. Internamente contiene a dos osciladores controlados por voltaje por lo que se podrá utilizar uno de ellos para generar la señal de reloj y el restante para el circuito PLL que se utilizará en la separación de datos.

La frecuencia de oscilación deberá ser de 4 MHz. de forma que al ser dividida 2 ó 4 veces en los flip-flops tipo D se obtenga una señal simétrica de 2 ó 1 MHz. La frecuencia de salida podrá ser seleccionada por medio de dos "SPST DIP Switches" como se indica en la figura 17.

Por recomendación del fabricante (20), la línea de control de frecuencia deberá estar en cero lógico y la línea de rango en uno lógico, esto último será conseguido por medio de una resistencia de 1K conectada a Vcc.

El hecho de que se utilice un circuito del tipo Schottky y no uno de menor consumo de corriente, se debe primordialmente a las exigencias de linealidad en el control de frecuencia para el circuito PLL que se utiliza en la separación de datos. Este circuito ha sido tomado de las notas de aplicación editadas por la compañía Western Digital para su familia de integrados FD 179X-02.

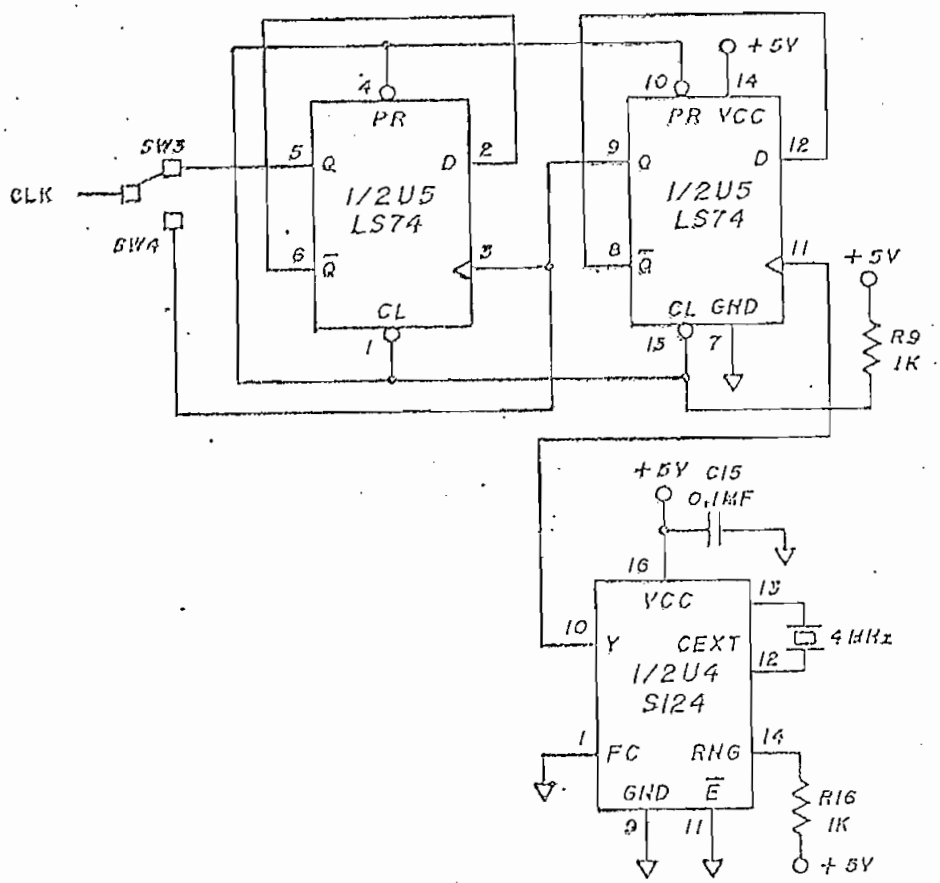
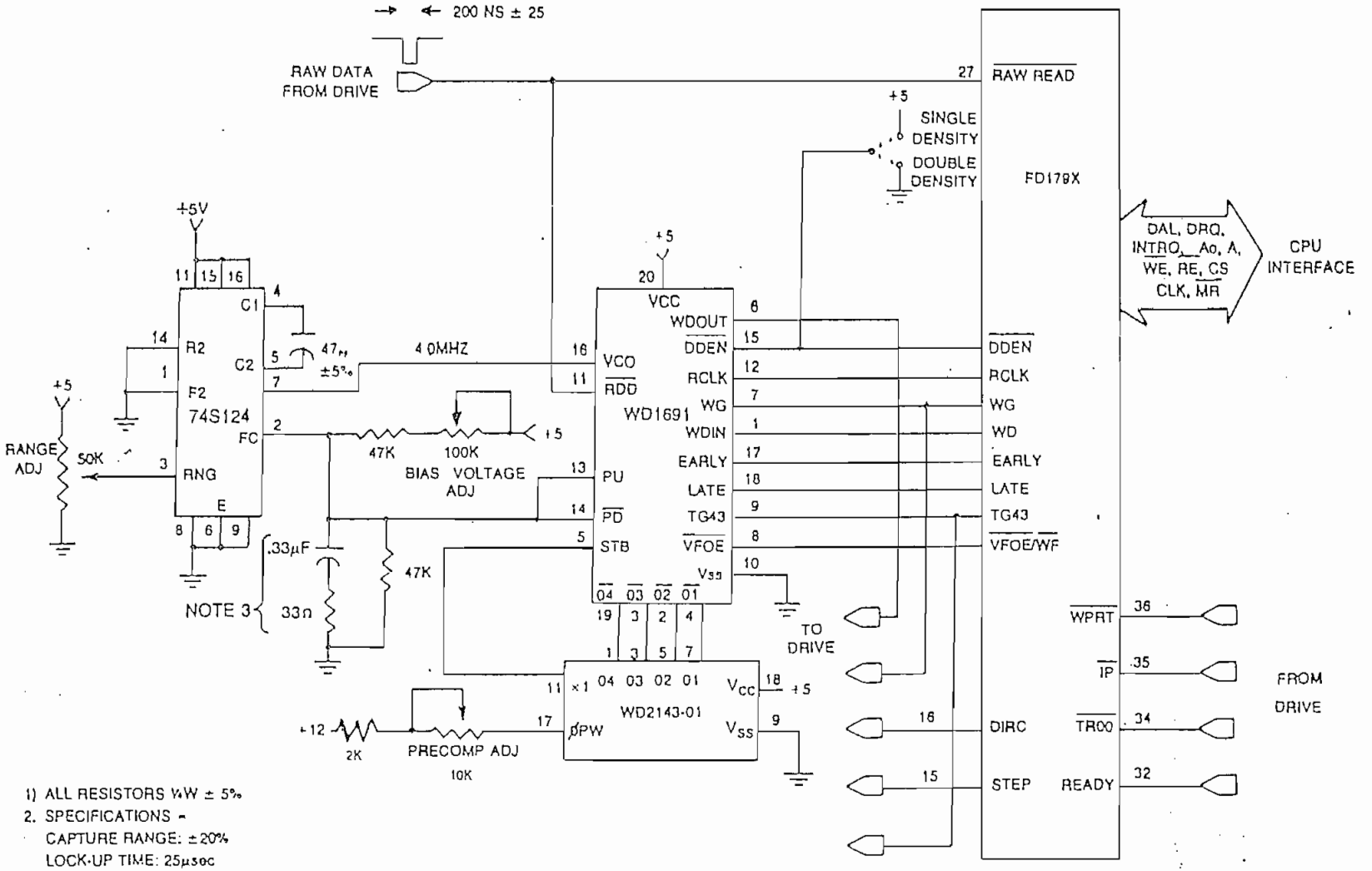


FIGURA 17

CIRCUITO GENERADOR DE LA SEÑAL DE RELOJ

Las ventajas de utilizar el circuito que se muestra en la figura 18 para la separación de datos son múltiples:

- Se utilizan sólamente tres integrados.
- Precompensación en la escritura encluida.
- Confiabilidad en la separación de datos por utilizar un



1) ALL RESISTORS ¼W ± 5%
2. SPECIFICATIONS -
CAPTURE RANGE: ±20%
LOCK-UP TIME: 25µsec
(ALL ONE'S PATTERN, MFM)

3) FOR 5 1/4" .68µf 68Ω ; 8 .33µf 33Ω

FIGURA 18

circuito PLL.

- Las exigencias en las tolerancias de las resistencias y condensadores externos no son mayores.
- La cantidad de precompensación deseada puede ser ajustada en forma sencilla.
- Trabaja en densidad simple o doble, con discos de 5¼" y de 8".
- Conexión directa con el integrado FD1797-02.

El integrado WD1691 consta de dos secciones:

1) Circuito de separación de datos.
2) Circuito de precompensación.

Este integrado está diseñado de forma que la precompensación en la escritura se realice sólamente en las pistas mayores a la número 43 siempre que se utilice densidad doble, es decir, cuando TG43 = 1L y $\overline{DDEN}$ = 0L. El funcionamiento de la sección de precompensación es muy similar al del circuito de la figura 6 que fuera explicado al estudiar los formatos de escritura.

La sección de separación de datos es habilitada solamente cuando el integrado FD1797-02 espera recibir datos confiables de la lectora/escritora es decir, cuando $\overline{VFOE/WF}$ = 0L y WRITE GATE = 0L. Si las condiciones anteriores se cumplen, las líneas PU y $\overline{PD}$ incrementarán o disminuirán la frecuencia de oscilación del integrado 74S124, de forma que los pulsos prove-

nientes de la lectora/escritora ocurran en el centro de cada ciclo de la señal de salida RCLK. Debe notarse que el circuito no produce una separación completa de los bits de datos y de reloj, sino que ésta se realiza en el integrado FD1797-02 a partir de la señal RCLK y aquella leída del disco.

Cabe anotar que el circuito original que consta en las notas de aplicación adolece de un error pues se considera que para trabajar con discos de 5¼" no hace falta cambiar la frecuencia del VCO, cuando uno de los requerimientos del integrado WD1691 es precisamente el de que la frecuencia central de oscilación del VCO sea de 2 MHz. cuando se trabaja con discos de 5¼". En el circuito de la figura 18 se ha corregido el error por medio de un condensador de 47 pF. que debe ser conectado -cuando se trabaje con discos de 5¼"- en paralelo con el condensador original externo del integrado 74S124. De esta forma se reduce la frecuencia de oscilación del VCO a la mitad, es decir, a 2 MHz.

La calibración del circuito de la figura 18 debe realizarse de la siguiente manera:

1) Desconectar el integrado WD1691.
2) Configurar el circuito para utilizarlo con discos de 8 ó 5¼ pulgadas, según se requiera.
3) Obtener un voltaje de 1.4 voltios en la línea de control de frecuencia (FC) variando el potenciómetro correspondiente.

4) Ajustar el potenciómetro de rango (RNG) hasta obtener una frecuencia de oscilación de 2 MHz. para discos de 5¼" o de 4 MHz. para discos de 8".

5) Conectar nuevamente el integrado WD1691.

Otro de los requerimientos del circuito de la figura 18 es el de que los pulsos provenientes de la lectora/escritora tengan una duración de 200 nSeg. ± 25 por lo que para asegurar compatibilidad con cualquier lectora/escritora se utilizará un monoestable que produzca pulsos de esta duración a partir de los pulsos recibidos.

Debido a la relativamente corta duración de los pulsos requeridos, se utilizará uno de los monoestables del integrado 74123. En este tipo de monoestables la duración del pulso puede ser calculada con la siguiente fórmula:

$$t_w = 0{,}28 \times R_t \times C_{ext} \times (1 + 0{,}7/R_t)$$

Utilizando la fórmula anterior para valores típicos de R_t se determinó que C_{ext} deberá ser de un valor menor a 1000 pF, en estos casos el fabricante recomienda escoger los valores de R_t y C_{ext} gráficamente (21). De acuerdo a las gráficas proporcionadas se llegó a determinar que el pulso de 200 nSeg. puede ser obtenido con un condensador de 39 pF y una resistencia de 10 KΩ Una vez obtenidos estos valores se decidió utilizar como resistencia externa a un potenciómetro de 10 KΩ en serie con una resistencia de 5,1 KΩ, de esta forma el ancho del pulso po

drá ser regulado con precisión, variando el valor de la resistencia externa dentro de amplios límites pero siempre dentro de los valores recomendados por el fabricante.

El segundo de los monoestables del integrado 74123 podrá ser utilizado para producir un pulso en la línea $\overline{\text{MASTER RESET}}$ del integrado FD1797-02 cada vez que la señal $\overline{\text{RESET}}$ del equipo MEK6800D2 se ponga en cero lógico. De esta forma el controlador de discos será inicializado conjuntamente con el equipo de microprocesador.

El fabricante especifica que la duración mínima del pulso en la línea $\overline{\text{MASTER RESET}}$ debe ser de 50 µSeg., pero teniendo en cuenta que al finalizar el pulso, el integrado FD1797-02 carga y ejecuta automáticamente un comando que hace regresar a la cabeza de grabación a la pista ØØ, se ha creido conveniente alargar la duración del pulso a unos 275 µSeg. de forma que la lectora/escritora pueda estar lista para aceptar el comando. La salida del monoestable podrá ser utilizada también para inicializar los circuitos de interface con el equipo de microprocesador.

Aplicando la fórmula -ya mencionada- para este tipo de monoestables, se llegó a determinar que un pulso de 275 µSeg. puede ser obtenido con los siguientes valores de resistencia y condensador externos:

$R_t = 12\ K\Omega$

$C_{ext} = 0{,}082\ \mu F$

3.2 DISEÑO DE LA INTERFACE.

En los procesos de interface entre el controlador y la lectora/escritora se seguirán las recomendaciones del fabricante de esta última, es decir, para cada salida del controlador se deberán utilizar compuertas de colector abierto, capaces de soportar una corriente de salida de 35 mA. o más en cero lógico, además, cada entrada deberá ser terminada por una resistencia de 150 Ω conectada a Vcc. y a una compuerta del tipo "Schmitt Trigger". (22).

Por las razones mencionadas se utilizarán compuertas 7438 en las salidas que necesiten inversión y 7417 en las que no. Para las entradas se utilizarán compuertas del tipo 74LS14 junto con resistencias de 150 Ω conectadas a Vcc.

En cuanto a la interface entre el controlador y el equipo MEK6800D2 se utilizará una PIA como principal elemento de la interface pues esta dotará de una mayor agilidad a ciertas secciones de los programas de soporte, además de que sus múltiples líneas periféricas podrán ser utilizadas para controlar los procesos de interface, transmitir datos desde o hacia el microprocesador y establecer las condiciones de funcionamiento del con

trolador.

El circuito de interface entre el controlador y la lectora/escritora se muestra en la figura 19. Podrá notarse -cuando se hable de los programas de soporte- que utilizando este circuito se consigue la mayor agilidad posible en los programas de transmisión y recepción de datos, permitiendo de esta manera que el microprocesador pueda trabajar a una frecuencia menor a la máxima para él permitida sin que se pierdan datos escritos o leídos del disco. Esta característica es de mucha importancia pues el equipo MEK6800D2 trabaja con una frecuencia de 614,4 KHz.

Debe notarse que se ha preferido utilizar un tipo de interface en la que la transmisión de datos se realiza "por programa" dejando de lado la utilización de DMA, pues su uso incrementaría el número de integrados utilizados en la interface en cambio, el circuito de la figura 19 hace uso de componentes incluidos en el equipo MEK6800D2, como son la PIA y los circuitos utilizados para el direccionamiento de la misma, además, la interface podrá ser fácilmente conectada al equipo ya que este dispone de salidas para el efecto.

A continuación se explica el funcionamiento del circuito:

La transmisión de datos entre el controlador y el equi

po MEK6800D2 se realiza por medio de las líneas PBØ - PB7 que deberán ser programadas como salidas o entradas, dependiendo de que se envíe un dato al controlador o se reciba un dato del mismo. Las líneas PAØ y PA1 tienen que ser programadas como

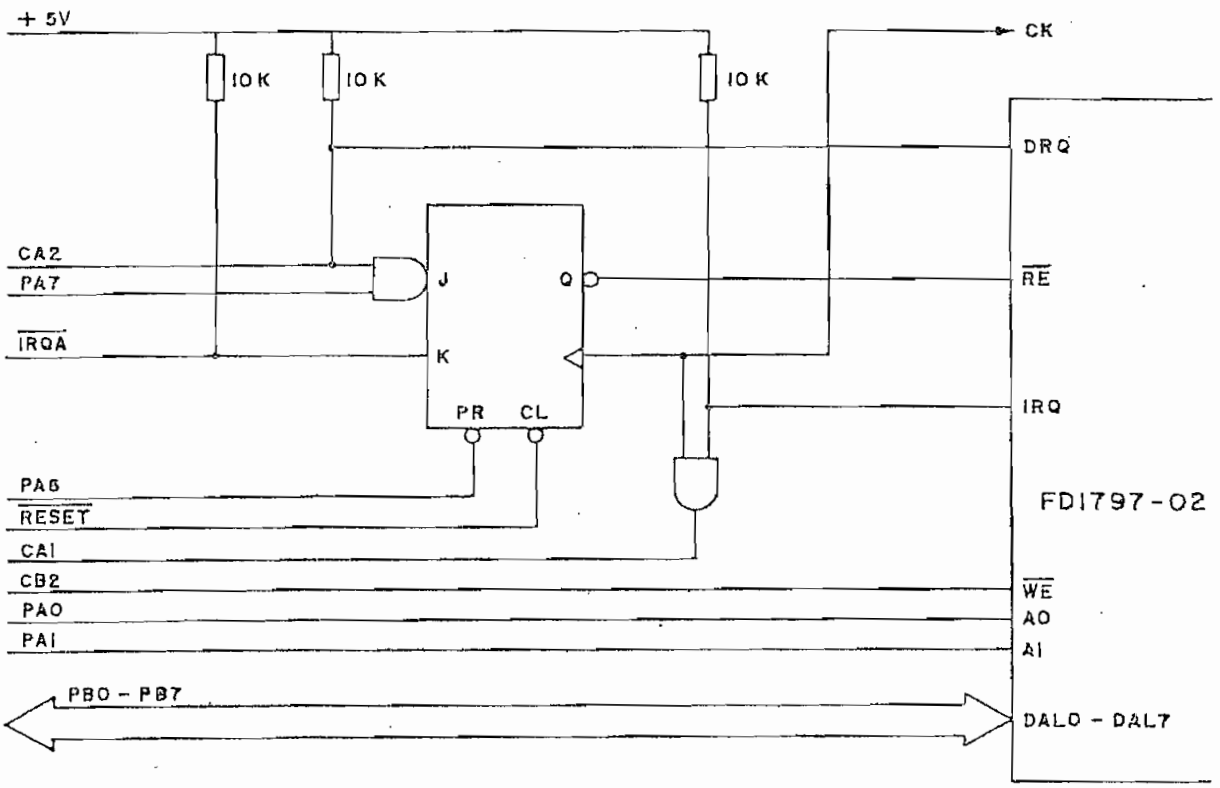


FIGURA 19

CIRCUITO DE INTERFACE

salidas y deberán apuntar al registro del integrado FD1797-02 que se requiera leer o escribir.

El control de escritura de datos en los registros del integrado FD1797-02 se lo realiza directamente por medio de la

línea de control CB2, mientras que para el control de lectura de datos desde el controlador hacia la PIA se utiliza un flip-flop JK en conjunto con algunas señales provenientes de la PIA y el controlador.

Para el control de lectura de datos hay que diferenciar dos casos: el primero en que se lee un registro del controlador en el momento en que se necesite de ello y el segundo -utilizando en la lectura del disco- en que el controlador ha recibido un byte de datos y por lo tanto requiere que el microprocesador lo lea antes de que el siguiente byte sea leído del disco. Esta última condición es indicada por el controlador por medio de un uno lógico en su salida DRQ.

De acuerdo a lo indicado, la línea $\overline{RE}$ deberá controlarse en forma tanto asincrónica como sincrónica con DRQ, lo que se ha conseguido con un flip-flop cuya salida puede ser puesta asincrónicamente en cero lógico bajo el control de la línea PA6 proveniente de la PIA, así pues, si se requieren leer datos en este modo de operación se deberá escribir un cero lógico en PA6, leer el dato y poner nuevamente PA6 en uno lógico. Si un cero lógico fue previamente escrito en PA7 -como requiere este modo de operación- las entradas J y K estarán en cero y uno lógico respectivamente por lo que $\overline{RE}$ regresará a uno lógico con la siguiente transición positiva de CK.

En la segunda forma de operación, un uno lógico debe

ser escrito en PA6 y PA7 y los bits 3-5 del registro de control A deben ser programados con 11Ø respectivamente, de esta forma cuando DRQ se ponga en uno lógico la salida del flip-flop se pondrá en cero lógico con la siguiente transición positiva de CK. Puesto que se han unido las líneas DRQ y CA2, al producirse la transición positiva de DRQ la salida $\overline{IRQA}$ se pondrá en cero lógico, manteniendo por tanto la salida del flip-flop en cero lógico. El microprocesador podrá reconocer que existe un dato listo, pues el bit 6 del registro de control A (CRA) se pondrá en uno lógico, deberá leerlo en el registro de datos B (DRB) e indicar que se lo ha leído por medio de una operación de lectura del registro de datos A (DRA) con lo que $\overline{IRQA}$ y CRA bit 6 regresarán a uno y cero lógico respectivamente.

Para el control de escritura, los bits 3-5 del registro de control B (CRB) deberán programarse con 1Ø1 respectivamente. De esta forma, cada vez que se realice un proceso de escritura en el registro de datos B (DRB), la PIA mandará un pulso por su salida CB2.

De manera similar a la lectura de datos, el controlador pondrá un uno lógico en su salida DRQ cada vez que esté listo para recibir un nuevo dato del microprocesador y escribirlo en el disco, por lo que será necesario deshabilitar el flip-flop de lectura escribiendo un cero y un uno lógico en PA7 y PA6 respectivamente. Un uno lógico en el bit 6 de CRA indicará al

microprocesador que el controlador está listo para recibir un nuevo dato, en ese momento se deberá escribir el dato en DRB y borrar la "bandera" leyendo el registro de datos A.

En la figura 19 se indica también la forma en que se han conectado las líneas IRQ del controlador y CA1 de la PIA. Recordando que la salida IRQ se pondrá en uno lógico al término de un comando o cuando se produce un error mientras se realiza el mismo, los bits 1 y 2 de CRA deberán ser programados como cero y uno lógico respectivamente, de forma que cuando IRQ se ponga en uno lógico, se informe de esta condición al microprocesador por medio de una bandera que en este caso es el bit 7 de CRA.

Cuando el microprocesador detecte un uno lógico en CRA bit 7, deberá borrar esta bandera leyendo DRA para luego leer el registro de "Status" del controlador y así determinar si se produjo un error o es que se ha terminado el comando.

3.3 EL CIRCUITO COMPLETO.

A continuación se presenta el diagrama de circuitos completo, en él se puede notar la presencia de algunos circuitos adicionales como es el caso del integrado 74LS138 que decodifica las señales de PA4 y PA5 para seleccionar la lectora/escritora deseada, de esta forma se podrá manejar hasta cuatro lectoras/escritoras con el mismo controlador. Puede observarse

también que la selección de una lectora/escritora puede ser mantenida en todo momento (SW2 cerrado) o sólamente cuando la línea $\overline{HLD}$ se ponga en cero lógico (SW1 cerrado), esta última forma de operación permite utilizar lectoras/escritoras -como la SA400- que bajan la cabeza de grabación al ser seleccionadas.

Por medio de PA2 se permite utilizar lectoras/escritoras que mantienen bajada la cabeza en todo momento y por lo tanto el controlador no necesita esperar a que se realice ese proceso. De utilizarse este tipo de lectoras/escritoras, un cero lógico deberá ser escrito en PA2, de forma que se inhabilite el disparo del monoestable que produce la demora para que se realice el proceso de bajado de la cabeza.

Algunas lectoras/escritoras no tienen la salida $\overline{READY}$, en esos casos se puede cerrar SW8 de forma que se considere que la lectora/escritora está lista cuando la señal $\overline{MON}$ del controlador se pone en cero lógico.

Lectoras/escritoras para discos de 5¼" o de 8" en densidad simple o doble pueden ser utilizadas como se indica en la tabla V.

El circuito ha sido armado, usando la técnica de "Wire Wrap", sobre una placa VECTOR 4609 diseñada para albergar cir-

TABLA V

TAMAÑO	DENSIDAD	SW3	SW4	SW5	SW6	SW7	C9
5¼"	Simple	C	D	D	C	D	C
	Doble	C	D	C	C	D	C
8"	Simple	D	C	D	D	C	D
	Doble	D	C	C	D	C	D

SIMBOLOS UTILIZADOS:

C = Conectado

D = Desconectado

cuitos de interface de microcomputadoras. Sus dos salidas para conectores planos, han sido utilizadas para unir al circuito con el equipo MEK6800D2 -por medio de un cable de 50 conductores- y con la lectora/escritora -por medio de un cable de 34 conductores-.

La placa tiene líneas impresas para V_{CC} y tierra que han sido utilizadas en conjunto con los enlaces VECTOR T112-1 para llevar los voltajes de polarización hasta cada integrado. Terminales de tipo VECTOR R32 han sido utilizados para el cristal de 4 MHz y C9 de forma que puedan ser facilmente removidos de la placa. Resistencias, diodos y condensadores, han sido soldados a postes del tipo VECTOR T44-1 de forma que puedan ser conectados mediante la técnica de "Wire Wrap".

Condensadores de desacoplamiento han sido conectados siguiendo las recomendaciones del fabricante de la placa (23). Estos suman una capacidad total de 30 μF en los terminales para la fuente de 5 V y de 0,1 μF en la de 12 V.

La posición de los integrados ha sido estudiada para minimizar las distancias entre aquellos integrados con líneas comunes, tratando además de que los integrados con alto consumo de corriente o muy ruidosos, queden cerca de las entradas de voltaje. De esta forma se han eliminado, en lo posible, las capacidades parásitas y los ruidos, quedando la posición de los elementos en la placa, como se muestra en la figura 20.

Los voltajes de 5 y 12 voltios serán tomados de las fuentes de voltaje del equipo MEK6800D2, por lo que éstas deberán soportar un incremento mínimo de corriente de 1A para la fuente de 5 voltios y de 50 mA para la de 12 voltios. Operando normalmente, el circuito deberá tomar alrededor de 0,5 A y 15 mA de dichas fuentes.

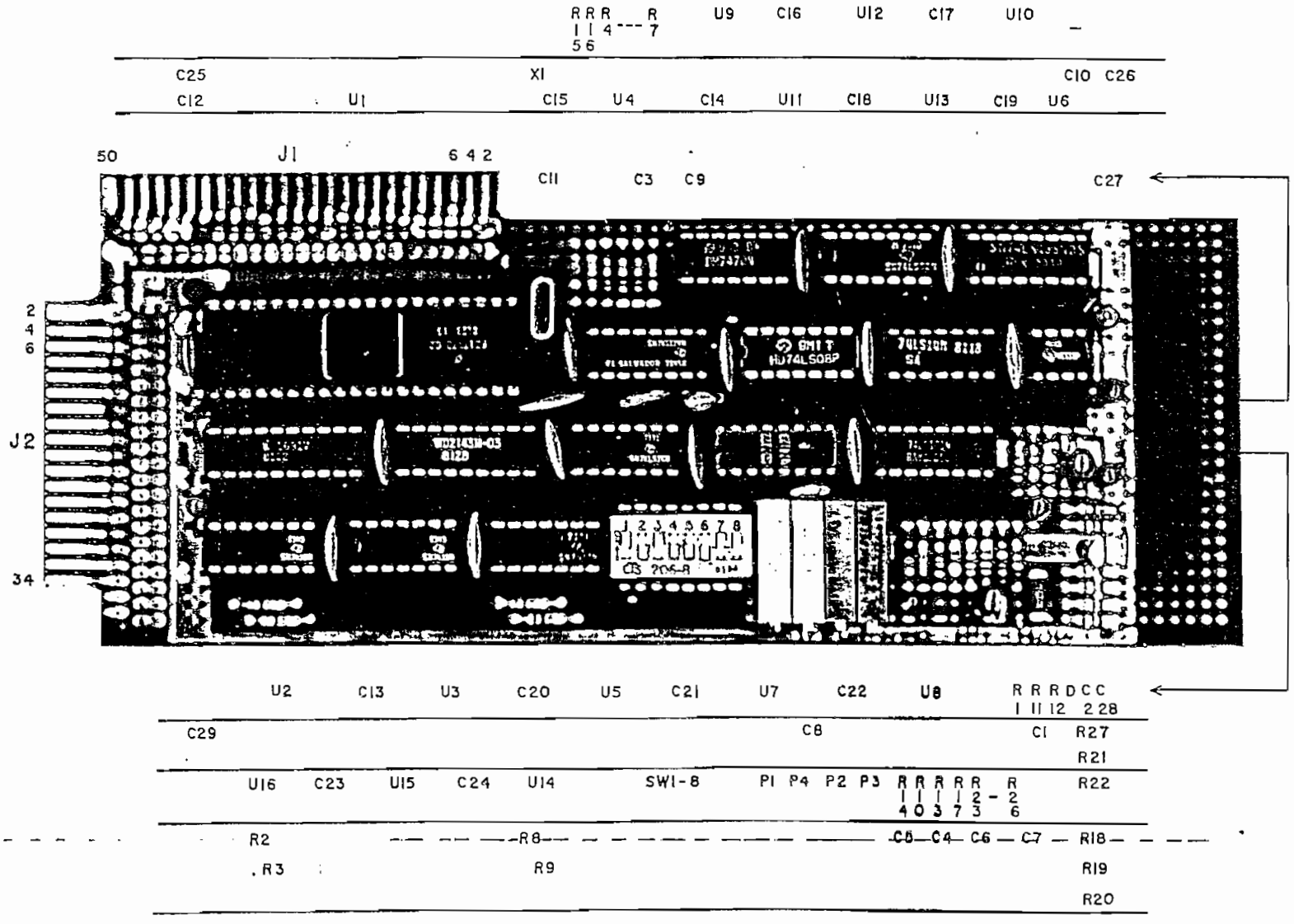


FIGURA 20

PLACA DE CIRCUITOS

C A P I T U L O C U A R T O

" PROGRAMAS DE SOPORTE "

4. LOS PROGRAMAS DE SOPORTE

Los programas de soporte complementan el funcionamiento del circuito, convirtiendo las instrucciones sencillas, utilizadas por el usuario, en una serie de procesos involucrados en la lectura o escritura de información en el disco. De esta forma los procesos de control, transmisión y almacenamiento de datos, resultan transparentes al usuario.

Para el desarrollo de los programas de soporte, se ha creído conveniente separarlos en tres categorías:

1.- Programas de control
2.- Programas operativos
3.- Programas de consola

El primer tipo de programas deben trasladar las instrucciones o comandos, propios del controlador, a subrutinas que además de ejecutar esos comandos, se encarguen de los procesos de interface. Por esta razón, cada una de estas subrutinas podrá ser utilizada como si se tratara de una instrucción que se envía al controlador.

Los programas operativos deberán hacer uso de las subrutinas de control para la grabación y lectura de los programas o datos del usuario. Deberán encargarse de la inicialización (creación del formato) de los discos que van a ser utilizados por primera vez; leer o escribir el número de sectores requeri

dos para almacenar la información del usuario y llevar un registro, tanto de las pistas y sectores ocupados por la información en el disco, como de las localidades de memoria a donde deberá ser leída dicha información.

Los programas de consola tendrán a su cargo el ingreso y salida de datos alfanuméricos que permitan establecer un diálogo entre el usuario y los programas operativos. Deberán, por tanto, mantener el control sobre los elementos de entrada/salida y estar en capacidad de "entender" las instrucciones impartidas por el usuario.

Al final de este capítulo se han incluído los programas terminados, tanto en código nemotécnico como en lenguaje de máquina, que serán utilizados -en conjunto con el controlador- para la lectura y escritura de discos floppy. Para el desarrollo de estos programas se ha partido siempre del diagrama de flujo correspondiente, por lo que, se ha creído conveniente presentar, en las siguientes páginas, a cada uno de los programas en diagrama de flujo junto con una pequeña explicación de su utilidad y de las condiciones de entrada/salida de los mismos.

El conjunto completo de programas han sido grabados en una memoria EPROM que será colocada en la posición U1Ø del equipo MEK6800D2, de forma que estos no ocupen el reducido espacio de memoria RAM de que se dispone actualmente, sin embargo, se han reservado los sectores 3-1Ø de la pista Ø para que en el

futuro se puedan almacenar allí la mayoría de los programas de soporte, dejando en la EPROM solamente aquellos indispensables para que el microprocesador pueda leer los programas restantes a memoria RAM. Con ello, se podrán almacenar en la EPROM todos aquellos programas que necesiten realmente estar siempre presentes en memoria.

4.1. PROGRAMAS DE CONTROL

A continuación se da una ligera explicación de los programas de control. El usuario podrá utilizar eventualmente los programas de control "principales" que han sido marcados con un asterisco (*). Estos programas se caracterizan porque al término de los mismos, los registros de la PIA permanecen sin cambios o son inicializados, de esta forma se los puede utilizar -uno tras otro- sin preocuparse por la condición de dichos registros.

4.1.1. INIC.- *

Esta subrutina inicializa los registros de la PIA y los registros asignados a las lectoras/escritoras. Por medio de ella, las líneas periféricas del lado A de la PIA son programadas como salidas y las del lado B como entradas; CRA y CRB son programados con $1E y $2C respectivamente (de acuerdo a lo explicado en el diseño de la interface) y, $44 es escrito en DRA con lo que se permite el disparo del monoestable de bajado de

la cabeza; se selecciona la lectora/escritora número cero; se apunta a "command register" del controlador y se inhabilita el flip-flop de lectura.

Los cinco registros asignados a las lectoras/escritoras son inicializados -con la subrutina INIC- a $ØØ. El primero de estos registros especifica el número de la lectora/escritora que se está utilizando y los cuatro restantes almacenan el número de pista sobre la cual se encuentra la cabeza de grabación de cada una de las lectoras/escritoras.

CONDICIONES DE ENTRADA**	CONDICIONES DE SALIDA**
Ninguna	A = $44 B = $2C X = DRA

**NOTA: Las condiciones de entrada/salida serán dadas solamente para los registros: A, B y X del microprocesador, incluyendo, cuando sea de importancia, el estado de uno o más bits del registro de código de condición (CCR) como se indica en los siguientes ejemplos:

A = 44	- acumulador A contiene el número 44 decimal (2C hexadecimal)-
B = $44	- acumulador B contiene el número 44 Hex.
A = (REGISTRO)	- acumulador A cargado con el contenido de un REGISTRO-
X = REGISTRO	- X contiene la dirección de un REGISTRO-

X = X + 256 - X contiene la suma de 256 Dec. más el valor de X de entrada.

3,X = $Ø3 - en la dirección 3 + (X) está almacenado el número 3 Hex.

CCR: Z = Ø - bit Z de CCR está en cero lógico-

C = 1 - bit C (Carry) de CCR está en uno lógico-

Debe entenderse además que las condiciones de entrada/salida se han incluído para ayudar al usuario que desea crear programas que utilicen las presentes subrutinas y por lo tanto deberá conocer la función de cada uno de los registros que se han creado en memoria y de los registros internos del controlador, luego de haber estudiado los programas terminados que se han incluído al final de este capítulo.

4.1.2. DISC.- *

Esta subrutina selecciona una lectora/escritora por medio de PA4 y PA5. El número de pista en la que se encuentra la lectora/escritora que se estuvo utilizando anteriormente se almacena en su registro correspondiente (de acuerdo a lo indicado en la subrutina INIC), a la vez que se recobra el número de pista en que se encuentra la lectora/escritora que se está seleccionando y se lo almacena en el registro de pista del controlador (Track Register). Por último se almacena el número de la lectora/escritora seleccionada en el registro reservado para el objeto.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
3,X = Número de lectora/escritora a seleccionarse	NORMALES: A = Pista en que se encuentra la L/E seleccionada B = Ø X = REGPIS + número de L/E seleccionada CCR: Z = 1; C = Ø DE ERROR: A = (3,X) B = $ØF X = X CCR: Z = Ø; C = 1

4.1.3. REGR.- *

La subrutina REGR posiciona la cabeza de grabación, de la lectora/escritora seleccionada, sobre la pista ØØ, utilizando para ello un comando "RESTORE" del controlador. REGR selecciona también una lectora/escritora por medio de la subrutina DISC, pues normalmente se necesitará que la cabeza de grabación de una lectora/escritora que se utiliza por primera vez, regrese a la pista ØØ. En caso de que no se requiera seleccionar una lectora/escritora, se puede utilizar la subrutina REGR desde la posición REGR1.

Al final de la subrutina se lee el registro de estado

del controlador ("Status Register") para averiguar si se produjo algún error en el proceso. El código de error es leído en el acumulador B del microprocesador, indicando que se ha producido un error por medio de un cero lógico en el bit Z de CCR. Esta forma de control de errores será utilizada -siempre que sea posible- en las subrutinas restantes, debiendo notar que el código de error será el mismo que el utilizado por el integrado FD1797-02 que puede encontrarse en los apéndices de la presente tesis.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
REGR: 3,X = Número de la L/E a seleccionarse	NORMALES: A = (DRA)
	B = (STATUS REGISTER)
REGR1: Ninguna.	X = X
	CCR: Z = 1
	DE ERROR: A = (DRA)
	B = (STATUS REGISTER)
	X = X
	CCR: Z = Ø

4.1.4. BUSQ.- *

Esta subrutina posicona la cabeza de grabación sobre una pista especificada. BUSQ será normalmente utilizada antes de leer o escribir un sector en el disco, por lo que se ha hecho que almacene también el número de sector en el registro correspondiente del controlador ("Sector Register").

Al estudiar el diagrama de flujo de la subrutina BUSQ, se encontrará que no se ha utilizado el comando de búsqueda ("SEEK") del controlador, sinó que se llega a la pista especificada "por programa". Esto se debe a que el comando SEEK mueve la cabeza de grabación "por pasos" cuya duración máxima es de 30 mSeg. mientras que la lectora/escritora SA400 -que se está utilizando- necesita como mínimo de 40 mSeg. (24).

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
	NORMALES:	A = Pista especificada
A = Pista especificada		B = Pista especificada
B = Sector especificado		X = X
	CCR:	Z = 1
	DE ERROR:	A = Pista especificada
		B = (STATUS REG.)
		X = X
	CCR:	Z = Ø

4.1.5. LEA.- *

LEA se utiliza para leer un sector del disco. Haciendo uso de la subrutina BUSQ, posiciona la cabeza de grabación sobre la pista especificada y lee el sector especificado a las localidades de memoria comprendidas entre aquella apuntada por el registro X del microprocesador y las 255 localidades siguientes.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	NORMALES: A = (DRA)
	B = (STATUS REG.)
A = Pista especificada	X = X + 256
B = Sector especificado	CCR: Z = 1
X = Comienzo del espacio de RAM a donde se lee el sector.	DE ERROR: A = (DRA)
	B = (STATUS REG.)
	X = Indeterminado
	CCR: Z = Ø

4.1.6. ESCRIB.- *

Esta subrutina es utilizada para escribir un sector en el disco. Al igual que la subrutina LEA, posiciona la cabeza de grabación sobre la pista especificada y luego escribe el sector especificado con el contenido de las 256 localidades consecutivas de memoria que comienzan desde aquella apuntada por el registro X del microprocesador.

A pesar de la optimización de que ha sido objeto la subrutina ESCRIB y de las ventajas que representa la utilización de la PIA como elemento de interface, ESCRIB es la subrutina que limita la frecuencia mínima de reloj que puede ser utilizada en el microprocesador. Desde luego, esto depende de la lectora/escritora utilizada y por lo tanto del tiempo entre bits de datos (como fuera indicando en la sección 1.3.2.).

Tomando en cuenta el número de ciclos de reloj que se

utilizan en la subrutina ESCRIB -en lenguaje de máquina- para almacenar un byte de datos en el disco y las características del controlador, se ha llegado a determinar que la frecuencia mínima utilizable es:

$$f_m = 25/t_{service\ (WRITE)}$$

De acuerdo a esta fórmula, la frecuencia mínima es el caso de discos de 5¼", en densidad simple, ($t_{service\ (WRITE)}$ = 47,Ø µSeg.) es de 532 KHz. (25). Puesto que el equipo MEK6800D2 trabaja a 614,4 KHz., no se necesitará de ningún cambio en la frecuencia de reloj mientras se utilice la lectora/escritora SA400.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
	NORMALES:	A = (DRA)
A = Pista especificada		B = (STATUS REG.)
B = Sector especificado		X = X + 256
X = Comienzo del espacio	CCR:	Z = 1
de RAM de donde se	DE ERROR:	A = (DRA)
lee el sector a es-		B = (STATUS REG.)
cribirse.		X = Indeterminado
	CCR:	Z = Ø

4.1.7. VERIF.- *

Esta subrutina se utiliza para verificar el último sector escrito, por lo que supone que la cabeza de grabación se

encuentra sobre la pista que se quiere verificar y que el controlador contiene el número de sector que es objeto de esta verificación.

VERIF busca sólamente errores de CRC (lo que cubre la mayoría de los errores posibles) efectuando para ello un proceso de lectura del sector pero sin almacenar los bytes leídos en memoria.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
Ninguna	NORMALES:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = 1
	DE ERROR:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = Ø

4.1.8. CHKRDY.- *

CKKRDY debe utilizarse para averiguar la condición de la lectora/escritora, es decir, si se encuentra lista para realizar algún proceso.

La subrutina considera que una lectora/escritora está lista si la línea $\overline{\text{READY}}$ está en ØL, si este no es el caso, mueve la cabeza de grabación un paso adelante y un paso atrás,

con lo que se consigue que la línea $\overline{HLD}$ del controlador se ponga en ØL y por lo tanto se tenga la seguridad de que una lectora/escritora está seleccionada (ver sección 3.3.), su cabeza bajada y su motor encendido ($\overline{MON}$ = ØL); luego, la subrutina busca nuevamente un ØL en la línea $\overline{READY}$, indicando que hay un error si esto no se produce.

