

T621.383
9937
1989

C. UNIVERSIDAD TECNICA FEDERICO SANTA MARIA
CENTRO EDUCACIONAL "REY BALDUINO DE BELGICA"
TALCAHUANO

PROTOTIPO DE UN KIT PARA DESARROLLO.

TRABAJO PARA OPTAR AL TITULO DE
TECNICO UNIUERSITARIO EN ELECTRONICA

PROFESORES GUIAS :

GONZALO HORMAZABAL A.

SERGIO BRICEÑO M.

ALUMNO : RODRIGO GUERRA OPAZO.

T621.383
9937
1989
C-1

1688
1989

ESTE TRABAJO LO DEDICO CON MUCHO
CARIÑO A MIS PADRES, PATRICIA Y
PROFESORES QUE SI SU COLABORACION
NO HABRIA LLEGADO A FELIZ TERMINO.

INDICE

INTRODUCCION

-TITULO PRETENCIOSO.....	1
-PUNTO DE PARTIDA.....	1
-QUE LOGRAR.....	1
-ALCANCES INTERESANTES.....	2

CAPITULO I.- SDK-85

-CARACTERISTICAS DEL HARDWARE.....	3
-CARACTERISTICAS DEL SOFTWARE.....	3

CAPITULO II.- VISION GLOBAL DEL TRABAJO A REALIZAR

-MEDIO DE COMUNICACION.....	5
-HARDWARE.....	6
-SOFTWARE.....	6

CAPITULO III.- CIRCUITO FINAL JUSTIFICACION

-CIRCUITO IMPLEMENTADO.....	7
-8251 USART.....	9
-6264 RAM.....	13
-2732 EPROM.....	14
-74LS138 DECODIFICADORES.....	14
-BUFFER RS 232-C 1489 1488.....	14
-RELOJ 74LS90 74LS76.....	15

CAPITULO IV.- COMUNICACION SERIE

-CONECTOR.....	17
-NORMA RS 232-C.....	19

CAPITULO V.- SOFTWARE IMPLEMENTADO

-SUBRRUTINAS DEL PROGRAMA MONITOR DEL SDK-85.....	26
-FUNCIONES DEL CP/M UTILIZADAS.....	26
-SOFTWARE PARA EL SDK-85.....	30
-SOFTWARE PARA EL COMPUTADOR ADVANTAGE.....	30

CAPITULO VI.- APLICACIONES

-EJEMPLO PRACTICO.....	32
-IDEAS PARA IMPLEMENTAR.....	34

CAPITULO VII.- CONCLUSIONES

-AUTOCRITICA.....	35
-------------------	----

BIBLIOGRAFIA

APENDICE A: PROGRAMAS FUENTES

APENDICE B: ESPECIFICACIONES TECNICAS

INTRODUCCION

TITULO PRETENCIOSO

Mirándolo a grosso modo sí se puede decir que el título propuesto es pretencioso puesto que lo que se piensa lograr es un poco menos. Por disponibilidad de tiempo y horarios se consideraría aceptable lograr una comunicación eficiente entre el kit SDK-85 de INTEL y un sistema computacional mayor para poder usufructuar de algunos servicios de este último.

PUNTO DE PARTIDA

La forma de desarrollar este trabajo se puede considerar como punto de partida para desarrollar otro del mismo tipo, con otro procesador de mayores prestaciones en algún área determinada.

Esto no quiere decir que éste sea el procedimiento más correcto, pero para este caso particular se considera el más adecuado, puesto que en el sistema mayor escogido ya existen depuradores y ensambladores para los nemónicos del 8085 y también algunas características técnicas ya establecidas.

QUE LOGRAR

Si consideramos las características del kit de INTEL SDK-85, nos daremos cuenta de que la mayor desventaja que presenta es la de no tener un método permanente de almacenamiento de información; por este motivo y otros más de carácter técnico, se hace necesario la búsqueda de comunicación de éste, con un sistema mayor que posea medios de almacenamiento en lo posible magnético, además de contar con la ayuda de SOFTWARE especializado para la elaboración de programas compatibles con el procesador del SDK-85.

Para lograr este propósito debemos incrementar los servicios y/o capacidades del micro. En otras palabras, incrementar el HARDWARE y el SOFTWARE del micro en la siguiente forma:

Referente al HARDWARE será necesario agregar:

- Puerta serie
- Memoria RAM
- Memoria ROM

Referente al SOFTWARE, éste deberá ser capaz de manejar:

- Puerta serie.
- Transmisión y recepción de datos.
- Manejo y almacenamiento de datos.
- Protocolos necesarios para una eficiente comunicación.

ALCANCES INTERESANTES

Se considera muy interesante que la persona que piense en hacer uso de este pequeño sistema, tenga un básico dominio de algún tipo de ensamblador o quizás más específicamente, del par M80 - L80 que se utilizaron para los ejemplos presentados aquí.

Al hablar de un básico dominio se considera el poder ensamblar en forma adecuada y poder direccionar la información adecuadamente según los requerimientos impuestos.

Otro punto interesante es la correcta elección del cable de comunicación, puesto que el sistema está configurado de manera que con o sin conector funcione indistintamente.

Como dato práctico cualquier anomalía dentro del funcionamiento del computador ADVANTAGE, en especial el encuentro de los diskettes R/O por el sistema, es completamente solucionable por medio de ^C (control C) ;esto se debe exclusivamente a anomalías leves en el funcionamiento de éste a la fecha.

CAPITULO I. - SDK-85

CARACTERISTICAS DEL HARDWARE

El SDK-85 es un kit estructurado en base a la CPU 8085 de INTEL, que fue adquirido con fines didácticos para acercar más al estudiante a la tecnología imperante en ese tiempo.

Este kit contaba en forma original con:

- Sólo 256 bytes de memoria RAM; la cual podría ser incrementada en 256 bytes más, al agregar otro chip.
- 2 k. bytes de memoria ROM también con la posibilidad de ser incrementada en 2k.bytes más, al agregar otro chip.
- Como medio de comunicación con el medio exterior utiliza un pequeño teclado y displays de 7 segmentos.
- 6 puertos paralelo programables bidireccionales con la posibilidad de aumentar este número a 12.

Al hablar de "incremento de memoria (RAM o ROM, del kit) con solo agregar un chip", se quiere hacer notar otra característica física: originalmente los chip de memoria incluyen también puertos programables I/O, un timer programable y además el proceso de demultiplexeo dirección data entre las líneas ADO-AD7; de éstos, el 8155 contiene memoria RAM, el 8355 contiene memoria ROM y el 8755 contiene memoria PROM.

Como se ve, el kit está bien equipado en cuanto a puertos paralelo y sólo tiene la posibilidad, entre comillas, de una salida serie conformada en forma de loop de corriente cuyo baud rate es bajo y determinado, controlado exclusivamente por software aprovechando los terminales SID y SOD (Serial Input Data y Serial Output Data respectivamente de la CPU).

CARACTERISTICAS DEL SOFTWARE

En cuanto a SOFTWARE, cuenta con un pequeño programa monitor capaz de interactuar con el HARDWARE presente y con futuras expansiones; sus comandos son muy limitados, pero son los necesarios para cumplir con su propósito.

Dentro de las características del programa monitor se encuentran:

- Cuando se escribe en memoria éste lee la misma posición de memoria para comprobar el dato con lo que se acaba de escribir y así detectar cualquier anomalía - que no exista memoria o que ella esté fallando.

- Este maneja la escritura en los displays como una subrutina, ubicada en 02B7 HEX a la cual se le informa dónde se encuentra la data, en qué campo se quiere escribir; la data debe estar adecuadamente codificada.

- Otra subrutina importante es la de tiempo de retardo, ubicada en 05F1 HEX a la cual se entrega el registro DE cargado de acuerdo al tiempo de retardo.

CAPITULO II.- VISION GLOBAL DEL TRABAJO A REALIZAR

MEDIO DE COMUNICACION

Como ya habíamos dicho, el principal objetivo a lograr es un medio de comunicación eficiente entre el computador ADVANTAGE y el kit SDK-85, a fin de usufructuar del software disponible en el computador para desarrollar programas o rutinas en el kit SDK-85.

Este medio de comunicación debe ser rápido, seguro y, por supuesto, eficiente. Dentro de las posibilidades para lograr esto se tiene comunicación paralelo y serie.

Comunicación paralelo:

Esta consiste en transmitir una palabra (8 bit) simultáneamente de un equipo a otro; este tipo de comunicación es rápido y además usa un protocolo muy singular.

El equipo a transmitir avisa al otro por una línea de protocolo (strobe); el otro equipo lee la palabra y comunica al primero el término de lectura por una línea de protocolo (busy).

El cable empleado está compuesto por:

- 8 líneas de data
- 1 línea de tierra de señal
- 1 línea de tierra de chasis o protección
- líneas de protocolo. (una o más)

Comunicación serie:

En este tipo de comunicación la palabra (8 bit) se transmite bit a bit.

Existen dos tipos :Sincrónica y Asincrónica.

Comunicación serie sincrónica:

En este modo de comunicación se acompaña a la palabra a transmitir por una señal de reloj; esto quiere decir que la lectura de cada bit se hace en un estado determinado de la señal de reloj.

Además, es frecuente encontrar uno o más bytes de sincronismo cada cierta cantidad de información transmitida.

Comunicación serie asincrónica:

Este modo de comunicación no emplea señal de reloj, pero agrega una cierta cantidad determinada de bit a la palabra a transmitir de manera de sincronizar transmisor y receptor.

De estos tipos de comunicación se ha optado por la comunicación serie asincrónica por considerarse mas didáctica para el trabajo y la cual se explicará en detalle mas adelante.

HARDWARE

Por ende, para el uso de este tipo de comunicación, se deberá implementar en el SDK-85 el hardware necesario: un medio para transformar la información paralelo (CPU) a serie, adaptadores de niveles de la señal y forma de fijar el baud rate.

Una vez operativo el medio de comunicación, se hace necesario aumentar la memoria RAM para almacenar la data transferida.

SOFTWARE

Se hace necesario dejar en forma permanente el SOFTWARE requerido para manejar la puerta serie y la forma de almacenamiento de los datos, ya sea en la transmisión o la recepción. La solución más evidente para esto es instalar una memoria EPROM de capacidad suficiente para sostener este SOFTWARE y el de futuras ampliaciones.

Resumiendo en este punto podemos decir que necesitamos:

Memoria RAM: Para almacenar la data (información, programas) transferida para su posterior utilización.

Memoria EPROM: Indispensable para el almacenamiento del SOFTWARE de transferencia de data permanente en micro.

Medio de Comunicación SERIE/ASINCRONICA: En esta área existen unos cuantos chip especializados como la USART, UART, USIA, ASIA.

Se indicó en el dibujo como "-*" a los LED por no tener la posibilidad de representarlos en la forma convencional.

A continuación se detallan los componentes de mayor importancia utilizados:

- U1 = 74LS138 (decodificador)
- U2 = 8251 (usart)
- U3 = 1489 (buffer RS 232-C)
- U4 = 74LS90 (contador de decadas)
- U5 = 74LS138 (decodificador)
- U6 = 1488 (buffer RS 232-C)
- U7 = 74LS76 (flip flop JK)
- U8 = 74LS00 (compuertas NAND)
- U9 = 2732 (EPROM 4K x 8 Bit)
- U10 = 6264 (RAM 8K x 8 Bit)

Luego se explicará detalladamente los criterios y razonamientos empleados para la elección de cada uno de los circuitos integrados usados.