El hecho de mover la cabeza de grabación un paso adelante y otro atrás ayuda también a un mejor posicionamiento de la cabeza en una pista del disco, por esta razón, la subrutina CHKRDY puede ser utilizada también luego de haberse producido un error en la lectura o escritura del disco.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
Ninguna	NORMALES:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = 1
	DE ERROR:	A = (DRA)
		B = (STATUS REG.)
		X = X
		CCR: Z = Ø

4.1.9. ENTR.-

Por medio de esta subrutina se programan las líneas periféricas del lado B de la PIA como Entradas.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = $2C
	X = X

4.1.10. SALID.-

SALID realiza el proceso inverso a ENTR, programando el lado B de la PIA como salidas.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = $2C
	X = X

4.1.11. DTARG.-

Esta subrutina escribe un 1L en PAØ y PA1, seleccionando por tanto a "DATA REGISTER" del controlador para cualquier proceso de lectura o escritura de este registro.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = (DRA)
	X = X

4.1.12. CMRG.-

Por medio de CMRG se escribe un ØL en PAØ y PA1, selec

cionando por tanto a "STATUS REGISTER" del controlador para un proceso de lectura y a "COMMAND REGISTER" para la escritura (como se indica en los apéndices de la presente tesis.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
Ninguna	B = (DRA)
	X = X

4.1.13. LEARG.-

Esta subrutina se utiliza para leer el registro del controlador apuntado por PAØ y PA1.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = (DRA)
Ninguna	B = (REGISTRO APUNTADO POR PAØ, PA1)
	X = X

4.1.14. MCMD.-

MCMD escribe un comando o un dato en el registro apuntado por PAØ y PA1.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = A
A = Comando o dato.	B = $2C
	X = X

4.1.15. MCSP.-

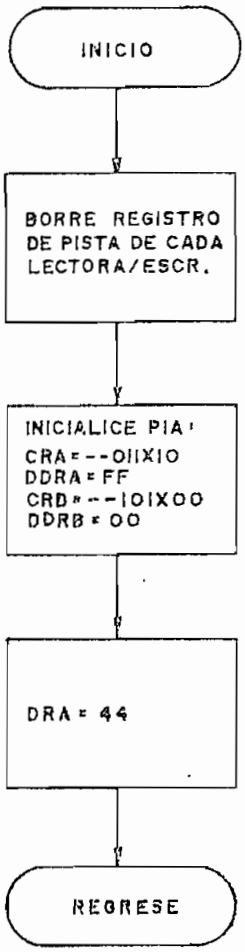
Esta subrutina manda un comando al controlador y espera a que se lo termine, luego lee el registro de estado del controlador ("STATUS REGISTER") para averiguar si se produjo algún error mientras se ejecutaba el comando.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = (DRA)
A = Comando	B = (STATUS REGISTER)
	X = X

4.1b. DIAGRAMAS DE FLUJO DE LOS PROGRAMAS DE CONTROL.

A continuación se presentarán los diagramss de flujo correspondientes a los programas de control.

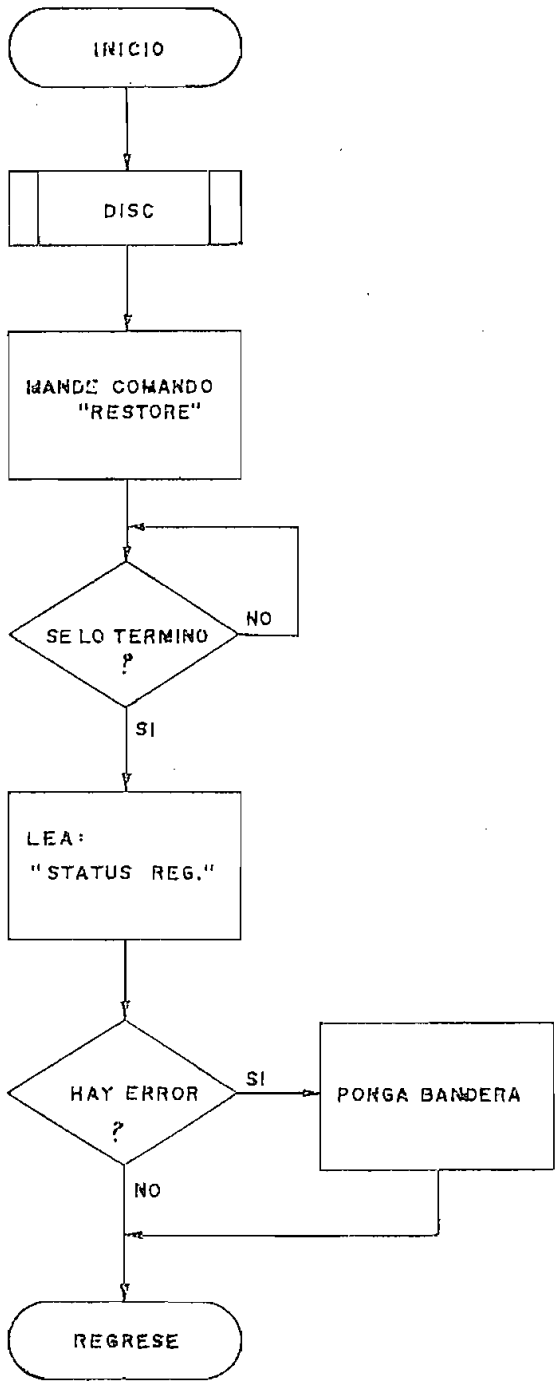
4.1b.1. INIC.


INICIO
BORRE REGISTRO DE PISTA DE CADA LECTORA/ESCR.
INICIALICE PIA:
CRA = --011X10
DDRA = FF
CRB = --101X00
DDRB = 00
DRA = 44
REGRESE

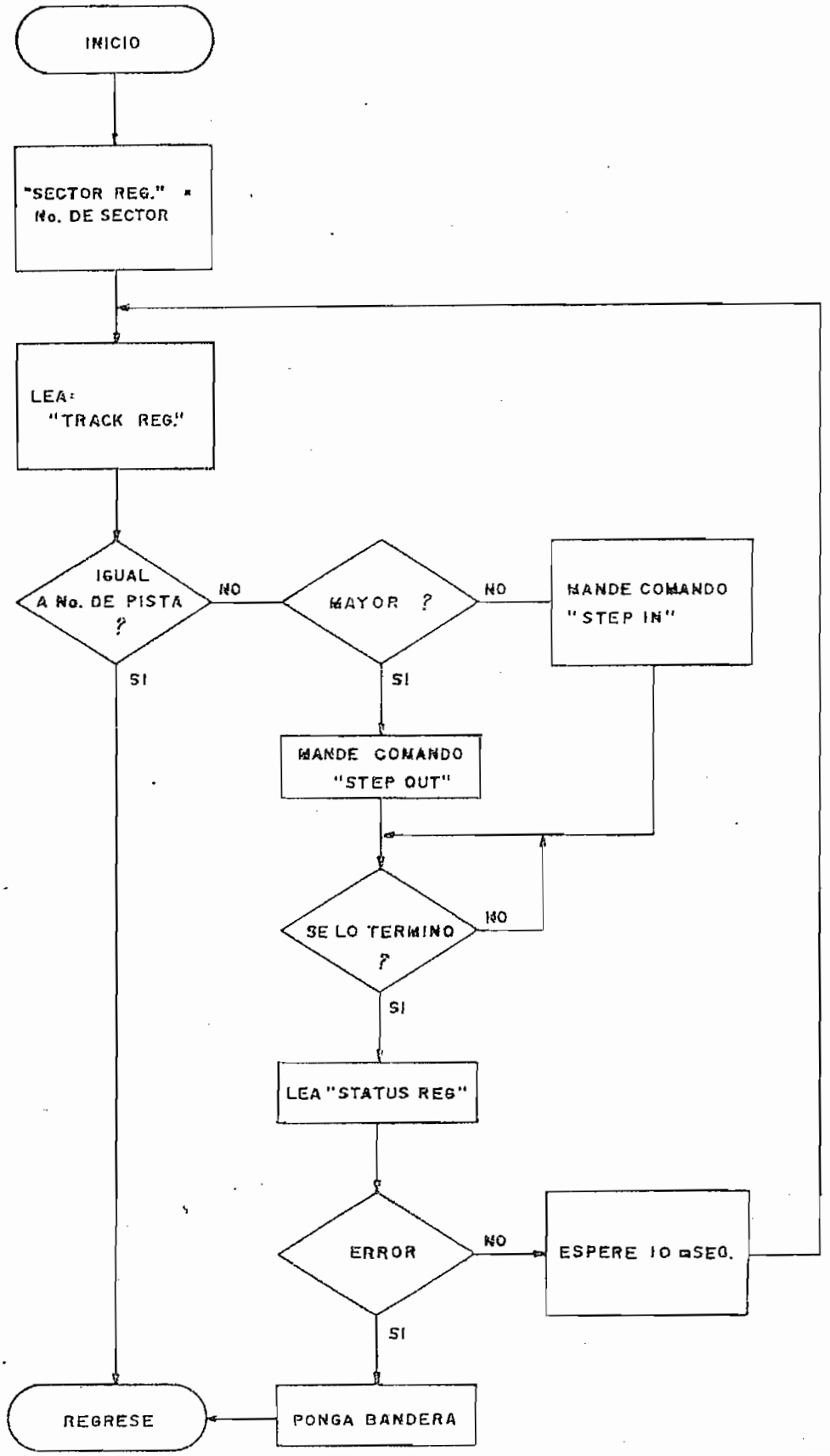
4.1b.2. DISC.


INICIO
ALMACENE No. DE PISTA DE LEC/ES. QUE SE UTILIZABA
SELECCIONE LA NUEVA LEC/ESC.
ALMACENE SU No. DE PISTA EN "TRACK REG."
REGRESE

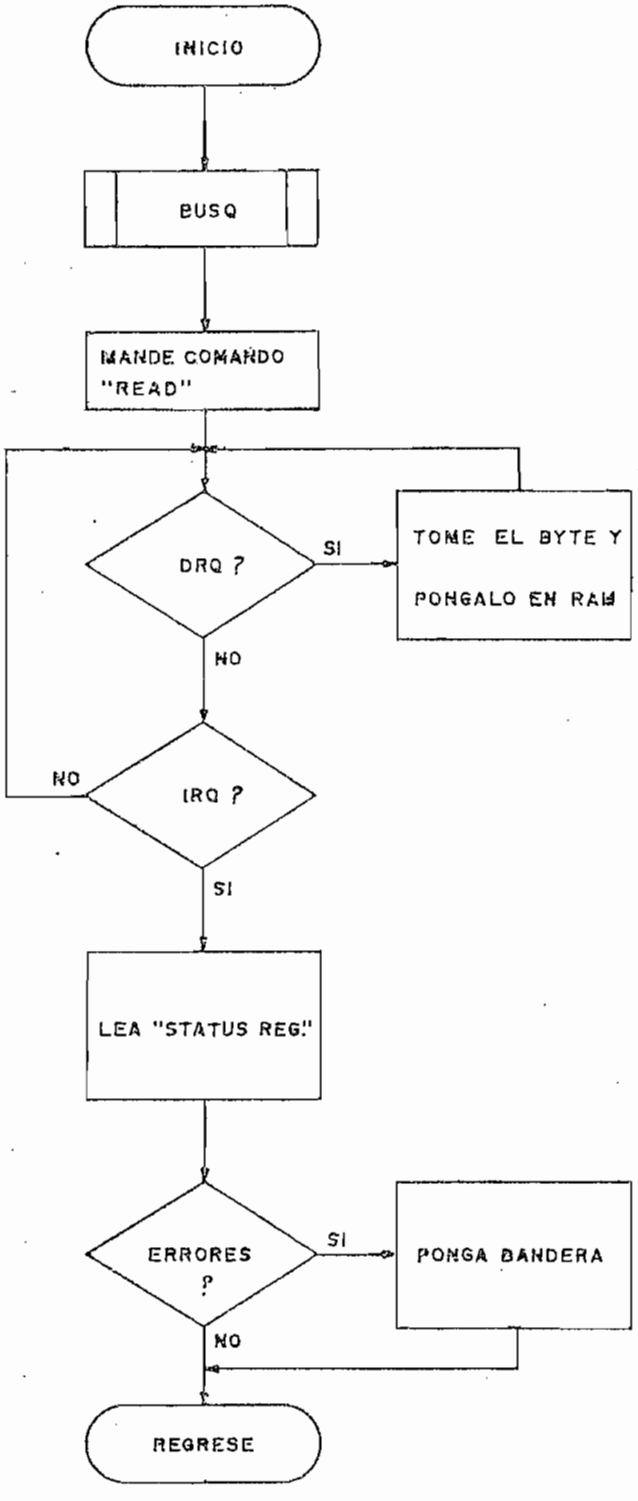
4.1b.3. REGR.


INICIO
DISC
MANDE COMANDO "RESTORE"
SE LO TERMINO ?
NO
SI
LEA: "STATUS REG."
HAY ERROR ?
SI
PONGA BANDERA
NO
REGRESE

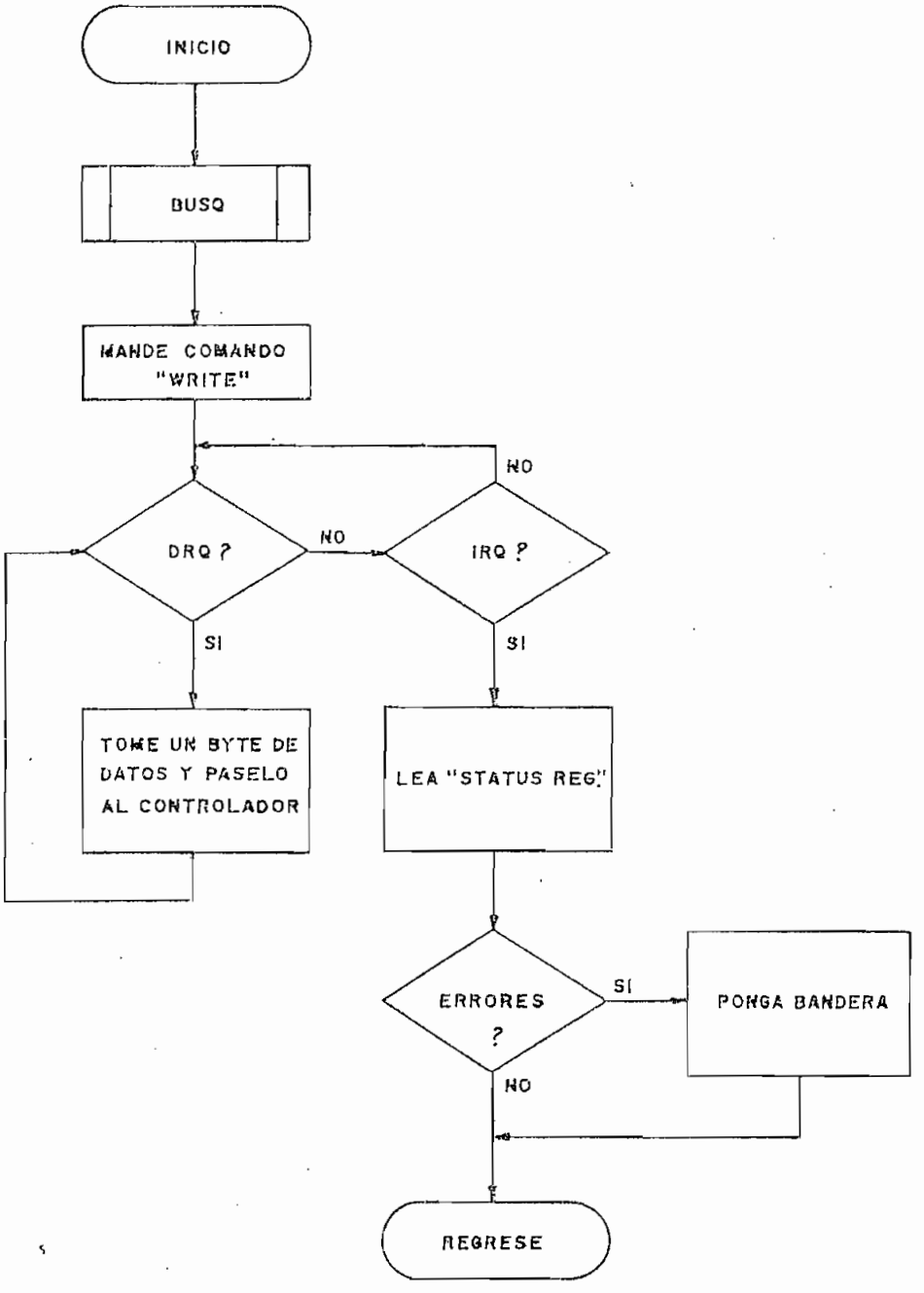
4.1b.4. BUSQ.


INICIO
"SECTOR REG." = No. DE SECTOR
LEA: "TRACK REG."
IGUAL A No. DE PISTA ?
NO
MAYOR ?
NO
MANDE COMANDO "STEP IN"
SI
SI
MANDE COMANDO "STEP OUT"
SE LO TERMINO ?
NO
SI
LEA "STATUS REG"
ERROR
NO
ESPERE 10 mSEG.
SI
PONGA BANDERA
REGRESE

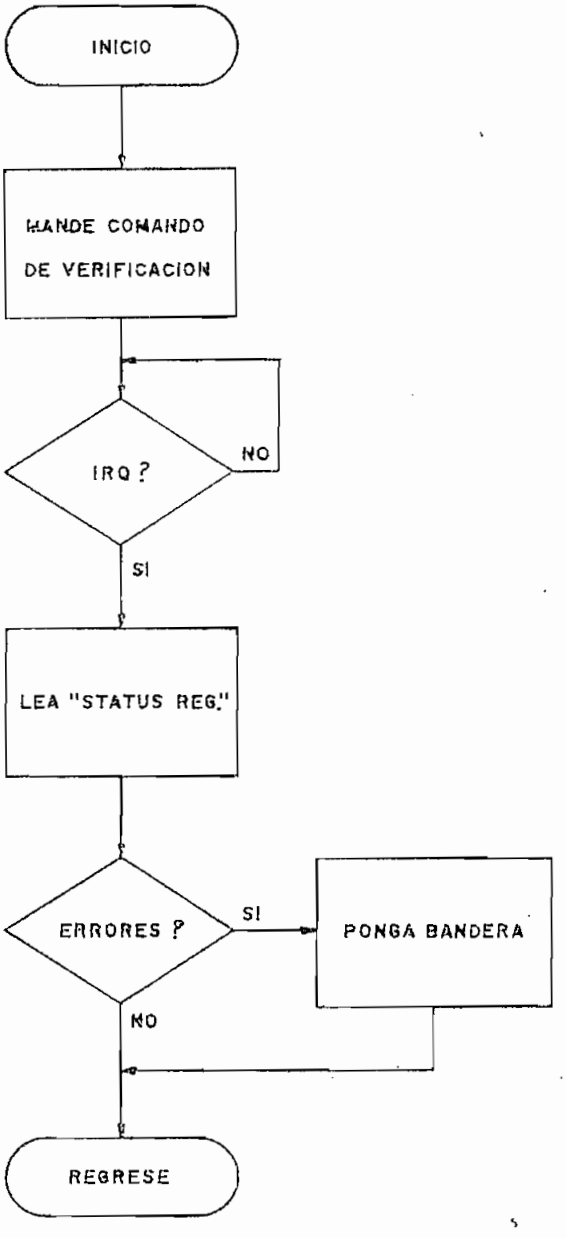
4.1b.5. LEA.


INICIO
BUSQ
MANDE COMANDO "READ"
DRQ ?
SI
TOME EL BYTE Y PONGALO EN RAM
NO
IRQ ?
NO
SI
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

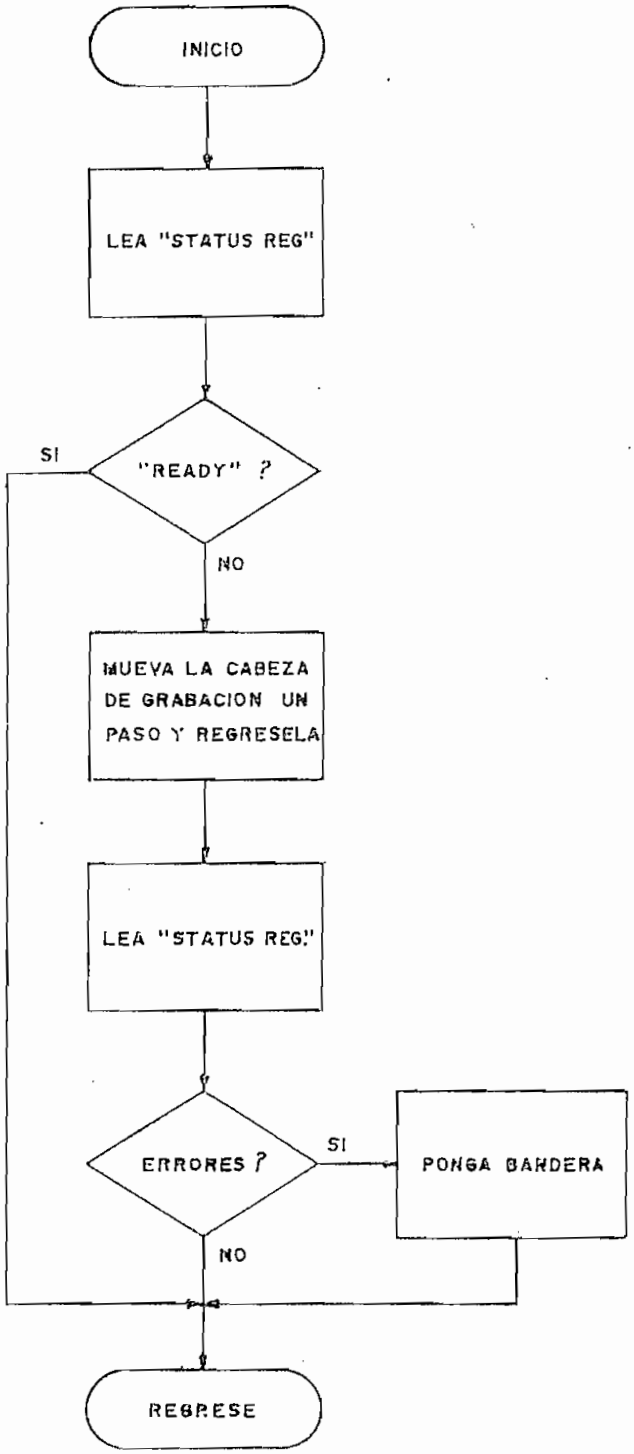
4.1b.6. ESCRIB.


INICIO
BUSQ
MANDE COMANDO "WRITE"
DRQ ?
NO
IRQ ?
NO
SI
SI
TOME UN BYTE DE DATOS Y PASELO AL CONTROLADOR
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

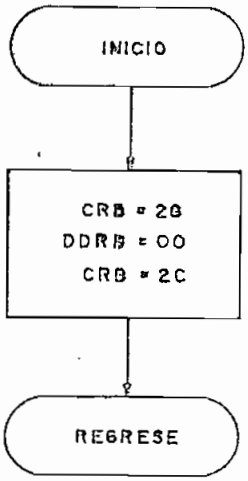
4.1b.7. VERIF.


INICIO
MANDE COMANDO
DE VERIFICACION
IRQ ?
NO
SI
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

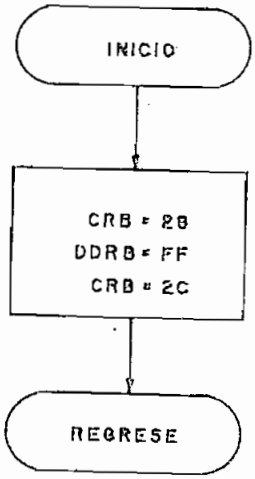
4.1b.8. CHKRDY.


INICIO
LEA "STATUS REG"
"READY" ?
SI
NO
MUEVA LA CABEZA DE GRABACION UN PASO Y REGRESELA
LEA "STATUS REG."
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

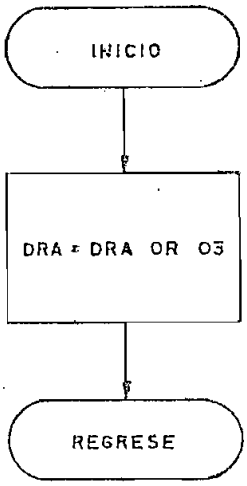
4.1b.9. ENTR.



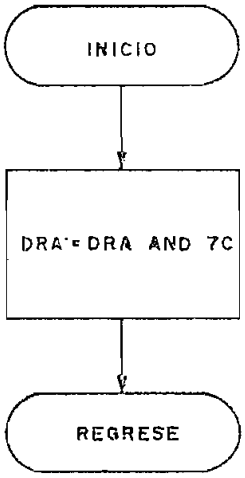
4.1b.10. SALID.



4.1b.11. DTARG.



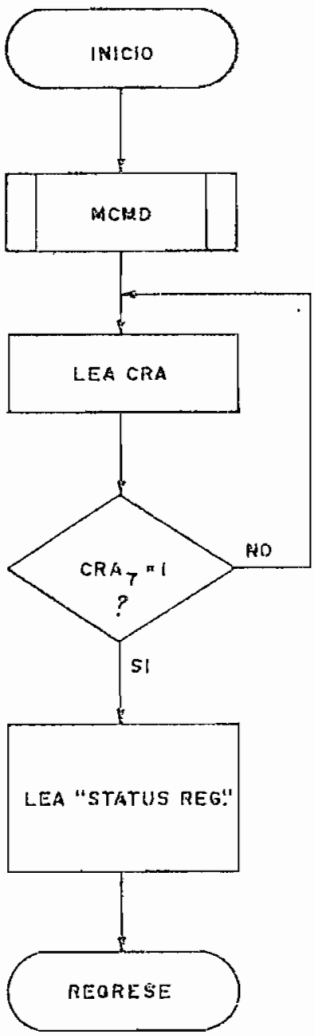
4.1b.12. CMRG.



4.1b.13. LEARG.

4.1b.14. MCMD.

4.1b.15. MCSP.



4.2. PROGRAMAS OPERATIVOS.

Seguidamente se da una ligera explicación de los programas operativos. Como en el caso de los programas de control, se darán las condiciones de entrada/salida para cada programa, incluyendo -cuando sea de importancia- el contenido de uno o más registros creados en memoria.

Los programas operativos podrán ser utilizados por el usuario siempre que se satisfagan las condiciones de entrada, la PIA se encuentre inicializada (por medio de la subrutina INIC), se haya seleccionado una lectora/escritora y el registro de pista del controlador contenga el número de pista sobre la cual se encuentra la cabeza de grabación de la lectora/escritora seleccionada (pudiendo utilizar para ello la subrutina REGR).

4.2.1. FORMAT.-

Por medio de esta subrutina se crea el formato en aquellos discos que van a ser utilizados por primera vez. El formato utilizado es de densidad simple y divide al disco en 35 pistas, subdivididas a su vez en 10 sectores de 256 bytes cada uno. La capacidad total de almacenamiento por disco será de 87,5 K bytes, de acuerdo a lo indicado en la sección 1.3.2.

En la sección 1.3.2. se indicó además la forma en que

se crea el formato en el disco y la función de cada uno de los bytes utilizados para ello. Este complejo proceso se reduce -utilizando el integrado FD1797-02- a escribir en cada pista los bytes indicados a continuación:

Nº DE BYTES	CONTENIDO	FUNCION
6	$ØØ	Fin de GAP IV (a partir del pulso índice)
1	$FC	Marca de pulso índice.
16	$FF	Comienzo de GAP I
* 6	$ØØ	Fin de GAPS I/III
1	$FE	Marca de bytes de identificación.
1	$XX	Número de pistas ($ØØ - $22)
1	$ØØ	Número de lado.
1	$ØX	Número de sector ($Ø1 - $ØA)
1	$Ø1	256 bytes/sector
1	$F7	Código para escritura de 2 bytes de CRC
11	$FF	Comienzo de GAP II
6	$ØØ	Fin de GAP II
1	$FB	Marca de campo de Datos
256	$E5	Datos
1	$F7	Código para escritura de 2 bytes de CRC.
14	$FF	Comienzo de GAP III/IV
72	$FF	Continuación de GAP IV

* NOTA: Los bytes indicados deberán ser escritos para cada sector en la pista (1Ø veces por pista).

Si bien, entre los programas de control no existe uno

para la escritura de una pista completa, se ha utilizado para ello a la subrutina ESCRIB desde la posición ESCRI2, almacenando previamente el código del comando "Write Track" en el acumulador A. La forma de utilización de este comando se puede encontrar en los apéndices de la presente tesis y su estudio ayudará a la mejor comprensión de la subrutina FORMAT.

Cabe anotar que en la creación del formato se ha utilizado la técnica de "sectores entrelazados", en ella la posición física de los sectores en una pista, no sigue una secuencia numérica, sinó que se intercalan sectores entre aquellos numéricamente consecutivos, con el objeto de proveer al sistema principal de un tiempo suficiente como para reorganizar los datos leídos y decidir si se debe leer el siguiente sector o no.

En el caso de discos de 5¼" la posición de los sectores en una pista sigue usualmente la siguiente secuencia: 1,3,5,7, 9,2,4,6,8,10 (secuencia Impar/Par), con ello, el tiempo requerido para leer dos o más sectores consecutivos puede ser reducido practicamente a la mitad del que se requeriría al no utilizar la técnica de sectores entrelazados, desde luego, lo anterior sucederá sólamente en el caso en que se tomen decisiones o se realice algún proceso cada vez que se termine la lectura de un sector.

Una vez creado el formato en el disco, FORMAT para directamente a la subrutina VRFDIS para verificar todos los sectores e inicializar el directorio del disco.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA	
Ninguna	NORMALES:	A = (DRA) B = (STATUS REG.) X = MEMPIS + 512 CCR: Z = 1
	DE ERROR1:	A = 3 B = (STATUS REG.) X = Indeterminado CCR: Z = Ø
	DE ERROR2:	A = (DRA) B = (STATUS REG.) X = Indeterminado CCR: Z = Ø

NOTA: ERROR1 = Error en la lectura de un sector.
ERROR2 = Error en la escritura del directorio.

4.2.2. VRFDIS.-

Esta subrutina puede ser considerada como parte de FORMAT y se la utiliza para verificar todos los sectores del disco e inicializar el directorio. Durante la verificación se realiza un proceso de lectura de todos los sectores pero, los datos leídos no son almacenados en RAM sinó que se buscan errores de CRC.

El directorio está constituído por los sectores 1 y 2 de la pista Ø, aunque solamente se utilizan los 350 primeros bytes de los 512 disponibles. Cada byte del directorio apunta a un sector en el disco y su contenido especifica el número de programa o grupo de datos al que pertenece dicho sector. Así por ejemplo, si en el byte número 12 del directorio se ha al

macenado el número 33 hexadecimal, significará que el sector número 2 de la pista 1, le corresponde al programa o grupo de datos número 33. Poniendo lo anterior en forma de una ecuación se tiene:

DIRECTORIO = Número de programa

Pista*10 + Sector

CONDICIONES DE ENTRADA		CONDICIONES DE SALIDA
	NORMALES:	A = (DRA) B = (STATUS REG.) X = MEMPIS + 512 CCR: Z = 1
Ninguna	DE ERROR1:	A = 3 B = (STATUS REG.) X = Indeterminado CCR: Z = Ø
	DE ERROR2:	A = (DRA) B = (STATUS REG.) X = Indeterminado CCR: Z = Ø

NOTA: ERROR1 = Error de lectura de un sector
ERROR2 = Error en la escritura del directorio.

4.2.3. GRBPR.-

Esta subrutina se utiliza para grabar un programa o un grupo de datos en el disco. El número de programa y las direcciones de memoria en donde comienza y termina el programa (o grupo de datos), son directamente preguntados al usuario por medio de las subrutinas de consola: PREGPD y PREGCF. Si el usuario no desea que estos datos sean ingresados desde el tecla

do sino que prefiere especificarlos dentro de alguno de sus programas, puede utilizar la subrutina GRBPR desde la posición GRBPRØ siempre que cumpla con las condiciones de entrada indicadas.

Las direcciones de memoria donde comienza y termina el programa, se adjuntan al comienzo del primer sector asignado para la grabación y son seguidas por los bytes que conforman el programa o grupo de datos. Los sectores asignados -cuyo número depende de la longitud del programa-, son siempre aquellos sectores libres (sin datos) que se encuentran lo más cerca posible al sector 1 de la pista Ø, es decir aquellos "más externos".

Si el número de programa que se está grabando ya ha sido utilizado, pero ocupa un menor número de sectores en el disco, GRBPR borra automáticamente los sectores sobrantes, de forma que puedan ser utilizados en grabaciones que se realicen posteriormente, con ello se logra una mejor utilización del espacio disponible en el disco.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
GRBPR: Ninguna	NORMALES: A = (DRA)
GRBPRØ: NPROG = Número de programa	B = (STATUS REG.)
COMADD = Dirección de memoria donde comienza el programa.	X = MEMPIS + 512
	FIN = 1
	CCR: Z = 1

FINADD = Dirección de memoria donde termina el programa.

DE ERROR1: A = (DRA)
B = (STATUS REG.)
X = MEMPIS + 512
FIN = Ø
CCR: Z = 1

DE ERROR2: A = 3
B = (STATUS REG.)
X = Indeterminado
CCR: Z = Ø ; C = 1

DE ERROR3: A = 3
B = (STATUS REG.)
X = Indeterminado
CCR: Z = Ø ; C = Ø

DE ERROR4: A = (DRA)
B = (STATUS REG.)
X = Indeterminado
CCR: Z = Ø ; C = Ø

NOTA: ERROR1 = Error de disco lleno.
ERROR2 = Error de lectura del directorio.
ERROR3 = Error de verificación del directorio o sector.
ERROR4 = Error de escritura del directorio o sector.

4.2.4. LEAPRG.-

La subrutina LEAPRG permite leer un programa del disco. El número de programa o grupo de datos es directamente preguntado al usuario, por medio de la subrutina de consola PREGPD,

pero si el usuario desea definir este número dentro de alguno de sus programas, puede utilizar la subrutina LEAPRG desde la posición LEAPRØ, para ello sólo tendrá que cumplir con las condiciones de entrada que se indican.

Como se indicó en la sección 4.2.3., los cuatro primeros bytes del primer sector asignado al programa contiene las localidades de memoria en las que éste comienza y termina, por lo que LEAPRG las utiliza, durante la lectura, para que el programa o grupo de datos sea repuesto a su posición original (aquella que ocupaba cuando fue grabado).

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	NORMALES: A = (DRA)
	B = (STATUS REG.)
	X = (FINADD)
	CCR: Z = 1 ; C = Ø
	DE ERROR1: A = 3
	B = (STATUS REG.)
LEAPRG: Ninguna	X = Indeterminado
LEAPRØ: NPROG = Número de programa	CCR: Z = Ø ; C = 1
	DE ERROR2: A = Nº de programa
	B = 35
	X = MEMPIS + 350
	CCR: Z = Ø ; C = Ø
	DE ERROR3: A = 3
	B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = Ø

NOTA: ERROR1 = Error en la lectura del directorio.

ERROR2 = Programa no existe o está mal grabado.

ERROR3 = Error en la lectura de un sector del programa.

4.2.5. BORPRG.-

Utilizando esta subrutina se pueden borrar programas del disco, de igual forma que las subrutinas GRBPR y LEAPRG el usuario puede ingresar el número de programa a borrarse desde alguno de sus programas -utilizando para ello la subrutina BORPRG desde la posición BORPRØ- ó desde el teclado -utilizando la subrutina BORPRG desde le comienzo.

Cabe anotar que el programa no es propiamente borrado del disco, sino que se lo elimina del directorio, de esta forma los sectores que estuvieron ocupados por el programa podrán ser utilizados en alguna grabación posterior.

CONDICIONES DE ENTRADA

BORPRG: Ninguna

BORPRØ: NPROG = Número de programa

CONDICIONES DE SALIDA

NORMALES: A = (DRA)

B = (STATUS REG.)

X = MEMPIS + 512

CCR: Z = 1 ; C = Ø

DE ERROR1: A = Nº de programa

B = 35

X = MEMPIS + 35Ø

CCR: Z = Ø ; C = Ø

DE ERROR2: A = 3

B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = 1

DE ERROR3: A = 3

B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = Ø

DE ERROR4: A = (DRA)

B = (STATUS REG.)

X = Indeterminado

CCR: Z = Ø ; C = Ø

NOTA: ERROR1 = Programa no existe en el disco.

ERROR2 = Error de lectura del directorio.

ERROR3 = Error de verificación del directorio.

ERROR4 = Error de escritura del directorio.

4.2.6. LEADIR.-

Esta subrutina se utiliza para leer el directorio del disco a memoria. Debe recordarse que el directorio está constituído por los sectores 1 y 2 de la pista Ø.

CONDICIONES DE ENTRADA CONDICIONES DE SALIDA

NORMALES: A = (DRA)

	B = (STATUS REG.)
	X = MEMPIS + 512
	CCR: Z = 1
Ninguna	DE ERROR: A = (DRA)
	B = (STATUS REG.)
	X = Indeterminado
	CCR: Z = Ø

4.2.7. DIRINI.-

Esta subrutina lee el directorio a memoria, utilizando para ello la subrutina LEADIR. En caso de que se produzca un error de lectura, intenta leerlo dos veces adicionales y una vez que se tiene el directorio en memoria, pasa directamente a la subrutina ENCPRG.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	NORMALES: A = (NPROG)
	B = Indeterminado
	X = MEMPIS + 1Ø*(PISTA)
	+ (SECTOR) - 1
Ninguna	CCR: Z = 1
	DE ERROR1: A = (NPROG)
	B = $23
	X = MEMPIS + 35Ø
	CCR: Z = Ø
	DE ERROR2: A = 3

B = (STATUS REG.)

X = Indeterminado

CCR: C = 1

NOTA: ERROR1 = Error de disco lleno.

ERROR2 = Error de lectura del directorio.

4.2.8. ENCPRG.-

Por medio de la subrutina ENCPRG se encuentran los números de pista y de sector que han sido asignados a un programa. Una vez encontrado el primer sector asignado, se podrán encontrar los siguientes, utilizando esta subrutina desde la posición ENCPR1.

Si en algún momento se pide a la subrutina que encuentre un programa que no se encuentre en el directorio, o que se trate de buscarlo por sobre la pista 34 sector 1Ø, la subrutina pondrá el código de error de disco lleno o programa inexistente: $23.

Debe notarse que ENCPRG supone que el directorio ha sido leído y que los registros de pista y de sector se han inicializado con Ø1 y Ø1 respectivamente.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
ENCPRG: A = Nº de programa a encontrarse.	NORMALES: A = A
X = MEMPIS + 1Ø	B = Indeterminado

ENCPR1: A = Nº de programa a encontrarse

X = MEMPIS + 1Ø*(PISTA)

+ (SECTOR) - 1

X = MEMPIS + 1Ø*(PISTA) +(SECTOR) - 1

CCR: Z = 1

DE ERROR: A = A

B = $23

X = MEMPIS + 35Ø

CCR: Z = Ø

4.2.9. ALMAC.-

Esta subrutina se utiliza para escribir en memoria un número determinado de bytes con un valor especificado por el usuario.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
A = Valor especificado	A = A
B = Nº de bytes a escribir	B = 1
X = Dirección de memoria desde donde se empieza.	X = X+B

4.2.1Ø. INIVAR.-

Esta subrutina se utiliza para inicializar los registros de pista y de sector con $Ø1, haciendo además que el registro X apunte al byte número once del directorio.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
Ninguna	A = 1
	B = B

X ≐ MEMPIS + 1Ø

CCR: N = Ø ; Z = Ø ; V = Ø

4.2.11. INTEN.-

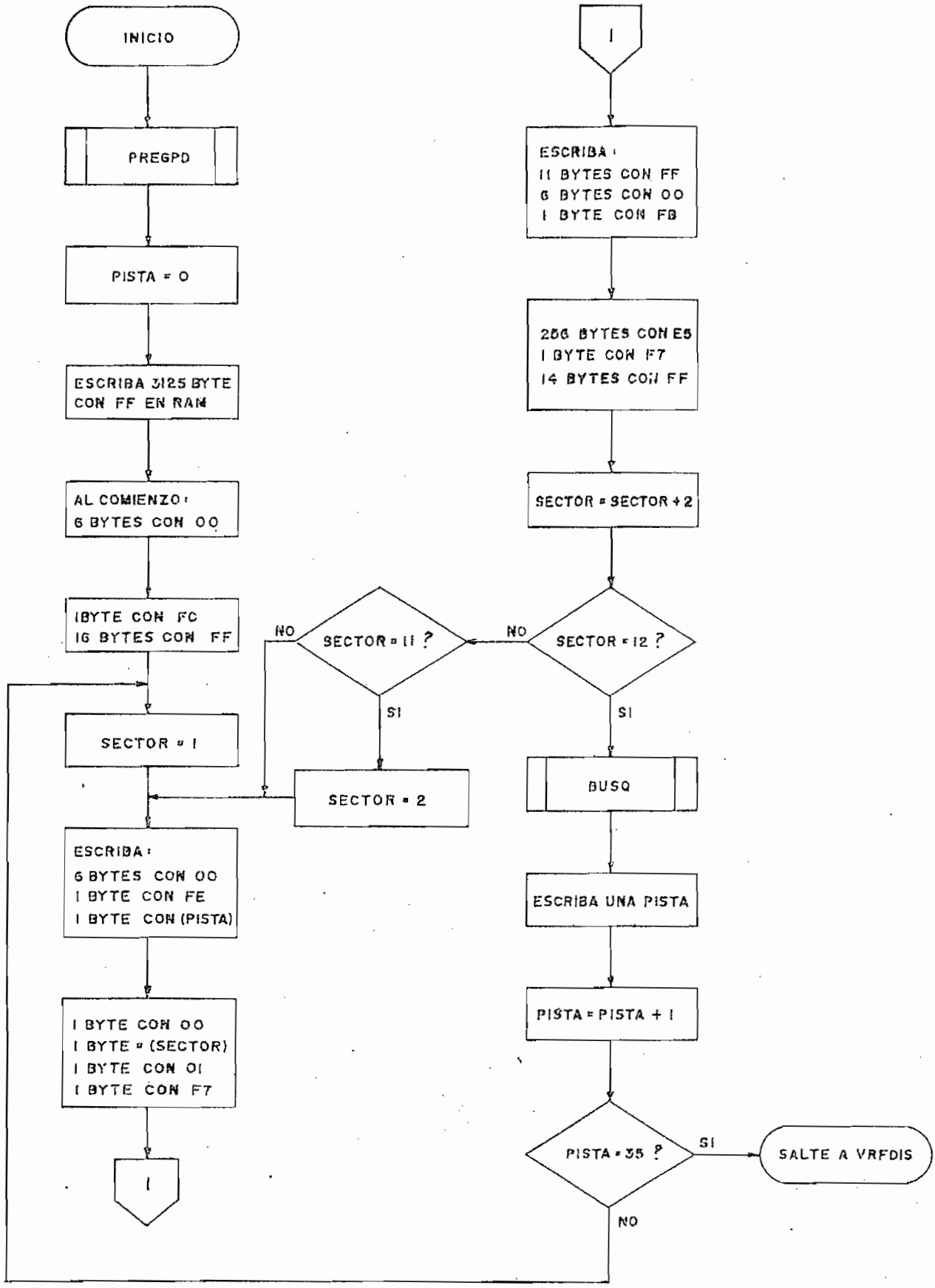
INTEN es un contador de intentos de lectura, además pondrá un indicador en el momento en que se hayan cumplido tres intentos.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
	A = (INTENT)
	B = B
	X = X
	CCR: Z = 1 (3 Intentos)
	X = Ø (≠ 3 Intentos)

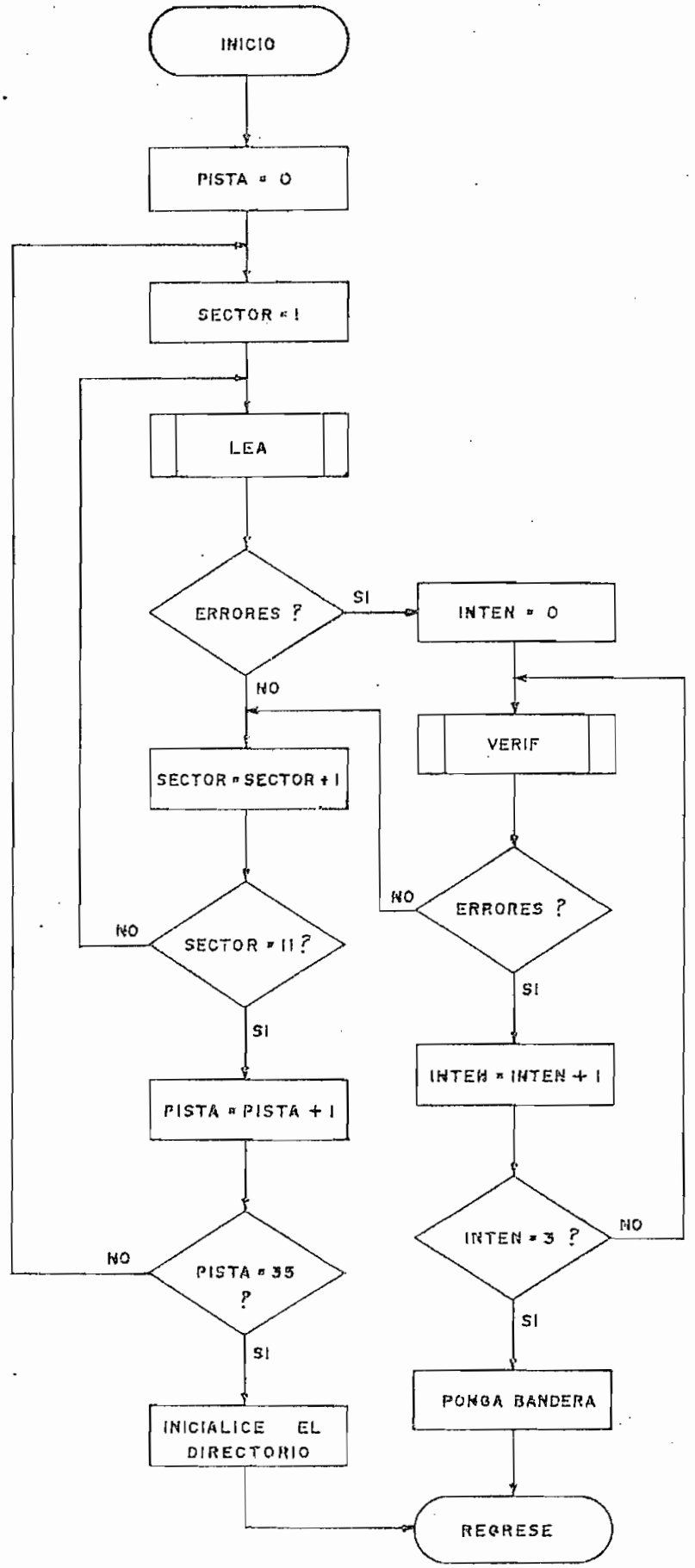
4.2b. DIAGRAMAS DE FLUJO DE LOS PROGRAMAS OPERATIVOS.

A continuación se presentan los diagramas de flujo correspondientes a los programas operativos.

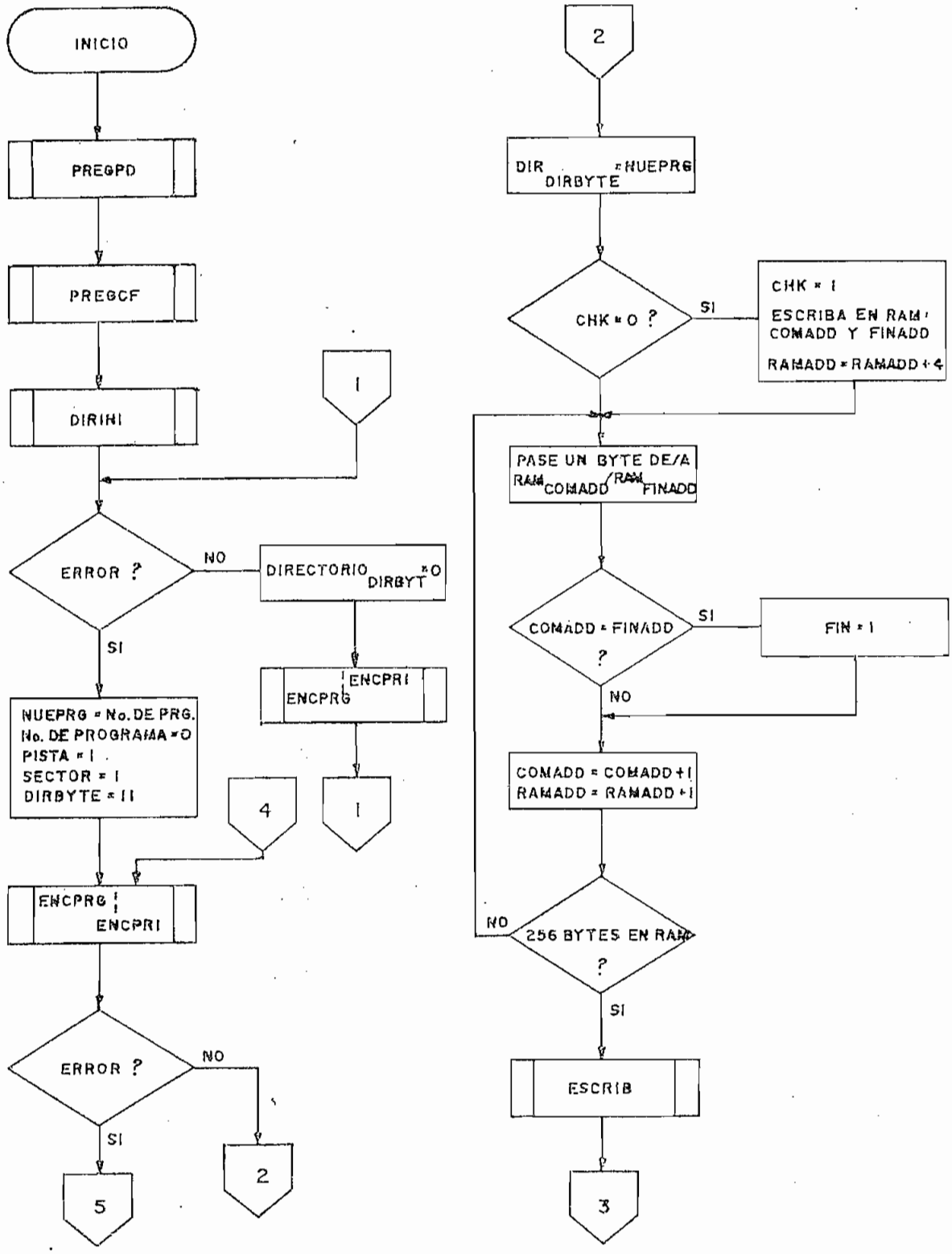
4.2b.1. FORMAT.


INICIO
PREGPD
PISTA = 0
ESCRIBA 3125 BYTE CON FF EN RAM
AL COMIENZO: 6 BYTES CON 00
1BYTE CON FC 16 BYTES CON FF
SECTOR = 1
ESCRIBA: 6 BYTES CON 00 1 BYTE CON FE 1 BYTE CON (PISTA)
1 BYTE CON 00 1 BYTE = (SECTOR) 1 BYTE CON 01 1 BYTE CON F7
1
1
ESCRIBA: 11 BYTES CON FF 6 BYTES CON 00 1 BYTE CON FB
256 BYTES CON E5 1 BYTE CON F7 14 BYTES CON FF
SECTOR = SECTOR + 2
SECTOR = 12 ?
NO
SI
SECTOR = 11 ?
NO
SI
SECTOR = 2
BUSQ
ESCRIBA UNA PISTA
PISTA = PISTA + 1
PISTA = 35 ?
SI
SALTE A VRFDIS
NO

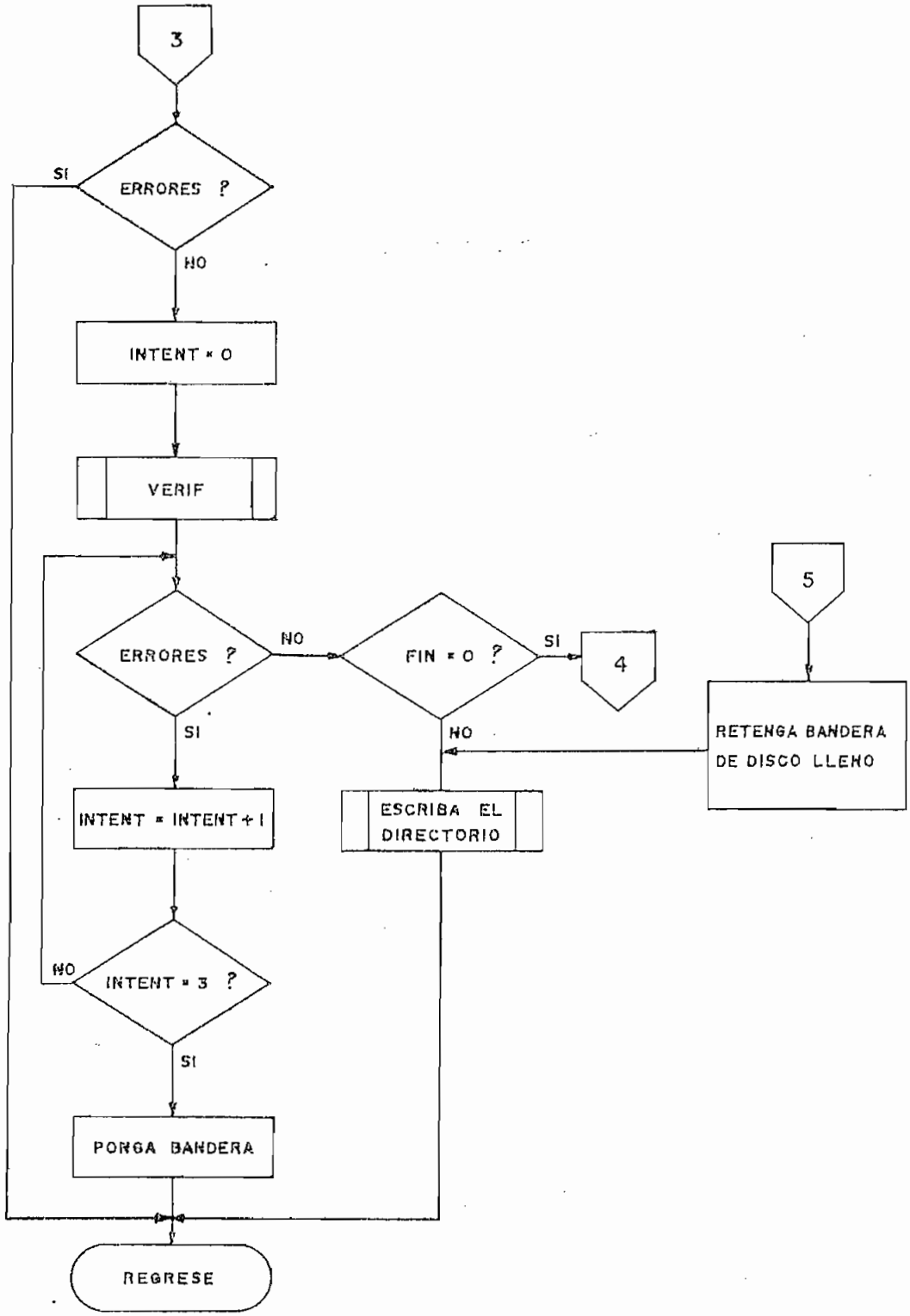
4.2b.2. VRFDIS.


INICIO
PISTA = 0
SECTOR = 1
LEA
ERRORES ?
SI
NO
INTEN = 0
VERIF
SECTOR = SECTOR + 1
NO
ERRORES ?
NO
SECTOR = 11 ?
SI
SI
INTEN = INTEN + 1
PISTA = PISTA + 1
INTEN = 3 ?
NO
NO
PISTA = 35 ?
SI
SI
PONGA BANDERA
INICIALICE EL DIRECTORIO
REGRESE

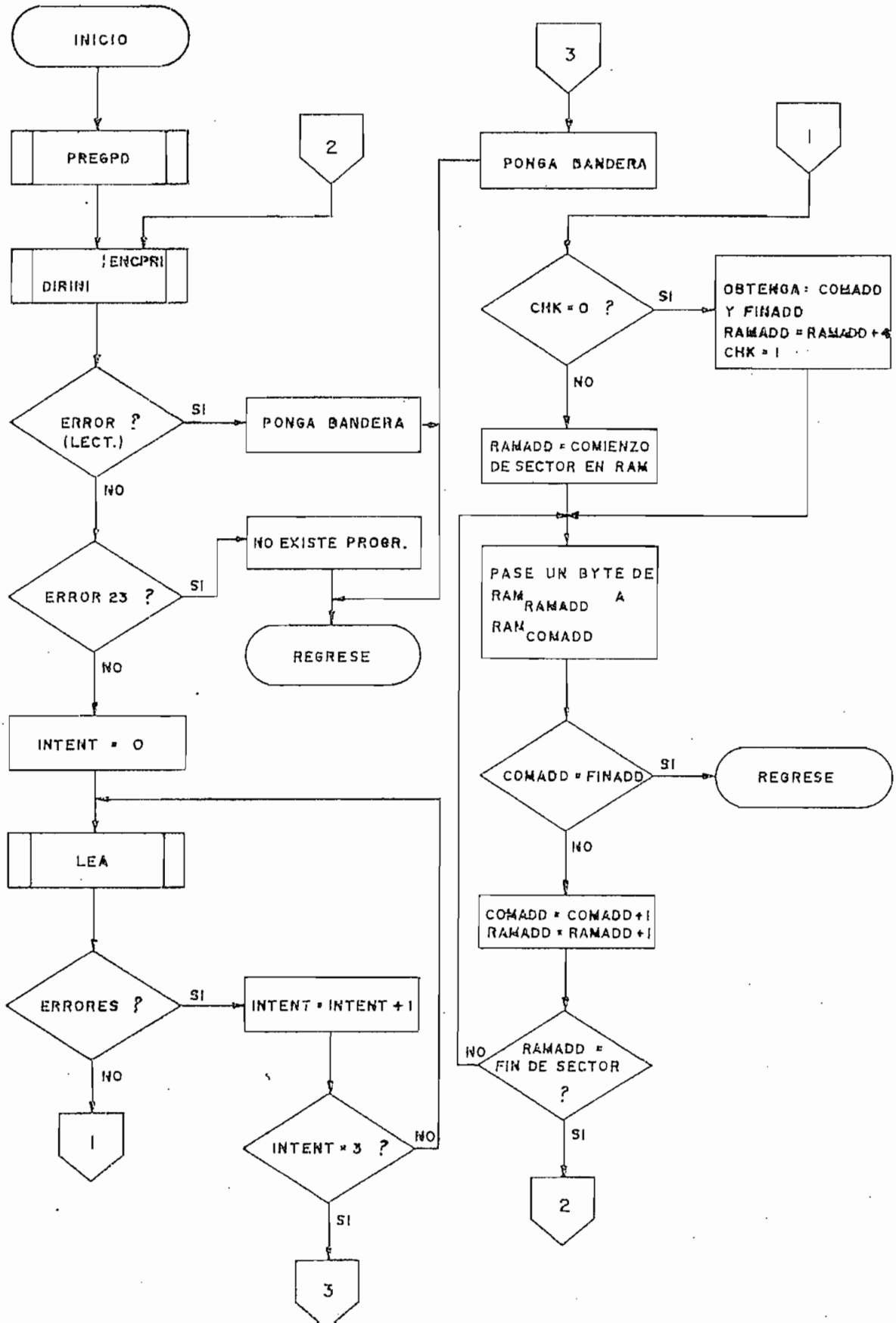
4.2b.3. GRBPRG.


INICIO
PREGPD
PREGCF
DIRINI
1
ERROR ?
NO
DIRECTORIO DIRBYT =0
ENCPRI
ENCPRG
1
SI
NUEPRG = No. DE PRG.
No. DE PROGRAMA = 0
PISTA = 1
SECTOR = 1
DIRBYTE = 11
4
ENCPRG
ENCPRI
ERROR ?
NO
SI
5
2
2
DIR DIRBYTE = NUEPRG
CHK = 0 ?
SI
CHK = 1
ESCRIBA EN RAM
COMADD Y FINADD
RAMADD = RAMADD + 4
PASE UN BYTE DE/A
RAM COMADD / RAM FINADD
COMADD = FINADD ?
SI
FIN = 1
NO
COMADD = COMADD + 1
RAMADD = RAMADD + 1
NO
256 BYTES EN RAM ?
SI
ESCRIB
3

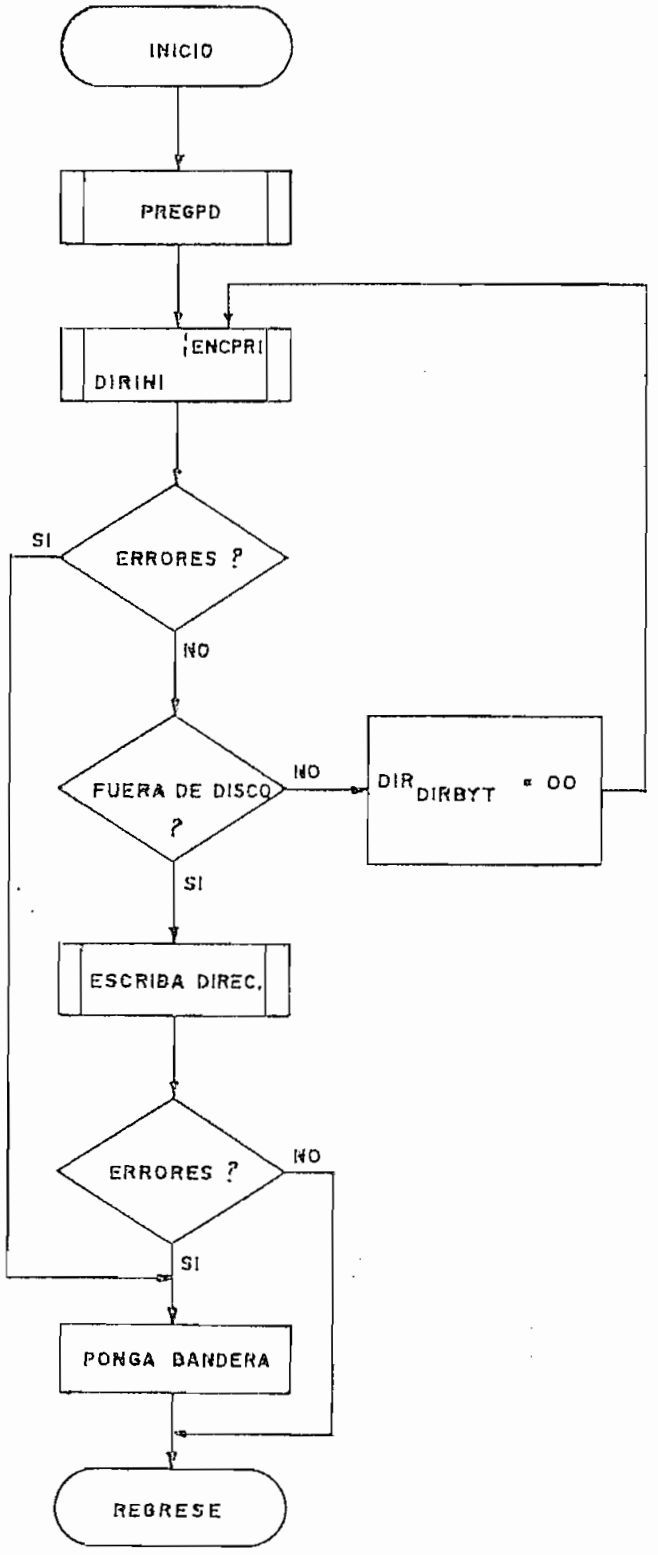
4.2b.3. GRBPRG.(Cont.)



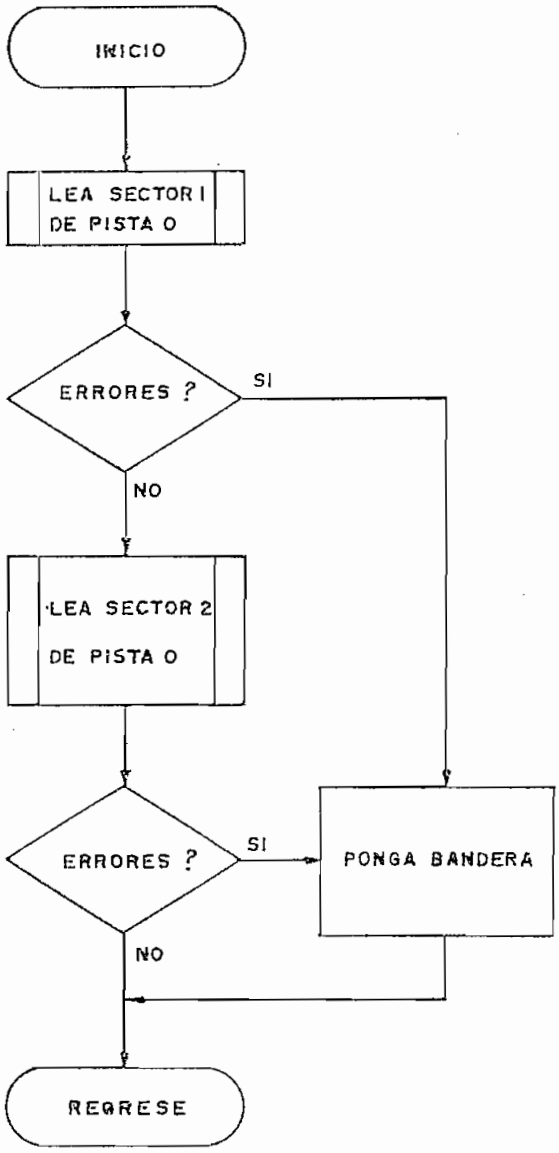
4.2b.4. LEAPRG.


INICIO
PREGPD
2
ENCPRI
DIRINI
ERROR ? (LECT.)
SI
PONGA BANDERA
NO
ERROR 23 ?
SI
NO EXISTE PROGR.
REGRESE
NO
INTENT = 0
LEA
ERRORES ?
SI
INTENT = INTENT + 1
NO
1
INTENT = 3 ?
NO
SI
3
3
PONGA BANDERA
1
CHK = 0 ?
SI
OBTENGA: COMADD Y FINADD
RAMADD = RAMADD + 4
CHK = 1
NO
RAMADD = COMIENZO DE SECTOR EN RAM
PASE UN BYTE DE RAM RAMADD A RAM COMADD
COMADD = FINADD
SI
REGRESE
NO
COMADD = COMADD + 1
RAMADD = RAMADD + 1
RAMADD = FIN DE SECTOR ?
NO
SI
2

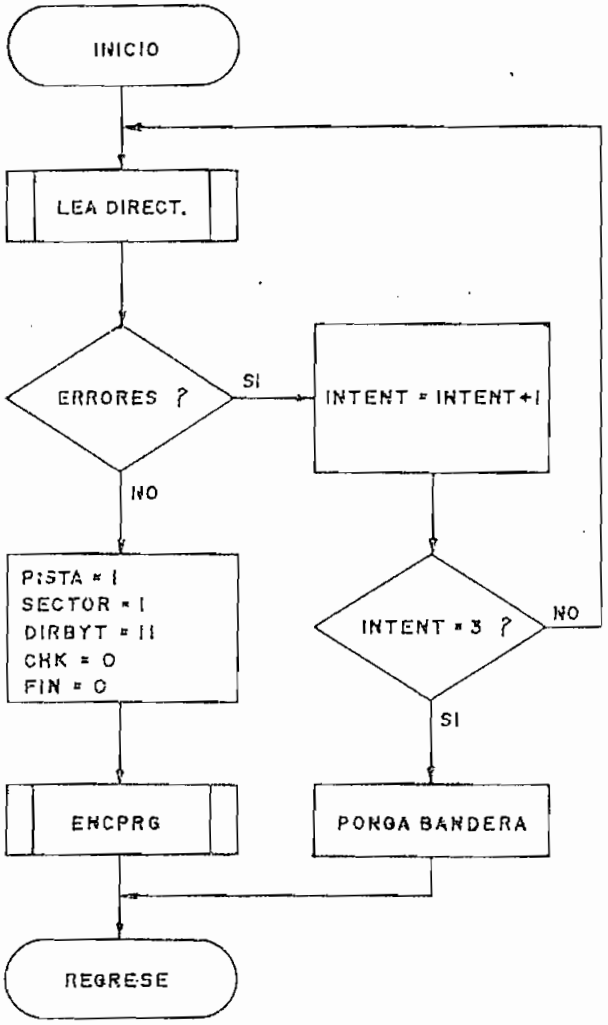
4.2b.5. BORPRG.


INICIO
PREGPD
ENCPRI
DIRINI
ERRORES ?
SI
NO
FUERA DE DISCO ?
NO
DIR DIRBYT = 00
SI
ESCRIBA DIREC.
ERRORES ?
NO
SI
PONGA BANDERA
REGRESE

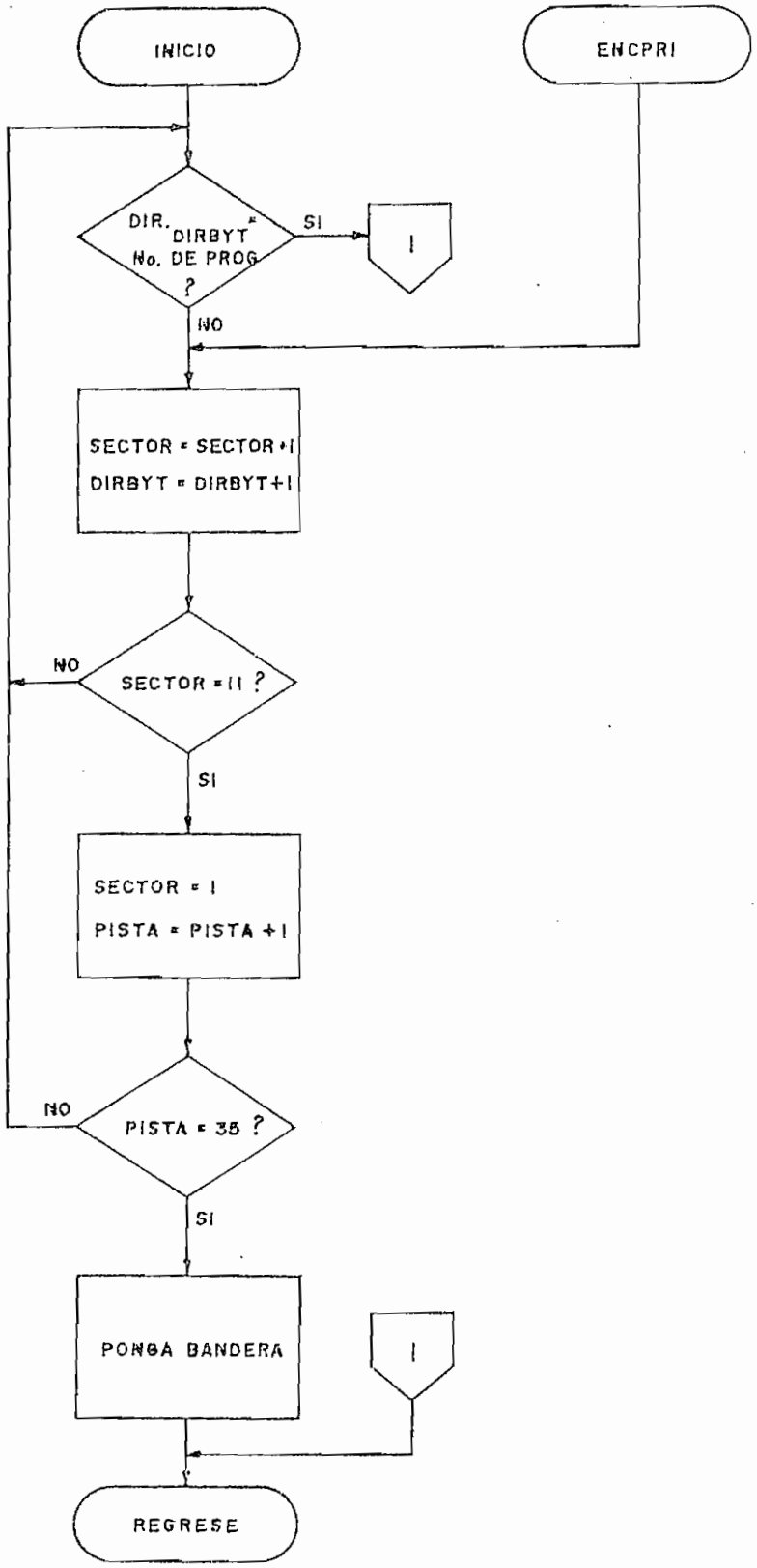
4.2b.6. LEADIR.


INICIO
LEA SECTOR 1
DE PISTA 0
ERRORES ?
SI
NO
LEA SECTOR 2
DE PISTA 0
ERRORES ?
SI
PONGA BANDERA
NO
REGRESE

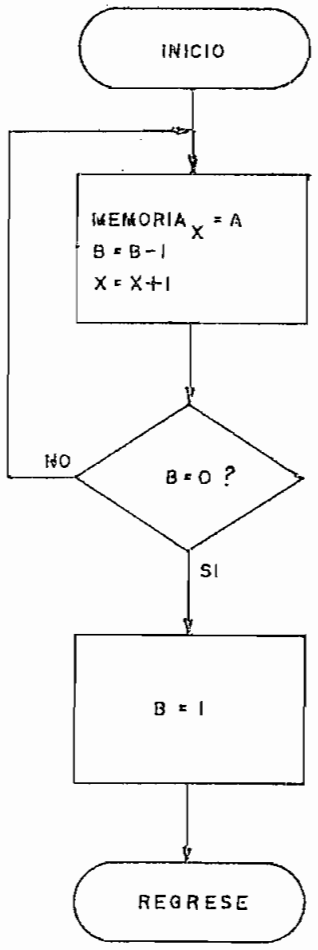
4.2b.7. DIRINI.


INICIO
LEA DIRECT.
ERRORES ?
SI
INTENT = INTENT+1
NO
PISTA = 1
SECTOR = 1
DIRBYT = 11
CHK = 0
FIN = 0
INTENT = 3 ?
NO
SI
ENCPRG
PONGA BANDERA
REGRESE

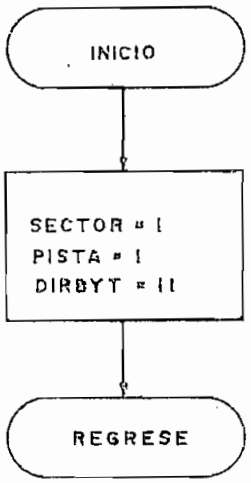
4.2b.8. ENCPRG.


INICIO
ENCPRI
DIR. DIRBYT No. DE PROG ?
SI
1
NO
SECTOR = SECTOR+1
DIRBYT = DIRBYT+1
NO
SECTOR = 11 ?
SI
SECTOR = 1
PISTA = PISTA +1
NO
PISTA = 35 ?
SI
PONGA BANDERA
1
REGRESE

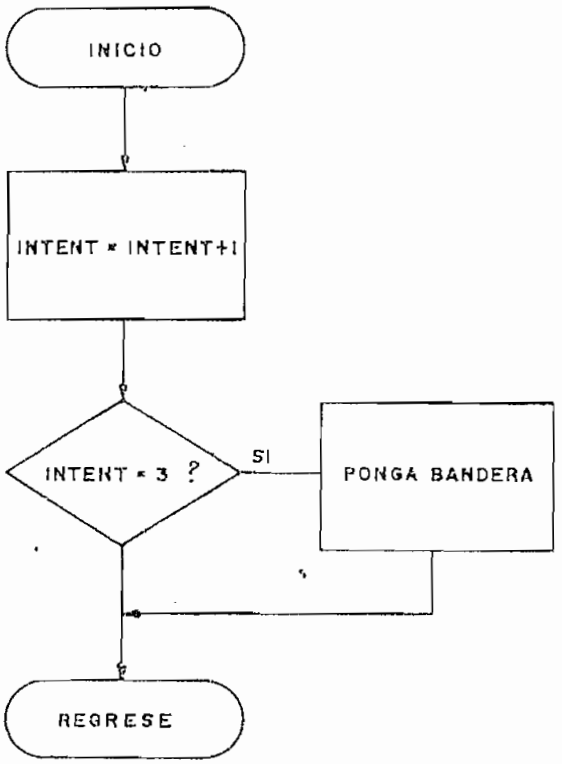
4.2b.9. ALMAC.



4.2b.10. INIVAR.



4.2b.11. INTEN.



4.3. PROGRAMAS DE CONSOLA

Los programas de consola se encargan de los procesos de comunicación con el usuario, en el presente caso controlan el teclado y la pantalla para el ingreso y salida de datos, por esta razón los programas de consola deberán ser modificados cada vez que se realice un cambio en esos elementos de entrada/salida.

Como en los otros tipos de programas se darán -siempre que sea posible- las condiciones de entrada/salida. Deberá notarse que algunos de los programas de consola se manejan directamente desde el teclado y que una vez finalizados regresan a él, por lo que en ellos no se podrá hablar de condiciones de entrada/salida.

4.3.1. PREGPD.-

Esta subrutina se utiliza para preguntar al usuario el número de lectora/escritora a utilizarse y el número de programa que se requiere para algún proceso. La subrutina pondrá una "D" en la pantalla para preguntar el número de lectora/escritora; una vez ingresado un dígito correcto (Ø $\leq$ dígito $\leq$ 3) verá si esa lectora/escritora está ya seleccionada, y de no ser así, la seleccionará por medio de la subrutina DISC. Una vez realizado el proceso anterior PREGPD pondrá una "A" en la pantalla para preguntar a qué número de programa se hace referencia, en este caso esperará el ingreso de dos dígitos (ØØ - FF) y almacenará este número en el registro NPROG.

Cada vez que se seleccione una nueva lectora/escritora, PREGPD averiguará en los registros de pista, si la cabeza de esa lectora/escritora se encuentra sobre la pista ØØ, si este es el caso mandará un comando de regreso a la pista ØØ, evitando de esta forma que se produzcan errores debido a que en la subrutina INIC se inicializaron todos los registros de pista con $ØØ. Si se produce un error mientras se busca la pista ØØ, la subrutina PREGPD regresará un nivel de subrutina más alto que aquel desde el cual fue llamada.

Mientras PREGPD esté pidiendo datos, el usuario tendrá la opción de "escapar" de la subrutina, aplastando la tecla E, en este caso se hará un salto a la subrutina PPRIN en su posición ESCAPE.

CONDICIONES DE ENTRADA		CONDICIONES DE SALIDA
	NORMALES:	A = Nº de programa B = 4 últimos bits de A X = (BPADR)

Ninguna	DE ERROR:	A = (DRA)
		B = (STATUS REG.)
		X = REGPIS + # de lectora/escritora seleccionada
		SP = SP + 4
		CCR: Z = Ø

4.3.2. PREGCF.-

Por medio de esta subrutina se preguntan al usuario las direcciones de memoria en donde comienza un programa y donde termina el mismo, estas direcciones son convertidas a números

hexadecimales de dos bytes para su almacenamiento en los registros COMADD y FINADD creados en memoria.

PREGCF pondrá una "C" en la pantalla para preguntar la dirección en donde comienza un programa y una "F" al preguntar la dirección en donde finaliza. Cuatro dígitos deberán ser ingresados para cada una de estas direcciones y como sucede en la subrutina PREPD, el usuario tiene la opción de "escapar" de la subrutina por medio de la tecla E.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
Ninguna	A = Byte menos significativo de (FINADD) B = 4 últimos bits de A. X = (FINADD)

4.3.3. INGRES.-

Esta subrutina se utiliza para el ingreso de datos desde el teclado. El número de dígitos que ingresen no podrá ser mayor que seis o menor a uno y serán almacenados en los registros creados en memoria: DSBUF - DSBUF+5.

INGRES escribe también una letra en la pantalla: A - F (códigos ØA - ØF), de esta forma el usuario podrá reconocer qué tipo de dato se le está pidiendo. Si se requiere que en la pantalla aparezca "-" ó un espacio en blanco, se podrán utilizar los códigos: 1Ø y 11 respectivamente.

Los cuatro primeros datos ingresados podrán ser convertidos a un número hexadecimal de dos bytes, para ello deberá

hacerse uso de la subrutina de monitor BLDX. Esta subrutina toma los datos almacenados en los registros: DSBUF - DSBUF+3, los convierte a dos bytes hexadecimales y guarda el byte más significativo en el registro BPADR y el menos significativo en el registro BPADR+1, por último el número completo es también almacenado en el registro X del microprocesador.

CONDICIONES DE ENTRADA	CONDICIONES DE SALIDA
A = Número de dígitos a ingresar	A = Indeterminado
B = Código para la pantalla	B = 4 últimos bits de último dato
	X = (XDBCHK)
	CCR: Z = 1

4.3.4. TECDIS.-

TECDIS tiene funciones múltiples, puede actuar como subrutina para el ingreso de datos (en cuyo caso es llamada desde la subrutina INGRES) o como programa que controla las opciones que tiene el usuario para el manejo de la información en sus discos, en este caso la función de cada una de las teclas es diferente a la que tenían bajo el control de los programas de monitor.

Mientras está actuando como subrutina (CHK≠Ø), se permite unicamente el ingreso de las teclas numéricas (Ø - F) y de la tecla de control: E, esta tecla hará que la subrutina regrese al monitor si interrupciones del tipo NMI están habilitadas, si este no es el caso, la subrutina regresará al programaPPRIN

en su posición ESCAPE. Debe notarse que las interrupciones NMI son inhabilitadas por medio de la subrutina de monitor DISNMI.

Mientras TECDIS esté actuando como programa de control de las opciones del teclado (CHK=Ø), sólamente permitirá el ingreso de las teclas de control: L,G,V,N,M. La función de cada una de estas teclas se explica a continuación:

- Tecla L: Lectura de un programa desde el disco a memoria. (Transfiere el control al programa LEER).
- Tecla G: Grabación de un programa en el disco. (Programa GRABAR).
- Tecla V: Borrado de un programa en el disco. (Programa BORRAR).
- Tecla N: Nuevo disco, creación del formato en él. (Programa NUEDIS).
- Tecla M: Monitor, escape al monitor. (Programa EREST del monitor).

Para TECDIS no se darán las condiciones de entrada/salida, pues cuando se lo utiliza como subrutina se la deberá llamar desde INGRES y las condiciones de salida serán iguales que para esa subrutina.

4.3.5. LEER.-

LEER hace uso de LEAPRG para leer un programa del disco,

de producirse un error, lo indicará poniendo el código del mismo en la pantalla (el código ØØ indicará que la lectura se efectuó sin problemas). Una vez terminado el proceso, LEER transferirá el control al programa TECDIS.

4.3.6. GRABAR.-

Este programa se utiliza para la grabación de datos ó programas en el disco, de igual forma que en el programa LEER al final del proceso se pondrá el código de error en la pantalla y se transferirá el control a TECDIS.

4.3.7. BORRAR.-

Por medio de BORRAR se pueden eliminar programas del directorio del disco. Para su funcionamiento utiliza la subrutina BORPRG poniendo el código de error en la pantalla y transfiriendo el control a TECDIS una vez terminada esta subrutina.

4.3.8. NUEDIS.-

NUEDIS debe utilizarse sólamente para aquellos discos que no contengan información o en los que ésta no tenga importancia. Puesto que NUEDIS utiliza a la subrutina FORMAT para su funcionamiento, el usuario tendrá la posibilidad de "escapar" de NUEDIS mientras se estén pidiendo los números de lectora/escritora y de volúmen, aplastando para ello la tecla E.

Si se produce un error, el usuario deberá intentar nue

vamente la creación del formato y, de mantenerse el error, desechar el disco que se esté utilizando.

4.2.9. PPRIN.-

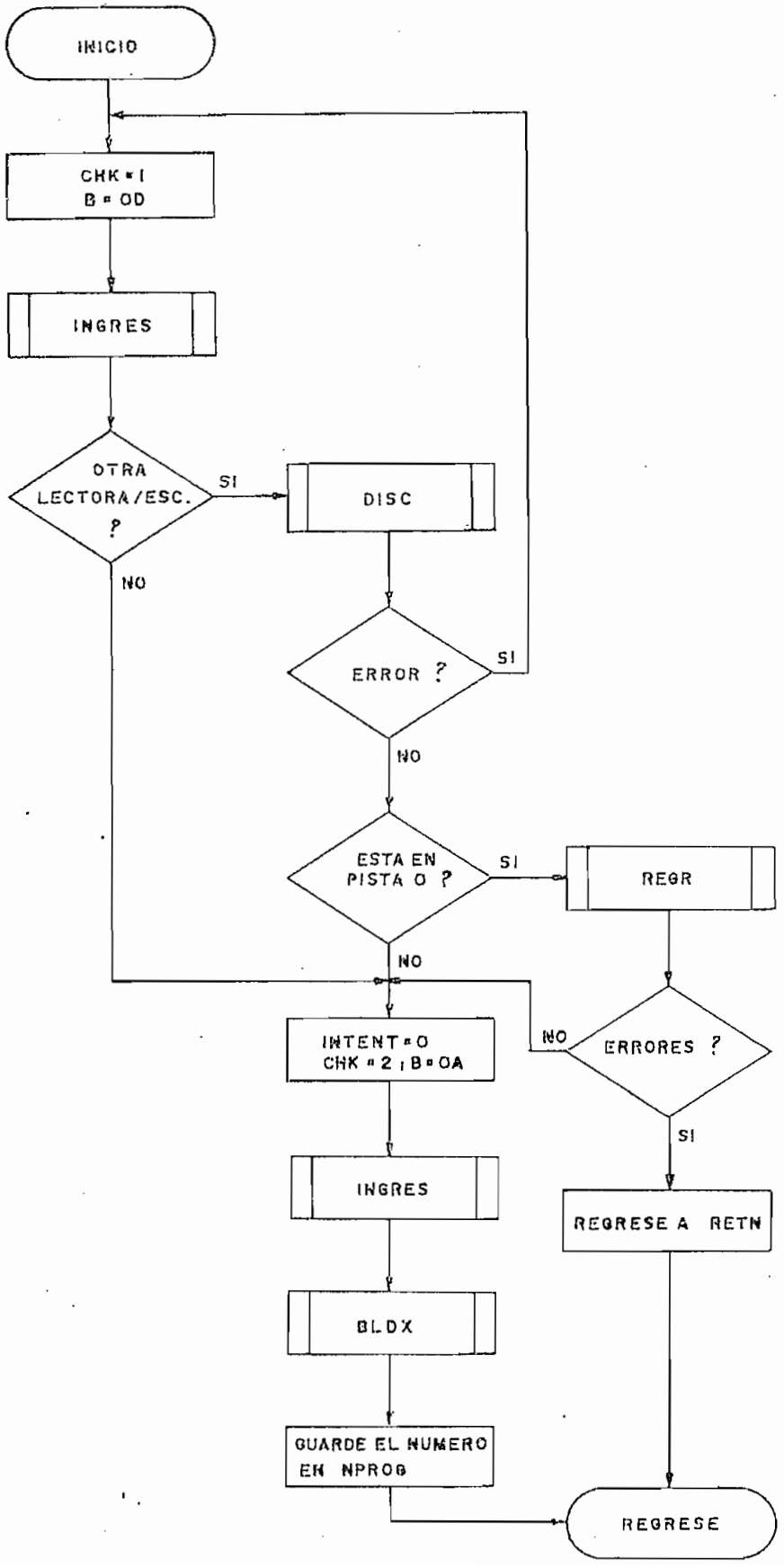
Este programa se encarga de los procesos de inicialización para el funcionamiento de la interface y el controlador de discos floppy y constituye por tanto la única entrada hacia el control y utilización de las lectoras/escritoras que se utilicen con la presente tesis.