USART 8251

Este es un CI especializado para la comunicación serie. Su función es ayudar a la CPU a transmitir y/o recibir data en forma serie. Puede configurarse para trabajar en forma sincrónica o en forma asincrónica, en este segundo caso se usará en forma asincrónica, por lo cual se hace necesario generar un reloj para fijar el baud rate, el cual tiene que ser muy estable para no tener pérdida de sincronismo; no hay que olvidar que este tipo de comunicación se sincroniza por medio del START BIT y el STOP BIT, por ende si hay diferencia entre el reloj de transmisión y el de recepción podrá también haber problemas en la transmisión o recepción.

Este reloj a la vez de ser muy estable es conveniente que tenga la opcionalidad de cambiar de frecuencia según sea el baud rate requerido para una determinada transmisión o recepción; para este caso específico se tomó como referencia el reloj con el cual trabaja la CPU (3.072 Mhz), el cual está implementado en base a un cristal. Hay que tener presente también que la señal de reloj que se aplica a la USART es dividida interiormente por uno de tres posibles factores, estos son: 1 - 16 - 64; el factor de división se determina al inicializar la USART.

La elección de los factores externos y obtención de los mismos se verán en el próximo punto a tratar.

A continuación se entregan los baud rate posibles de obtener con las divisiones internas y externas:

FACTOR EXTERNO	FACTOR INTERNO	BOUN RATE
10	16	19200
20	16	9600
10	64	4800
20	64	2400
40	64	1200

Los factores de división externos son seleccionados manualmente por el usuario, la selección se hace en cuatro pin de dos bases distintas ubicadas bajo y hacia la derecha mirando el micro en su posición normal de trabajo.

La selección se hace utilizando un común y conectando éste a los otros pins con lo que se logra los siguientes factores de división de izquierda a derecha :10 , 20 , 40.

El factor de división 5 se ha eliminado para lograr los baud rate requeridos con los otros tres factores.

INICIALIZACION DE LA USART

El terminal #12 de la USART se denomina CONTROL - DATOS y funciona conjuntamente con las instrucciones de escritura y de lectura de la USART ;al poner un cero lógico en este terminal, la USART queda en condiciones de recibir o transmitir información; si, por el contrario se le aplica un uno lógico a este terminal, las operaciones de lectura o de escritura que se realicen serán exclusivamente de comandos, ya sea para inicializar a la USART (escritura) o para revisar estados (lectura).

Para inicializar la USART (en el modo asincrónico) hay que entregarle dos palabras, una de modo y la otra de comando.El formato es el mismo para éstas que para la palabra de estado y se detalla a continuación:

Cada byte : D7 - D6 - D5 - D4 - D3 - D2 - D1 - D0

El formato de la palabra de modo es el siguiente:

D1 - D0 : Estos bits están asociados a los factores de división y a la elección de trabajo de la USART ya sea sincrónica o asincrónica.La siguiente tabla muestra lo anterior.

**FACTORES
BAUD RATE**

	D 1	D 0
MODO SINCRONICO	0	0
DIVISION X 1	0	1
DIVISION X 16	1	0
DIVISION X 64	1	1

D3 - D2 : Estos bits están asociados al largo del carácter como se muestra en la tabla siguiente:

LARGO DE
PALABRAS

	D3	D2
5 BITS	0	0
6 BITS	0	1
7 BITS	1	0
8 BITS	1	1

D4 : Este bit está asociado a la habilitación o deshabilitación de la paridad, si éste es un "1" lógico la paridad queda habilitada, pero si es un "0" lógico la paridad queda deshabilitada.

D5 : Este bit está asociado a la generación y verificación del bit de paridad; si éste es "0" entonces será impar y si es "1" será par.

D7 - D6 : estos dos bits están asociados a la cantidad de STOP BIT o tiempos de STOP BIT en relación al reloj de transmisión - recepción ; se especifica en la tabla siguiente:

BITS DE STOP

	D7	D6
INVALIDADO	0	0
1 BIT	0	1
1.5 BIT	1	0
2 BIT	1	1

El formato de la palabra de comando es el siguiente:

D0 : Este bit al ser "1" log. permite la habilitación de la parte transmisora de la USART y si es "0" log. deshabilita ésta.

D1 : Con este bit se tiene control directo sobre el terminal de salida DTR (terminal de datos listo, pin # 24) de la USART; si éste es "1" log. fuerza el terminal DTR a "0" log. y por el contrario si este bit es "0" log. fuerza el terminal DTR a "1" log.

D2 : Este bit al ser "1" log. habilita la parte receptora de la USART , por el contrario, si es "0" log, la deshabilita.

D3 : Este bit se denomina SEND BREAK CHARACTER y su traducción podría ser "mandato de corte del carácter", al ser éste "1" log. se detiene la transmisión y el terminal TXD (transmisión de datos pin # 19) toma el estado "0" log. ; si este bit es "1" log. la transmisión de datos es normal.

D4 : Este bit se denomina ERROR RESET (reset de flags de error); si es "1" log. resetea todos los flags internos de USART y si éste es "0" log. todos los flags actúan normalmente.

D5 : Este bit tiene control directo sobre el terminal de salida RTS (requerimiento de mandato pin # 23) de la USART ; si éste es "1" log. fuerza el terminal RTS a "0" log. y si, por el contrario este bit es "0" log. fuerza el terminal RTS a "1" log.

D6 : Este bit se denomina INTERNAL RESET (reset interno de la USART) ; si éste es "1" log. se produce un reset interno total de la USART quedando ésta en condiciones de ser reinicializada nuevamente y si éste es "0" log. se tiene normal operación.

D7 : Este bit es de uso en el modo sincrónico de la USART; para nuestro caso no tiene mayor importancia; se denomina ENTER HUNT MODE (entrar en el modo de caza) y se refiere a la caza o captura del byte de sincronismo. Un "1" log. en este bit habilita esta captura y un "0" log. la deshabilita.

Una vez concluida la estraga de estos dos bytes la USART está inicializada, cualquier otra palabra que se envíe posteriormente a estas dos se tomará como palabra de comando.