En su posición ESCAPE se inicializa el "STACK"; se inhabilita interrupciones NMI; se borra la pantalla y se transfiere el control a TECDIS.

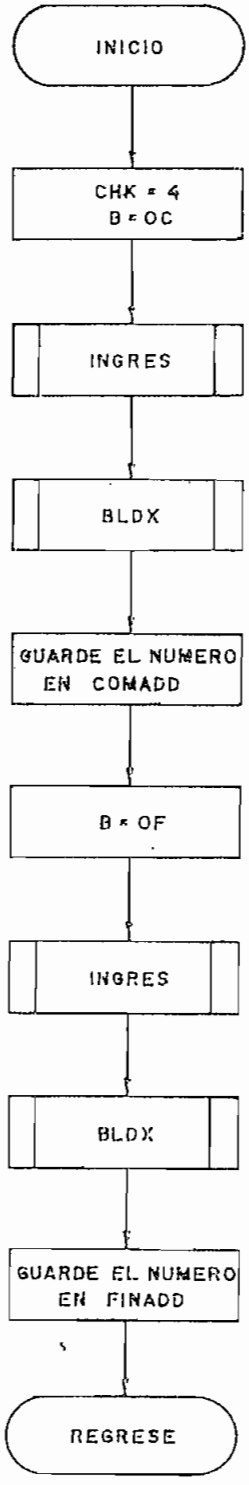
4.3b. DIAGRAMAS DE FLUJO DE LOS PROGRAMAS DE CONSOLA.

A continuación se presentan los diagramas de flujo correspondientes a los programas de consola.

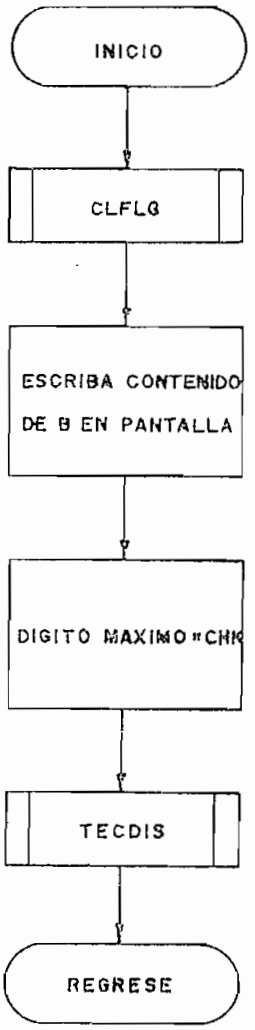
4.3b.1. PREGPD.


INICIO
CHK = I
B = OD
INGRES
OTRA LECTORA/ESC. ?
SI
DISC
NO
ERROR ?
SI
NO
ESTA EN PISTA O ?
SI
REGR
NO
INTENT = O
CHK = 2 ; B = OA
NO
ERRORES ?
SI
INGRES
REGRESE A RETN
BLDX
GUARDE EL NUMERO EN NPROG
REGRESE

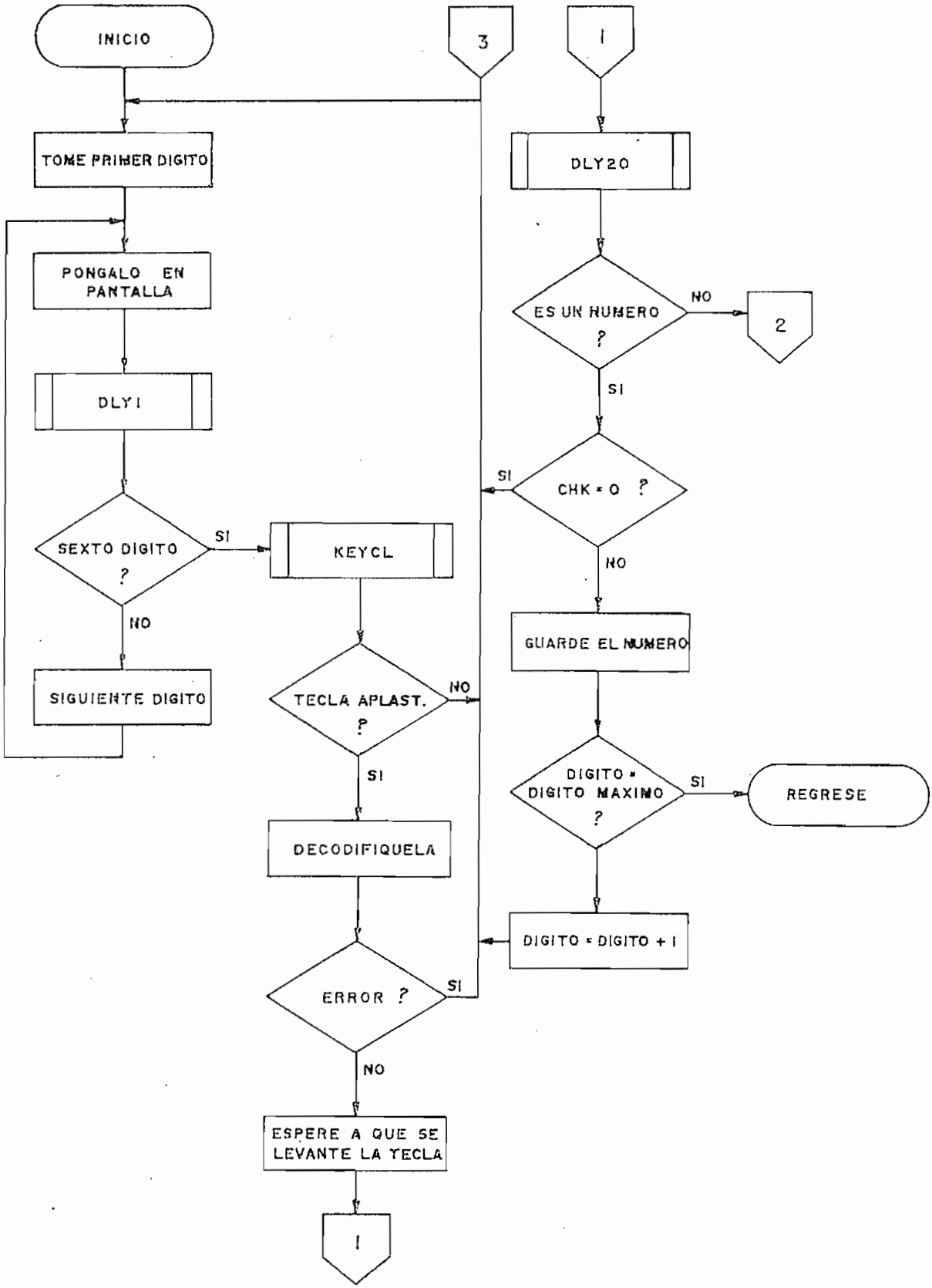
4.3b.2. PREGCF.


INICIO
CHK = 4
B = OC
INGRES
BLDX
GUARDE EL NUMERO
EN COMADD
B = OF
INGRES
BLDX
GUARDE EL NUMERO
EN FINADD
REGRESE

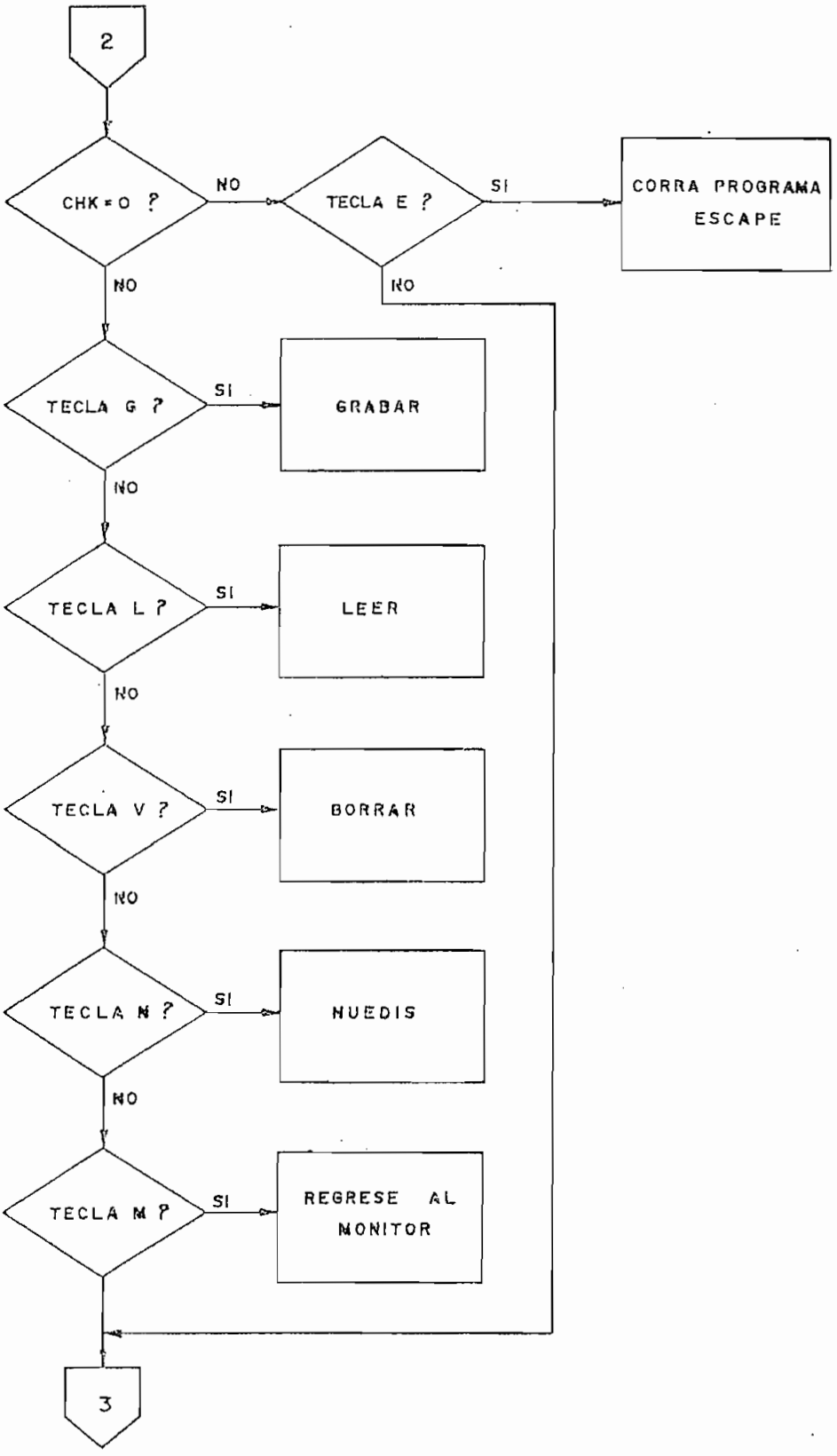
4.3b.3. INGRES.


INICIO
CLFLG
ESCRIBA CONTENIDO
DE B EN PANTALLA
DIGITO MAXIMO=CHK
TECDIS
REGRESE

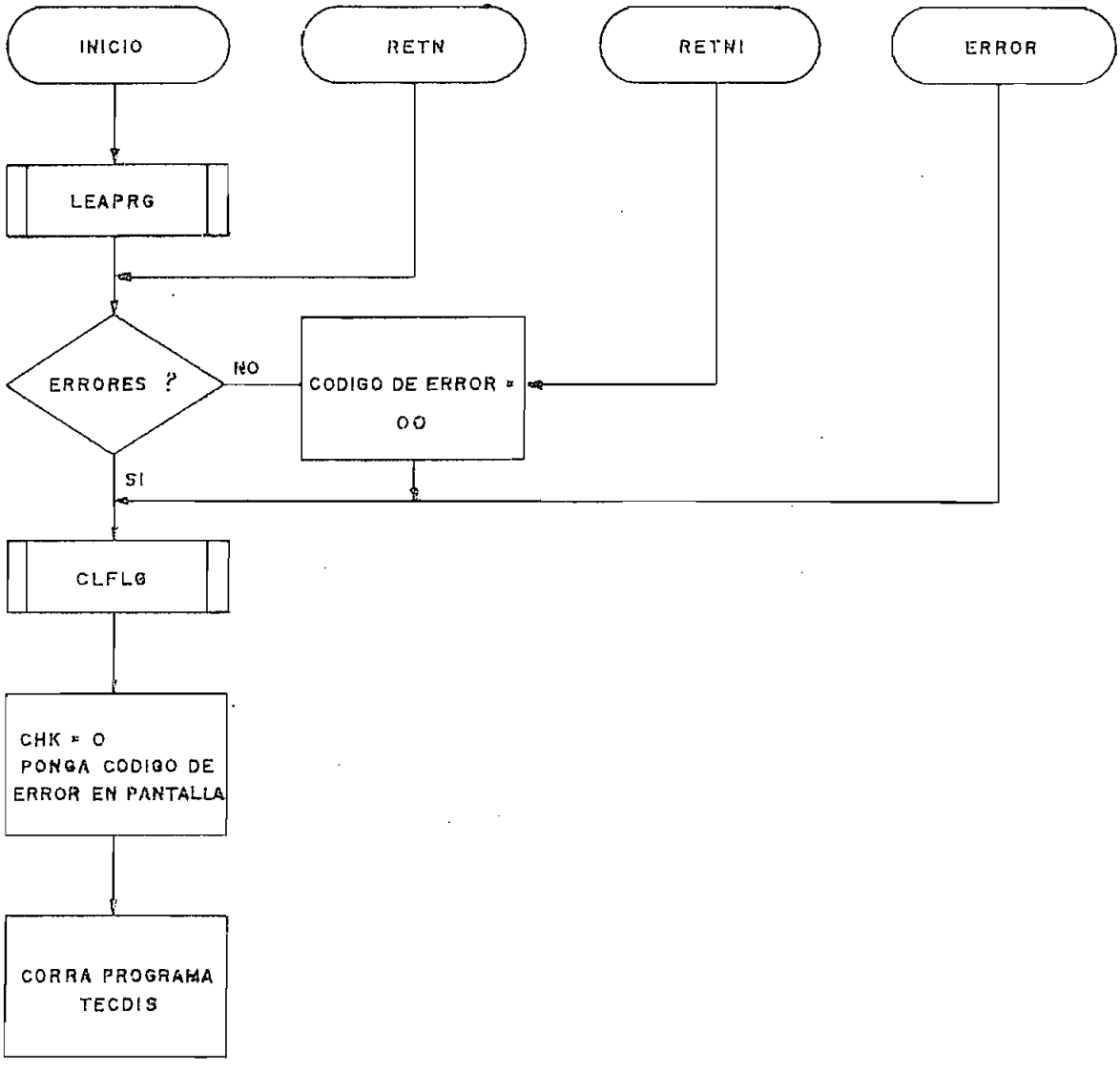
4.3b.4. TECDIS.


INICIO
3
1
TOME PRIMER DIGITO
DLY20
PONGALO EN PANTALLA
ES UN NUMERO ?
NO
2
SI
DLY1
SI
CHK = 0 ?
SEXTO DIGITO ?
SI
KEYCL
NO
NO
GUARDE EL NUMERO
SIGUIENTE DIGITO
TECLA APLAST. ?
NO
SI
DIGITO = DIGITO MAXIMO ?
SI
REGRESE
DECODIFIQUELA
DIGITO = DIGITO + 1
ERROR ?
SI
NO
ESPERE A QUE SE LEVANTE LA TECLA
1

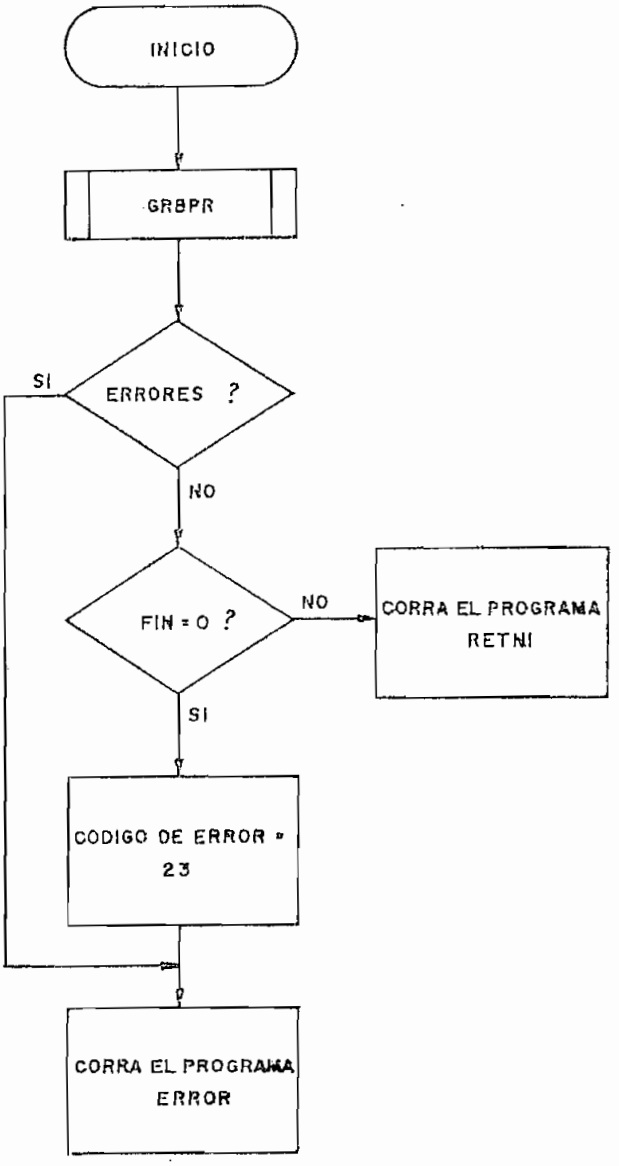
4.3b.4. TECDIS.(Cont.)



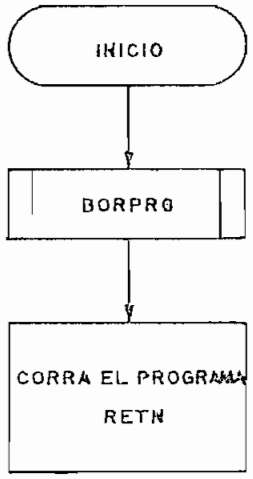
4.3b.5. LEER.


INICIO
RETN
RETNI
ERROR
LEAPRG
ERRORES ?
NO
CODIGO DE ERROR =
00
SI
CLFLG
CHK = 0
PONGA CODIGO DE
ERROR EN PANTALLA
CORRA PROGRAMA
TECDIS

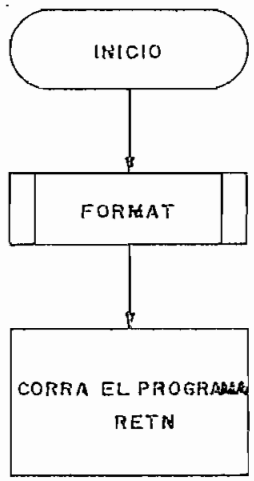
4.3b.6. GRABAR.


INICIO
GRBPR
ERRORES ?
SI
NO
FIN = 0 ?
NO
CORRA EL PROGRAMA RETNI
SI
CODIGO DE ERROR = 23
CORRA EL PROGRAMA ERROR

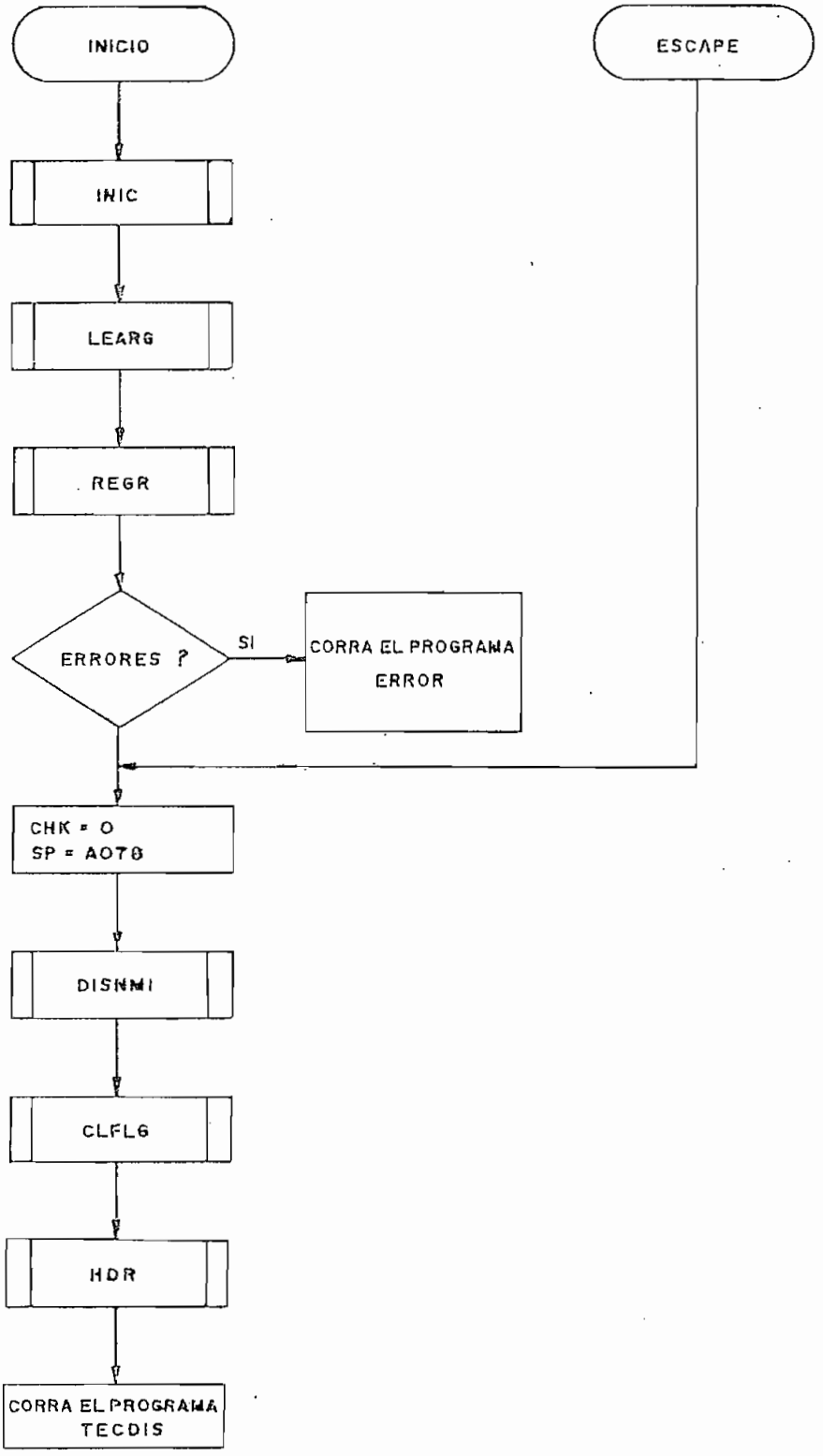
4.3b.7. BORRAR.



4.3b.8. NUEDIS.



4.3b.9. PPRIN.


INICIO
ESCAPE
INIC
LEARG
REGR
ERRORES ?
SI
CORRA EL PROGRAMA ERROR
CHK = 0
SP = A078
DISNMI
CLFLG
HDR
CORRA EL PROGRAMA TECDIS

4.4. CONJUNTO COMPLETO DE PROGRAMAS DE SOPORTE.

Antes de presentar el conjunto completo de programas de soporte -en código mnemotécnico y en lenguaje de máquina- será conveniente dar una ligera explicación de cada uno de los registros que han sido creados en memoria para el funcionamiento de los programas de soporte:

- DISACT = Número de la lectora/escritora actualmente seleccionada.
- REGPIS = Cuatro registros que contienen el número de pista en la que se encuentra cada una de las cuatro lectoras/escritoras.
- PISTA = Número de pista en el que se está trabajando.
- SECTOR = Número de sector que se utiliza.
- NPROG = Número de programas sobre el que se realiza un proceso.
- CHK = En el programa TECDIS: CHK=ØØ permite el ingreso de las teclas de control.
  CHK=n permite el ingreso de n teclas numéricas
  En LEAPRG y GRBPR: CHK=ØØ indica que el primer sector asignado a un programa se lee o escribe por primera vez.
  CHK=Ø1 En otros casos.

- FIN = Indicador de que el último byte de un programa, ha sido pasado para su grabación en el disco.
- INTENT = Contador de intentos de lectura.
- COMADD = Dirección de memoria en donde comienza un programa.
- FINADD = Dirección de memoria en donde termina un programa.
- RAMADD = Dirección de memoria a/de la cual se pasa un byte del/al programa.
- XDRBYT = Dirección de memoria que apunta a un byte del directorio.
- XDBCHK = Dirección de memoria en donde se almacenará el último dígito que ingrese desde la subrutina INGRES.
- DISBUF = Ocho registros que contienen los dígitos que se muestran en la pantalla.
- XKEYBF = Dirección de memoria en donde se almacenará el siguiente dígito que ingrese desde el teclado.
- SCNCNT = Contiene un bit en 1L que va desplazándose ala derecha para encender el siguiente dígito en la pantalla.
- BPADR = Registro de 16 bits en donde se almacenan los dos bytes que se crean a partir de los cuatro primeros dígitos almacenados en DISBUF.
- XDSBUF = Dirección de memoria de un dígito en DISBUF que se saca a la pantalla.

```
A00C                        ORG     $A00C
A00C 00          DISBUF     FCB     0,0,0,0,0,0,0,0
A01A                        ORG     $A01A
A01A 00 00       XKEYBF     FDB     0
A01C 00          SCNCNT     FCB     0
A01D 00          VFLAG      FCB     0
A01E 00 00       BPADR      FDB     0
A020 00 00       XDSBUF     FDB     0
A031                        ORG     $A031
A031             RAM        EQU     *
A031 00          DISACT     FCB     0
A032 00 00       REGPIS     FDB     0,0
A036 00          PISTA      FCB     0
A037 00          SECTOR     FCB     0
A038 00          NPROG      FCB     0
A039 00          CHK        FCB     0
A03A 00          FIN        FCB     0
A03B 00          INTENT     FCB     0
A03C 00 00       COMADD     FDB     0
A03E 00 00       FINADD     FDB     0
A040 00 00       RAMADD     FDB     0
A042 00 00       XDRBYT     FDB     0
A044 00 00       XDBCHK     FDB     0
                 *
2200                        ORG     $2200
2200 00 00       RAMSEC     FDB     0
                 *
                 * EQUATES
                 *
22FF             FINSEC     EQU     $22FF
E3CA             DIGTBL     EQU     $E3CA
E3DC             KEYTBL     EQU     $E3DC
8020             DISREG     EQU     $8020
8022             SCNREG     EQU     $8022
E0E0             DLY1       EQU     $E0E0
E0DD             DLY20      EQU     $E0DD
E12F             KEYCL      EQU     $E12F
E13A             KEYCL1     EQU     $E13A
E08D             RESTAR     EQU     $E08D
E084             DISNMI     EQU     $E084
E0B2             CLFLG      EQU     $E0B2
E0D7             HDR        EQU     $E0D7
E0E4             BLDX       EQU     $E0E4
E31C             REGSTS     EQU     $E31C
2000             MEMPIS     EQU     $2000
2C37             FINPIS     EQU     $2C37
2017             MEMSEC     EQU     $2017
                 *
00F4             WTKMND     EQU     $F4
008C             RDCMND     EQU     $8C
00AC             WTCMND     EQU     $AC
001B             SKCMND     EQU     $1B
000B             RSCMND     EQU     $0B
```

```
007B                SOCMND  EQU     $7B
005B                SICMND  EQU     $5B
009C                MSCLCT  EQU     $9C
0010                MSCBSQ  EQU     $10
00DC                MSCESC  EQU     $DC
0098                MSCVER  EQU     $98
0088                MSCRST  EQU     $88
0080                MSCRDY  EQU     $80
000C                SECMAX  EQU     $0C
000B                SECMXI  EQU     $0B
0023                PISMAX  EQU     $23
                    *
8004                DRA     EQU     $8004
8005                CRA     EQU     $8005
8006                DDRB    EQU     $8006
8007                CRB     EQU     $8007
8006                DRB     EQU     $8006
8004                DDRA    EQU     $8004
                    *
                    * JUMP TABLE (TABLA DE SALTOS)
                    *
C000                        ORG     $C000
C000 7E C0 4A       DLEA    JMP     LEA
C003 7E C0 C1       DESCR   JMP     ESCRIB
C006 7E C1 46       DVERIF  JMP     VERIF
C009 7E C1 4F       DREGR   JMP     REGR
C00C 7E C1 51       DREGR1  JMP     REGR1
C00F 7E C1 13       DDISC   JMP     DISC
C012 7E C1 58       DCHK    JMP     CHKRDY
C015 7E C0 2B       DINIC   JMP     INIC
C018 7E C0 36       DWARM   JMP     INIC3
C01B 7E C0 76       DBUSQ   JMP     BUSQ
C01E 7E C0 C5       DESCR2  JMP     ESCRI2
                    *
                    * DEMORA
                    *
                    * Demora de 10 milisegundos para BUSQ (614.4 KHz)
                    *
C021 86 02          DEMORA  LDA A   #$02        INICIALICE EL CONTADOR
C023 4A             DEMOR2  DEC A
C024 26 FD                  BNE     DEMOR2
C026 5A                     DEC B
C027 26 F8                  BNE     DEMORA      SE HA PRODUCIDO LA DEMORA?
C029 20 54                  BRA     BUSQ2       SI, REGRESE

                    *
                    * INIC
                    *
                    * INICIALICE PIA Y REGISTROS PARA DISCOS
                    *
C02B CE A0 31       INIC    LDX     $DISACT     APUNTE A LAS VARIABLES
C02E C6 05                  LDA B   #$5         NO. DE VARIABLES A BORRAR
C030 6F 00          INIC2   CLR     0,X         BORRE LA VARIABLE
C032 08                     INX                 APUNTE A LA SIGIENTE VARIABLE
```

```
C033 5A                  DEC B
C034 26 FA               BNE     INIC2      CONTINUE HASTA TERMINAR
C036 CE 80 04   INIC3    LDX     #DRA       APUNTE A LA PIA
C039 6F 01               CLR     1,X        OBTENGA ACCESO A DRA
C03B 86 FF               LDA A   #$FF       PROGRAME PA0-PA7 COMO SALIDAS
C03D A7 00               STA A   0,X
C03F 86 1E               LDA A   #$1E       OBTENGA ACCESO A DRA
C041 A7 01               STA A   1,X
C043 86 44               LDA A   #$44       APUNTE A COMREG,DISCO 0, INHABILITE
C045 A7 00               STA A   0,X        F-F DE LECTURA Y HABILITE HLT
C047 8D 6A               BSR     ENTR       PROGRAME PB0-PB7 COMO ENTRADAS
C049 39                  RTS
                *
                * LEA
                *
                * LEA UN SECTOR DEL DISCO
                *
C04A 8D 2A      LEA      BSR     BUSQ       PONGA EL DISCO EN LA PISTA CORECTA
C04C 86 8C               LDA A   #RDCMND    CARGUE UN COMANDO DE LECTURA
C04E 0F         LEA2     SEI                INHABILITE INTERRUPCIONES
C04F 8D 50               BSR     MCMD       MANDE EL COMANDO
C051 8D 4B               BSR     DTARG      APUNTE A "DATA REGISTER" DE WD
C053 CA 80               ORA B   #$80       HABILITE FLIP-FLOP DE LECTURA
C055 F7 80 04            STA B   DRA
C058 86 C0               LDA A   #$C0       MASCARA PARA IRQ O DRQ DE WD
C05A B5 80 05   LEA3     BIT A   CRA        EXISTE UN DRQ O IRQ DE WD?
C05D 27 FB               BEQ     LEA3       NO, TRATE NUEVAMENTE
C05F 2B 0B               BMI     LEA4       ES UN IRQ?
C061 F6 80 06            LDA B   DRB        ENTONCES ES UN DRQ, LEA EL DATO
C064 E7 00               STA B   0,X        GUARDE EL DATO EN MEMORIA
C066 F6 80 04            LDA B   DRA        BORRE LA BANDERA
C069 08                  INX                APUNTE AL SIGUIENTE BYTE DE MEMORIA
C06A 20 EE               BRA     LEA3       CONTINUE HASTA TERMINAR
C06C 8D 39      LEA4     BSR     CMRG       INHABILITE F-F Y APUNTE A STATUS REG
C06E 8D 04               BSR     ELEARG     LEA STATUS REG DE WD Y BORRE BANDERA
C070 C5 9C               BIT B   #MSCLCT    ENCUENTRE ERROMES DE LECTURA
C072 0E                  CLI                HABILITE INTERRUPCIONES
C073 39                  RTS
                *
                * ESTE ES UN ENLACE A LEARG
                *
C074 20 7F      ELEARG   BRA     LEARG
                *
                * BUSQ
                *
                * Busque la pista especificada
                *
C076 36         BUSQ     PSH A              GUARDE LA PISTA ESPECIFICADA
C077 17                  TBA                ACC. A CON EL SECTOR ESPECIFICADO
C078 8D 24               BSR     DTARG      APUNTE A "DATA REGISTER" DE WD
C07A 7A 80 04            DEC     DRA        *APUNTE A "SECTOR REGISTER"
C07D 8D 2F               BSR     MCMD       GUARDE EL NUMERO DE SECTOR EN WD
C07F 8D 26      BUSQ2    BSR     CMRG       APUNTE A "COMMAND REGISTER" DE WD
C081 7C 80 04            INC     DRA        *APUNTE A "TRACK REGISTER"
```

```
C084 8D 6F                BSR    LEARG      LEA SU CONTENIDO
C086 7A 80 04             DEC    DRA        *APUNTE A "COMMAND REGISTER" DE WD
C089 32                   PUL A             RECOBRE NO. DE PISTA
C08A 11                   CBA               COMPARELA CON LA DE WD
C08B 27 10                BEQ    BUSQ5      IGUALES
C08D 36                   PSH A
C08E 2E 04                BGT    BUSQ3      PISTA ESPECIFICADA ES MAYOR
C090 86 7B                LDA A  #$OCMND    CARGUE UN COMANDO "STEP OUT"
C092 20 02                BRA    BUSQ4      ENVIELO A WD
C094 86 5B        BUSQ3   LDA A  #$ICMND    CARGUE UN COMANDO "STEP IN"
C096 8D 56        BUSQ4   BSR    MCSF       MANDE EL COMANDO
C098 C4 10                AND B  #MSCBSQ    ENCUENTRE ERRORES
C09A 27 85                BEQ    DEMORA     ESPERE 10 mSEG SI NO HAY ERROR
C09C 32                   PUL A             DEJE EL 'STACK' SIN CAMBIOS
C09D 39           BUSQ5   RTS
                  *
                  * DTARG
                  *
                  * APUNTE CON PA0 Y PA1 A "DATA REGISTER" DE WD
                  *
C09E F6 80 04     DTARG   LDA B  DRA        LEA EL CONTENIDO ANTERIOR
C0A1 CA 03                ORA B  #$03       PONGA PA0 Y PA1 EN 1L
C0A3 F7 80 04     DTARG2  STA B  DRA        *ALMACENE EL RESULTADO
C0A6 39                   RTS
                  *
                  * CMRG
                  *
                  * APUNTE A "STATUS REGISTER" Y "COMMAND REGISTER DE WD
                  * E INHABILITE FLIP-FLOP DE LECTURA
                  *
C0A7 F6 80 04     CMRG    LDA B  DRA        LEA EL CONTENIDO ANTERIOR
C0AA C4 7C                AND B  #$7C       PONGA PA0,PA1,Y PA7 EN OL
C0AC 20 F5                BRA    DTARG2     ALMACENE EL RESULTADO
                  *
                  * MCMD
                  *
                  * MANDE COMANDO O DATO A WD
                  *
C0AE 8D 59        MCMD    BSR    SALID      PROGRAME LADO B DE PIA COMO SALIDAS
C0B0 B7 80 06             STA A  DRB        MANDE EL COMANDO O DATO
                  *
                  * ENTR
                  *
                  * PROGRAME PB0-PB7 COMO ENTRADAS
                  *
C0B3 C6 28        ENTR    LDA B  #$28       OBTENGA ACESO A DDRB
C0B5 F7 80 07             STA B  CRB
C0B8 7F 80 06             CLR    DDRB       PROGRAME LADO B COMO ENTRADAS
C0BB C6 2C        ENTR2   LDA B  #$2C       OBTENGA ACESO A DRB
C0BD F7 80 07             STA B  CRB
C0C0 39                   RTS
                  *
                  * ESCRIB
                  *
```

```
                        * ESCRIBA UN SECTOR
                        *
C0C1 8D E3              ESCRIB  BSR     BUSQ        BUSQUE LA PISTA ESPECIFICADA
C0C3 86 AC                      LDA A   #WTCMND     CARGUE UN COMANDO DE ESCRITURA
C0C5 0F                 ESCRI2  SEI                 INHABILITE INTERUPCIONES
C0C6 8D E6                      BSR     MCMD        MANDE EL COMANDO
C0C8 8D D4                      BSR     DTARG       APUNTE A "DATA REGISTER"
C0CA 8D 3D                      BSR     SALID       PROGRAME LADO B COMO SALIDAS
C0CC 86 C0                      LDA A   #$C0        MASCARA PARA IRQ O DRQ
C0CE E6 00                      LDA B   0,X         TENGA LISTO EL PRIMER DATO
C0D0 B5 80 05           ESCRI3  BIT A   CRA         DRQ O IRQ DE WD?
C0D3 27 FB                      BEQ     ESCRI3      NO, TRATE NUEVAMENTE
C0D5 2B 0B                      BMI     ESCRI4      IRQ?
C0D7 F7 80 06                   STA B   DRB         ES DRQ, ESCRIBA UN DATO
C0DA F6 80 04                   LDA B   DRA         BORRE LA BANDERA
C0DD 08                         INX                 APUNTE AL SIGUIENTE DATO
C0DE E6 00                      LDA B   0,X         CARGUELO
C0E0 20 EE                      BRA     ESCRI3      CONTINUE HASTA TERMINAR
C0E2 8D CF              ESCRI4  BSR     ENTR        PROGRAME LADO B COMO ENTRADAS
C0E4 8D C1                      BSR     CMRG        APUNTE A "STATUS REG"
C0E6 8D 0D                      BSR     LEARG       LEA ESE REGISTRO
C0E8 C5 DC                      BIT B   #MSCESC     ENCUENTRE ERRORES
C0EA 0E                         CLI                 HABILITE INTERUPCIONES
C0EB 39                         RTS
                        *
                        * ESTE ES UN ENLACE A MCMD
                        *
C0EC 20 C0              EMCMD   BRA     MCMD
                        *
                        * MCSP
                        *
                        * MANDE UN COMANDO Y ESPERE QUE SE LO TERMINE
                        *
C0EE 8D BE              MCSP    BSR     MCMD        MANDE EL COMANDO
C0F0 B6 80 05           MCSP2   LDA A   CRA         IRQ DE WD?
C0F3 2C FB                      BGE     MCSP2       PRUEBE NUEVAMENTE
                        *
                        * LEARG
                        *
                        * LEA UN REGISTRO DE WD
                        *
C0F5 B6 80 04           LEARG   LDA A   DRA         LEA EL CONTENIDO ANTERIOR
C0F8 84 BF                      AND A   #$BF        PONGA PA6 EN 0
C0FA B7 80 04                   STA A   DRA
C0FD F6 80 06                   LDA B   DRB         LEA EL REGISTRO
C100 8A 40                      ORA A   #$40        PONGA PA6 EN 1
C102 B7 80 04                   STA A   DRA
C105 B6 80 04                   LDA A   DRA         BORRE BANDERA IRQ SI EXISTE
C108 39                         RTS
                        *
                        * SALID
                        *
                        * PROGRAME PB0-PB7 COMO SALIDAS
                        *
```

```
C109 C6 28      SALID   LDA B   #$28        OBTENGA ACESO A DDRB
C10B F7 80 07           STA B   CRB
C10E 7A 80 06           DEC     DDRB        PROGRAME LADO B COMO SALIDAS
C111 20 A8              BRA     ENTR2       OBTENGA ACESO A DRB
                *
                * DISC
                *
                * SELECCIONE EL DISCO ESPECIFICADO
                *
C113 A6 03      DISC    LDA A   3,X         ENCUENTRE EL NO DE DISCO
C115 36                 PSH A               GUARDELO
C116 81 03              CMP A   #$3         ASEGURESE DE QUE ES MENOR QUE 4
C118 23 05              BLS     DISC2       OK
C11A C6 0F              LDA B   #$0F        PONGA CODIGO DE ERROR
C11C 32                 PUL A               DEJE EL 'STACK' SIN CAMBIOS
C11D 0D                 SEC                 PONGA BANDERA
C11E 39                 RTS
C11F 8D 48      DISC2   BSR     ENCPIS      X APUNTA A REG DE PISTA DISCO ANT.
C121 7C 80 04           INC     DRA         *APUNTE A "TRACK REG"
C124 8D CF              BSR     LEARG       LEA LA PISTA DEL DISCO ANTERIOR
C126 E7 00              STA B   0,X         GUARDE EL NUMERO
C128 84 CF              AND A   #$CF        SELECCIONE EL DISCO 0
C12A B7 80 04           STA A   DRA
C12D 32                 PUL A               RECOBRE NUMERO DE DISCO DESEADO
C12E B7 A0 31           STA A   DISACT      SELECCIONE NUEVO DISCO
C131 48                 ASL A               POSICIONE LOS BITS PARA PIA
C132 48                 ASL A
C133 48                 ASL A
C134 48                 ASL A
C135 BA 80 04           ORA A   DRA         SELECCIONE EL DISCO
C138 B7 80 04           STA A   DRA
C13B 8D 2C              BSR     ENCPIS      APUNTE X A REG DE PISTA DISCO ACTUAL
C13D A6 00              LDA A   0,X         ENCUENTRE EN QUE PISTA ESTA
C13F 8D AB              BSR     EMCMD       PONGLA EN "TRACK REG" DE WD
C141 7A 80 04           DEC     DRA         *APUNTE A "STATUS REG"
C144 5F                 CLR B               INDIQUE QUE NO HAY ERRORES
C145 39                 RTS
                *
                * VERIF
                *
                * VERIFIQUE EL ULTIMO SECTOR ESCRITO
                *
C146 86 8C      VERIF   LDA A   #RDCMND     CARGUE UN COMANDO DE LECTURA
C148 0F         VERIF2  SEI                 INHABLITE INTERRUPCIONES
C149 8D A3              BSR     MCSP        MANDE EL COMANDO
C14B C5 98              BIT B   #MSCVER     ERRORES?
C14D 0E                 CLI                 HABILITE INTERRUPCIONES
C14E 39                 RTS
                *
                * REGR
                *
                * REGRESE LA CABEZA A LA PISTA 00
                *
C14F 8D C2      REGR    BSR     DISC        CAMBIE A DISCO ESPECIFICADO
```

```
C151 86 0B      REGR1   LDA A  #RSCMND   CARGUE EL COMANDO
C153 8D 99              BSR    MCSP      MANDELO
C155 C5 88              BIT B  #MSCRST   ERRORES?
C157 39                 RTS
                *
                * CHKRDY
                *
                * CHEQUEE QUE EL DISCO ESTE LISTO
                *
C158 8D 9B      CHKRDY  BSR    LEARG     LEA "STATUS REG"
C15A C5 80              BIT B  #$80      LISTO?
C15C 27 0A              BEQ    CHKRY2
C15E 86 7B              LDA A  #SOCMND   CARGUE UN COMANDO "STEP OUT"
C160 8D 8C              BSR    MCSP      MANDELO
C162 86 58              LDA A  #SICMND   CARGUE UN COMANDO "STEP IN"
C164 8D 88              BSR    MCSP
C166 C5 80              BIT B  #MSCRDY   ERRORES?
C168 39         CHKRY2  RTS
                *
                * ENCPIS
                *
                * APUNTE X AL REGISRO DE PISTA DE DISCO SELECCIONADO
                *
C169 CE A0 32   ENCPIS  LDX    #REGPIS   APUNTE X A REGISTROS DE PISTA
C16C F6 A0 31           LDA B  DISACT    ENCUNTRE NO. DE DISCO ACTUAL
C16F 27 04      ENCPS2  BEQ    ENCPS3    X APUNTE REG DE DISCO ACTUAL
C171 08                 INX              NO, INCREMENTE X
C172 5A                 DEC B
C173 20 FA              BRA    ENCPS2    CONTINUE HASTA ENCONTRARLO
C175 39         ENCPS3  RTS
                *
                * FORMAT.
                *
                * CREACION DEL FORMATO
                *
C176 BD C3 DF   FORMAT  JSR    PREGPD    PREGUNTE No. DE DISCO Y DE VOLUMEN
C179 0F                 SEI              INHABILITE INTERRUPCIONES
C17A 7F A0 36           CLR    PISTA     COMIENCE CON LA PISTA 00
C17D CE 20 00           LDX    #MEMPIS   APUNTE AL ESPACIO DE RAM DE TRABAJO
C180 C6 FF              LDA B  #$FF      INICIALICE TODO ESTE ESPACIO CON FF
C182 E7 00      FORM1   STA B  0,X
C184 08                 INX              APUNTE AL SIGUIENTE BYTE
C185 8C 2C 37           CPX    #FINPIS   INICIALIZACION COMPLETA?
C188 26 F8              BNE    FORM1     NO
C18A CE 20 00           LDX    #MEMPIS   REGRESE
C18D 4F                 CLR A            PONGA 6 CEROS
C18E C6 06              LDA B  #$06
C190 8D 77              BSR    ALMAC
C192 86 FC              LDA A  #$FC      CARGUE LA MARCA DE PULSO INDICE
C194 A7 00              STA A  0,X       ALMACENE
C196 CE 20 17   FORM2   LDX    #MEMSEC   APUNTE AL COMIENZO DEL PRIMER SECTO
C199 86 01              LDA A  #$01      COMIENCE DESDE EL SECTOR 01
C19B B7 A0 37   FORM4   STA A  SECTOR
C19E 4F         FORM3   CLR A            ALMACENE 6 CEROS
```

```
C19F C6 06                LDA B  #$6
C1A1 8D 66                BSR    ALMAC
C1A3 86 FE                LDA A  #$FE        MARCA DE BYTES DE IDENTIFICACION
C1A5 8D 62                BSR    ALMAC       ALMACENELA Y APUNTE A SIGUIENTE BY
C1A7 B6 A0 36             LDA A  PISTA       CARGUE EL NUMERO DE PISTA ACTUAL
C1AA 8D 5D                BSR    ALMAC       ALMACENELA
C1AC 6F 00                CLR    0,X         LADO=0
C1AE 08                   INX                AFUNTE AL SIGUIENTE BYTE
C1AF B6 A0 37             LDA A  SECTOR      CARGUE EL NUMERO DEL SECTOR ACTUAL
C1B2 8D 55                BSR    ALMAC
C1B4 86 01                LDA A  #1          INDICADOR DE LONGITUD DE SECTOR
C1B6 8D 51                BSR    ALMAC
C1B8 86 F7                LDA A  #$F7        INDICADOR DE ESCRITURA DE 2 CRC'S
C1BA 8D 4D                BSR    ALMAC
C1BC 86 FF                LDA A  #$FF        ESCRIBA 11 BYTES CON FF
C1BE C6 0B                LDA B  #11
C1C0 8D 47                BSR    ALMAC
C1C2 4F                   CLR A
C1C3 C6 06                LDA B  #6          ESCRIBA 6 CEROS
C1C5 8D 42                BSR    ALMAC
C1C7 86 FB                LDA A  #$FB        CARGUE LA MARCA DE DATOS
C1C9 8D 3E                BSR    ALMAC
C1CB 86 E5                LDA A  #$E5        ESCRIBA 256 DATOS CON EL VALOR E5
C1CD 5F                   CLR B
C1CE 8D 39                BSR    ALMAC
C1D0 86 F7                LDA A  #$F7        INDICADOR DE ESCRITURA DE 2 CRC'S
C1D2 8D 35                BSR    ALMAC
C1D4 86 FF                LDA A  #$FF        ESCRIBA 14 BYTES CON FF
C1D6 C6 0E                LDA B  #14
C1D8 8D 2F                BSR    ALMAC
C1DA 7C A0 37             INC    SECTOR      CONTINUE CON EL SIGUIENTE SECTOR
C1DD 7C A0 37             INC    SECTOR
C1E0 B6 A0 37             LDA A  SECTOR      ENCUENTRE EN QUE SECTOR ESTA
C1E3 81 0C                CMP A  #SECMAX     SE HAN ESCRITO TODOS LOS SECTORES?
C1E5 27 08                BEQ    FORM5       SI, ESCRIBA UNA PISTA EN EL DISCO
C1E7 81 0B                CMP A  #SECMXI     TERMINO CON LOS SECTORES IMPARES?
C1E9 26 B3                BNE    FORM3
C1EB 86 02                LDA A  #2          COMIENCE CON LOS SECTORES PARES
C1ED 20 AC                BRA    FORM4
C1EF B6 A0 36  FORM5      LDA A  PISTA       CARGUE EL NUMERO DE PISTA ACTUAL
C1F2 BD C0 76             JSR    BUSQ        BUSQUE ESTA PISTA EN EL DISCO
C1F5 CE 20 00             LDX    #MEMPIS     APUNTE AL ESPACIO DE RAM A GRABARS
C1F8 86 F4                LDA A  #WTKMND     CARGUE COMANDO DE ESCRITURA DE PIS
C1FA BD C0 C5             JSR    ESCRI2      ESCRIBA TODA UNA PISTA
C1FD 7C A0 36             INC    PISTA       CONTINUE CON LA SIGUIENTE PISTA
C200 B6 A0 36             LDA A  PISTA       ENCUENTRE EN PSITA ESTA
C203 81 23                CMP A  #PISMAX     SE HAN ESCRITO TODAS LAS PISTAS?
C205 26 8F                BNE    FORM2       NO
C207 20 08                BRA    VRFDIS      VERIFIQUE TODOS LOS SECTORES
                 *
                 * ALMAC
                 *
                 * ALMACENE B BYTES CON EL VALOR DE A, COMENZANDO DESDE X
                 *
```

```
C209 A7 00      ALMAC   STA A  0,X        ALMACENE EL VALOR DE A
C20B 08                 INX               APUNTE AL SIGUIENTE BYTE
C20C 5A                 DEC B             B = NUMERO DE BYTES POR ALMACENAR
C20D 26 FA              BNE    ALMAC      CONTINUE HASTA TERMINAR
C20F 5C                 INC B             B=1 LISTO PARA ALMACENAR UN BYTE
C210 39                 RTS
                *
                * VRFDIS
                *
                * VERIFIQUE EL FORMATO EN EL DISCO E INICIALICE DIRECTORIO
                *
C211 7F A0 36   VRFDIS  CLR    PISTA      COMIENCE DESDE LA PISTA 00
C214 C6 01      VRFDS1  LDA B  #1         Y DESDE EL SECTOR NUMERO 1
C216 F7 A0 37   VRFDS2  STA B  SECTOR
C219 B6 A0 36           LDA A  PISTA      ENCUENTRE EN QUE PISTA SE ENCUENTRA
C21C CE 20 00           LDX    #MEMPIS    APUNTE AL ESPACIO DE RAM DE TRABAJO
C21F BD C0 00           JSR    DLEA       LEA EL SECTOR DE DISCO
C222 26 4A              BNE    VRFDS9     INTENTE LEER 3 VECES MAS
C224 F6 A0 37   VRFD11  LDA B  SECTOR     CONTINUE CON EL SIGUIENTE SECTOR
C227 5C                 INC B
C228 C1 0B              CMP B  #11        SE HAN LEIDO LOS 10 SECTORES/PISTA?
C22A 26 EA              BNE    VRFDS2     NO, CONTINUE CON EL SIGUIENTE SECTOR
C22C 7C A0 36           INC    PISTA      CONTINUE CON LA SIGUIENTE PISTA
C22F B6 A0 36           LDA A  PISTA      ENCUENTRE LA PISTA ACTUAL
C232 81 23              CMP A  #35        SE HAN LEIDO LAS 35 PISTAS?
C234 26 DE              BNE    VRFDS1     NO, ENTONCES LEA ESTA PISTA
C236 4F                 CLR A             INICIALICE EL DIRECTORIO EN PISTA 00
C237 5F                 CLR B             ALMACENE 512 BYTES CON 00 EN RAM
C238 CE 20 00           LDX    #MEMPIS
C23B 8D CC              BSR    ALMAC
C23D 5F                 CLR B
C23E 8D C9              BSR    ALMAC
C240 CE 20 00   VRFDS3  LDX    #MEMPIS    APUNTE AL DIRECTORIO EN RAM
C243 BD C0 03           JSR    DESCR      ESCRIBA 256 BYTES EN SECTOR 1
C246 26 25              BNE    VRFDS8     REGRESE SI HAY ERROR DE ESCRITURA
C248 7F A0 3B           CLR    INTENT     INICIALICE EL CONTADOR DE INTENTOS
C24B BD C0 06   VRFDS4  JSR    DVERIF     VERIFIQUE EL SECTOR ESCRITO
C24E 27 07              BEQ    VRFDS6     NO HAY ERROR
C250 BD C3 4B           JSR    INTEN      INTENTE HASTA TRES VECES
C253 26 F6              BNE    VRFDS4
C255 5D         VRFDS5  TST B             PONGA BANDERA
C256 39                 RTS
C257 4F         VRFDS6  CLR A             PISTA 0
C258 C6 02              LDA B  #2         SECTOR 2
C25A BD C0 03           JSR    DESCR      256 BYTES A PISTA 0 SECTOR 2
C25D 26 0E              BNE    VRFDS8     REGRESE SI HAY ERROR DE ESCRITURA
C25F 7F A0 3B           CLR    INTENT     INICIALICE EL CONTADOR DE INTENTOS
C262 BD C0 06   VRFDS7  JSR    DVERIF     VERIFIQUE EL SECTOR ESCRITO
C265 27 06              BEQ    VRFDS8     REGRESE SI NO HAY ERROR
C267 BD C3 4B           JSR    INTEN      INTENTE HASTA TRES VECES
C26A 26 F6              BNE    VRFDS7
C26C 5D                 TST B             PONGA BANDERA
C26D 39         VRFDS8  RTS
C26E 7F A0 3B   VRFDS9  CLR    INTENT     INICIALICE EL CONTADOR DE INTENTOS
```

```
C271 BD C0 06  VRFD10   JSR     DVERIF     VERIFIQUE EL SECTOR
C274 27 AE             BEQ     VRFD11     CONTINUE SI NO HAY ERROR
C276 BD C3 4B          JSR     INTEN      INTENTE HASTA TRES VECES
C279 26 F6             BNE     VRFD10
C27B 5D                TST B              PONGA BANDERA
C27C 39                RTS
                 *
                 * GRBPR
                 *
                 * SUBRUTINA PARA GRABAR UN PROGRAMA EN EL DISCO
                 *
C27D BD C3 DF  GRBPR   JSR     PREGPD     PREGUNTE POR No. DE PROGR. Y DISCO
C280 BD C4 12          JSR     PREGCF     PREGUNTE POR COMIENZO Y FIN DE PROG
C283 7F A0 3A  GRBPR0  CLR     FIN        INICIALICE LA VARIABLE
C286 BD C3 0C          JSR     DIRINI     LEA DIRECTORIO Y BUSQUE EL PROGRAMA
C289 25 6D             BCS     GRBPR9     ERROR DE LECTURA
C28B 26 07             BNE     GRBPR3     PROGRAMA AUN NO EXISTE
C28D 6F 00     GRBPR2  CLR     0,X        BORRE EL PROGRAMA DEL DIRECTORIO
C28F BD C3 23          JSR     ENCPR1     BUSQUE EL SIGUIENTE SECTOR ASIGNADO
C292 27 F9             BEQ     GRBPR2     CONTINUE HASTA HALLAR EL ULTIMO
C294 BD C3 3F  GRBPR3  JSR     INIVAR     INICIALICE VARIABLES
C297 4F                CLR A              ENCUENTRE UN SECTOR NO UTILIZADO
C298 BD C3 1E          JSR     ENCPRO
C29B 26 63     GRBPR4  BNE     GRBP12     DISCO LLENO
C29D B6 A0 38  GRBPR5  LDA A   NPROG      CARGUE EL NUMERO DE PROGRAMA
C2A0 A7 00             STA A   0,X        PONGALO EN EL DIRECTORIO (RAM)
C2A2 FF A0 42          STX     XDRBYT     GUARDE EL APUNTADOR DE DIRECTORIO
C2A5 CE 22 00          LDX     #RAMSEC    APUNTE AL SECTOR EN RAM
C2A8 7D A0 39          TST     CHK        PRIMER SECTOR A GRABARSE?
C2AB 26 12             BNE     GRBPR6     NO
C2AD 7C A0 39          INC     CHK
C2B0 FE A0 3C          LDX     COMADD     4 PRIMEROS BYTES DEL SECTOR INDICAN
C2B3 FF 22 00          STX     RAMSEC     EL COMIENZO Y FIN DEL PROGRAMA
C2B6 FE A0 3E          LDX     FINADD
C2B9 FF 22 02          STX     RAMSEC+2
C2BC CE 22 04          LDX     #RAMSEC+4  INDIQUE QUE YA HAY 4 BYTES EN RAM
C2BF FF A0 40  GRBPR6  STX     RAMADD
C2C2 FE A0 3C          LDX     COMADD     APUNTE A UN BYTE DE PROGRAMA
C2C5 A6 00             LDA A   0,X        CARGUELO
C2C7 BC A0 3E          CPX     FINADD     ULTIMO BYTE?
C2CA 26 03             BNE     GRBPR7     NO
C2CC 7C A0 3A          INC     FIN
C2CF 08        GRBPR7  INX                APUNTE AL SIGUIENTE BYTE DE PROGRAMA
C2D0 FF A0 3C          STX     COMADD     GUARDE EL APUNTADOR
C2D3 FE A0 40          LDX     RAMADD     APUNTE A UN BYTE DE RAM
C2D6 A7 00             STA A   0,X        ALMACENE ALLI EL VALOR LEIDO
C2D8 08                INX                SIGUIENTE BYTE DE RAM
C2D9 8C 23 00          CPX     #FINSEC+1  FIN DE SECTOR EN RAM?
C2DC 26 E1             BNE     GRBPR6     NO, CONTINUE HASTA TERMINAR
C2DE CE 22 00          LDX     #RAMSEC    INICIALICE VARIABLES PARA GRABACION
C2E1 B6 A0 36          LDA A   PISTA
C2E4 F6 A0 37          LDA B   SECTOR
C2E7 BD C0 03          JSR     DESCR      ESCRIBA UN SECTOR EN EL DISCO
C2EA 26 0D             BNE     GRBP10     ERROR DE ESCRITURA
```

```
C2EC 7F A0 3B          CLR     INTENT    INICIALICE CONTADOR DE INTENTOS
C2EF BD C0 06  GRBPR8  JSR     DVERIF    VERIFIQUE EL ULTIMO SECTOR ESCRITO
C2F2 27 06             BEQ     GRBP11    NO HAY ERROR
C2F4 8D 55             BSR     INTEN     INTENTE HASTA TRES VECES
C2F6 26 F7             BNE     GRBPR8
C2F8 5D        GRBPR9  TST B             PONGA BANDERA
C2F9 39        GRBP10  RTS
C2FA 4F        GRBP11  CLR A
C2FB 7D A0 3A          TST     FIN       SE HA GRABADO TODO EL PROGRAMA?
C2FE 27 05             BEQ     GRBP13    NO, CONTIUE HASTA TERMINAR
C300 C6 01     GRBP12  LDA B   #1        ESCRIBA EL DIRECTORIO EN EL DISCO
C302 7E C2 40          JMP     VRFIS3
C305 FE A0 42  GRBP13  LDX     XURBYT    RECOBRE EL APUNTADOR DE DIRECTORIO
C308 8D 19             BSR     ENCPR1    ENCUENTRE UN NUEVO SECTOR VACIO
C30A 20 8F             BRA     GRBPR4    REPITA EL PROCESO
               *
               * DIRINI
               *
               * LEA EL DIRECTORIO, INICIALICE VARIABLES Y VAYA A ENCPR
               *
C30C 8D 46     DIRINI  BSR     LEADIR    LEA EL DIRECTORIO
C30E 27 06             BEQ     DIRIN2    NO HAY ERROR
C310 8D 39             BSR     INTEN     INTENTE HASTA TRES VECES
C312 26 F8             BNE     DIRINI
C314 0D                SEC               PONGA BANDERA
C315 39                RTS
C316 8D 27     DIRIN2  BSR     INIVAR    INICIALICE VARIABLES
C318 7F A0 39          CLR     CHK
C31B B6 A0 38          LDA A   NPROG     CARGUE EL NUMERO DE PROGRAMA
               *
               * ENCPRG
               *
               * ENCUENTRE NUMEROS DE PISTA Y SECTOR ASIGNADOS AL PROGRAMA
               *
C31E A1 00     ENCPRG  CMP A   0,X       ES No. DE PROGRAMA IGUAL A DIR(X)?
C320 26 01             BNE     ENCPR1    NO, CONTINUE HASTA ENCONTRARLO
C322 39                RTS
C323 7C A0 37  ENCPR1  INC     SECTOR    APUNTE AL SIGUIENTE SECTOR
C326 08                INX               SIGUIENTE BYTE DE DIRECTORIO
C327 F6 A0 37          LDA B   SECTOR    CARGUE EL NUMERO DEL SECTOR
C32A C1 0B             CMP B   #$0B      ES MAYOR QUE 10?
C32C 26 F0             BNE     ENCPRG    NO, VEA SI ESTA ASIGNADO
C32E C6 01             LDA B   #1        REGRESE AL SECTOR No. 1
C330 F7 A0 37          STA B   SECTOR
C333 7C A0 36          INC     PISTA     APUNTE A LA SIGUIENTE PISTA
C336 F6 A0 36          LDA B   PISTA     CARGUE EL NUMERO DE PISTA
C339 C1 23             CMP B   #35       ES MAYOR A 34?
C33B 26 E1             BNE     ENCPRG    NO
C33D 5D                TST B             PONGA BANDERA
C33E 39                RTS
               *
               * INIVAR
               *
               * INICIALICE VARIABLES
```

```
                    *
C33F 86 01          INIVAR  LDA A  #1
C341 B7 A0 37               STA A  SECTOR     COMIENCE CON EL SECTOR #1
C344 B7 A0 36               STA A  PISTA      COMIENCE CON LA PISTA #1
C347 CE 20 0A               LDX    #MEMPIS+10 APUNTE A BYTE 11 DE DIRECTORIO
C34A 39                     RTS
                    *
                    * INTEN
                    *
                    * SUBRUTINA PARA PERMITIR HASTA 3 INTENTOS
                    *
C34B 7C A0 3B       INTEN   INC    INTENT     INCREMENTE EL CONTADOR
C34E B6 A0 3B               LDA A  INTENT     CARGUE EL NUMERO DE INTENTOS
C351 81 03                  CMP A  #3         COMPARELO CON 3
C353 39                     RTS
                    *
                    * LEADIR
                    *
                    * LEA EL DIRECTORIO DEL DISCO
                    *
C354 CE 20 00       LEADIR  LDX    $MEMPIS    COMIENZO DEL ESPACIO DE RAM
C357 4F                     CLR A             PISTA 0
C358 C6 01                  LDA B  #1         SECTOR #1
C35A BD C0 00               JSR    DLEA       LEA ESTE SECTOR A RAM
C35D 26 06                  BNE    LEADR1     ERROR DE LECTURA
C35F 4F                     CLR A             PISTA 0
C360 C6 02                  LDA B  #2         SECTOR No. 2
C362 BD C0 00               JSR    DLEA       LEA ESTE SECTOR A CONTINUACION
C365 39             LEADR1  RTS
                    *
                    * BORPRG
                    *
                    * BORRE UN PROGRAMA DEL DISCO (REALMENTE SOLO DEL DIRECTORIO
                    *
C366 8D 77          BORPRG  BSR    PREGPD     PREGUNTE POR No. DE DISCO Y PROGRAM
C368 8D A2          BORPRO  BSR    DIRINI     LEA DIRECTORIO Y BUSQUE EL PROGRAMA
C36A 25 8C                  BCS    GRBPR9     ERROR DE LECTURA
C36C 26 09                  BNE    BORPR2     PROGRAMA NO EXISTE AUN
C36E 6F 00          BORPR1  CLR    0,X        BORRE EL BYTE DEL DIRECTORIO
C370 8D B1                  BSR    ENCPR1     ENCUENTRE EL SIGUIENTE BYTE A BORRA
C372 27 FA                  BEQ    BORPR1     CONTINUE HASTA TERMINAR
C374 4F                     CLR A             GRABE EL NUEVO DIRECTORIO
C375 20 89                  BRA    GRBP12
C377 39             BORPR2  RTS
                    *
C378 7E C2 F8       EGRBP9  JMP    GRBPR9
                    *
                    * LEAPRG
                    *
                    * SUBRUTINA PARA LEER UN PROGRAMA DEL DISCO
                    *
C37B 8D 62          LEAPRG  BSR    PREGPD     PREGUNTE POR No. DE DISCO Y PROGRAM
C37D 8D 8D          LEAPRO  BSR    DIRINI     LEA DIRECTORIO Y BUSQUE EL PROGRAMA
C37F 25 F7                  BCS    EGRBP9     ERROR DE LECTURA
```

```
C381 26 5B      LEAPR1  BNE     LEAPR5      PROGRAMA NO EXISTE O FUE MAL GRABADO
C383 7F A0 38           CLR     INTENT      INICIALICE EL CONTADOR DE INTENTOS
C386 FF A0 42           STX     XIRBYT      GUARDE EL APUNTADOR DE DIRECTORIO
C389 CE 22 00   LEAPR2  LDX     #RAMSEC     APUNTE AL SECTOR EN RAM
C38C B6 A0 36           LDA A   PISTA       LEA EL SECTOR DEL DISCO
C38F F6 A0 37           LDA B   SECTOR
C392 BD C0 00           JSR     DLEA
C395 27 06              BEQ     LEAPR3      NO HAY ERROR DE LECTURA
C397 8D B2              BSR     INTEN       INTENTE HASTA 3 VECES
C399 26 EE              BNE     LEAPR2
C39B 5D                 TST B
C39C 39                 RTS
C39D CE 22 00   LEAPR3  LDX     #RAMSEC     APUNTE AL SECTOR EN RAM
C3A0 7D A0 39           TST     CHK         PRIMER SECTOR LEIDO?
C3A3 26 12              BNE     LEAPR4      NO
C3A5 7C A0 39           INC     CHK
C3A8 FE 22 00           LDX     RAMSEC      4 PRIMEROS BYTES DEL SECTOR INDICAN
C3AB FF A0 3C           STX     COMADD      EL COMIENZO Y FIN DEL PROGRAMA
C3AE FE 22 02           LDX     RAMSEC+2
C3B1 FF A0 3E           STX     FINADD
C3B4 CE 22 04           LDX     #RAMSEC+4   SE HAN LEIDO YA 4 BYTES DE RAM
C3B7 FF A0 40   LEAPR4  STX     RAMADD
C3BA A6 00              LDA A   0,X         CARGUE UN BYTE DE RAM
C3BC FE A0 3C           LDX     COMADD      APUNTE AL BYTE DE PROGRAMA
C3BF A7 00              STA A   0,X         ALMACENE ALLI EL BYTE CARGADO
C3C1 BC A0 3E           CPX     FINADD      ULTIMO BYTE?
C3C4 27 18              BEQ     LEAPR5      SI
C3C6 08                 INX                 APUNTE AL SIGUIENTE BYTE DE PROGRAMA
C3C7 FF A0 3C           STX     COMADD      GUARDE EL APUNTADOR
C3CA FE A0 40           LDX     RAMADD      CARGUE EL APUNTADOR DE RAM
C3CD 08                 INX                 APUNTE AL SIGUIENTE BYTE
C3CE 8C 23 00           CPX     #FINSEC+1   FINAL DEL SECTOR EN RAM?
C3D1 26 E4              BNE     LEAPR4      NO, CONTINUE HASTA TERMINAR
C3D3 B6 A0 38           LDA A   NPROG       RECOBRE EL No. DE PROGRAMA
C3D6 FE A0 42           LDX     XIRBYT      RECOBRE EL APUNTADOR DE DIRECTORIO
C3D9 BD C3 23           JSR     ENCPR1      ENCUENTRE OTRO SECTOR CON PROGRAMA
C3DC 20 A3              BRA     LEAPR1      REPITA EL PROCESO
C3DE 39         LEAPR5  RTS
                *
                * PREGPD
                *
                * PREGUNTE POR NUMERO DE PROGRAMA Y DISCO
                *
C3DF 86 01      PREGPD  LDA A   #1          PERMITA EL INGRESO DE UN DATO
C3E1 C6 0D              LDA B   #$0D        PONGA "D" EN LA PANTALLA
C3E3 8D 46              BSR     INGRES      SUBRUTINA DE INGRESO DE DATOS
C3E5 F1 A0 31           CMP B   DISACT      ES # DE DISCO IGUAL AL ACTUAL ?
C3E8 27 15              BEQ     PREGP1      SI
C3EA CE A0 09           LDX     #DISBUF-3   APUNTE AL DATO LEIDO PARA DISC
C3ED BD C0 0F           JSR     DDISC       SELECCIONE EL NUEVO DISCO
C3F0 26 ED              BNE     PREGPD      DISCO SELECCIONADO MAYOR QUE 3
C3F2 4D                 TST A               SE ENCUENTRA EN LA PISTA 00 ?
C3F3 26 0A              BNE     PREGP1      NO, ENTONCES YA FUE UTILIZADO
C3F5 BD C0 0C           JSR     DREGR1      REGRESE A LA PISTA 00
```

```
C3F8 27 05                BEQ     PREGP1     CONTINUE SI NO HAY ERROR
C3FA 31                   INS                REGRESE AL PROGRAMA RETN
C3FB 31                   INS
C3FC 31                   INS
C3FD 31                   INS
C3FE 39                   RTS
C3FF 86 02        PREGP1  LDA A   #2         PERMITA EL INGRESO DE DOS DATOS
C401 7F A0 38             CLR     INTENT     INICIALICE LA VARIABLE
C404 C6 0A                LDA B   #$0A       PONGA "A" EN LA PANTALLA
C406 8D 23                BSR     INGRES     SUBRUTINA DE INGRESO DE DATOS
C408 BD E0 E4             JSR     BLDX       ARME UN NUMERO DE 8 BITS DE DATOS
C40B B6 A0 1E             LDA A   BFADR      CARGUELO
C40E B7 A0 38             STA A   NPROG      ALMACENELO EN NUMERO DE PROGRAMA
C411 39                   RTS
                  *
                  * PREGCF
                  *
                  * PREGUNTE POR COMIENZO Y FIN DE PROGRAMA
                  *
C412 86 04        PREGCF  LDA A   #4         PERMITA EL INGRESO DE CUATRO DATOS
C414 C6 0C                LDA B   #$0C       PONGA "C" EN LA PANTALLA
C416 8D 13                BSR     INGRES     SUBRUTINA DE INGRESO DE DATOS
C418 BD E0 E4             JSR     BLDX       ARME UN NUMERO DE 16 BITS DE DATOS
C41B FF A0 3C             STX     COMADD     GUARDELO COMO COMIENZO DE PROGRAMA
C41E 86 04                LDA A   #4         PERMITA EL INGRESO DE CUATRO DATOS
C420 C6 0F                LDA B   #$0F       PONGA "F" EN LA PANTALLA
C422 8D 07                BSR     INGRES     SUBRUTINA DE INGRESO DE DATOS
C424 BD E0 E4             JSR     BLDX       ARME UN NUMERO DE 16 BITS DE DATOS
C427 FF A0 3E             STX     FINADD     GUARDELO COMO FINAL DE PROGRAMA
C42A 39                   RTS
                  *
                  * INGRES
                  *
                  * SUBRUTINA DE INGRESO DE DATOS
                  * ESCRIBE TAMBIEN UN DATO EN PANTALLA
                  *
C42B B7 A0 39     INGRES  STA A   CHK        INDIQUE QUE SE VAN A INGRESAR DATOS
C42E BD E0 B2             JSR     CLFLG      BORRE LA PANTALLA
C431 CE A0 0B             LDX     #DISBUF-1  APUNTE AL REGISTRO DE PANTALLA -1
C434 E7 01                STA B   1,X        ESCRIBA LA LETRA EN PANTALLA
C436 F6 A0 39             LDA B   CHK        CALCULE LA POSICION DE DIGITO MAXIMO
C439 08           INGRE1  INX
C43A 5A                   DEC B
C43B 26 FC                BNE     INGRE1     CONTINUE HASTA TERMINAR
C43D FF A0 44             STX     XDBCHK     ALMACENE EL RESULTADO
                  *
                  * TECDIS
                  *
                  * SUBRUTINA DE MANEJO DE TECLADO Y DISPLAY
                  *
C440 CE A0 0C     TECDIS  LDX     #DISBUF    APUNTE AL REGISTRO DE PANTALLA
C443 A6 00        TECDS1  LDA A   0,X        TOME EL PRIMER DIGITO
C445 4C                   INC A              ALISTELO PARA DECODIFICARLO
C446 08                   INX                APUNTE AL SIGUIENTE DIGITO
```

```
C447 FF A0 20          STX    XDSBUF      GUARDE EL APUNTADOR
C44A CE E3 C9          LDX    #DIGTBL-1   APUNTE A LA TABLA DE DECODIFICACION
C44D 08        TECDS2  INX
C44E 4A                DEC A              ENCUENTRE EL CODIGO PARA PANTALLA
C44F 26 FC             BNE    TECDS2      CONTINUE HASTA ENCONTRARLO
C451 7F 80 22          CLR    SCNREG      BORRE LA PANTALLA
C454 A6 00             LDA A  0,X         CARGUE EL CODIGO
C456 B7 80 20          STA A  DISREG      MANDELO A LA PANTALLA
C459 B6 A0 1C          LDA A  SCNCNT      CARGUE EL NUMERO DE DIGITO
C45C B7 80 22          STA A  SCNREG      SELECCIONE EL DIGITO
C45F CE 00 4D          LDX    #$4D        PREPARESE PARA ESPERAR 1 MS.
C462 BD E0 E0          JSR    DLY1        ESPERE 1 MS.
C465 FE A0 20          LDX    XDSBUF      RECOBRE EL APUNTADOR
C468 8C A0 12          CPX    #DISBUF+6   ULTIMO DIGITO ?
C46B 27 05             BEQ    TECDS3      SI, VAYA AL TECLADO
C46D 74 A0 1C          LSR    SCNCNT      SELECCIONE EL SIGUIENTE DIGITO
C470 20 D1             BRA    TECDS1      CONTINUE HASTA EL ULTIMO DIGITO
C472 86 20     TECDS3  LDA A  #$20        INICIALICE EL APUNTADOR DE DIGITO
C474 B7 A0 1C          STA A  SCNCNT
C477 BD E1 2F  TECDS4  JSR    KEYCL       BUSQUE UNA TECLA APLASTADA
C47A 27 C4             BEQ    TECDIS      NO SE HA APLASTADO UNA TECLA
C47C BD E0 DD          JSR    DLY20       ESPERE 20 MS.
C47F CE 80 20          LDX    #DISREG     APUNTE AL REGISTRO DE PANTALLA/TECL
C482 86 01             LDA A  #$01        PRIMERA FILA
C484 A7 02             STA A  2,X
C486 BD E1 3A  TECDS5  JSR    KEYCL1      BUSQUE LA TECLA APLASTADA
C489 26 0A             BNE    TECDS6      TECLA ENCONTRADA
C48B A6 02             LDA A  2,X         BORRE BANDERA NM1
C48D 81 20             CMP A  #$20        ULTIMA FILA ?
C48F 27 AF             BEQ    TECDIS      SI, VAYA A LA PANTALLA
C491 68 02             ASL    2,X         SIGUIENTE FILA
C493 20 F1             BRA    TECDS5      CONTINUE HASTA TERMINAR
C495 5F        TECDS6  CLR B              INICIALICE EL CONTADOR
C496 CE E3 DC          LDX    #KEYTBL     APUNTE A LA TABLA DE TECLAS
C499 A1 00     TECDS7  CMP A  0,X         ENCONTRO LA TECLA EN LA TABLA ?
C49B 27 09             BEQ    TECDS8      SI, VEA SI ES UN NUMERO O NO
C49D 8C E3 F4          CPX    #KEYTBL+24  FIN DE LA TABLA ?
C4A0 27 9E             BEQ    TECDIS      SI, VAYA A LA PANTALLA
C4A2 08                INX                AVANCE EN LA PANTALLA
C4A3 5C                INC B
C4A4 20 F3             BRA    TECDS7      CONTINUE HASTA TERMINAR
C4A6 BD E1 2F  TECDS8  JSR    KEYCL       SE HA LEVANTADO LA TECLA ?
C4A9 26 FB             BNE    TECDS8      NO
C4AB BD E0 DD          JSR    DLY20       ESPERE 20 MS.
C4AE C1 0F             CMP B  #$0F        ES UN NUMERO HEXADECIMAL ?
C4B0 2E 17             BGT    TECD11      NO, DECODIFIQUE EL COMANDO
C4B2 7D A0 39          TST    CHK         SE ESTA PIDIENDO UN NUMERO ?
C4B5 27 89             BEQ    TECDIS      NO, REGRESE A LA PANTALLA
C4B7 FE A0 1A          LDX    XKEYBF      RECOBRE EL APUNTADOR DE TECLADO
C4BA E7 00             STA B  0,X         GUARDE EL NUMERO
C4BC BC A0 44          CPX    XDECHK      ESTAN YA TODOS LOS DIGITOS PEDIDOS
C4BF 27 07             BEQ    TECD10      SI, REGRESE
C4C1 08                INX                APUNTE AL SIGUIENTE DIGITO
C4C2 FF A0 1A          STX    XKEYBF      GUARDE EL APUNTADOR
```

```
C4C5 7E C4 40  TECD69  JMP     TECD15     ESPERE AL SIGUIENTE DIGITO
C4C8 39        TECD10  RTS
C4C9 7D A0 39  TECD11  TST     CHK        SE ESTA PIDIENDO UN NUMERO ?
C4CC 27 06             BEQ     TECD12     NO, ENTONCES ES OPCION DE TECLADO
C4CE C1 15             CMP B   #$15       ES LA TECLA E ?
C4D0 27 2A             BEQ     ESCAPE     SI, ESCAPE DE LA OPCION DE TECLADO
C4D2 20 F1             BRA     TECD69     REGRESE A LA PANTALLA
C4D4 CE C4 BE  TECD12  LDX     #TECD15-32 ENCUENTRE LA OPCION EN LA TABLA
C4D7 08        TECD14  INX
C4D8 08                INX
C4D9 5A                DEC B
C4DA 26 FB             BNE     TECD14
C4DC 6E 00             JMP     0,X        VAYA A LA OPCION SELECCIONADA
C4DE 20 E5     TECD15  BRA     TECD69     TECLA P
C4E0 20 2B             BRA     LEER       TECLA L
C4E2 20 4E             BRA     NUEDIS     TECLA N
C4E4 20 47             BRA     BORRAR     TECLA V
C4E6 20 06             BRA     EREST      TECLA M
C4E8 20 DB             BRA     TECD69     TECLA E
C4EA 20 D9             BRA     TECD69     TECLA R
C4EC 20 31             BRA     GRABAR     TECLA G
               *
C4EE 7E E0 8D  EREST   JMP     RESTAR     ESTE ES UN ENLACE A RESTAR
               *
               * PPRIN
               *
               * PROGRAMA PRINCIPAL PARA EL CONTROL DE DISCOS FLOPPY
               *
C4F1 BD C0 15  PPRIN   JSR     DINIC      INICIALICE PIA Y REGISTROS DE DISC
C4F4 BD C0 F5          JSR     LEARG      BORRE BANDERA IRQ SI EXISTE
C4F7 BD C0 0C          JSR     DREGR1     PONGA LA CABEZA EN LA PISTA 00
C4FA 26 17             BNE     ERROR      INDIQUE EL ERROR
C4FC 7F A0 39  ESCAPE  CLR     CHK        COMIENCE CON OPCIONES DE TECLADO
C4FF 8E A0 78          LDS     #$A078     INICIALICE EL STACK
C502 BD E0 84          JSR     DISNMI     INHABILITE INTERRUPCIONES NMI
C505 BD E0 B2          JSR     CLFLG      BORRE LA PANTALLA
C508 BD E0 D7          JSR     HDR        ESCRIBA "-" EN LA PANTALLA
C50B 20 B8             BRA     TECD69     CONTROLE EL TECLADO Y PANTALLA
               *
               * LEER
               *
               * PROGRAMA PARA LEER DATOS O PROGRAMAS DEL DISCO
               *
C50D BD C3 7B  LEER    JSR     LEAPRG     VAYA A LA SUBRUTINA DE LECTURA
C510 26 01     RETN    BNE     ERROR      INDIQUE EL ERROR
C512 5F        RETN1   CLR B              INDIQUE QUE NO HAY ERROR
C513 BD E0 B2  ERROR   JSR     CLFLG      BORRE LA PANTALLA
C516 7F A0 39          CLR     CHK        PREPARECE PARA OPCIONES DE TECLADO
C519 17                TBA                CODIGO (DE ERROR O NO) EN ACC. A
C51A BD E3 1C          JSR     REGST5     PONGA EL CODIGO EN PANTALLA
C51D 20 A6             BRA     TECD69     CONTROLE EL TECLADO Y PANTALLA
               *
               * GRABAR
               *
```

```
                    * PROGRAMA PARA GRABAR DATOS O PROGRAMAS EN EL DISCO
                    *
C51F BD C2 7D       GRABAR  JSR     GRBPR     VAYA A LA SUBRUTINA DE GRABACION
C522 26 EF                  BNE     ERROR     INDIQUE EL ERROR
C524 C6 23                  LDA B   #$23      CODIGO DE DISCO LLENO
C526 7D A0 3A               TST     FIN       ESTA EL DISCO LLENO ?
C529 27 E8                  BEQ     ERROR     SI, INDIQUE EL ERROR
C52B 20 E5                  BRA     RETN1     INDIQUE QUE NO HAY ERROR
                    *
                    * BORRAR
                    *
                    * PROGRAMA PARA BORRAR UN PROGRAMA DEL DISCO
                    *
C52D BD C3 66       BORRAR  JSR     BORPRG    VAYA A LA SUBRUTINA DE BORRADO
C530 20 DE                  BRA     RETN      CHEQUEE SI HAY ERRORES E INDIQUELO
                    *
                    * NUEDIS
                    *
                    * PROGRAMA PARA DISCOS NUEVOS, CREA EL FORMATO EN ELLOS
                    *
C532 BD C1 76       NUEDIS  JSR     FORMAT    VAYA A LA SUBRUTINA DE FORMATO
C535 20 D9                  BRA     RETN      CHEQUEE SI HAY ERRORES E INDIQUELO
                    *
                            END
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

ALMAC	C209	BLDX	E0E4	BORPR0	C368	BORPR1	C36E	BORPR2	C377
BORPRG	C366	BORRAR	C52D	BPADR	A01E	BUSQ	C076	BUSQ2	C07F
BUSQ3	C094	BUSQ4	C096	BUSQ5	C09D	CHK	A039	CHKRDY	C158
CHKRY2	C168	CLFLG	E0B2	CMRG	C0A7	COMADD	A03C	CRA	8005
CRB	8007	DBUSG	C01B	DCHK	C012	DDISC	C00F	DDRA	8004
DDRB	8006	DEMOR2	C023	DEMORA	C021	DESCR	C003	DESCR2	C01E
DIGTBL	E3CA	DINIC	C015	DIRIN2	C316	DIRINI	C30C	DISACT	A031
DISBUF	A00C	DISC	C113	DISC2	C11F	DISNMI	E084	DISREG	8020
DLEA	C000	DLY1	E0E0	DLY20	E0DD	DRA	8004	DRB	8006
DREGR	C009	DREGR1	C00C	DTARG	C09E	DTARG2	C0A3	DVERIF	C006
DWARM	C018	EGRBP9	C378	ELEARG	C074	EMCMD	C0EC	ENCPIS	C169
ENCPR1	C323	ENCPRG	C31E	ENCPS2	C16F	ENCPS3	C175	ENTR	C0B3
ENTR2	C0BB	EREST	C4EE	ERROR	C513	ESCAPE	C4FC	ESCR12	C0C5
ESCRI3	C0D0	ESCRI4	C0E2	ESCRIB	C0C1	FIN	A03A	FINADD	A03E
FINPIS	2C37	FINSEC	22FF	FORM1	C182	FORM2	C196	FORM3	C19E
FORM4	C19B	FORM5	C1EF	FORMAT	C176	GRABAR	C51F	GRBP10	C2F9
GRBP11	C2FA	GRBP12	C300	GRBP13	C305	GRBPR	C27D	GRBPR0	C283
GRBPR2	C28D	GRBPR3	C294	GRBPR4	C29B	GRBPR5	C29D	GRBPR6	C28F
GRBPR7	C2CF	GRBPR8	C2EF	GRBPR9	C2F8	HIR	E0D7	INGRE1	C439
INGRES	C42B	INIC	C02B	INIC2	C030	INIC3	C036	INIVAR	C33F
INTEN	C34B	INTENT	A03B	KEYCL	E12F	KEYCL1	E13A	KEYTBL	E3DC
LEA	C04A	LEA2	C04E	LEA3	C05A	LEA4	C06C	LEADIR	C354
LEADR1	C365	LEAPR0	C37D	LEAPR1	C381	LEAPR2	C389	LEAPR3	C39D
LEAPR4	C3B7	LEAPR5	C3DE	LEAPRG	C37B	LEARG	C0F5	LEER	C50D
MCMD	C0AE	MCSP	C0EE	MCSP2	C0F0	MEMPIS	2000	MEMSEC	2017
MSCBSQ	0010	MSCESC	00DC	MSCLCT	009C	MSCRDY	0080	MSCRST	0088
MSCVER	0098	NPROG	A038	NUEDIS	C532	PISMAX	0023	PISTA	A036
PPRIN	C4F1	PREGCF	C412	PREGP1	C3FF	PREGPD	C3DF	RAM	A031
RAMADD	A040	RAMSEC	2200	RDCMND	008C	REGPIS	A032	REGR	C14F
REGR1	C151	REGST5	E31C	RESTAR	E080	RETN	C510	RETN1	C512
RSCMND	000B	SALIU	C109	SCNCNT	A01C	SCNREG	8022	SECMAX	000C
SECMXI	000B	SECTOR	A037	SICMND	0058	SKCMND	001B	SOCMND	007B
TECD10	C4C8	TECD11	C4C9	TECD12	C4D4	TECD14	C4D7	TECD15	C4DE
TECDIS	C440	TECDS1	C443	TECDS2	C44D	TECDS3	C472	TECDS4	C477
TECDS5	C486	TECDS6	C495	TECDS7	C499	TECDS8	C4A6	TECDS9	C4C5
VERIF	C146	VERIF2	C148	VFLAG	A010	VRFD10	C271	VRFD11	C224
VRFDIS	C211	VRFDS1	C214	VRFDS2	C216	VRFDS3	C240	VRFDS4	C24B
VRFDS5	C255	VRFDS6	C257	VRFDS7	C262	VRFDS8	C26D	VRFDS9	C26E
WTCMND	00AC	WTKMND	00F4	XDBCHK	A044	XDRBYT	A042	XDSBUF	A020
XKEYBF	A01A								

C A P I T U L O Q U I N T O

"UTILIZACION Y EJEMPLO DE APLICACION"

5. UTILIZACION Y EJEMPLO DE APLICACION.

5.1. UTILIZACION.

Antes de utilizar el circuito desarrollado en la presente tesis, se deberá verificar las conexiones que van a la placa del microprocesador y a la lectora/escritora. Para ello se ha seguido con la convención utilizada en el equipo MEK6800D2 es decir, si los componentes de una placa están en la parte superior de la misma, los cables de varios conductores, deberán salir hacia abajo de la placa. En caso de duda se deberá estudiar los diagramas de circuitos de las diversas placas.

Si se piensa utilizar una nueva lectora/escritora con el circuito, se deberá estudiar la sección 4.3. y seleccionar las diversas opciones de operación que concuerden con las características de la nueva lectora/escritora.

Una vez concluídos los procesos de verificación anteriormente indicados, se puede proceder a la utilización del equipo completo, para ello se deberán seguir los siguientes pasos:

1.- Encender el equipo MEK6800D2 y aplastar el botón de "RESET" del mismo.

2.- Esperar 10 segundos y encender la lectora/escritora.

3.- Insertar un disco y correr el programa que se en-

cuentra en C4F1 (PPRIN).

4.- Aplastar la tecla correspondiente al proceso que se desea realizar:

- Tecla M: Se la utilizará para regresar al control del monitor, el funcionamiento del mismo no ha sido modificado y por lo tanto se podrá trabajar con él, en la forma usual. Si el usuario desea realizar algún proceso en el disco, deberá regresar al paso 3.
- Tecla L: Lectura de un programa. El usuario será preguntado por el número de lectora/escritora (1 dígito) y por el número de programa (2 dígitos), el programa será leído y aparecerá el código de error o no, en la pantalla (ØØ indicará que no se ha producido un error).
- Tecla G: Grabación de un programa. Los números de lectora/escritora y de programa serán preguntados al usuario, seguidamente se preguntarán las direcciones (4 dígitos) donde comienza y termina el programa, se lo grabará en el disco y el código de error aparecerá en la pantalla.
- Tecla V: Borrado de un programa. Se preguntarán los números de lectora/escritora y de programa, el programa será borrado del directorio y se pondrá el código de error en la pantalla.
- Tecla N: Nuevo disco. El usuario será preguntado por los números de lectora/escritora y de vo-

lúmen del disco (2 dígitos), se creará el formato en el disco y se verificarán todos los sectores del mismo, por último se pondrá el código de error en la pantalla.

- Tecla E: Escape. Esta tecla sólamente será aceptada mientras se esté pidiendo un dato al usuario. Una vez aplastada se regresará al paso 4.

El número de programa puede ser cualquiera comprendido entre Ø1 y FF, el número ØØ ha sido reservado para recuperar el último programa borrado, esto es, si accidentalmente se borra un programa del disco, se lo puede recuperar leyendo el programa ØØ.

Un programa que se graba no deberá ocupar las direcciones de memoria comprendidas entre 2ØØØ - 2FFF pues este espacio es utilizado para el control de la información en el disco. Si no se va a crear el formato en un disco nuevo, se podrán utilizar también las direcciones de memoria 23ØØ - 2FFF.

Debe recordarse que los discos pueden ser protegidos contra escritura por lo que será conveniente tomar esta precaución en aquellos que contengan información importante.

5.2. EJEMPLO DE APLICACION.

El siguiente ejemplo se ha diseñado para programar

EPROMS del tipo 2758 y 2716. El circuito y los programas de soporte no son más que versiones modificadas de un programador de EPROMS 2708 que fuera explicado en una revista de electrónica (26).

5.2.1. EL CIRCUITO.

El circuito utiliza una PIA como elemento de interface que ocupa las direcciones de memoria BØØØ - BØØ3. En el equipo MEK6800D2 se la puede direccionar en estas posiciones de memoria haciendo las siguientes conexiones:

MEK6800D2		PIA
AØ	-	RSØ
A1	-	RS1
A12	-	CS1
$\overline{\text{STACK}}$	-	$\overline{\text{CS2}}$

Una vez conectada la PIA a la placa del microprocesador, se deberá armar el circuito de la figura 21 para completar el programador de EPROMS.

De la figura 21 se deduce que el lado "A" de la PIA se utiliza para la transmisión de datos, por lo que éstas líneas periféricas deberán ser programadas como entradas o salidas según se lea o escriba en la EPROM. El lado "B" se utiliza para sacar las direcciones de memoria a la EPROM en conjunto con los flip-flops tipo D del integrado 74LS75. Produciendo un pulso

positivo en CB2 se almacenan en los flip-flops los 3 bits más significativos de la dirección de memoria, para luego sacar los 8 bits restantes directamente del lado "B" de la PIA.

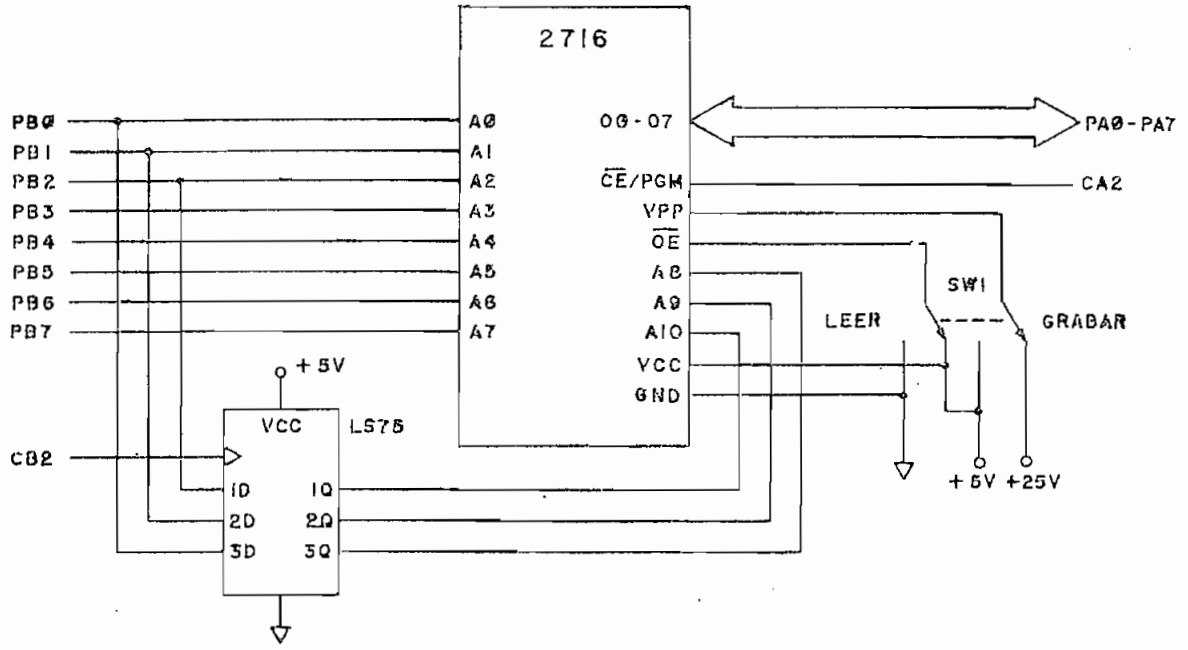


FIGURA 21

PROGRAMADOR DE EPROMS

Por medio de SW1 se enviarán 5 voltios a Vpp y un cero lógico a $\overline{CS}$ para procesos de lectura y 25 voltios a Vpp, y un uno lógico a $\overline{CS}$ para procesos de grabación.

CA2 deberá estar en cero lógico para leer la EPROM, mientras que un pulso positivo de 50 mSeg deberá enviarse por

esta línea para cada dato que se grabe.

5.2.2. LOS PROGRAMAS DE SOPORTE Y SU UTILIZACION.

Los programas de soporte sirven para leer, grabar y verificar el contenido de una EPROM, estos programas se han incluído al final de este capítulo y han sido grabados como programa Ø1 en el disco ØØ.

La función de cada una de las subrutinas utilizadas se podrá comprender leyendo los listados incluídos, por lo que será preferible estudiar la forma de utilización de las subrutinas principales.

5.2.2.1. VBLNK.- ($ØØ1ØØ)

Esta subrutina se utiliza para verificar que una EPROM esté "limpia" es decir, que todas sus localidades de memoria contengan el número FF hexadecimal. Al igual que en la subrutina VRFY, al finalizar el proceso se hace un salto al monitor -si no se ha producido un error- o de producirse se regresa a él, por medio de una interrupción SWI, en este último caso se puede analizar la causa del error por medio de las siguientes condiciones de regreso:

X = Dirección del byte errado en la EPROM.

A = Byte leído de la EPROM.

B = Dato esperado.

5.2.2.2. LEA.- ($Ø1Ø2)

LEA se utiliza para transferir el contenido de la EPROM a memoria, antes de utilizar esta subrutina se deberán definir los valores de las siguientes variables: PRMAD ($Ø1FØ); BEGAD ($Ø1F2) y ENDAD ($Ø1F4), las mismas en las que se deberán definir las siguientes direcciones de memoria:

PRMAD = Dirección inicial en la EPROM (usualmente ØØØØ).
BEGAD = Dirección inicial en memoria.
ENDAD = Dirección final de memoria (ENDAD = BEGAD + número de bytes a leer, escribir o verificar).

Una vez efectuado el proceso de lectura, LEA hace un salto directamente al monitor.

5.2.2.3. ESCRIB.- ($Ø1Ø4)

Esta subrutina se utiliza para grabar la EPROM, antes de utilizarla se deberán definir las variables PRMAD, BEGAD y ENDAD como en la sección 5.2.2.2., los datos a escribirse en la EPROM deberán estar entre las direcciones de memoria BEGAD y ENDAD, por último se deberá colocar SW1 en la posición GRABAR y correr el programa.

Al finalizar el proceso, ESCRIB hace un salto directamente al monitor y el usuario deberá colocar SW1 en la posición LEER nuevamente.

5.2.2.4. VRFY.- ($Ø1Ø6)

VRFY se utiliza para verificar lo grabado en la EPROM, comparando los datos almacenados en memoria con aquellos contenidos por la EPROM. Si se utiliza después de las subrutinas LEA o ESCRIBA, no será necesario definir nuevamente las variables: PRMAD, BEGAD y ENDAD.

El usuario podrá determinar si se produjo un error o no, como en la sección 5.2.2.1.