RESET DE LA USART

Existen dos formas de realizar esto:

- Por hardware.
- por software.

Reset por Hardware : El pin # 21 de ésta es el terminal de reset , al ser llevado a "1" log. se produce el reset total de la USART quedando en condiciones de ser inicializada.

Reset por Software : Como se vio anteriormente, es posible resetear la USART por medio de una palabra de comando (40 Hex), esto es cuando la USART ya ha sido inicializada.

Ahora, en el caso contrario, el fabricante recomienda dos alternativas (éstas deben ser entregadas como palabras de control):

- enviar tres ceros sucesivos y luego un 40 HEX.
- enviar dos 80 HEX. sucesivos y luego un 40 HEX.

LECTURA DE LA PALABRA DE ESTADO

Considerando el mismo formato que para las palabras de modo y comando se tiene:

D0 : El estado de este bit es el mismo que tiene la terminal TX RDY (transmisor listo pin #15) de la USART.

D1 : El estado de este bit es el mismo que tiene el terminal RX RDY (receptor listo pin #14) de la USART.

D2 : El estado de este bit es el mismo que tiene el terminal TXE (transmisor vacío pin #18) de la USART.

D3 : Este bit contiene el estado del flag de error de paridad; su nivel activo es "1" log. La activación de este flag se produce al detectar error de paridad en la recepción de una palabra y no altera el normal funcionamiento de la USART.

D4 : Este bit contiene el estado del flag de error de sobrecarga; su nivel activo es "1" log. La activación de este flag se produce cuando la CPU no lee un carácter disponible y la USART carga otro sobre el anterior, y su activación no altera el normal funcionamiento de la USART.

D5 : Este bit contiene el estado del flag de error de enmascaramiento que tiene única relación con el modo sincrónico de la USART.

D6 : El estado de este bit es el mismo que tiene el terminal SYNDET (detector de sincronismo pin #16) de la USART; éste es de uso exclusivo del modo sincrónico.

D7 : El estado de este bit es el mismo que tiene el terminal de entrada DSR (listo el aparato de poner datos pin #22) de la USART ;esta entrada es utilizada por lo general en la comunicación con modem.

RAM 6264

Esta es una memoria RAM estática de 8 K bytes por 8 bit con un tiempo de acceso max de 100 n Seg.

Se optó por ésta, debido a su capacidad de almacenamiento, además por ser estática no hay problema alguno en su instalación (refresco).

EPROM 2732

Esta es una memoria EPROM de 4 K bytes por 8 bit con un tiempo de acceso max de 200 n Seg.

Se optó por ésta por su capacidad y, además, por ser memoria EPROM es posible borrarla en caso de querer hacer cambios o agregar más rutinas.

DECODIFICADOR 74LS138

Se optó por este decodificador, puesto que se encontró conveniente dejar disponibles otras direcciones, aparte de las utilizadas.

Las direcciones utilizadas como puertas son las siguientes:

80 Hex. Puerta I/O correspondiente a datos de la USART.
81 Hex. Puerta I/O correspondiente a comandos de la USART.

Las direcciones utilizadas como posiciones de memoria son las siguientes:

8000 - 8FFF Hex. Direcciones de la memoria EPROM.
A000 - BFFF Hex. Direcciones de la memoria RAM.

Quedando disponibles las siguientes direcciones :

Puertas.- 90 - A0 - B0 - C0 - D0 - E0 - F0.
Memoria.- 9000 - 9FFF Hex.
 C000 - DFFF Hex.
 E000 - FFFF Hex.

BUFFER RS 232-C 1488 1489

A este punto ya hemos elegido un tipo de comunicación, serie, y dentro de esta una modalidad de trabajo, asincrónica. Ahora falta estandarizar este tipo de comunicación y la norma que se encuentra más adecuada a nuestra elección es la norma RS 232-C.

Esta se explicará en detalle más adelante, por ahora sólo veremos los dos chip que permiten adecuar los niveles TTL del

micro SDK-85 a esta norma.

1488 : se denomina QUAR MDTL LINE DRIVE (cuatro manejadores de línea). Este es normalmente utilizado en esta norma, puesto que cumple con las exigencias de ésta.

Dentro de este chip encontramos; haciendo una comparación con TTL :

- Una compuerta NAND de una entrada.
- Tres compuertas NAND de dos entradas.

Su función es adaptar los niveles TTL del micro a los de la norma.

1489 : Se denomina QUAR MDTL LINE RECEIVERS (cuatro receptores de línea). Este es normalmente utilizado en esta norma, puesto que cumple con las exigencias de ésta.

Dentro de este chip encontramos, haciendo una comparación con TTL :

- Cuatro compuertas NAND de una entrada.

Su función es adaptar los niveles de la norma a los niveles TTL del micro.

Este buffer en particular posee según la literatura, un terminal de control de histéresis, pero personalmente diría que es para determinar el estado de la salida si el terminal de entrada está al aire.

Por ejemplo : Si la entrada está al aire, la salida del buffer será "1" log (con el terminal de control de histéresis al aire). Si por el contrario se fija un "1" log es este terminal (por medio de una resistencia, de valor adecuado, a Vcc) y el terminal de entrada del buffer está al aire ; la salida del buffer será "0" log.

RELOJ 74LS90 74LS76

Para fijar el baud rate de la USART se le debe aplicar una señal de reloj de transmisión y de recepción; éstas señales deben ser muy estables, es por esto que se toma la señal de reloj de la CPU (3.072 M Herz) y se procedió a dividirla por 4 distintos factores (factores externos).

Esta señal de reloj (3.072 M Hz) se ingresó primero al 74LS90 en el cual se dividió por 5 y luego por 2. La señal resultante (3.072 M Hz /5 /2) se inserta al 74LS76 y es dividida por 2 y nuevamente por 2.