```
                        *
                        * ESTE PROGRAMA PUEDE UTILIZARSE EN CUALQUIER DIRECCION DE
                        * MEMORIA SIN QUE SE REQUIERAN CAMBIOS.
                        *
01F0                            ORG     $01F0
01F0 00 00              PRMAD   FDB     0
01F2 00 00              BEGAD   FDB     0
01F4 00 00              ENDAD   FDB     0
01F6 00 00              PRMAT   FDB     0
01F8 00 00              MEMAD   FDB     0
                        *
                        *
B000                    PAD     EQU     $B000
B001                    PAC     EQU     $B001
B002                    PBD     EQU     $B002
B003                    PBC     EQU     $B003
E08D                    JBUG    EQU     $E08D
                        *
                        *
                        * TABLA DE SALTOS
                        *
0100                            ORG     $0100
0100 20 06              VBLNK   BRA     VBO         VERIFIQUE EPROM BORRADA
0102 20 11              LEA     BRA     RDO         LEA EPROM A MEMORIA
0104 20 67              ESCRIB  BRA     BRNO        PROGRAME LA EPROM
0106 20 52              VRFY    BRA     VRFYL       VERIFIQUE LO PROGRAMADO
                        *
                        * VBO
                        *
                        * VERIFIQUE QUE LA EPROM HAYA SIDO BORRADA
                        *
0108 8D 1F              VBO     BSR     INIT        INICIALICE PIA PARA LECTURA
010A C6 FF                      LDA B   #$FF        TODA LA EPROM DEBE CONTENER $FF
010C 8D 38              VLP     BSR     VERI        LEA UN DATO DE EPROM Y COMPARELO
010E 8C 08 00                   CPX     #$0800      SE HA VERIFICADO TODA LA EPROM ?
0111 26 F9                      BNE     VLP         NO, CONTINUE HASTA TERMINAR
0113 20 42                      BRA     JJBG        REGRESE AL MONITOR
                        *
                        * RDO
                        *
                        * LEA EL CONTENIDO DE LA EPROM A MEMORIA
                        *
0115 8D 12              RDO     BSR     INIT        INICIALICE PIA PARA LECTURA
0117 FF 01 F8           RD1     STX     MEMAD       ALMACENE EL APUNTADOR DE MEM.
011A 8D 35                      BSR     ENTR        TOME UN DATO DE LA EPROM
011C FE 01 F8                   LDX     MEMAD       APUNTE A MEMORIA
011F A7 00                      STA A   0,X         GUARDE EL DATO EN MEMORIA
0121 BC 01 F4                   CPX     ENDAD       ULTIMO DATO ?
0124 27 31                      BEQ     JJBG        SI, REGRESE AL MONITOR
0126 08                         INX                 INCREMENTE EL APUNTADOR DE MEMORIA
0127 20 EE                      BRA     RD1         CONTINUE HASTA TERMINAR
                        *
                        * INIT
```

```
                    *
                    * PULSO
                    * SUBRUTINA PARA GENERAR UN PULSO DE PROGRAMACION
                    *
015C 86 3C          PULSO   LDA A  #$3C      "PRENDA" EL PULSO
015E B7 B0 01               STA A  PAC
0161 CE 0E FF               LDX    #$0EFF    PREPARESE PARA MANTENERLO 50 MSeg.
0164 09             PULS1   DEX              PRODUZCA LA DEMORA
0165 26 FD                  BNE    PULS1     CONTINUE HASTA TERMINAR
0167 86 34                  LDA A  #$34      "APAGUE" EL PULSO
0169 B7 B0 01               STA A  PAC
016C 39                     RTS
                    *
                    * BRNO
                    *
                    * SUBRUTINA DE PROGRAMACION DE LA EPROM
                    *
016D 8D BA          BRNO    BSR    INIT      INICIALICE LA PIA
016F 86 30                  LDA A  #$30      PROGRAME LADO A DE PIA, SALIDAS
0171 B7 B0 01               STA A  PAC
0174 7A B0 00               DEC    PAD
0177 86 34                  LDA A  #$34
0179 B7 B0 01               STA A  PAC
017C FF 01 F8       BRN1    STX    MEMAD     ALMACENE EL APUNTADOR DE MEMORIA
017F 8D 29                  BSR    ADROUT    PONGA LA DIRECCION PARA LA EPROM
0181 FE 01 F8               LDX    MEMAD     RECOBRE EL APUNTADOR DE MEMORIA
0184 A6 00                  LDA A  0,X       CARGUE UN BYTE DE MEMORIA
0186 B7 B0 00               STA A  PAD       MANDELO A LA EPROM
0189 8D D1                  BSR    PULSO     MANDE UN PULSO DE PROGRAMACION
018B FE 01 F8               LDX    MEMAD     RECOBRE EL APUNTADOR DE MEMORIA
018E BC 01 F4               CPX    ENDAD     SE HA PROGRAMADO TODO LO PEDIDO?
0191 27 C4                  BEQ    JJBG      SI, REGRESE AL MONITOR
0193 08                     INX              APUNTE AL SIGUIENTE BYTE DE MEM.
0194 20 E6                  BRA    BRN1      CONTINUE HASTA TERMINAR
                    *
                    * VFYO
                    *
                    * SUBRUTINA PARA VERIFICAR QUE LO ESCRITO EN LA EPROM SEA
                    * IGUAL A LO QUE SE TIENE EN MEMORIA
                    *
0196 8D 91          VFYO    BSR    INIT      INICIALICE PIA PARA LECTURA
0198 FF 01 F8       VFY1    STX    MEMAD     ALMACENE EL APUNTADOR DE MEMORIA
019B E6 00                  LDA B  0,X       TOME UN BYTE DE MEMORIA
019D 8D AA                  BSR    VERI      COMPARELO CON UNO DE LA EPROM
019F FE 01 F8               LDX    MEMAD     CARGUE EL APUNTADOR DE MEMORIA
01A2 BC 01 F4               CPX    ENDAD     SE HA VERIFICADO TODO LO PEDIDO?
01A5 27 B0                  BEQ    JJBG      SI, REGRESE AL MONITOR
01A7 08                     INX              CONTINUE HASTA TERMINAR
01A8 20 EE                  BRA    VFY1
                    *
                    * ADROUT
                    *
                    * SUBRUTINA PARA SACAR UNA DIRECCION A LA EPROM
                    *
```

```
                    * PULSO
                    * SUBRUTINA PARA GENERAR UN PULSO DE PROGRAMACION
                    *
015C 86 3C          PULSO   LDA A  #$3C      "PRENDA" EL PULSO
015E B7 B0 01               STA A  PAC
0161 CE 0E FF               LDX    #$0EFF    PREPARESE PARA MANTENERLO 50 MSeg
0164 09             PULS1   DEX              PRODUZCA LA DEMORA
0165 26 FD                  BNE    PULS1     CONTINUE HASTA TERMINAR
0167 86 34                  LDA A  #$34      "APAGUE" EL PULSO
0169 B7 B0 01               STA A  PAC
016C 39                     RTS
                    *
                    * BRNO
                    *
                    * SUBRUTINA DE PROGRAMACION DE LA EPROM
                    *
016D 8D BA          BRNO    BSR    INIT      INICIALICE LA PIA
016F 86 30                  LDA A  #$30      PROGRAME LADO A DE PIA, SALIDAS
0171 B7 B0 01               STA A  PAC
0174 7A B0 00               DEC    PAD
0177 86 34                  LDA A  #$34
0179 B7 B0 01               STA A  PAC
017C FF 01 F8       BRN1    STX    MEMAD     ALMACENE EL APUNTADOR DE MEMORIA
017F 8D 29                  BSR    ADROUT    PONGA LA DIRECCION PARA LA EPROM
0181 FE 01 F8               LDX    MEMAD     RECORRE EL APUNTADOR DE MEMORIA
0184 A6 00                  LDA A  0,X       CARGUE UN BYTE DE MEMORIA
0186 B7 B0 00               STA A  PAD       MANDELO A LA EPROM
0189 8D D1                  BSR    PULSO     MANDE UN PULSO DE PROGRAMACION
018B FE 01 F8               LDX    MEMAD     RECORRE EL APUNTADOR DE MEMORIA
018E BC 01 F4               CPX    ENDAD     SE HA PROGRAMADO TODO LO PEDIDO?
0191 27 C4                  BEQ    JJBG      SI, REGRESE AL MONITOR
0193 08                     INX              APUNTE AL SIGUIENTE BYTE DE MEM.
0194 20 E6                  BRA    BRN1      CONTINUE HASTA TERMINAR
                    *
                    * VFYO
                    *
                    * SUBRUTINA PARA VERIFICAR QUE LO ESCRITO EN LA EPROM SEA
                    * IGUAL A LO QUE SE TIENE EN MEMORIA
                    *
0196 8D 91          VFYO    BSR    INIT      INICIALICE PIA PARA LECTURA
0198 FF 01 F8       VFY1    STX    MEMAD     ALMACENE EL APUNTADOR DE MEMORIA
019B E6 00                  LDA B  0,X       TOME UN BYTE DE MEMORIA
019D 8D AA                  BSR    VERI      COMPARELO CON UNO DE LA EPROM
019F FE 01 F8               LDX    MEMAD     CARGUE EL APUNTADOR DE MEMORIA
01A2 BC 01 F4               CPX    ENDAD     SE HA VERIFICADO TODO LO PEDIDO?
01A5 27 B0                  BEQ    JJBG      SI, REGRESE AL MONITOR
01A7 08                     INX              CONTINUE HASTA TERMINAR
01A8 20 EE                  BRA    VFY1
                    *
                    * ADROUT
                    *
                    * SUBRUTINA PARA SACAR UNA DIRECCION A LA EPROM
                    *
01AA CE B0 00       ADROUT  LDX    #PAD      APUNTE A LA PIA
```

```
01AD B6 01 F6        LDA A   FRMAT       MSB DE DIRECCION DE LA EPROM
01B0 A7 02           STA A   2,X         MANDELO AL "LATCH"
01B2 86 3C           LDA A   #$3C        MANDE UN PULSO DE ALMACENAMIENTO
01B4 A7 03           STA A   3,X
01B6 86 34           LDA A   #$34        "APAGUE" EL PULSO
01B8 A7 03           STA A   3,X
01BA B6 01 F7        LDA A   FRMAT+1     LSB DE DIRECCION DE LA EPROM
01BD A7 02           STA A   2,X         MANDELO A LA EPROM
01BF FE 01 F6        LDX     FRMAT       CARGUE LA DIRECCION COMPLETA
01C2 08              INX                 INCREMENTELA
01C3 FF 01 F6        STX     FRMAT
01C6 39              RTS
                     END
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADROUT	01AA	BEGAD	01F2	BRNO	016D	BRN1	017C	ENDAD	01F4
ENTR	0151	ESCRIB	0104	INIT	0129	JBUG	E08D	JJBG	0157
LEA	0102	MEMAD	01F8	PAC	B001	PAD	B000	PBC	B003
PBD	B002	PRMAD	01F0	PRMAT	01F6	PULS0	015C	PULS1	0164
RD0	0115	RD1	0117	VB0	0108	VBLNK	0100	VER1	0149
VFY0	0196	VFY1	0198	VLP	010C	VRFY	0106	VRFYL	015A
VRT	0150								

COMENTARIOS Y CONCLUSIONES

Una vez armado el circuito desarrollado en la presente tesis, se pudo constatar que el integrado 74S124 (U4) producía mucho ruido, interfiriendo con la operación normal del circuito PLL separador de datos, esto hizo que se cambie C14 por un condensador de tantalio de 4.7 µF.

Otro error se encontró en las notas de aplicación del integrado FD1797-02 y es el de que los pulsos en la señal leída del disco deben tener una duración de 400 nSeg. para discos de 5¼" y no de 200 nSeg. como se indica en estas notas. El error se ha corregido ajustando el potenciómetro P4.

Originalmente la subrutina BUSQ fue diseñada para localizar una pista en el disco -por programa- con el objeto de producir una demora de 10 mSeg. al pasar de una pista a otra pero, al existir la posibilidad de utilizar el circuito con una lectora/escritora más rápida que al no necesitar de esta demora, pemita la utilización del comando "SEEK" propio del controlador para localizar una pista en el disco, se ha creído conveniente presentar aquí las modificaciones que se deberían hacer en dicho caso, consiguiendo con ello que la subrutina BUSQ sea más corta, rápida y confiable:

1.- Eliminar la subrutina DEMORA (Que forma parte de BUSQ)
2.- Modificar la subrutina BUSQ como se indica a continuación:

```
CØ8D   BD CØ 9E   JSR      DTARG    APUNTE A "DATA REGISTER"
CØ9Ø   BD CØ AE   JSR      MCMD     ALMACENE EL NUMERO DE PISTA
CØ93   BD CØ A7   JSR      CMRG     APUNTE A "COMMAND REGISTER"
CØ96   86 1B      LDA A    #SKCMND  CARGUE UN COMANDO "SEEK"
CØ98   BD CØ EE   JSR      MCSP     MANDELO Y ESPERE TERMINARLO
CØ9B   39         RTS
```

Puesto que el equipo MEK6800D2 no dispone más que de 512 bytes de memoria mientras que el controlador de discos floppy necesita de alrededor de 4 K bytes para la creación del formato en el disco, se ha ampliado la memoria del equipo a 32 K utilizando para ello una placa producida por la compañía DIGITAL RESEARCH COMPUTERS que deberá ser utilizada mientras se concluye una tesis que se está desarrollando para el objeto.

Por último cabe anotar que el sistema terminado ha sido probado extensamente, pudiendo comprobar sus excelentes características de velocidad, confiabilidad y facilidad de manejo, superando -en la mayoría de los casos- las metas propuestas para la presente tesis.

A P E N D I C E 1

"ESPECIFICACIONES DE LA FAMILIA DE CONTROLADORES FD179X-02"

FD 179X-02 Floppy Disk Formatter/Controller Family

MAY 1980

FEATURES

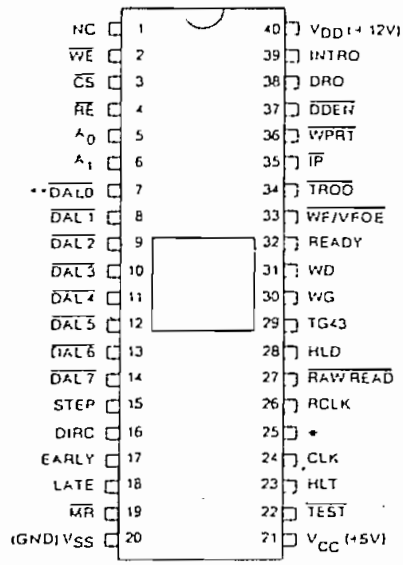
- TWO VFO CONTROL SIGNALS
- SOFT SECTOR FORMAT COMPATIBILITY
- AUTOMATIC TRACK SEEK WITH VERIFICATION
- ACCOMMODATES SINGLE AND DOUBLE DENSITY FORMATS
  IBM 3740 Single Density (FM),
  IBM System 34 Double Density (MFM)
- READ MODE
  Single/Multiple Sector Read with Automatic Search or Entire Track Read
  Selectable 128 Byte or Variable length Sector
- WRITE MODE
  Single/Multiple Sector Write with Automatic Sector Search
  Entire Track Write for Diskette Formatting
- SYSTEM COMPATIBILITY
  Double Buffering of Data 8 Bit Bi-Directional Bus for Data, Control and Status
  DMA or Programmed Data Transfers
  All Inputs and Outputs are TTL Compatible
  On-Chip Track and Sector Registers/Comprehensive Status Information
- PROGRAMMABLE CONTROLS
  Selectable Track to Track Stepping Time
  Side Select Compare
- WRITE PRECOMPENSATION
- WINDOW EXTENSION
- INCORPORATES ENCODING/DECODING AND ADDRESS MARK CIRCUITRY
- FD1792/4 IS SINGLE DENSITY ONLY
- FD1795/7 HAS A SIDE SELECT OUTPUT

179X-02 FAMILY CHARACTERISTICS

FEATURES	1791	1793	1795	1797
Single Density (FM)	X	X	X	X
Double Density (MFM)	X	X	X	X
True Data Bus		X		X
Inverted Data Bus	X		X	
Write Precomp	X	X	X	X
Side Selection Output			X	X

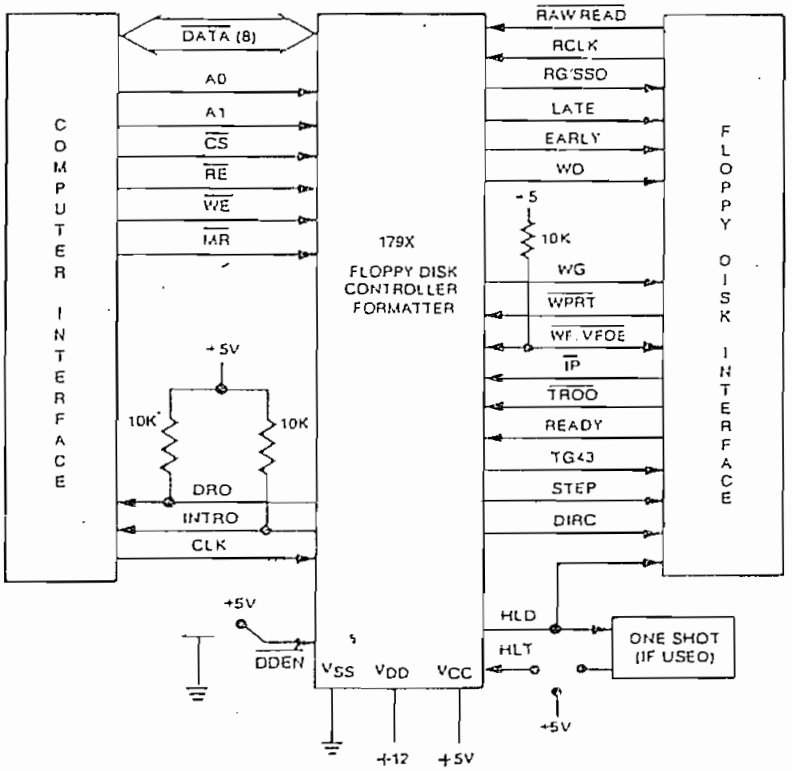
APPLICATIONS

FLOPPY DISK DRIVE INTERFACE
SINGLE OR MULTIPLE DRIVE CONTROLLER/FORMATTER
NEW MINI-FLOPPY CONTROLLER



*1791/3 = RG 1795/7 = SSO
**1793/7 TRUE BUS

PIN CONNECTIONS



FD179X SYSTEM BLOCK DIAGRAM

functions of a Floppy Disk Formatter/Controller in a single chip implementation. The FD179X, which can be considered the end result of both the FD1771 and FD1781 designs, is IBM 3740 compatible in single density mode (FM) and System 34 compatible in Double Density Mode (MFM). The FD179X contains all the features of its predecessor the FD1771, plus the added features necessary to read/write and format a double density diskette. These include address mark detection, FM and MFM encode and decode logic, window extension, and write precompensation. In order to maintain compatibility, the FD1771, FD1781, and FD179X designs were made as close as possible with the computer interface, instruction set, and I/O registers being identical. Also, head load signments vary by only a few pins from any one to another.

The processor interface consists of an 8-bit bidirectional bus for data, status, and control word transfers. The FD179X is set up to operate on a multiplexed bus with other bus-oriented devices.

The FD179X is fabricated in N-channel Silicon Gate MOS technology and is TTL compatible on all inputs and outputs. The 1793 is identical to the 1791 except the DAL lines are TRUE for systems that utilize true data busses.

The 1795/7 has a side select output for controlling double sided drives, and the 1792 and 1794 are "Single Density Only" versions of the 1791 and 1793. On these devices, DDEN must be left open.

PIN OUTS

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
1	NO CONNECTION	NC	Pin 1 is internally connected to a back bias generator and must be left open by the user.
19	MASTER RESET	$\overline{MR}$	A logic low on this input resets the device and loads HEX 03 into the command register. The Not Ready (Status Bit 7) is reset during $\overline{MR}$ ACTIVE. When $\overline{MR}$ is brought to a logic high a RESTORE Command is executed, regardless of the state of the Ready signal from the drive. Also, HEX 01 is loaded into sector register.
20	POWER SUPPLIES	V_{SS}	Ground
21		V_{CC}	+5V ±5%
40		V_{DD}	+12V ±5%
COMPUTER INTERFACE:			
2	WRITE ENABLE	$\overline{WE}$	A logic low on this input gates data on the DAL into the selected register when $\overline{CS}$ is low.
3	CHIP SELECT	$\overline{CS}$	A logic low on this input selects the chip and enables computer communication with the device.
4	READ ENABLE	$\overline{RE}$	A logic low on this input controls the placement of data from a selected register on the DAL when $\overline{CS}$ is low.
5,6	REGISTER SELECT LINES	A0, A1	These inputs select the register to receive/transfer data on the DAL lines under $\overline{RE}$ and $\overline{WE}$ control: A1 A0 $\overline{RE}$ $\overline{WE}$ 0 0 Status Reg Command Reg 0 1 Track Reg Track Reg 1 0 Sector Reg Sector Reg 1 1 Data Reg Data Reg
7-14	DATA ACCESS LINES	$\overline{DAL0}$-$\overline{DAL7}$	Eight bit inverted Bidirectional bus used for transfer of data, control, and status. This bus is receiver enabled by $\overline{WE}$ or transmitter enabled by $\overline{RE}$.
24	CLOCK	CLK	This input requires a free-running square wave clock for internal timing reference, 2 MHz for 8" drives, 1 MHz for mini-drives.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
38	DATA REQUEST	DRQ	This open drain output indicates that the DR contains assembled data in Read operations, or the DR is empty in Write operations. This signal is reset when serviced by the computer through reading or loading the DR in Read or Write operations, respectively. Use 10K pull-up resistor to +5.
39	INTERRUPT REQUEST	INTRQ	This open drain output is set at the completion of any command and is reset when the STATUS register is read or the command register is written to. Use 10K pull-up resistor to +5.
FLOPPY DISK INTERFACE:			
15	STEP	STEP	The step output contains a pulse for each step.
16	DIRECTION	DIRC	Direction Output is active high when stepping in, active low when stepping out.
17	EARLY	EARLY	Indicates that the WRITE DATA pulse occurring while Early is active (high) should be shifted early for write precompensation.
18	LATE	LATE	Indicates that the write data pulse occurring while Late is active (high) should be shifted late for write precompensation.
22	$\overline{\text{TEST}}$	$\overline{\text{TEST}}$	This input is used for testing purposes only and should be tied to +5V or left open by the user unless interfacing to voice coil actuated motors.
23	HEAD LOAD TIMING	HLT	When a logic high is found on the HLT input the head is assumed to be engaged.
25	READ GATE (1791/3)	RG	A high level on this output indicates to the data separator circuitry that a field of zeros (or ones) has been encountered, and is used for synchronization.
25	SIDE SELECT OUTPUT (1795, 1797)	SSO	The logic level of the Side Select Output is directly controlled by the 'S' flag in Type II or III commands. When S = 1, SSO is set to a logic 1. When S = 0, SSO is set to a logic 0. The Side Select Output is only updated at the beginning of a Type II or III command. It is forced to a logic 0 upon a MASTER RESET condition.
26	READ CLOCK	RCLK	A nominal square-wave clock signal derived from the data stream must be provided to this input. Phasing (i.e. RCLK transitions) relative to RAW READ is important but polarity (RCLK high or low) is not.
27	$\overline{\text{RAW READ}}$	$\overline{\text{RAW READ}}$	The data input signal directly from the drive. This input shall be a negative pulse for each recorded flux transition.
28	HEAD LOAD	HLD	The HLD output controls the loading of the Read-Write head against the media.
29	TRACK GREATER THAN 43	TG43	This output informs the drive that the Read/Write head is positioned between tracks 44-76. This output is valid only during Read and Write Commands.
30	WRITE GATE	WG	This output is made valid before writing is to be performed on the diskette.

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
31	WRITE DATA	WD	A 250 ns (MFM) or 500 ns (FM) pulse per flux transition. WD contains the unique Address marks as well as data and clock in both FM and MFM formats.
32	READY	READY	This input indicates disk readiness and is sampled for a logic high before Read or Write commands are performed. If Ready is low the Read or Write operation is not performed and an interrupt is generated. Type I operations are performed regardless of the state of Ready. The Ready input appears in inverted format as Status Register bit 7.
33	$\overline{\text{WRITE FAULT}}$ / VFO ENABLE	$\overline{\text{WF}}$/$\overline{\text{VFOE}}$	This is a bi-directional signal used to signify writing faults at the drive, and to enable the external PLO data separator. When WG = 1, Pin 33 functions as a WF input. If WF = 0, any write command will immediately be terminated. When WG = 0, Pin 33 functions as a VFOE output. VFOE will go low during a read operation after the head has loaded and settled (HLT = 1). On the 1795/7, it will remain low until the last bit of the second CRC byte in the ID field. VFOE will then go high until 8 bytes (MFM) or 4 bytes (FM) before the Address Mark. It will then go active until the last bit of the second CRC byte of the Data Field. On the 1791/3, VFOE will remain low until the end of the Data Field.
34	$\overline{\text{TRACK 00}}$	$\overline{\text{TR00}}$	This input informs the FD179X that the Read/Write head is positioned over Track 00.
35	$\overline{\text{INDEX PULSE}}$	$\overline{\text{IP}}$	This input informs the FD179X when the index hole is encountered on the diskette.
36	$\overline{\text{WRITE PROTECT}}$	$\overline{\text{WPRT}}$	This input is sampled whenever a Write Command is received. A logic low terminates the command and sets the Write Protect Status bit.
37	$\overline{\text{DOUBLE DENSITY}}$	$\overline{\text{DDEN}}$	This pin selects either single or double density operation. When $\overline{\text{DDEN}}$ = 0, double density is selected. When $\overline{\text{DDEN}}$ = 1, single density is selected. This line must be left open on the 1792/4

ORGANIZATION

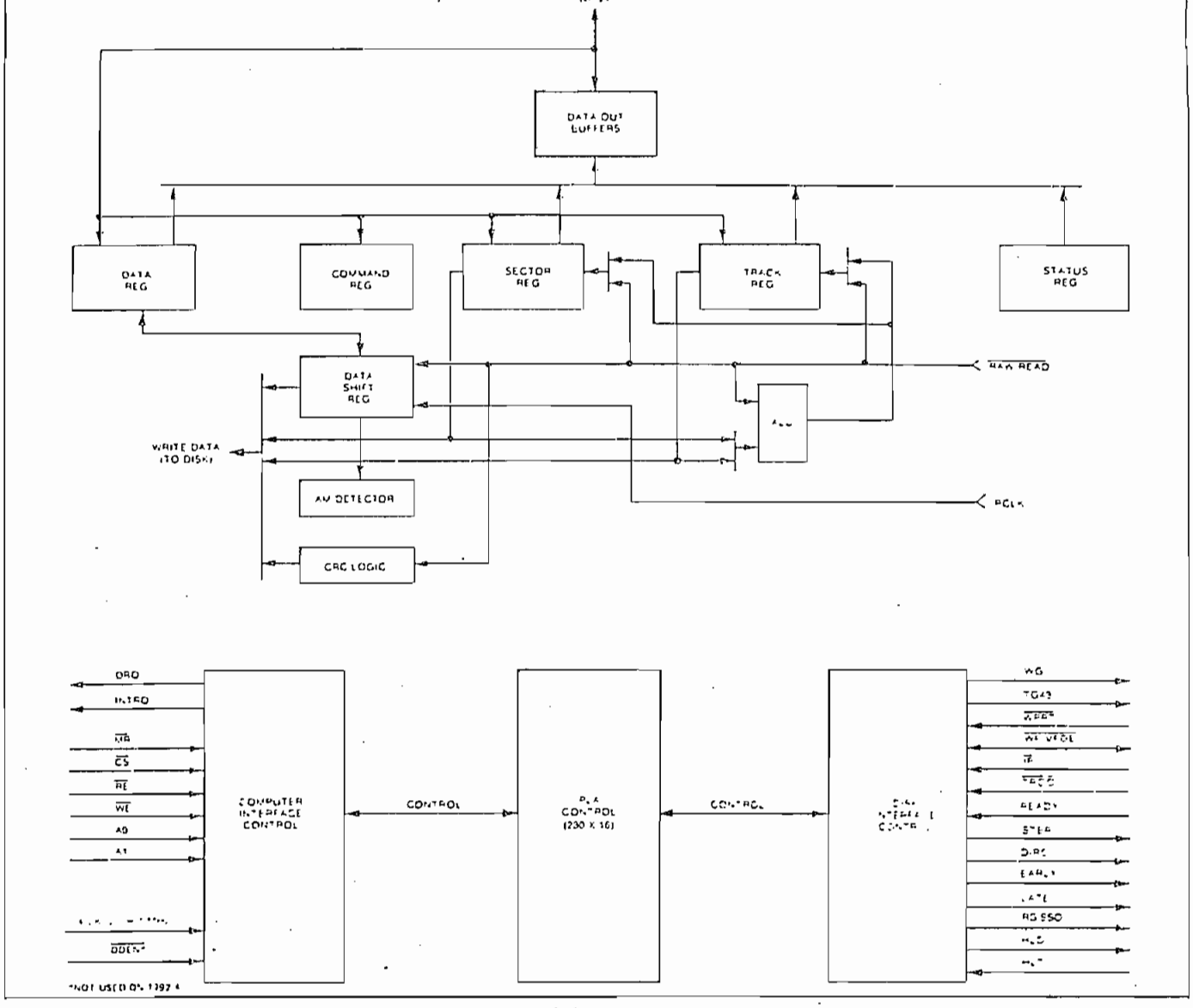
The Floppy Disk Formatter block diagram is illustrated on page 5. The primary sections include the parallel processor interface and the Floppy Disk interface.

Data Shift Register—This 8-bit register assembles serial data from the Read Data input ($\overline{\text{RAW READ}}$) during Read operations and transfers serial data to the Write Data output during Write operations.

Data Register—This 8-bit register is used as a holding register during Disk Read and Write operations. In Disk Read operations the assembled data byte is transferred in parallel to the Data Register from the Data Shift Register. In Disk Write operations information is transferred in parallel from the Data Register to the Data Shift Register.

When executing the Seek command the Data Register holds the address of the desired Track position. This register is loaded from the DAL and gated onto the DAL under processor control.

Track Register—This 8-bit register holds the track number of the current Read/Write head position. It is incremented by one every time the head is stepped in (towards track 76) and decremented by one when the head is stepped out (towards track 00). The contents of the register are compared with the recorded track number in the ID field during disk Read, Write, and Verify operations. The Track Register can be loaded from or transferred to the DAL. This Register should not be loaded when the device is busy.