La selección de estos factores se realiza en cuatro PIN de dos bases ubicadas abajo y hacia la derecha de la USART mirando el micro en su forma normal de trabajo.

Descripción de los PIN de izquierda a derecha:

- El de la extrema izquierda es el común o punto común de las señales TXC y RXC de la USART (pin 9 y 25 respectivamente).

- El siguiente es el factor por 10 de la señal de reloj, el próximo es el factor por 20 y el de la derecha es el factor por 40.

Hay que tener presente que el baud rate definitivo lo determinan los factores de división internos de la USART.

CAPITULO IV.- COMUNICACION SERIE

CONECTOR

Se empleó un conector DB 25 hembra en ángulo recto soldado en la placa del SDK-85.

En el cual se utilizó:

PIN	DESCRIPCION
1	TIERRA CHASIS.
2	TXD (TRANSMISION DE DATOS).
3	RXD (RECEPCION DE DATOS).
4	RST (PETICION PARA PONER DATOS).
5	CTS (LIMPIEZA DE PETICION).
6	DSR (LISTO EQUIPO DE PONER DATOS).
7	TIERRA SEÑAL.
10	-12V DISPONIBLES (BAJA CORRIENTE).
20	DTR (LISTO EL EQUIPO TERMINAL DE DATOS).

Haciendo un poco referencia a la norma RS 232-C :

Existe un detalle que no se mencionó y que personalmente considero interesante; éste se refiere a la configuración de equipos al implementar comunicación serie bajo esta norma y en especial a la correcta elección del cable de comunicación.

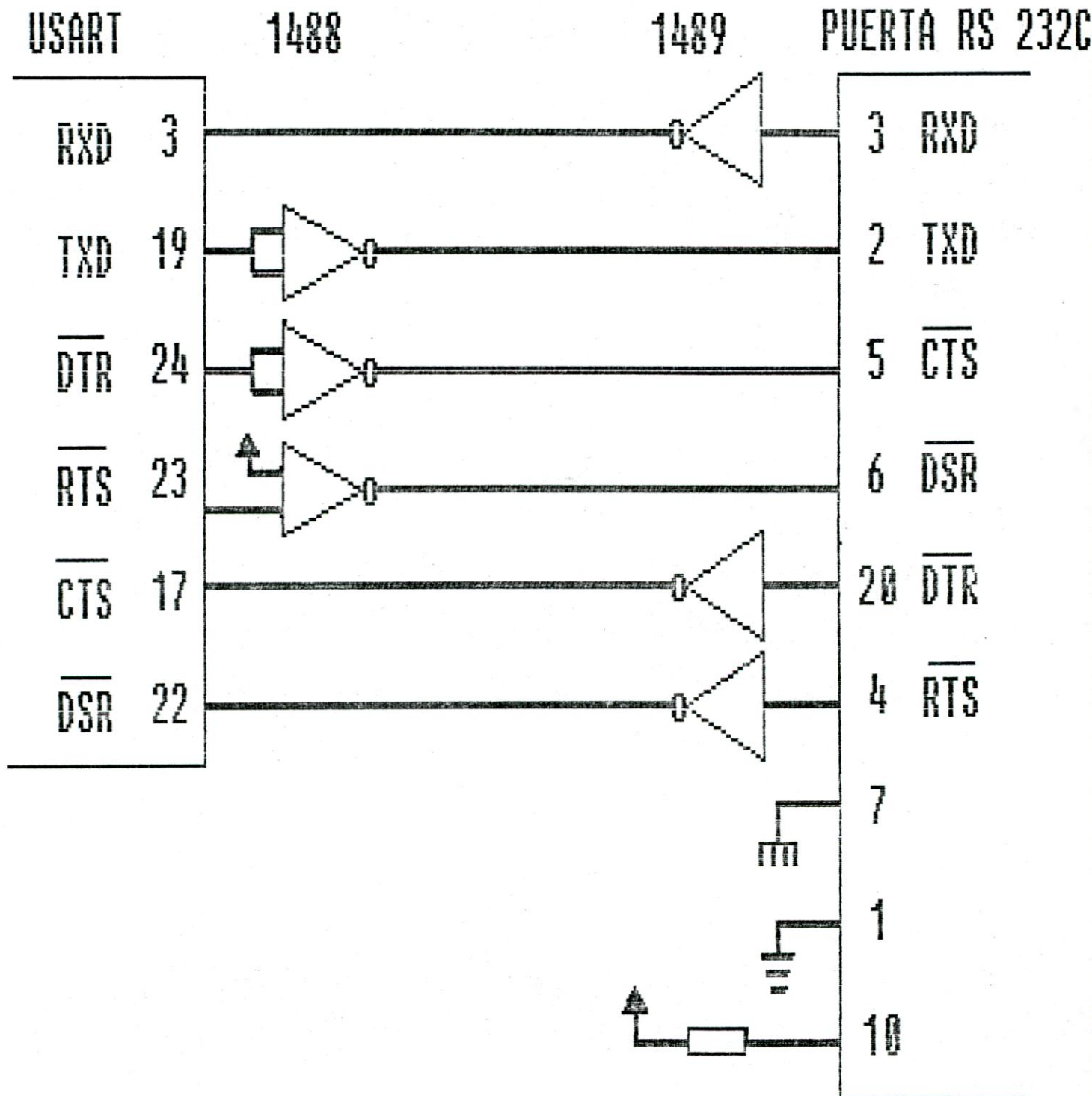
Se distinguen dos tipos :

- Los equipos configurados como modem : se distinguen porque el terminal de transmisión de datos es el pin # 3 del conector DB 25 y el pin # 2 es el terminal de recepción de datos.

- Los equipos configurados como terminal : se distinguen porque el terminal de transmisión de datos es el pin # 2 del conector, y el pin # 3 es el terminal de recepción de datos.

En los dos casos anteriores, los terminales de control mantienen su ubicación normal.