FD179X BLOCK DIAGRAM

Sector Register (SR)—This 8-bit register holds the address of the desired sector position. The contents of the register are compared with the recorded sector number in the ID field during disk Read or Write operations. The Sector Register contents can be loaded from or transferred to the DAL. This register should not be loaded when the device is busy.

Command Register (CR)—This 8-bit register holds the command presently being executed. This register should not be loaded when the device is busy unless the new command is a force interrupt. The command register can be loaded from the DAL, but not read onto the DAL.

Status Register (STR)—This 8-bit register holds device Status information. The meaning of the Status bits is a function of the type of command previously executed. This register can be read onto the DAL, but not loaded from the DAL.

CRC Logic—This logic is used to check or to generate the 16-bit Cyclic Redundancy Check (CRC). The polynomial is: $G(x) = x^{16} + x^{12} + x^5 + 1$.

The CRC includes all information starting with the address mark and up to the CRC characters. The CRC register is preset to ones prior to data being shifted through the circuit.

Arithmetic/Logic Unit (ALU)—The ALU is a serial comparator, incrementer, and decrementer and is used for register modification and comparisons with the disk recorded ID field.

Timing and Control—All computer and Floppy Disk Interface controls are generated through this logic. The internal device timing is generated from an external crystal clock.

The FD1791/3 has two different modes of operation according to the state of $\overline{DDEN}$. When $\overline{DDEN} = 0$ double density (MFM) is assumed. When $\overline{DDEN} = 1$, single density (FM) is assumed.

AM Detector—The address mark detector detects ID, data and index address marks during read and write operations.

PROCESSOR INTERFACE

The interface to the processor is accomplished through the eight Data Access Lines ($\overline{DAL}$) and associated control signals. The $\overline{DAL}$ are used to transfer Data, Status, and Control words out of, or into the FD179X. The $\overline{DAL}$ are three state buffers that are enabled as output drivers when Chip Select (CS) and Read Enable ($\overline{RE}$) are active (low logic state) or act as input receivers when $\overline{CS}$ and Write Enable ($\overline{WE}$) are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and $\overline{CS}$ is made low. The address bits A1 and A0, combined with the signals $\overline{RE}$ during a Read operation or $\overline{WE}$ during a Write operation are interpreted as selecting the following registers:

A1-A0	READ ($\overline{RE}$)	WRITE ($\overline{WE}$)
0 0	Status Register	Command Register
0 1	Track Register	Track Register
1 0	Sector Register	Sector Register
1 1	Data Register	Data Register

During Direct Memory Access (DMA) types of data transfers between the Data Register of the FD179X and the processor, the Data Request (DRQ) output is used in Data Transfer control. This signal also appears as status bit 1 during Read and Write operations.

On Disk Read operations the Data Request is activated (set high) when an assembled serial input byte is transferred in parallel to the Data Register. This bit is cleared when the Data Register is read by the processor. If the Data Register is read after one or more characters are lost, by having new data transferred into the register prior to processor readout, the Lost Data bit is set in the Status Register. The Read operation continues until the end of sector is reached.

On Disk Write operations the data Request is activated when the Data Register transfers its contents to the Data Shift Register, and requires a new data byte. It is reset when the Data Register is loaded with new data by the processor. If new data is not loaded at the time the next serial byte is required by the Floppy Disk, a byte of zeroes is written on the diskette and the Lost Data bit is set in the Status Register.

At the completion of every command an INTRQ is generated. INTRQ is reset by either reading the status register or by loading the command register with a new command. In addition, INTRQ is generated if a Force Interrupt command condition is met.

FLOPPY DISK INTERFACE

The 179X has two modes of operation according to the state of $\overline{DDEN}$ (Pin 37). When $\overline{DDEN}$ = 1, single density is selected. In either case, the CLK input (Pin 24) is at 2 MHz. However, when interfacing with the mini-floppy, the CLK input is set at 1 MHz for both single density and double density. When the clock is at 2 MHz, the stepping rates of 3, 6, 10, and 15 ms are obtainable. When CLK equals 1 MHz these times are doubled.

HEAD POSITIONING

Five commands cause positioning of the Read-Write head (see Command Section). The period of each positioning step is specified by the r field in bits 1 and 0 of the command word. After the last directional step an additional 15 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. Note that this time doubles to 30 ms for a 1 MHz clock. If $\overline{TEST}$ = 0, there is zero settling time. There is also a 15 ms head settling time if the E flag is set in any Type II or III command.

The rates (shown in Table 1) can be applied to a Step-Direction Motor through the device interface.

Step—A 2 μs (MFM) or 4 μs (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output.

Direction (DIRC)—The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid 12 μs before the first stepping pulse is generated.

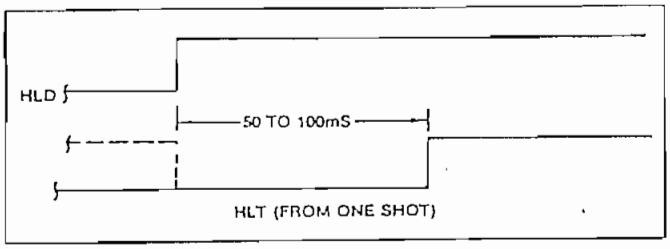
When a Seek, Step or Restore command is executed an optional verification of Read-Write head position can be performed by setting bit 2 (V = 1) in the command word to a logic 1. The verification operation begins at the end of the 15 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. The FD179X must find an ID field with correct track number and correct CRC within 5 revolutions of the media; otherwise the seek error is set and an INTRQ is generated.

Table 1. STEPPING RATES

CLK	2 MHz	2 MHz	1 MHz	1 MHz	2 MHz	1 MHz
$\overline{DDEN}$	0	1	0	1	x	x
R1 R0	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=1	$\overline{TEST}$=0	$\overline{TEST}$=0
0 0	3 ms	3 ms	6 ms	6 ms	184μs	368μs
0 1	6 ms	6 ms	12 ms	12 ms	190μs	380μs
1 0	10 ms	10 ms	20 ms	20 ms	198μs	396μs
1 1	15 ms	15 ms	30 ms	30 ms	208μs	416μs

The Head Load (HLD) output controls the movement of the read/write head against the media. HLD is activated at the beginning of a Type I command if the h flag is set (h = 1), at the end of the Type I command if the verify flag (V = 1), or upon receipt of any Type II or III command. Once HLD is active it remains active until either a Type I command is received with (h = 0 and V = 0); or if the FD179X is in an idle state (non-busy) and 15 index pulses have occurred.

which is used for the head engage time. When HLT = 1, the FD179X assumes the head is completely engaged. The head engage time is typically 30 to 100 ms depending on drive. The low to high transition on HLD is typically used to fire a one shot. The output of the one shot is then used for HLT and supplied as an input to the FD179X.



HEAD LOAD TIMING

When both HLD and HLT are true, the FD179X will then read from or write to the media. The "and" of HLD and HLT appears as a status bit in Type I status.

In summary for the Type I commands: if h = 0 and V = 0, HLD is reset. If h = 1 and V = 0, HLD is set at the beginning of the command and HLT is not sampled nor is there an internal 15 ms delay. If h = 0 and V = 1, HLD is set near the end of the command, an internal 15 ms occurs, and the FD179X waits for HLT to be true. If h = 1 and V = 1, HLD is set at the beginning of the command. Near the end of the command, after all the steps have been issued, an internal 15 ms delay occurs and the FD179X then waits for HLT to occur.

For Type II and III commands with E flag off, HLD is made active and HLT is sampled until true. With E flag on, HLD is made active, an internal 15 ms delay occurs and then HLT is sampled until true.

DISK READ OPERATIONS

Sector lengths of 128, 256, 512 or 1024 are obtainable in either FM or MFM formats. For FM, $\overline{\text{DDEN}}$ should be placed to logical "1." For MFM formats, $\overline{\text{DDEN}}$ should be placed to a logical "0." Sector lengths are determined at format time by a special byte in the "ID" field. If this Sector length byte in the ID field is zero, then the sector length is 128 bytes. If 01 then 256 bytes. If 02, then 512 bytes. If 03, then the sector length is 1024 bytes. The number of sectors per track as far as the FD179X is concerned can be from 1 to 255 sectors. The number of tracks as far as the FD179X is concerned is from 0 to 255 tracks. For IBM 3740 compatibility, sector lengths are 128 bytes with 26 sectors per track. For System 34 compatibility (MFM), sector lengths are 256 bytes/sector with 26 sectors/track; or lengths of 1024 bytes/sector with 8 sectors/track. (See Sector Length Table.)

For read operations, the FD179X requires $\overline{\text{RAW READ}}$ Data (Pin 27) signal which is a 250 ns pulse per flux transition and a Read clock (RCLK) signal to indicate flux transition spacings. The RCLK (Pin 26) signal is provided by some drives but if not it may be counter techniques. In addition, a Read Gate Signal is provided as an output (Pin 25) which can be used to inform phase lock loops when to acquire synchronization. When reading from the media in FM. RG is made true when 2 bytes of zeroes are detected. The FD179X must find an address mark within the next 10 bytes; otherwise RG is reset and the search for 2 bytes of zeroes begins all over again. If an address mark is found within 10 bytes, RG remains true as long as the FD179X is deriving any useful information from the data stream. Similarly for MFM, RG is made active when 4 bytes of "00" or "FF" are detected. The FD179X must find an address mark within the next 16 bytes, otherwise RG is reset and search resumes.

During read operations (WG = 0), the $\overline{\text{VFOE}}$ (Pin 33) is provided for phase lock loop synchronization. $\overline{\text{VFOE}}$ will go active when:

a) Both HLT and HLD are True
b) Settling Time, if programmed, has expired
c) The 179X is inspecting data off the disk

If $\overline{\text{WF}}/\overline{\text{VFOE}}$ is not used, leave open or tie to a 10K resistor to +5.

DISK WRITE OPERATION

When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the FD179X before the Write Gate signal can be activated.

Writing is inhibited when the $\overline{\text{Write Protect}}$ input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set. The Write Fault input, when activated, signifies a writing fault condition detected in disk drive electronics such as failure to detect write current flow when the Write Gate is activated. On detection of this fault the FD179X terminates the current command, and sets the Write Fault bit (bit 5) in the Status Word. The $\overline{\text{Write Fault}}$ input should be made inactive when the Write Gate output becomes inactive.

For write operations, the FD179X provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of 500 ns pulses in FM ($\overline{\text{DDEN}}$ = 1) and 250 ns pulses in MFM ($\overline{\text{DDEN}}$ = 0). Write Data provides the unique address marks in both formats.

Also during write, two additional signals are provided for write precompensation. These are EARLY (Pin 17) and LATE (Pin 18). EARLY is active true when the WD pulse appearing on (Pin 30) is to be written early. LATE is active true when the WD pulse is to be written LATE. If both EARLY and LATE are low when the WD pulse is present, the WD pulse is to be written at nominal. Since write precompensation values vary from disk manufacturer to disk manufacturer, the actual value is determined by several one shots or delay lines which are located external to the FD179X. The write precompensation signals EARLY and LATE are valid for the duration of WD in both FM and MFM formats.

is required the FD179X samples the Ready input. If this input is logic low the command is not executed and an interrupt is generated. All Type I commands are performed regardless of the state of the Ready input. Also, whenever a Type II or III command is received, the TG43 signal output is updated.

COMMAND DESCRIPTION

The FD179X will accept eleven commands. Command words should only be loaded in the Command Register when the Busy status bit is off (Status bit 0). The one exception is the Force Interrrupt command. Whenever a command is being executed, the Busy status bit is set. When a command is completed, an interrupt is generated and the Busy status bit is reset. The Status Register indicates whether the completed command encountered an error or was fault free. For ease of discussion, commands are divided into four types. Commands and types are summarized in Table 2.

Table 2. COMMAND SUMMARY

		BITS							
TYPE	COMMAND	7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	u	h	V	r_1	r_0
I	Step In	0	1	0	u	h	V	r_1	r_0
I	Step Out	0	1	1	u	h	V	r_1	r_0
II	Read Sector	1	0	0	m	F_2	E	F_1	0
II	Write Sector	1	0	1	m	F_2	E	F_1	a_0
III	Read Address	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrrupt	1	1	0	1	I_3	I_2	I_1	I_0

Note: Bits shown in TRUE form.

Table 3. FLAG SUMMARY

TYPE I COMMANDS

h = Head Load Flag (Bit 3)

h = 1, Load head at beginning
h = 0, Unload head at beginning

V = Verify flag (Bit 2)

V = 1, Verify on destination track
V = 0, No verify

r_1r_0 = Stepping motor rate (Bits 1-0)

Refer to Table 1 for rate summary

u = Update flag (Bit 4)

u = 1, Update Track register
u = 0, No update

TYPE II & III COMMANDS

m = Multiple Record flag (Bit 4)

m = 0, Single Record
m = 1, Multiple Records

a_0 = Data Address Mark (Bit 0)

a_0 = 0, FB (Data Mark)
a_0 = 1, F8 (Deleted Data Mark)

E = 15 ms Delay (2MHz)

E = 1, 15 ms delay
E = 0, no 15 ms delay

(F_2) S = Side Select Flag (1791/3 only)

S = 0, Compare for Side 0
S = 1, Compare for Side 1

(F_1) C = Side Compare Flag (1791/3 only)

C = 0, disable side select compare
C = 1, enable side select compare

(F_1) S = Side Select Flag

(Bit 1, 1795/7 only)

S = 0 Update SSO to 0
S = 1 Update SSO to 1

(F_2) b = Sector Length Flag

(Bit 3, 1975/7 only)

	Sector Length Field			
	00	01	10	11
b = 0	256	512	1024	128
b = 1	128	256	512	1024

Table 5. FLAG SUMMARY

TYPE IV COMMAND

Ii = Interrupt Condition flags (Bits 3-0)

I0 = 1, Not-Ready to Ready Transition
I1 = 1, Ready to Not-Ready Transition
I2 = 1, Index Pulse
I3 = 1, Immediate Interrupt
$I_3 - I_0$ = 0, Terminate with no Interrupt

TYPE I COMMANDS

The Type I Commands include the Restore, Seek, Step, Step-In, and Step-Out commands. Each of the Type I Commands contains a rate field (r_0r_1), which determines the stepping motor rate as defined in Table 1.

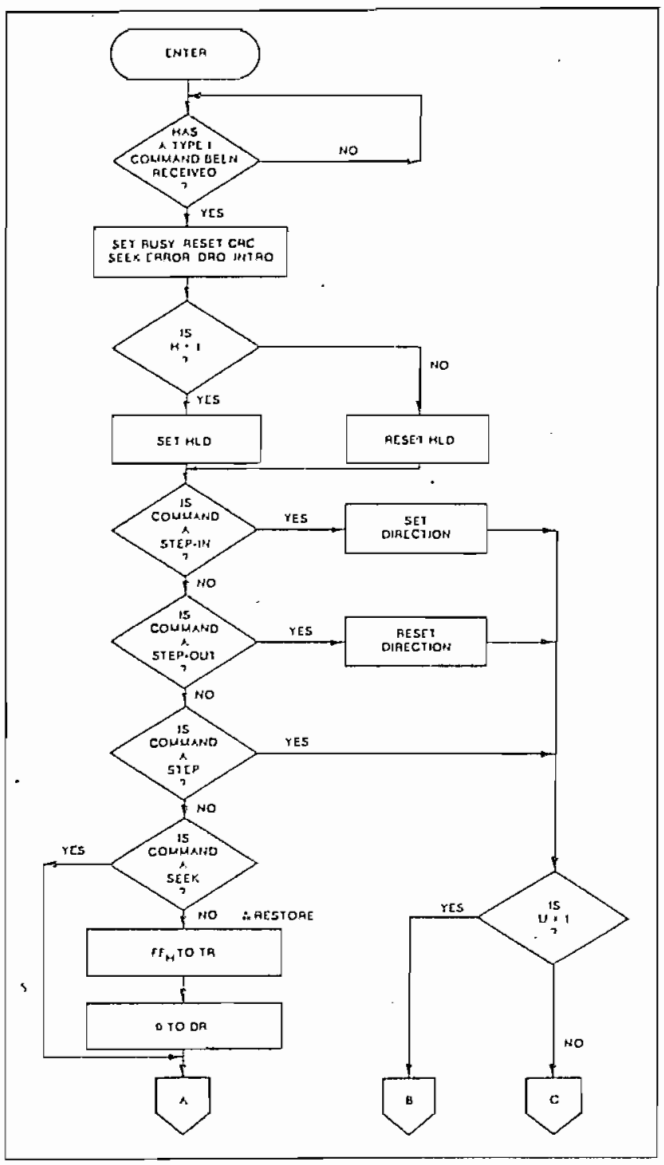
The Type I Commands contain a head load flag (h) which determines if the head is to be loaded at the beginning of the command. If h = 1, the head is loaded at the beginning of the command (HLD output is made active). If h = 0, HLD is deactivated. Once the head is loaded, the head will remain engaged until the FD179X receives a command that specifically disengages the head. If the FD179X is idle (busy = 0) for 15 revolutions of the disk, the head will be automatically disengaged (HLD made inactive).

The Type I Commands also contain a verification (V) flag which determines if a verification operation is to take place on the destination track. If V = 1, a verification is performed, if V = 0, no verification is performed.

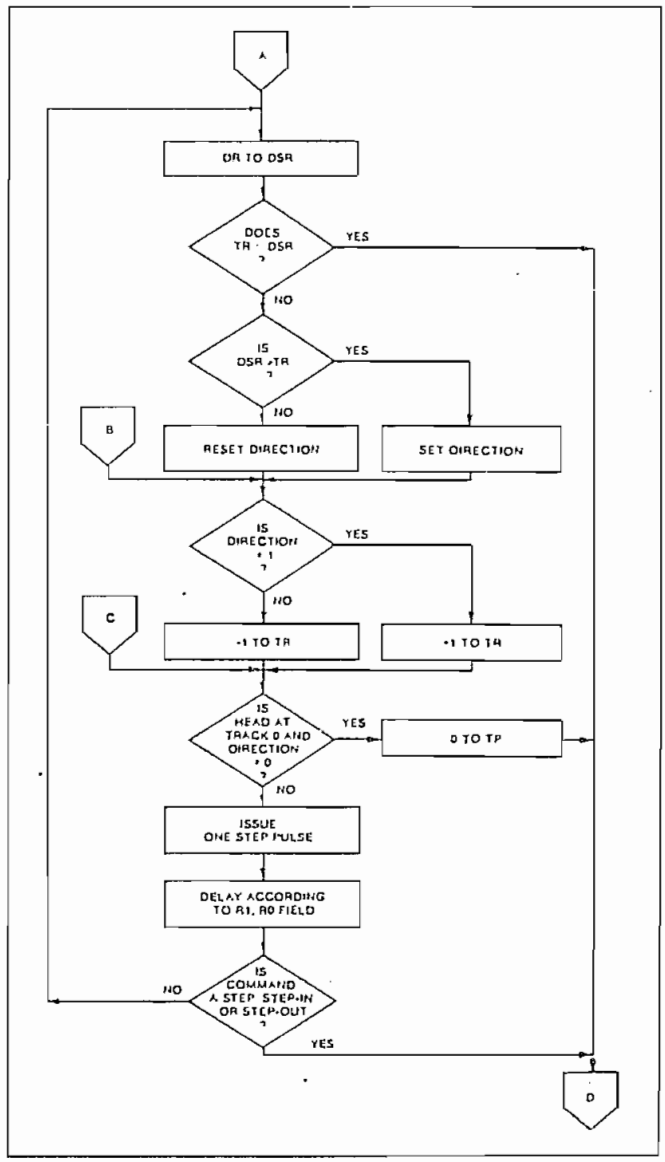
During verification, the head is loaded and after an internal 15 ms delay, the HLT input is sampled. When HLT is active (logic true), the first encountered ID field is read off the disk. The track address of the ID field is then compared to DR. If there is a match and a valid ID CRC, the verification is complete, an interrupt is generated and the Busy status bit is reset. If there is not a match but there is valid ID CRC, an interrupt is generated, and Seek Error Status bit (Status bit 4) is set and the Busy status bit is reset. If there is a match but not a valid CRC, the CRC error status bit is set (Status bit 3), and the next encountered ID field is read from the disk for the verification operation. If an ID field with a valid CRC cannot be found after four revolutions of the disk, the FD179X terminates the operation and sends an interrupt, (INTRQ).

The Step, Step-In, and Step-Out commands contain an Update flag (U). When U = 1, the track register is updated by one for each step. When U = 0, the track register is not updated.

On the 1795/7 devices, the SSO output is not affected during Type 1 commands, and an internal side compare does not take place when the (V) Verify Flag is on.



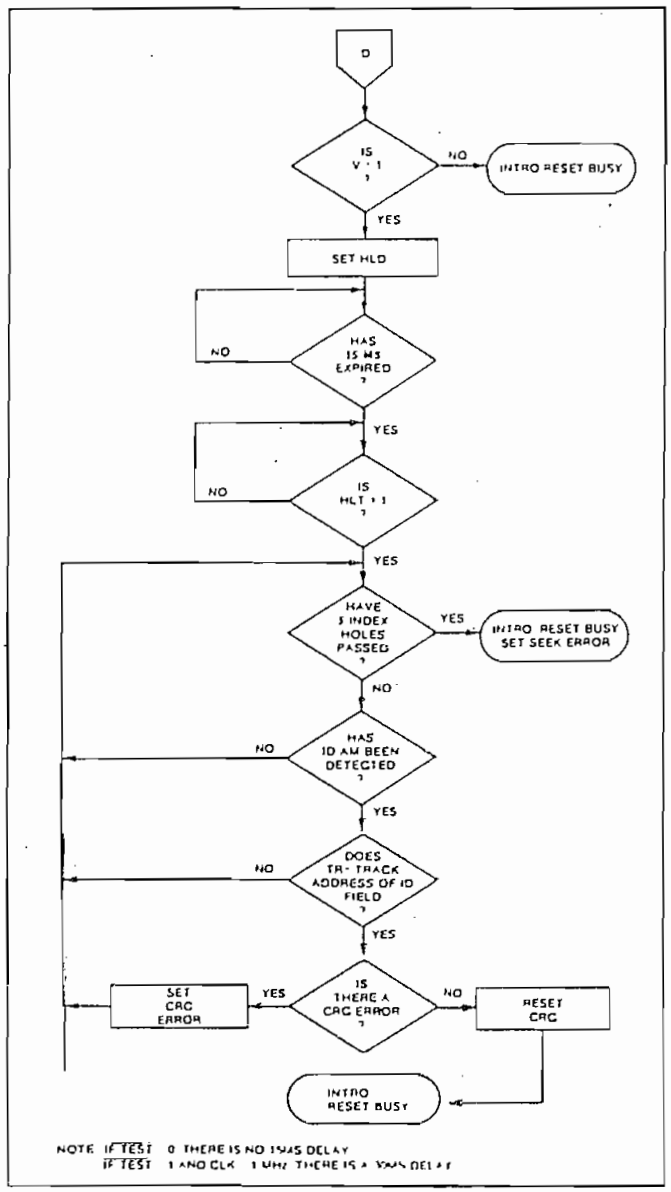
TYPE I COMMAND FLOW



TYPE I COMMAND FLOW

RESTORE (SEEK TRACK 0)

Upon receipt of this command the Track 00 ($\overline{TROO}$) input is sampled. If $\overline{TROO}$ is active low indicating the Read-Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TROO}$ is not active low, stepping pulses (pins 15 to 16) at a rate specified by the r_1r_0 field are issued until the $\overline{TROO}$ input is activated. At this time the Track Register is loaded with zeroes and an interrupt is generated. If the $\overline{TROO}$ input does not go active low after 255 stepping pulses, the FD179X terminates operation, interrupts, and sets the Seek error status bit. A verification operation takes place if the V flag is set. The h bit allows the head to be loaded at the start of command. Note that the Restore command is executed when $\overline{MR}$ goes from an active to an inactive state.



TYPE I COMMAND FLOW

SEEK

This command assumes that the Track Register contains the track number of the current position of the Read-Write head and the Data Register contains the desired track number. The FD179X will update the Track register and issue stepping pulses in the appropriate direction until the contents of the Track register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP

Upon receipt of this command, the FD179X issues one stepping pulse to the disk drive. The stepping motor direction is the same as in the previous step command. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. If the u flag is on, the Track Register is updated. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-IN

Upon receipt of this command, the FD179X issues one stepping pulse in the direction towards track 76. If the u flag is on, the Track Register is incremented by one. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-OUT

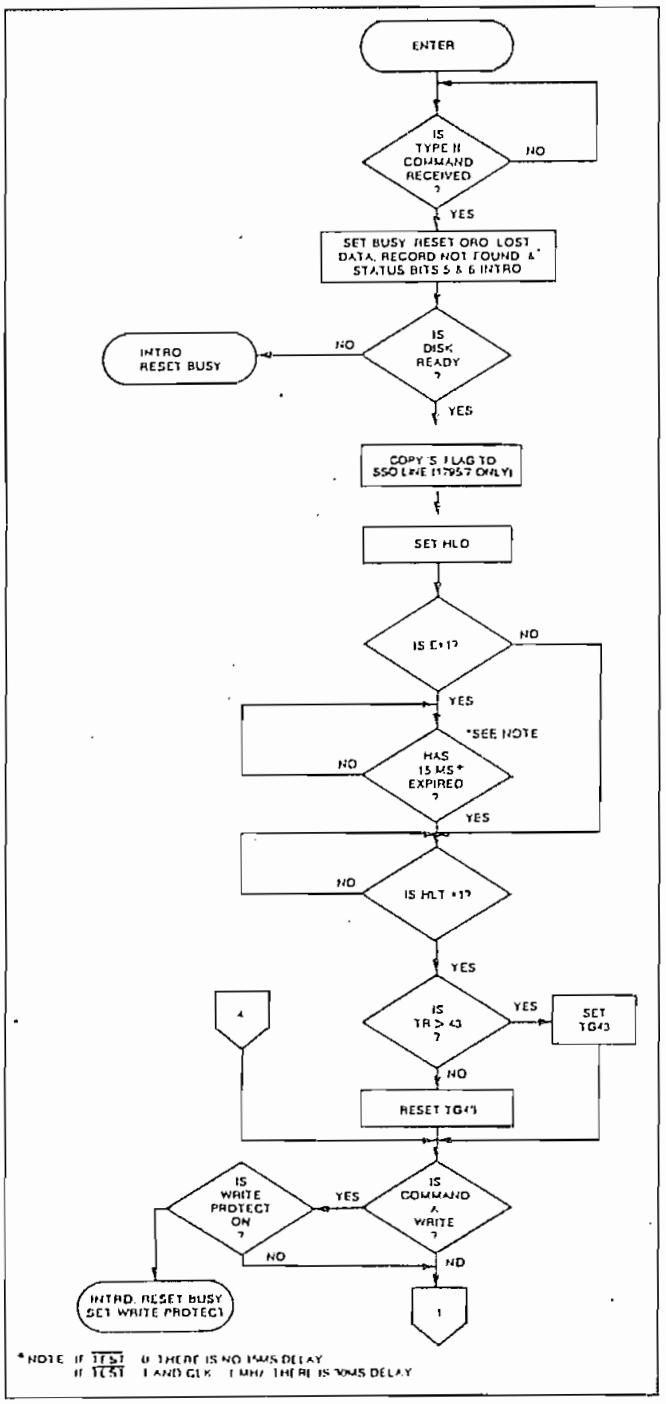
Upon receipt of this command, the FD179X issues one stepping pulse in the direction towards track 0. If the u flag is on, the Track Register is decremented by one. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

TYPE II COMMANDS

The Type II Commands are the Read Sector and Write Sector commands. Prior to loading the Type II Command into the Command Register, the computer must load the Sector Register with the desired sector number. Upon receipt of the Type II command, the busy status Bit is set. If the E flag = 1 (this is the normal case) HLD is made active and HLT is sampled after a 15 msec delay. If the E flag is 0, the head is loaded and HLT sampled with no 15 msec delay. The ID field and Data Field format are shown on page 13.

When an ID field is located on the disk, the FD179X compares the Track Number on the ID field with the Track Register. If there is not a match, the next en-

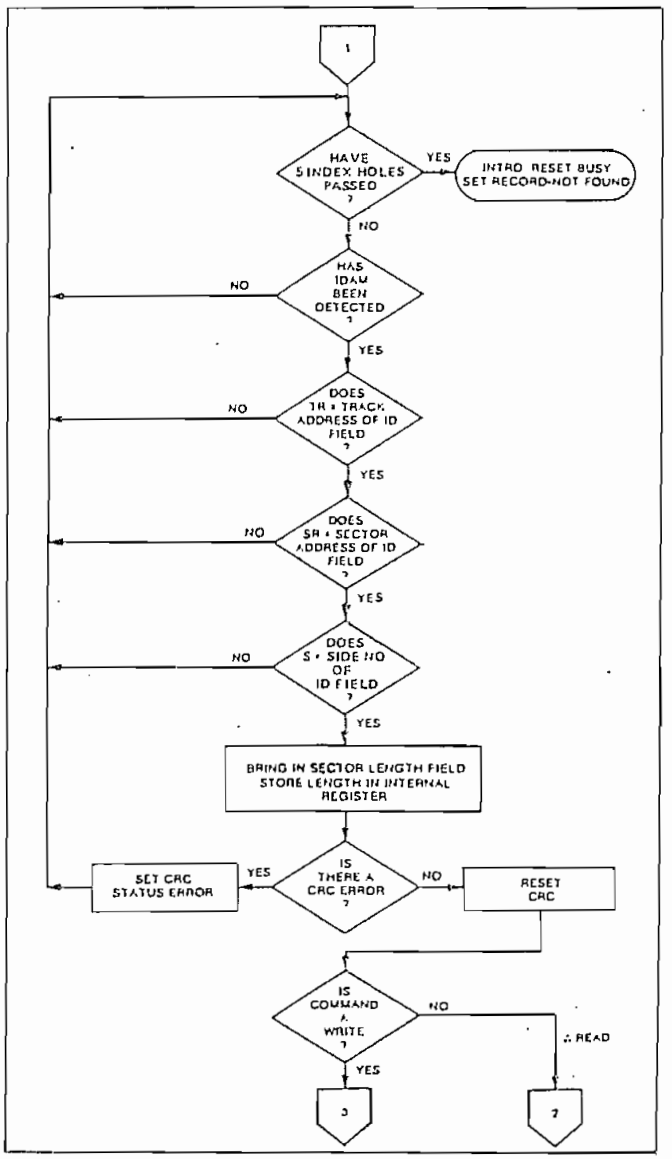
made. If there was a match, the Sector Number of the ID field is compared with the Sector Register. If there is not a Sector match, the next encountered ID field is read off the disk and comparisons again made. If the ID field CRC is correct, the data field is then located and will be either written into, or read from depending upon the command. The FD179X must find an ID field with a Track number, Sector number, side number, and CRC within four revolutions of the disk; otherwise, the Record not found status bit is set (Status bit 3) and the command is terminated with an interrupt.



TYPE II COMMAND

Sector Length Field (hex)	Number of Bytes in Sector (decimal)
00	128
01	256
02	512
03	1024

Each of the Type II Commands contains an (m) flag which determines if multiple records (sectors) are to be read or written, depending upon the command. If m = 0, a single sector is read or written and an interrupt is generated at the completion of the command. If m = 1, multiple records are read or written with the sector register internally updated so that an address verification can occur on the next record. The FD179X will continue to read or write multiple records and update the sector register until the sector regis-



TYPE II COMMAND

ter exceeds the number of sectors on the track or until the Force Interrupt command is loaded into the Command Register, which terminates the command and generates an interrupt.

If the Sector Register exceeds the number of sectors on the track, the Record-Not-Found status bit will be set.

The Type II commands also contain side select compare flags. When C = 0, no side comparison is made. When C = 1, the LSB of the side number is read off the ID Field of the disk and compared with the contents of the (S) flag. If the S flag compares with the side number recorded in the ID field, the 179X continues with the ID search. If a comparison is not made within 5 index pulses, the interrupt line is made active and the Record-Not-Found status bit is set.

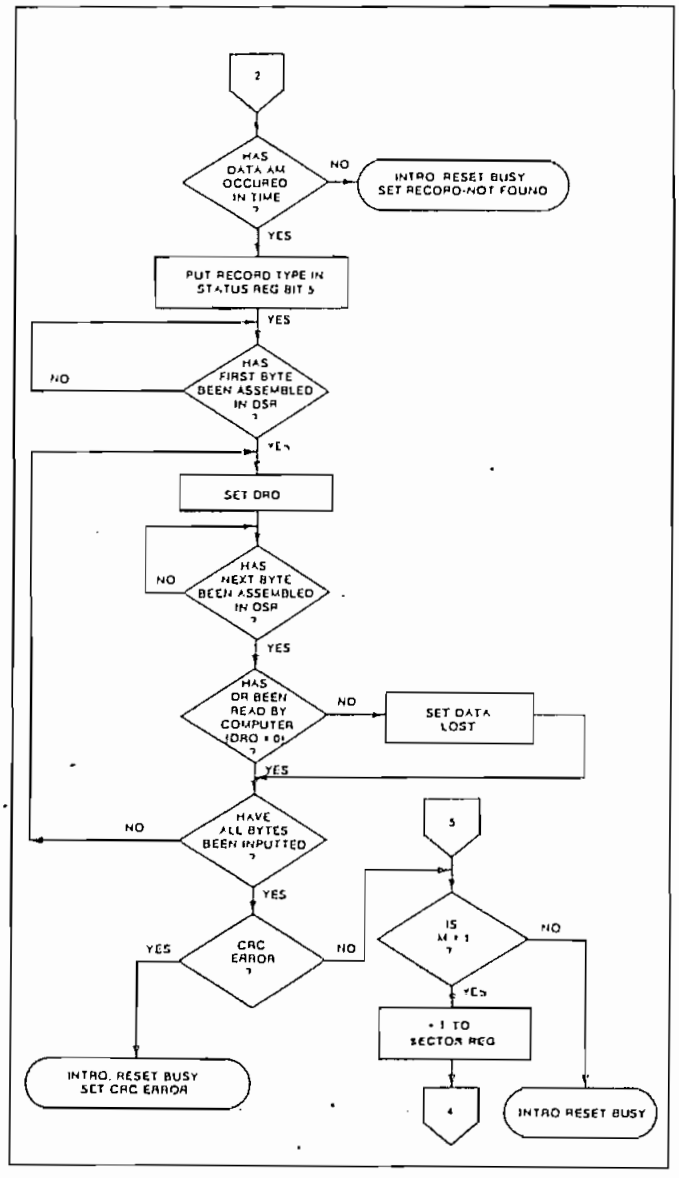
The 1795/7 READ SECTOR and WRITE SECTOR commands include a 'b' flag. The 'b' flag, in conjunction with the sector length byte of the ID Field, allows different byte lengths to be implemented in each sector. For IBM compatability, the 'b' flag should be set to a one. The 'b' flag allows direct ...

and is set or reset at the beginning of the command, dependent upon the value of this flag.

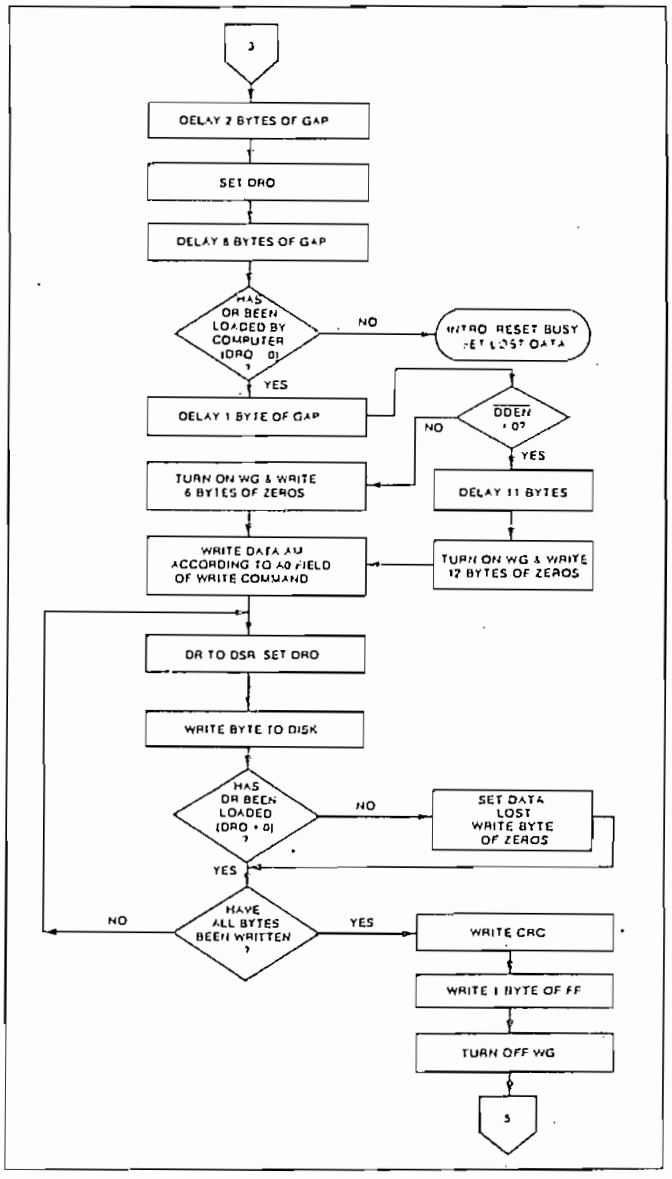
READ SECTOR

Upon receipt of the Read Sector command, the head is loaded, the Busy status bit set, and when an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, the data field is presented to the computer. The Data Address Mark of the data field must be found within 30 bytes in single density and 43 bytes in double density of the last ID field CRC byte; if not, the Record Not Found status bit is set and the operation is terminated.

When the first character or byte of the data field has been shifted through the DSR, it is transferred to the DR, and DRQ is generated. When the next byte is accumulated in the DSR, it is transferred to the DR and another DRQ is generated. If the Computer has not read the previous contents of the DR before a new character is transferred that character is lost and



TYPE II COMMAND



TYPE II COMMAND

tinues until the complete data field has been inputted to the computer. If there is a CRC error at the end of the data field, the CRC error status bit is set, and the command is terminated (even if it is a multiple record command).

At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Bit 5) as shown below:

STATUS BIT 5	
1	Deleted Data Mark
0	Data Mark

WRITE SECTOR

Upon receipt of the Write Sector command, the head is loaded (HLD active) and the Busy status bit is set. When an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, a DRQ is generated. The FD179X counts off 11 bytes in single density and 22 bytes in double density from the CRC field and the Write Gate (WG) output is made active if the DRQ is serviced (i.e., the DR has been loaded by the computer). If DRQ has not been serviced, the command is terminated and the Lost Data status bit is set. If the DRQ has been serviced, the WG is made active and six bytes of zeros in single density and 12 bytes in double density are then written on the disk. At this time the Data Address Mark is then written on the disk as determined by the a_0 field of the command as shown below:

a_0	Data Address Mark (Bit 0)
1	Deleted Data Mark
0	Data Mark

The FD179X then writes the data field and generates DRQ's to the computer. If the DRQ is not serviced in time for continuous writing the Lost Data Status Bit is set and a byte of zeros is written on the disk. The command is not terminated. After the last data byte has been written on the disk, the two-byte CRC is computed internally and written on the disk followed by one byte of logic ones in FM or in MFM. The WG output is then deactivated.

TYPE III COMMANDS

READ ADDRESS

Upon receipt of the Read Address command, the head is loaded and the Busy Status Bit is set. The disk, and the six data bytes of the ID field are assembled and transferred to the DR, and a DRQ is generated for each byte. The six bytes of the ID field are shown below:

TRACK ADDR	SIDE NUMBER	SECTOR ADDRESS	SECTOR LENGTH	CRC 1	CRC 2
1	2	3	4	5	6

Although the CRC characters are transferred to the computer, the FD179X checks for validity and the CRC error status bit is set if there is a CRC error. The Track Address of the ID field is written into the sector register. At the end of the operation an interrupt is generated and the Busy Status is reset.

READ TRACK

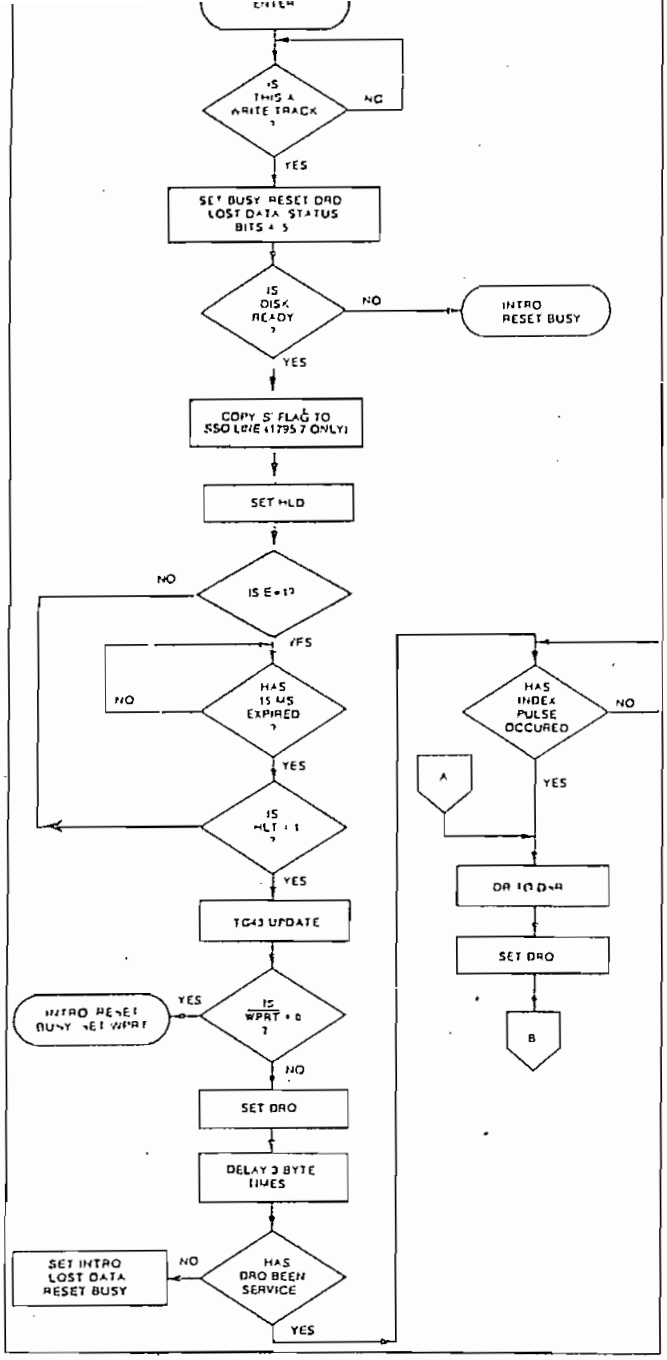
Upon receipt of the Read Track command, the head is loaded and the Busy Status bit is set. Reading starts with the leading edge of the first encountered index pulse and continues until the next index pulse. As each byte is assembled it is transferred to the Data Register and the Data Request is generated for each byte. No CRC checking is performed. Gaps are included in the input data stream. The accumulation of bytes is synchronized to each Address Mark encountered. Upon completion of the command, the interrupt is activated. RG is not activated during the Read Track Command. An internal side compare is not performed during a Read Track.