El siguiente esquema corresponde al alambrado utilizado:



NORMA RS 232-C

La norma RS 232-C fue definida por la EIA (Electrical Industry Association) en cooperación con los fabricantes de computadores y los fabricantes de Modem, con el objeto de normalizar los circuitos de interconexión, llamados circuitos de enlace de interfaz entre el equipo terminal de datos (ETD <equipo que transmite datos>) y el equipo de terminación del circuito de datos (ETCD <equipo que recibe los datos>).

A pesar de sus limitaciones (pequeñas distancias <15 metros> y baja velocidad de transmisión <20 K bit por seg>). Esta norma es muy utilizada en la actualidad en la comunicación de un computador y algunos de sus periféricos como : terminales de video, trazadores gráficos, modem y en la industria en control distribuido para la intercomunicación de los PLC.

Referente al nombre se denomina RS 232-C. La C se debe a que ésta es la tercera y última revisión.

Esta norma cubre los siguientes tres aspectos de la comunicación entre un ETD y su respectivo ETCD :

- Descripción funcional de las señales utilizadas en la comunicación.
- Características eléctricas de las señales.
- Características mecánicas de los conectores.

DESCRIPCION FUNCIONAL DE LAS SEÑALES

A continuación se hará la descripción en base a la nomenclatura utilizada por la norma.

Dentro del conjunto de las señales podemos distinguir cuatro grandes grupos :

- De datos.
- De control.
- De temporización.
- Las de masas.

En la siguiente figura se representa el número de las señales dentro del conector, el nemónico, el sentido de conexión entre el ETD y el ETCD y una breve descripción de sus significados.

PIN NUMERO	NEMONICO	DIRECCION	BREVE DESCRIPCION
1	AA	---	TERMINAL DE TIERRA
2	BA	HACIA ETCB	TRANSMISION DE DATOS
3	BB	HACIA ETD	RECEPCION DE DATOS
4	CA	HACIA ETCB	PETICION DE TRANSMISION
5	CB	HACIA ETD	PREPARADO PARA TRANSMITIR
6	CC	HACIA ETD	APARATO DE DATOS PREPARADO
7	AB	---	MASA COMUN DE LAS SEÑALES
8	CF	HACIA ETD	DETECTOR DE SEÑALES DE LINEA RECIBIDAS POR EL CANAL DE DATOS
9	---	---	RESERVADA PARA LA COMPROBACION DE LOS DATOS
10	---	---	RESERVADA PARA LA COMPROBACION DE LOS DATOS
11	---	---	SIN ASIGNACION
12	SCF	HACIA ETD	DETECTOR DE SEÑALES DE LINEA RECIBIDAS POR EL CANAL DE RESERVA DE DATOS
13	SCB	HACIA ETD	PREPARADO EL CANAL DE RESERVA PARA TRANSMITIR
14	SBA	HACIA ETCB	TRANSMISION DE DATOS POR EL CANAL DE RESERVA
15	DB	HACIA ETD	TEMPORIZACION PARA LOS ELEMENTOS DE SEÑAL EN LA TRANSMISION
16	SBB	HACIA ETD	RECEPCION DE DATOS POR EL CANAL DE RESERVA
17	DD	HACIA ETD	TEMPORIZACION PARA LOS ELEMENTOS DE SEÑAL EN LA RECEPCION
18	---	---	SIN ASIGNACION
19	SCA	HACIA ETCB	PETICION PARA TRANSMITIR POR EL CANAL DE RESERVA
20	CD	HACIA ETCB	TERMINAL DE DATOS PREPARADO
21	DG	HACIA ETCB	DETECTOR DE LA CALIDAD DE LAS SEÑALES DE DATOS
22	CE	HACIA ETD	INDICADOR DE LLAMADA
23	CH/CI	AMBAS	SELECTOR DE VELOCIDAD BINARIA CON ORIGEN ETD(CH) U ORIGEN ETCB (CI)
24	DA	HACIA ETCB	TEMPORIZACION PARA LOS ELEMENTOS DE SEÑAL EN LA TRANSMISION
25	---	---	SIN ASIGNACION

SEÑALES DE DATOS

BA. Transmisión de datos.

Es la línea utilizada para la transmisión de datos entre el ETD y el ETCD. Las restricciones que debe cumplir esta señal son las siguientes:

A) El ETD deberá poner esta señal a uno lógico entre la transmisión de caracteres o palabras y también cuando no se transmita datos.

B) Para que el ETD transmita datos se deberá cumplir que las señales CB, CC, CD y CA estén en cero lógico.

Normalmente, en los conectores comerciales se reconoce esta línea como TXD (datos a transmitir).

BB Recepción de datos.

Es la línea utilizada para la transmisión de los datos entre el ETCD y el ETD. Esta señal deberá estar en uno lógico mientras la señal CF esté en uno lógico. En un sistema half-duplex deberá estar en uno lógico cuando la señal CA esté en uno lógico.

En los conectores comerciales se reconoce esta línea como RXD (datos a recibir).

SBA Transmisión de datos por el canal de reserva.

Es equivalente a BA, pero por el canal de reserva. Dicho canal trabaja a velocidades inferiores.

SBB Recepción de datos por el canal de reserva.

Es equivalente a BB, pero por el canal de reserva. Dicho canal trabaja a velocidades inferiores.

Señales de control

CA Petición de transmitir.

Esta señal es enviada desde el ETD hacia el ETCD para indicarle, cuando la pone a cero lógico, que quiere realizar una transmisión. En un sistema half-duplex, un cero lógico inhibe la recepción.

Cuando se realiza sobre esta señal el cambio de uno a cero lógico, el ETCD responde cambiando la señal CB a cero lógico. Los datos a transmitir pueden ser, enviados solamente después de que el ETD detecte este cambio a cero lógico de CB.

Si la señal CA es cambiada a uno lógico, no podrá ser cambiada de nuevo a cero lógico hasta que el ETCD responda cambiando la señal CB a uno lógico. La señal CA es conocida normalmente como RST en conectores comerciales.

CB Preparado para transmitir.

Es enviada desde el ETCD hacia el ETD. El estado de esta señal indica si el ETCD está o no preparado para transmitir datos por el canal de datos.

Un uno lógico indica que el ETCD está en condiciones de transmitir datos por el canal de datos y un cero lógico indica lo contrario. Normalmente se conoce esta señal como CTS.

CC Aparato de datos preparado.

Esta señal es enviada por el ETCD hacia el ETD. El estado de esta señal indica si el ETCD está o no preparado para funcionar.

El estado uno lógico es sólo si el ETCD ha intentado establecer una comunicación por el canal después de haber cumplido con todas las temporizaciones necesarias y generado los tonos de respuesta. El estado cero lógico no indica que exista un canal de comunicación entre el ETCD y otro ETCD remoto, sino sólo el estado del ETCD local. Esta señal normalmente se conoce como DSR.

CD Terminal de datos preparado.

Esta señal es enviada desde el ETD hacia el ETCD. Un cero lógico en esta señal es necesario para mantener la comunicación entre el ETCD local y el ETCD remoto.

Si puesta a uno lógico indica al ETCD que deberá suspender la comunicación con el ETCD remoto al final de la transmisión que se está ejecutando en ese momento. Normalmente se le conoce como DTR.

CE Indicador de llamada.

Esta señal es enviada desde el ETCD hacia el ETD. El estado de esta señal indica si el ETCD está o no recibiendo una llamada.

Un cero lógico indica que el ETCD está recibiendo una llamada. La señal se pone a uno lógico en el intervalo entre

llamadas. Para que esta señal se ponga a cero lógico, la señal CD (DTR) deberá estar en cero lógico. Normalmente se le conoce como RI.

CF Detector de señales de línea recibidas por el canal de datos.

Es enviada desde el ETCD hacia el ETD. El estado de esta señal indica si las señales de línea recibidas por el canal de datos están o no dentro de los límites especificados en la recomendación pertinente para el ETCD. Un cero lógico indica que la señal recibida cumple con las especificaciones requeridas.

Normalmente se denomina a esta señal DCD (detector de portadora de datos).

CG Detector de la calidad en la señal de datos.

Esta señal va desde el ETCD hacia el ETD. El estado de esta señal indica si existe o no una cierta probabilidad de error en los datos recibidos por el canal de datos.

La calidad de señal indicada se ajusta a la recomendación pertinente sobre el ETCD. Un uno lógico indica que no hay motivos para creer que se ha producido un error. Un cero lógico indica que existe cierta probabilidad de error.

CH Selector de velocidad binaria.

Esta señal va desde el ETD hacia el ETCD. El estado de esta señal sirve para seleccionar una de las dos velocidades binarias de un ETCD sincrónico o una de las dos gamas de velocidades binarias en un ETCD asincrónico.

Un uno lógico causa la selección de la velocidad binaria o de la gama de velocidades binarias más elevada. Un cero lógico selecciona la más baja.

CI Selector de velocidad binaria.

Esta señal va desde el ETCD hacia el ETD. El estado de esta señal sirve para la selección de la velocidad binaria o de la gama de velocidades binarias en el ETD en función de la velocidad binaria utilizada en un ETCD sincrónico con dos velocidades binarias o de la gama de velocidades binarias utilizadas en un ETCD asincrónico con dos gamas de velocidades binarias.

Un uno lógico causa la selección de la velocidad binaria o gama de velocidades binarias más alta. Un cero lógico selecciona la más baja.

En una conexión determinada, sólo existirá una de las dos señales anteriores.

SCA Petición para transmitir por el canal de reserva.

Esta señal va desde el ETD hacia el ETCD. Su función es equivalente a la de la señal CA (petición de transmisión), pero para el canal de reserva.

SCB Preparado el canal de reserva para transmitir.

Esta señal va desde el ETCD hacia el ETD. Su función es equivalente a la de la señal CB, pero para el canal de reserva.

SCF Detector de señales de línea recibidas por el canal de reserva de datos.

Esta señal va desde el ETCD hacia el ETD. Su función es la misma que la señal CF pero por el canal de reserva.

Señales de temporización

DA Temporización, para los elementos de señal, en la transmisión.

Es una señal que va desde el ETD hacia el ETCD. El cambio de estado de cero a uno lógico en esta señal le indica al ETCD el centro de cada bit a transmitir.

DB Temporización, para los elementos de señal, en la transmisión.

Es una señal que va desde el ETCD hacia el ETD. El ETD deberá cambiar el estado de la línea BA (datos a transmitir) cuando se produzca una transición de uno a cero lógico en la señal DB.

DD Temporización para los elementos de señal en la recepción.

Esta señal va desde el ETCD hacia el ETD. La transición de estado cero a uno lógico en esta línea indica al ETD el centro del bit en la línea BB (datos recibidos). Esta señal será usada en el ETD para muestrear los datos recibidos.

Señales de masa

AA Señal de tierra.

Por esta señal se conectan las masas generales del ETD y el ETCD.

AB Masa común de las señales.

Es la señal de tierra o retorno común de forma que provee el potencial de referencia para todas las señales RS 232-C (excepto para la AA).

Los términos empleados en la descripción de las señales anteriores, así como algunas frases, han sido sacados de las recomendaciones V-24. Dicha recomendación es del CCITT y coincide en muchos aspectos con la norma RS 232-C.

Características eléctricas de las señales

La norma RS 232-C define las siguientes características eléctricas:

- A un uno lógico se le asignará -15 V.
- A un cero lógico se le asignará +15 V.

Estas señales serán válidas dentro de un rango de $\pm 3V$. a $\pm 15V$.

Características mecánicas de los conectores

Se designa como conector propio de esta norma al DB 25, pero en la actualidad es frecuente encontrar conectores DB 9 cumpliendo las exigencias de ésta.