WRITE TRACK

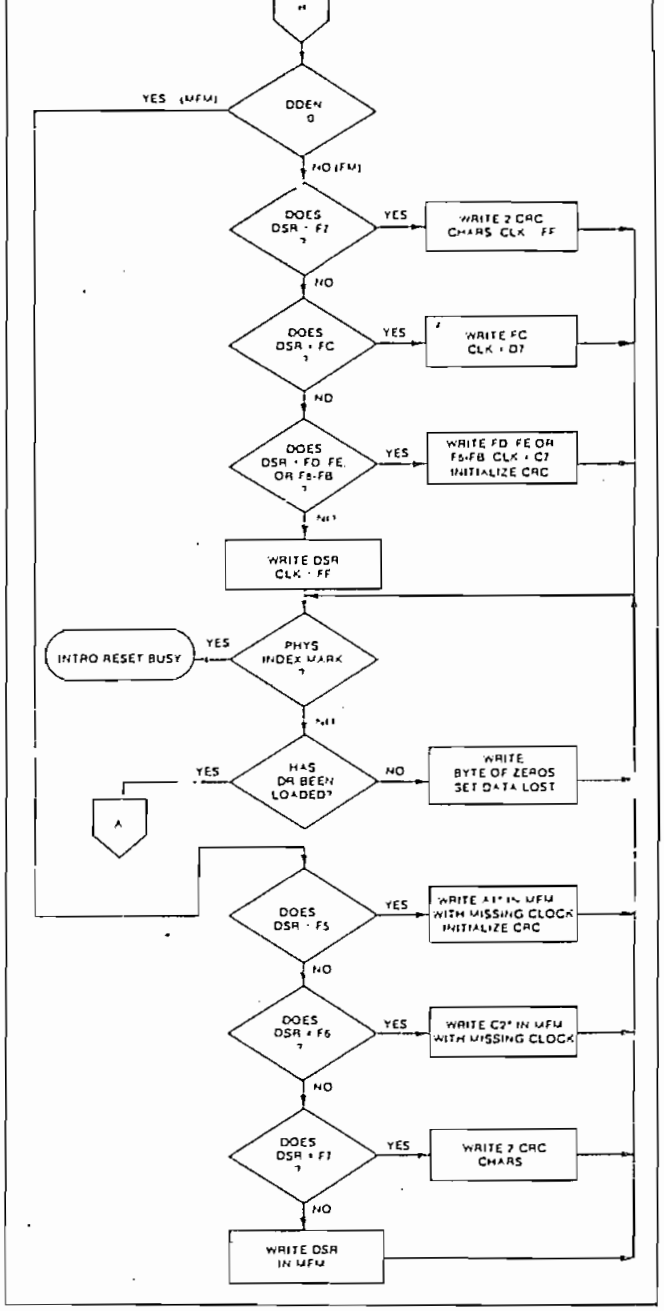
Upon receipt of the Write Track command, the head is loaded and the Busy Status bit is set. Writing starts with the leading edge of the first encountered index pulse and continues until the next index pulse, at which time the interrupt is activated. The Data Request is activated immediately upon receiving the command, but writing will not start until after the first byte has been loaded into the Data Register. If the DR has not been loaded by the time the index pulse is encountered the operation is terminated making the device Not Busy, the Lost Data Status Bit is set, and the Interrupt is activated. If a byte is not present in the DR when needed, a byte of zeros is substituted. Address Marks and CRC characters are written on the disk by detecting certain data byte patterns in the outgoing data stream as shown in the table below. The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR in FM or by receipt of F5 in MFM.

GAP III	ID AM	TRACK NUMBER	SIDE NUMBER	SECTOR NUMBER	SECTOR LENGTH	CRC 1	CRC 2	GAP II	DATA AM	DATA FIELD	CRC 1	CRC 2
ID FIELD										DATA FIELD		

In MFM only, IDAM and DATA AM are preceded by three bytes of A1 with clock transition between bits 4 and 5 missing.



TYPE III COMMAND WRITE TRACK



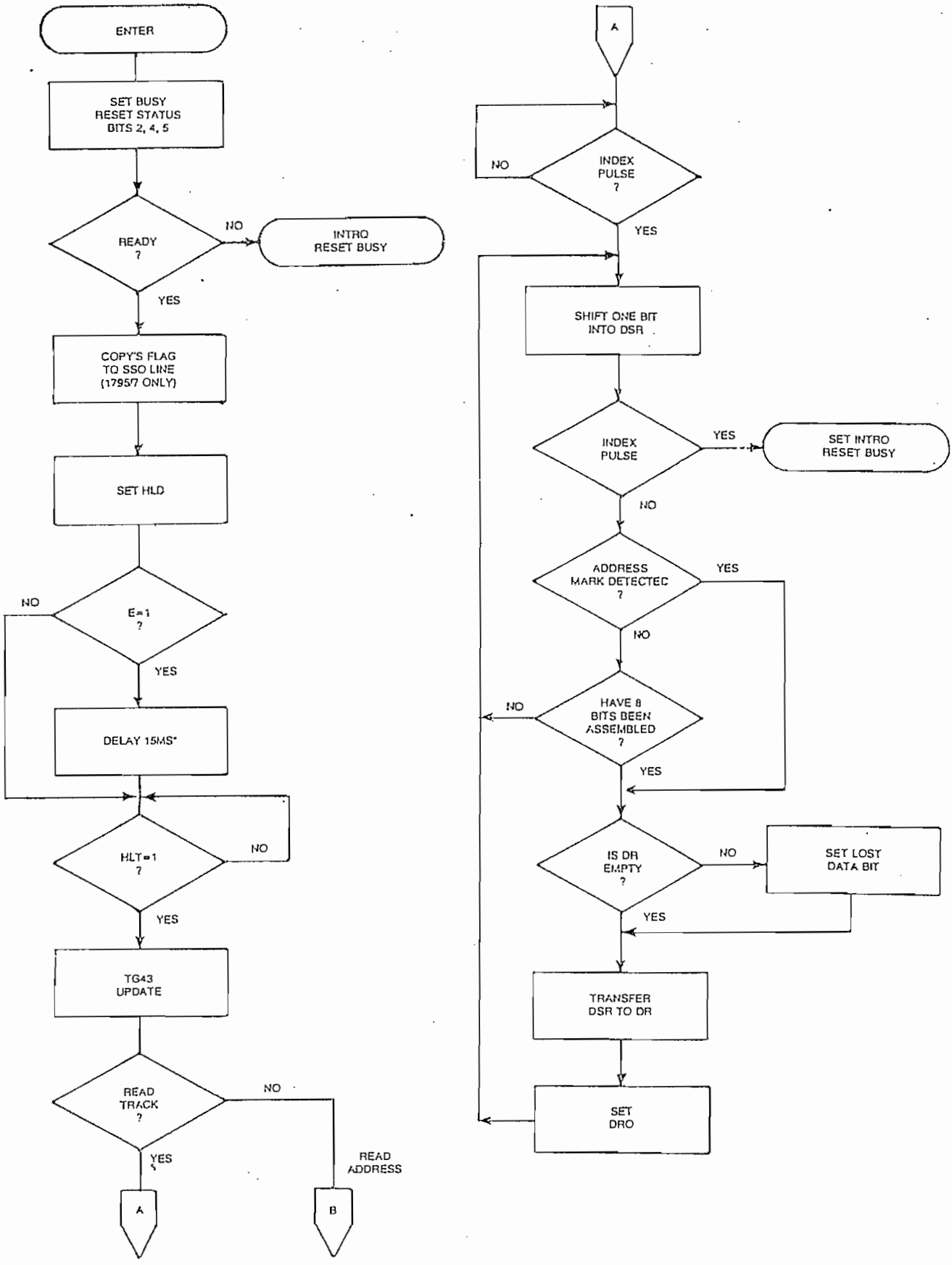
TYPE III COMMAND WRITE TRACK

CONTROL BYTES FOR INITIALIZATION

DATA PATTERN IN DR (HEX)	FD179X INTERPRETATION IN FM ($\overline{DDEN}$ = 1)	FD1791/3 INTERPRETATION IN MFM ($\overline{DDEN}$ = 0)
00 thru F4	Write 00 thru F4 with CLK = FF	Write 00 thru F4, in MFM
F5	Not Allowed	Write A1* in MFM, Preset CRC
F6	Not Allowed	Write C2** in MFM
F7	Generate 2 CRC bytes	Generate 2 CRC bytes
F8 thru FB	Write F8 thru FB, Clk = C7, Preset CRC	Write F8 thru FB, in MFM
FC	Write FC with Clk = D7	Write FC in MFM
FD	Write FD with Clk = FF	Write FD in MFM
FE	Write FE, Clk = C7, Preset CRC	Write FE in MFM
FF	Write FF with Clk = FF	Write FF in MFM

*Missing clock transition between bits 4 and 5

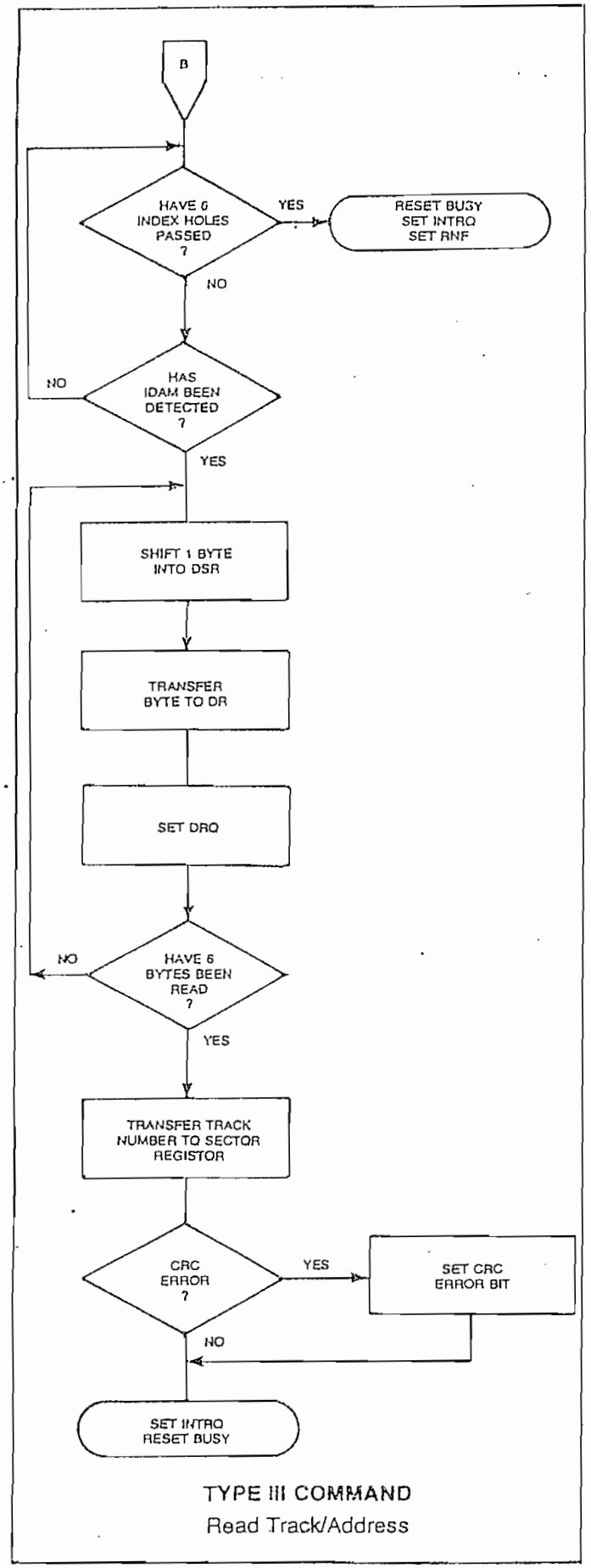
**Missing clock transition between bits 3 & 4



*If TEST = 0 NO DELAY

If TEST = 1 and CLK = 1 MHZ, 30 MS DELAY

TYPE III COMMAND
Read Track/Address



TYPE III COMMAND

Read Track/Address

TYPE IV COMMAND

FORCE INTERRUPT

This command can be loaded into the command register at any time. If there is a current command under execution (Busy Status Bit set), the command will be terminated and an interrupt will be generated when the condition specified in the I_0 through I_3 field is detected. The interrupt conditions are shown below:

I_0 = Not-Ready-To-Ready Transition
I_1 = Ready-To-Not-Ready Transition
I_2 = Every Index Pulse
I_3 = Immediate Interrupt (requires reset, see Note)

NOTE: If $I_0 - I_3 = 0$, there is no interrupt generated but the current command is terminated and busy is reset. *This is the only command that will enable the immediate interrupt to clear on a subsequent Load Command Register or Read Status Register.*

STATUS DESCRIPTION

Upon receipt of any command, except the Force Interrupt command, the Busy Status bit is set and the rest of the status bits are updated or cleared for the new command. If the Force Interrupt Command is received when there is a current command under execution, the Busy status bit is reset, and the rest of the status bits are unchanged. If the Force Interrupt command is received when there is not a current command under execution, the Busy Status bit is reset and the rest of the status bits are updated or cleared. In this case, Status reflects the Type I commands.

The format of the Status Register is shown below:

(BITS)							
7	6	5	4	3	2	1	0
S7	S6	S5	S4	S3	S2	S1	S0

Status varies according to the type of command executed as shown in Table 6.

(Refer to section on Type III commands for flow diagrams.)

Formatting the disk is a relatively simple task when operating programmed I/O or when operating under Formatting the disk is accomplished by positioning the R/W head over the desired track number and issuing the Write Track command. Upon receipt of the Write Track command, the FD179X raises the Data Request signal. At this point in time, the user loads the data register with desired data to be written on the disk. For every byte of information to be written on the disk, a data request is generated. This sequence continues from one index mark to the next index mark. Normally, whatever data pattern appears in the data register is written on the disk with a normal clock pattern. However, if the FD179X detects a data pattern of F5 thru FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation. For instance, in FM an FE pattern will be interpreted as an ID address mark (DATA-FE, CLK-C7) and the CRC will be initialized. An F7 pattern will generate two CRC characters in FM or MFM. As a consequence, the patterns F5 thru FE must not appear in the gaps, data fields, or ID fields. Also, CRC's must be generated by an F7 pattern.

Disks may be formatted in IBM 3740 or System 34 formats with sector lengths of 128, 256, 512, or 1024 bytes.

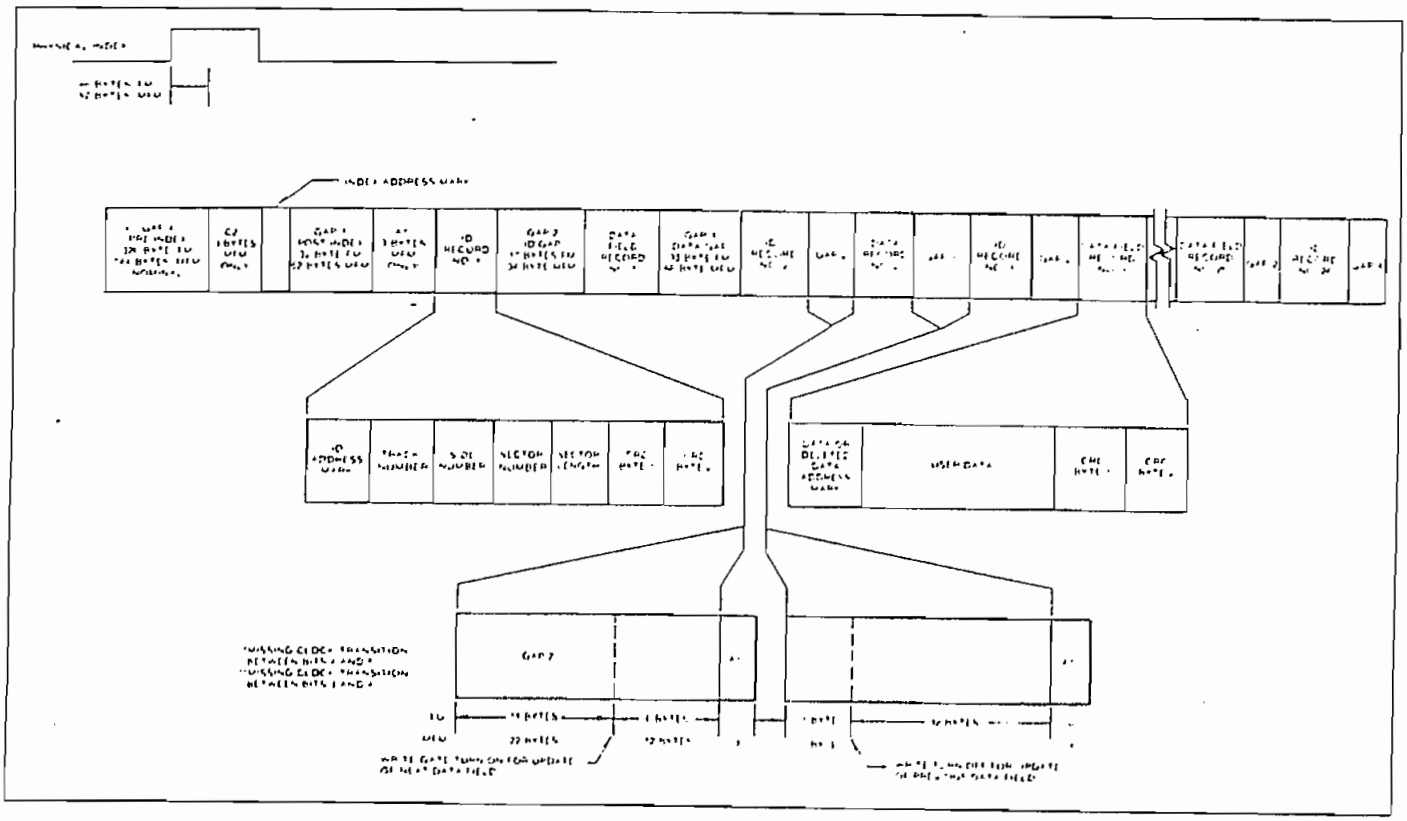
Shown below is the IBM single-density format with 128 bytes/sector. In order to format a diskette, the user must issue the Write Track command, and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
40	FF (or 00)[1]
6	00
1	FC (Index Mark)
26	FF (or 00)
6	00
1	FE (ID Address Mark)
1	Track Number
1	Side Number (00 or 01)
1	Sector Number (1 thru 1A)
1	00
1	F7 (2 CRC's written)
11	FF (or 00)
6	00
1	FB (Data Address Mark)
128	Data (IBM uses E5)
1	F7 (2 CRC's written)
27	FF (or 00)
247**	FF (or 00)

*Write bracketed field 26 times

**Continue writing until FD179X interrupts out. Approx. 247 bytes.

1-Optional '00' on 1795/7 only.



IBM TRACK FORMAT

256 BYTES/SECTOR

Shown below is the IBM dual-density format with 256 bytes/sector. In order to format a diskette the user must issue the Write Track command and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
80	4E
12	00
3	F6
1	FC (Index Mark)
50*	4E
12	00
3	F5
1	FE (ID Address Mark)
1	Track Number (0 thru 4C)
1	Side Number (0 or 1)
1	Sector Number (1 thru 1A)
1	01
1	F7 (2 CRCs written)
22	4E
12	00
3	F5
1	FB (Data Address Mark)
256	DATA
1	F7 (2 CRCs written)
54	4E
598**	4E

*Write bracketed field 26 times
**Continue writing until FD179X interrupts out. Approx. 598 bytes.

Variations in the IBM format are possible to a limited extent if the following requirements are met: sector size must be a choice of 128, 256, 512, or 1024 bytes; gap size must be according to the following table. Note that the Index Mark is not required by the 179X. The minimum gap sizes shown are that which is required by the 179X, with PLL lock-up time, motor speed variation, etc.; adding additional bytes.

	FM	MFM
Gap I	16 bytes FF	32 bytes 4E
Gap II	11 bytes FF	22 bytes 4E
*	6 bytes 00	12 bytes 00 3 bytes A1
Gap III	10 bytes FF	24 bytes 4E 3 bytes A1
**	4 bytes 00	8 bytes 00
Gap IV	16 bytes FF	16 bytes 4E

*Byte counts must be exact.
**Byte counts are minimum, except exactly 3 bytes of A1 must be written.

ELECTRICAL CHARACTERISTICS

MAXIMUM RATINGS

V_{DD} With Respect to V_{SS} (Ground) =15 to −0.3V

Max. Voltage to Any Input With Respect to V_{SS} =15 to −0.3V

Operating Temperature 0°C to 70°C

Storage Temperature −55°C to +125°C

V_{DD} = ID ma Nominal V_{CC} = 35 ma Nominal

OPERATING CHARACTERISTICS (DC)

TA.= 0°C to 70°C, V_{DD} = + 12V ± .6V, V_{SS} = OV, V_{CC} = + 5V ± .25V

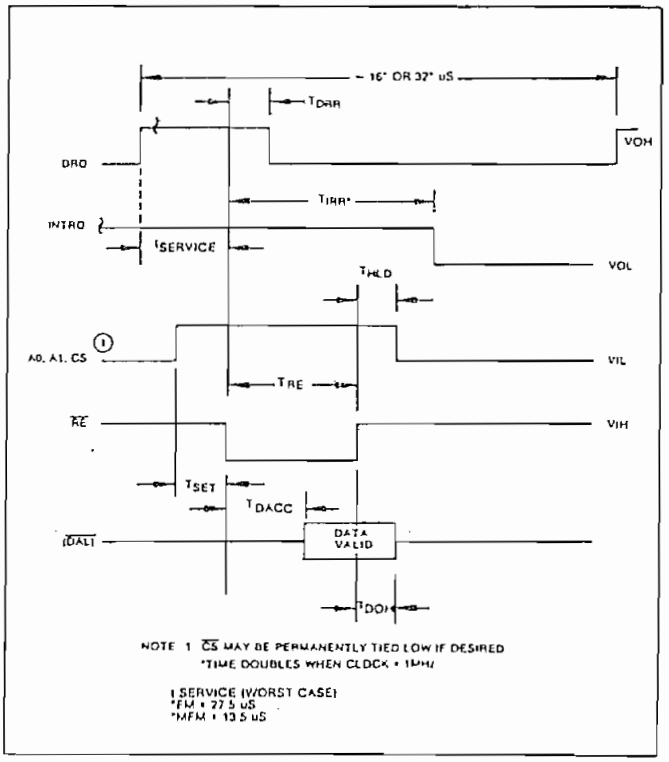
SYMBOL	CHARACTERISTIC	MIN.	MAX.	UNITS	CONDITIONS
I_{IL}	Input Leakage		10	μA	$V_{IN} = V_{DD}$
I_{OL}	Output Leakage		10	μA	$V_{OUT} = V_{DD}$
V_{IH}	Input High Voltage	2.6		V	
V_{IL}	Input Low Voltage		0.8	V	
V_{OH}	Output High Voltage	2.8		V	$I_O = -100\ \mu A$
V_{OL}	Output Low Voltage		0.45	V	$I_O = 1.6$ mA
P_D	Power Dissipation		0.5	W	

TIMING CHARACTERISTICS

T_A = 0°C to 70°C, V_{DD} = + 12V ± .6V, V_{SS} = 0V, V_{CC} = +5V ± .25V

READ ENABLE TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{RE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{RE}$	10			nsec	
TRE	$\overline{RE}$ Pulse Width	400			nsec	C_L = 50 pf
TDRR	DRQ Reset from $\overline{RE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{RE}$		500	3000	nsec	See Note 5
TDACC	Data Access from $\overline{RE}$			350	nsec	C_L = 50 pf
TDOH	Data Hold From $\overline{RE}$	50		150	nsec	C_L = 50 pf



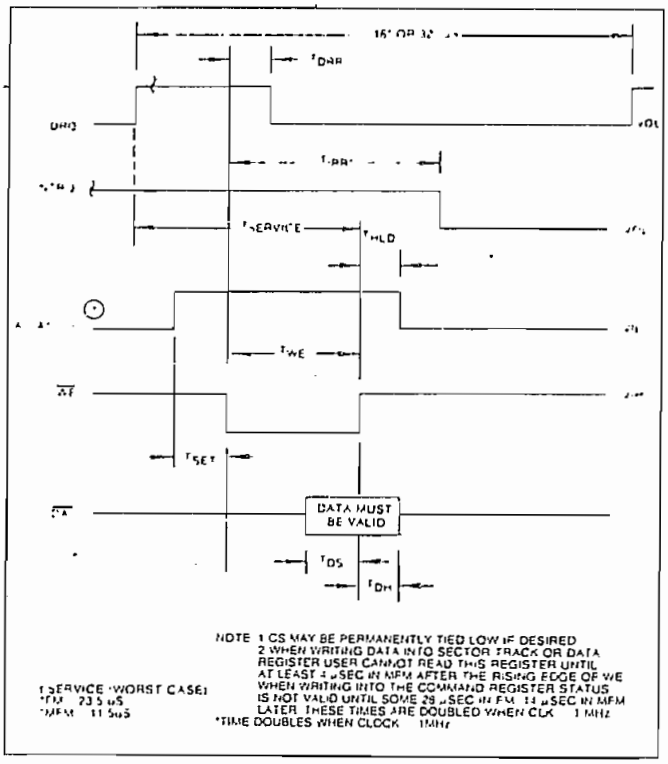
READ ENABLE TIMING

WRITE ENABLE TIMING

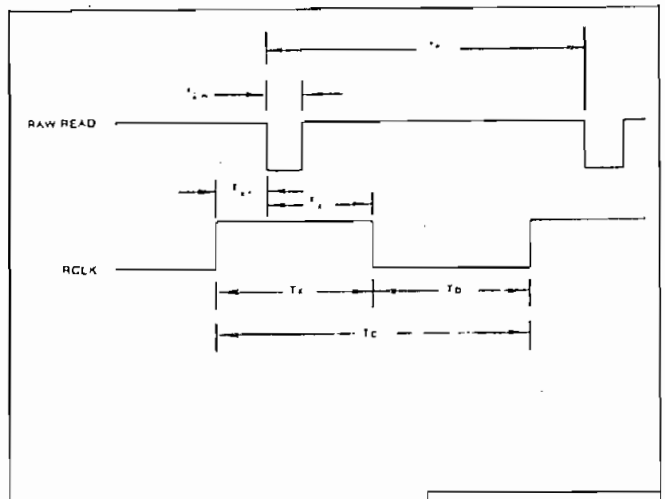
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TSET	Setup ADDR & CS to $\overline{WE}$	50			nsec	
THLD	Hold ADDR & CS from $\overline{WE}$	10			nsec	
TWE	$\overline{WE}$ Pulse Width	350			nsec	
TDRR	DRQ Reset from $\overline{WE}$		400	500	nsec	
TIRR	INTRQ Reset from $\overline{WE}$		500	3000	nsec	See Note 5
TDS	Data Setup to $\overline{WE}$	250			nsec	
TDH	Data Hold from $\overline{WE}$	70			nsec	

INPUT DATA TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Tpw	$\overline{Raw\ Read}$ Pulse Width	100	200		nsec	See Note 1
tbc	$\overline{Raw\ Read}$ Cycle Time		1500		nsec	1800 ns @ 70°C
Tc	RCLK Cycle Time		1500		nsec	1800 ns @ 70°C
Tx_1	RCLK hold to $\overline{Raw\ Read}$	40			nsec	See Note 1
Tx_2	$\overline{Raw\ Read}$ hold to RCLK	40			nsec	



WRITE ENABLE TIMING

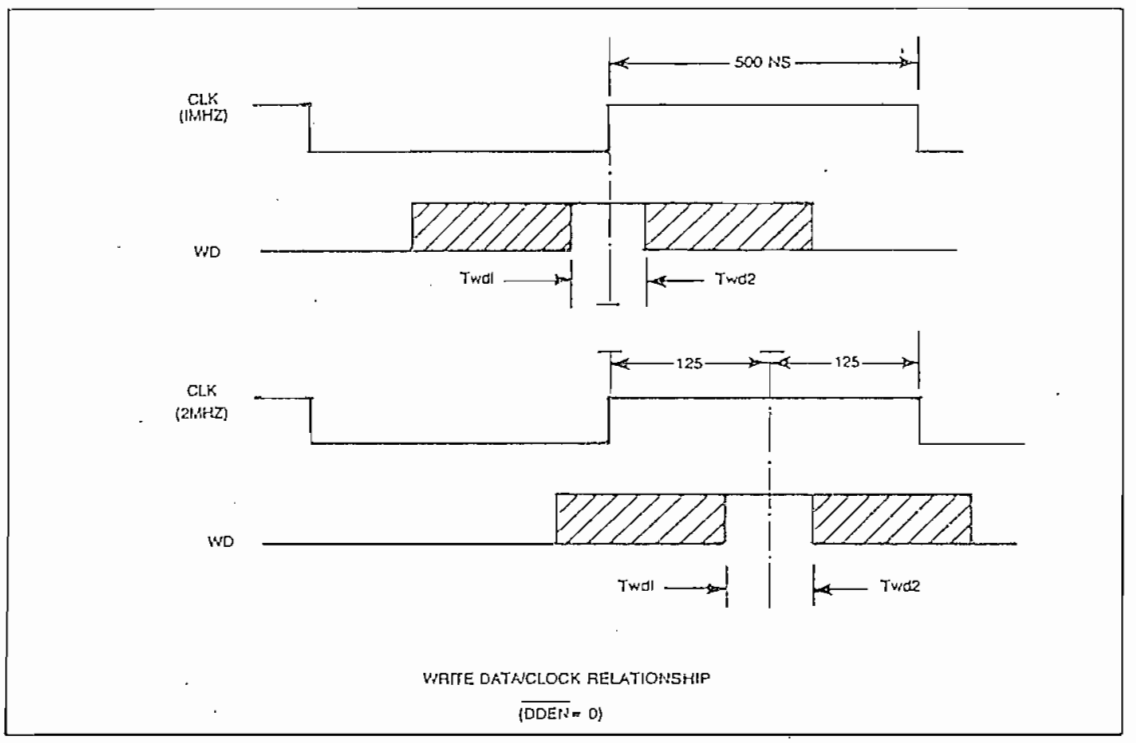


DISKETTE	MODE	$\overline{DDEN}$	CLK	NOMINAL T_x	NOMINAL T_b	NOMINAL T
8"	MFM	0	2 MHz	1 μs	1 μs	2 μs
8"	FM	1	2 MHz	2 μs	2μs	4 μs
5"	MFM	0	1 MHz	2 μs	2μs	4 μs
5"	FM	1	1 MHz	4 μs	4 μs	8 μs

INPUT DATA TIMING

WRITE DATA TIMING: (ALL TIMES DOUBLE WHEN CLK = 1 MHz)

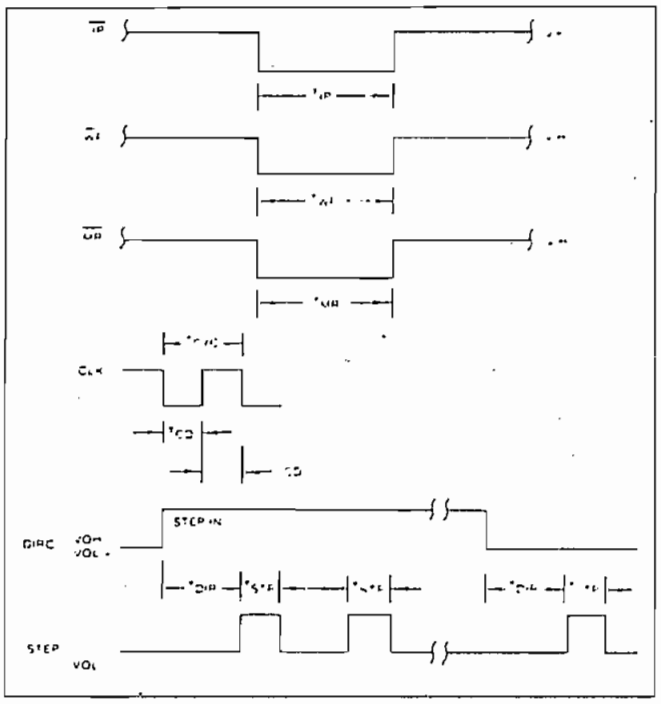
SYMBOL	CHARACTERISTICS	MIN.	TYP.	MAX.	UNITS	CONDITIONS
Twp	Write Data Pulse Width	450	500	550	nsec	FM
		150	200	250	nsec	MFM
Twg	Write Gate to Write Data		2		μsec	FM
			1		μsec	MFM
Tbc	Write data cycle Time		2,3, or 4		μsec	±CLK Error
Ts	Early (Late) to Write Data	125			nsec	MFM
Th	Early (Late) From Write Data	125			nsec	MFM
Twf	Write Gate off from WD		2		μsec	FM
			1		μsec	MFM
Twd1	WD Valid to Clk	100			nsec	CLK=1 MHZ
		50			nsec	CLK=2 MHZ
Twd2	WD Valid after CLK	100			nsec	CLK=1 MHZ
		30			nsec	CLK=2 MHZ



WRITE DATA TIMING

MISCELLANEOUS TIMING:

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
TCD_1	Clock Duty (low)	230	250	20000	nsec	
TCD_2	Clock Duty (high)	200	250	20000	nsec	
TSTP	Step Pulse Output	2 or 4			μsec	See Note 5
TDIR	Dir Setup to Step		12		μsec	± CLK ERROR
TMR	Master Reset Pulse Width	50			μsec	
TIP	Index Pulse Width	10			μsec	See Note 5
TWF	Write Fault Pulse Width	10			μsec	



MISCELLANEOUS TIMING

NOTES:

1. Pulse width on RAW READ (Pin 27) is normally 100-300 ns. However, pulse may be any width if pulse is entirely within window. If pulse occurs in both windows, then pulse width must be less than 300 ns for MFM at CLK = 2 MHz and 600 ns for FM at 2 MHz. Times double for 1 MHz.
2. A PPL Data Separator is recommended for 8" MFM.
3. tbc should be 2 μs, nominal in MFM and 4 μs nominal in FM. Times double when CLK = 1 MHz.
4. RCLK may be high or low during RAW READ (Polarity is unimportant).
5. Times double when clock = 1 MHz.

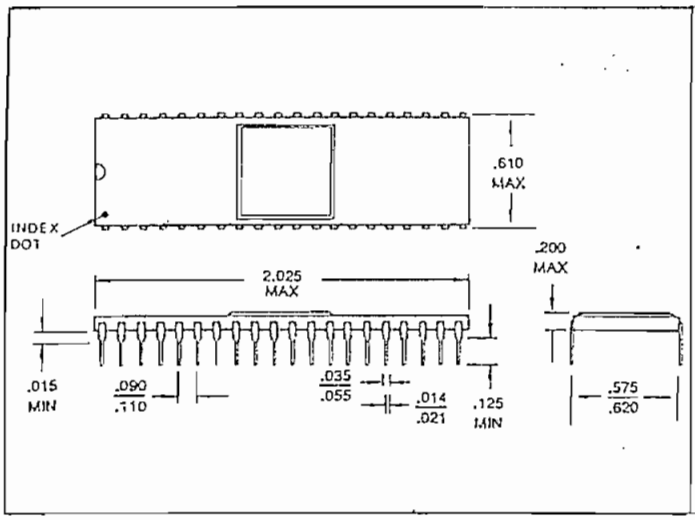
BIT	ALL TYPE I COMMANDS	READ ADDRESS	READ SECTOR	READ TRACK	WRITE SECTOR	WRITE TRACK
S7	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY
S6	WRITE PROTECT	0	0	0	WRITE PROTECT	WRITE PROTECT
S5	HEAD LOADED	0	RECORD TYPE	0	WRITE FAULT	WRITE FAULT
S4	SEEK ERROR	RNF	RNF	0	RNF	0
S3	CRC ERROR	CRC ERROR	CRC ERROR	0	CRC ERROR	0
S2	TRACK 0	LOST DATA	LOST DATA	LOST DATA	LOST DATA	LOST DATA
S1	INDEX	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

STATUS FOR TYPE I COMMANDS

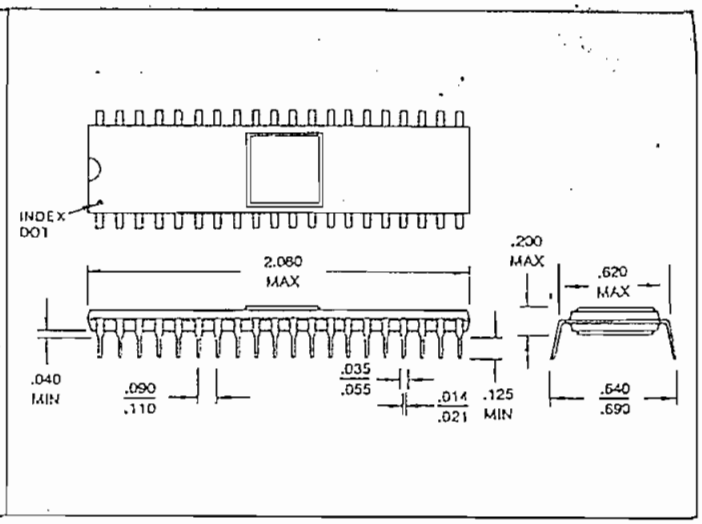
BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset it indicates that the drive is ready. This bit is an inverted copy of the Ready input and logically 'ored' with MR.
S6 PROTECTED	When set, indicates Write Protect is activated. This bit is an inverted copy of $\overline{WRPT}$ input.
S5 HEAD LOADED	When set, it indicates the head is loaded and engaged. This bit is a logical "and" of HLD and HLT signals.
S4 SEEK ERROR	When set, the desired track was not verified. This bit is reset to 0 when updated.
S3 CRC ERROR	CRC encountered in ID field.
S2 TRACK 00	When set, indicates Read/Write head is positioned to Track 0. This bit is an inverted copy of the $\overline{TROO}$ input.
S1 INDEX	When set, indicates index mark detected from drive. This bit is an inverted copy of the $\overline{IP}$ input.
S0 BUSY	When set command is in progress. When reset no command is in progress.

STATUS FOR TYPE II AND III COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset, it indicates that the drive is ready. This bit is an inverted copy of the Ready input and 'ored' with MR. The Type II and III Commands will not execute unless the drive is ready.
S6 WRITE PROTECT	On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated.
S5 RECORD TYPE/ WRITE FAULT	On Read Record: It indicates the record-type code from data field address mark. 1 = Deleted Data Mark. 0 = Data Mark. On any Write: It indicates a Write Fault. This bit is reset when updated.
S4 RECORD NOT FOUND (RNF)	When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated.
S3 CRC ERROR	If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated.
S2 LOST DATA	When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when updated.
S1 DATA REQUEST	This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated.
S0 BUSY	When set, command is under execution. When reset, no command is under execution.



FD179XA-02 CERAMIC PACKAGE



FD179XB-02 PLASTIC PACKAGE

This is a preliminary specification with tentative device parameters and may be subject to change after final product characterization is completed.

WESTERN DIGITAL
CORPORATION

3128 REDHILL AVENUE, BOX 2180
NEWPORT BEACH, CA 92663 (714) 557-3550,TWX 910-595-1139

A P E N D I C E 2

" CITAS BIBLIOGRAFICAS "

CITAS BIBLIOGRAFICAS

(1) PEATMAN John B., MICROCOMPUTER-BASED DESIGN, Editorial Mc Graw-Hill, 1977, p. 180.

(2) PEATMAN John B., MICROCOMPUTER-BASED DESIGN, Editorial Mc Graw-Hill, 1977, p. 182.

(3) PEATMAN John B., MICROCOMPUTER-BASED DESIGN, Editorial Mc Graw-Hill, 1977, p. 183.

(4) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.2.

(5) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.1.

(6) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.4.

(7) TEXAS INSTRUMENTS, TMS 9900 FLOPPY DISK CONTROLLER, 1977, p.24.

(8) WESTERN DIGITAL, FD179X-02 APPLICATION NOTES, 1980, p.3.

(9) SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977, p.23.

(10) PEATMAN John B., MICROCOMPUTER NASED DESIGN, Editorial Mc Graw-Hill, 1977, p.187.

(11) RALSTON Anthony, ENCYCLOPEDIA OF COMPUTER SCIENCE, Editorial VAN NOSTRAND REINHOLD, 1976, p.383.

(12) KRUTZ Ronald L., MICROPROCESSORS AND LOGIC DESIGN, Editorial John Wiley & Sons, 1980, p.418.

(13) WAKERLY John, Error Detecting Codes, Self-Checking Circuits and Applications, Editorial Elsevier North-Holland, 1978, p.32.

(14) KRUTZ Ronald L., MICROPROCESSORS AND LOGIC DESIGN, Editorial John Wiley & Sons, 1980, p.365.

(15) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.1,2.

(16) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.8.

(17) TEXAS INSTRUMENTS, The TTL Data Book, 1976, p. 6-82.

(18) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.5.

(19) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.6,8,9.

(20) TEXAS INSTRUMENTS, The TTL Data Book, 1976, p.7-123.

(21) TEXAS INSTRUMENTS, The TTL Data Book, 1976, p.6 - 81.

(22) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.14.

(23) VECTOR ELECTRONIC COMPANY, 80-81 CATALOG, 1980, p.IV-34.

(24) SHUGART ASSOCIATES, SA400 Minifloppy Diskette Storage Drive, 1977, p.9.

(25) WESTERN DIGITAL, 1981 Product Handbook, 1981, p.361.

(26) POPULAR ELECTRONICS, Agosto 1980, p.61.

BIBLIOGRAFIA

PEATMAN John B., MICROCOMPUTER-BASED DESIGN, Editorial Mc Graw-Hill, 1977.

SHUGART ASSOCIATES, SA400 MINIFLOPPY DISKETTE STORAGE DRIVE, 1977.

TEXAS INSTRUMENTS, TMS 9900 FLOPPY DISK CONTROLLER, 1977.

WESTERN DIGITAL, FD79X-02 APPLICATION NOTES, 1980.

RALSTON Anthony, ENCICLOPEDIA OF COMPUTER SCIENCE, Editorial VAN NOSTRAND REINHOLD, 1976.

KRUTZ Ronald L., MICROPROCESSORS AND LOGIC DESIGN, Editorial John Wiley & Sons, 1980.

WAKERLY John, ERROR DETECTING CODES, SELF-CHECKING CIRCUITS AND APPLICATIONS, Editorial Elsevier North-Holland, 1978.

TEXAS INSTRUMENTS, THE TTL DATA BOOK, 1976.

VECTOR ELECTRONIC COMPANY, 80-81 CATALOG, 1980.

WESTERN DIGITAL, 1981 PRODUCT HANDBOOK, 1981.

MOTOROLA SEMICONDUCTOR PRODUCTS INC., MICROPROCESSOR APPLICATIONS MANUAL, 1975.

TAUB Herbert & SCHILLING Donald, DIGITAL INTEGRATED ELECTRONICS, 1977.

SIPPL Charles & SIPPL Roger, COMPUTER DICTIONARY & HANDBOOK, 1980.

MOTOROLA SEMICONDUCTOR PRODUCTS INC., MC6843 FLOPPY DISK CONTROLLER, 1978.

MOTOROLA SEMICONDUCTOR PRODUCTS INC., MEK6800D2 MANUAL, 1977.