CAPITULO V.- SOFTWARE IMPLEMENTADO

SUBRRUTINAS DEL PROGRAMA MONITOR DEL SDK-85

La siguiente es la única subrutina del programa monitor que se ha utilizado para la elaboración del software presente en el SDK-85.

02B7 Hex. Esta dirección corresponde a una subrutina usada para escribir datos sobre los display, entregando adecuadamente la información de qué campo se quiere escribir (registro A), si se quiere o no punto decimal (registro B) y la ubicación de la data a escribir (registro HL).

En el registro A se debe informar qué campo se desea escribir -campo de data o campo de dirección; si este tiene valor cero, entonces se está seleccionando el campo de dirección y si éste tuviera valor uno se estaría seleccionando el campo de data.

En el registro B se indica el uso del punto decimal; si éste es cero, el punto decimal no es usado y si es uno el punto decimal es activado.

En el registro pareado HL se debe entregar la dirección donde está ubicada la data a usar por esta subrutina. Si se ha seleccionado el campo de data, la subrutina tomará sólo los dos primeros bytes y si, por el contrario, se ha seleccionado el campo de dirección se tomarán los cuatro primeros bytes de data a contar de esta dirección.

FUNCIONES DEL CP/M UTILIZADAS

Las siguientes son las funciones utilizadas en la elaboración del software para el computador ADVANTAGE:

Función N°1 Leer un carácter desde el teclado

Esta función lee un carácter desde el teclado y lo almacena en el acumulador. Si no se presiona tecla alguna, la ejecución del programa queda detenida.

El formato es el siguiente:

- Cargar el registro C con 01 Hex.
- Llamar la subrutina 05 Hex.

Funcion N'9 Imprimir caracteres en pantalla

Esta función fue la más utilizada, puesto que es la mejor forma de comunicarse con el usuario.

A esta función hay que entregarle en el registro pareado DE la dirección de la cadena de caracteres a imprimir. Dicha cadena debe ser terminada por un signo '\$' o su equivalente ASCII (24 Hex).

El formato es el siguiente:

- Cargar el registro C con 9 Hex.
- Cargar el registro pareado DE con la dirección de la cadena.
- Llamar la subrutina 5 Hex.

Función N'0F Abrir archivo de disco

Esta función permite abrir un archivo existente en el disco.

Se debe entregar en el registro DE la dirección del FCB y previo a su llamada es necesario escribir en el FCB el nombre del archivo a abrir.

Si el archivo a abrir no existe, la función responderá con FF Hex en el acumulador y si éste existe, devolverá en el acumulador la ubicación de éste en el DMA (0,1,2 o 3).

El formato es el siguiente:

- Cargar el registro C con 0F Hex.
- cargar el registro pareado DE con la dirección del FCB.
- Llamar la subrutina 05 Hex.

Función N'10 Cerrar archivo de disco

Esta función permite cerrar un archivo de disco previamente abierto.

Se debe entregar en el registro pareado DE la dirección del FCB y previo a su llamada se debe escribir en el FCB el nombre del archivo a cerrar.

Si el archivo a cerrar no es encontrado, la función responderá con FF Hex en el acumulador y si existe, devolverá en el acumulador el número de entrada al DMA (0,1,2 o 3).

El formato es el siguiente:

- Cargar el registro C con 10 Hex.
- Cargar el registro pareado DE con la dirección del FCB.
- Llamar la subrutina 05 HEX.

Función N°11 Buscar por primera vez

Esta función permite leer el directorio del disco por primera vez.

Pueden efectuarse dos tipos de búsquedas. Una del directorio en forma completa; para esto debe llenarse en el FCB el espacio correspondiente al nombre primario y secundario con 3F Hex. que corresponde a un signo comodín (?). El otro tipo de búsqueda corresponde a la búsqueda de un archivo en particular o grupos de archivos. Para esto se debe colocar el nombre del archivo a buscar o en el caso de grupos, se debe llenar con comodines las partes comunes de los archivos -todo esto en el FCB, por supuesto.

Se debe entregar en el registro DE la dirección del FCB. Si no es encontrado el archivo, la función devolverá FF Hex. en el acumulador y si es encontrado, devolverá en el acumulador la ubicación en el DMA (0,1,2 o 3).

El formato es el siguiente:

- Cargar el registro C con 11 Hex.
- Cargar el registro pareado DE con la dirección del FCB.
- Llamar la subrutina 05 Hex.

Función N°12 Seguir buscando

Esta función se ejecuta a continuación de la anterior para seguir buscando en el directorio. Los códigos de respuesta se mantienen igual, excepto el FF Hex que se asume como que no existe el archivo que se buscaba.

El formato es el siguiente:

- Cargar el registro C con 12 Hex.
- Llamar la subrutina 05 Hex.

Función N°14 Leer secuencialmente

Esta función permite leer secuencialmente un archivo de disco, siempre que se haya activado previamente; la data leída es copiada sobre el DMA. Si la función responde con 00 Hex en el acumulador, esto indica que no existe mas data por leer.

El formato es el siguiente:

- Cargar el registro C con 14 Hex.
- Cargar el registro pareado DE con la dirección del FCB.
- Llamar la subrutina 05 Hex.

Función N°15 Escribir secuencialmente

Esta función permite escribir secuencialmente un archivo de disco, siempre que se haya activado previamente; la data es copiada desde el DMA al disco. Si la función responde con 00 Hex en el acumulador, esto indica que la escritura se realizó sin problemas y si este valor es distinto de cero, quiere decir que probablemente existan problemas en la escritura.

El formato es el siguiente:

- Cargar el registro C con 15 Hex.
- Cargar el registro pareado DE con la dirección del FCB.
- Llamar la subrutina 05 Hex.

Función N°16 Crear un archivo en el disco

Esta función permite crear un archivo en el disco y es equivalente a la de abrir un archivo en el sentido de dejar activado el archivo para lectura o escritura.

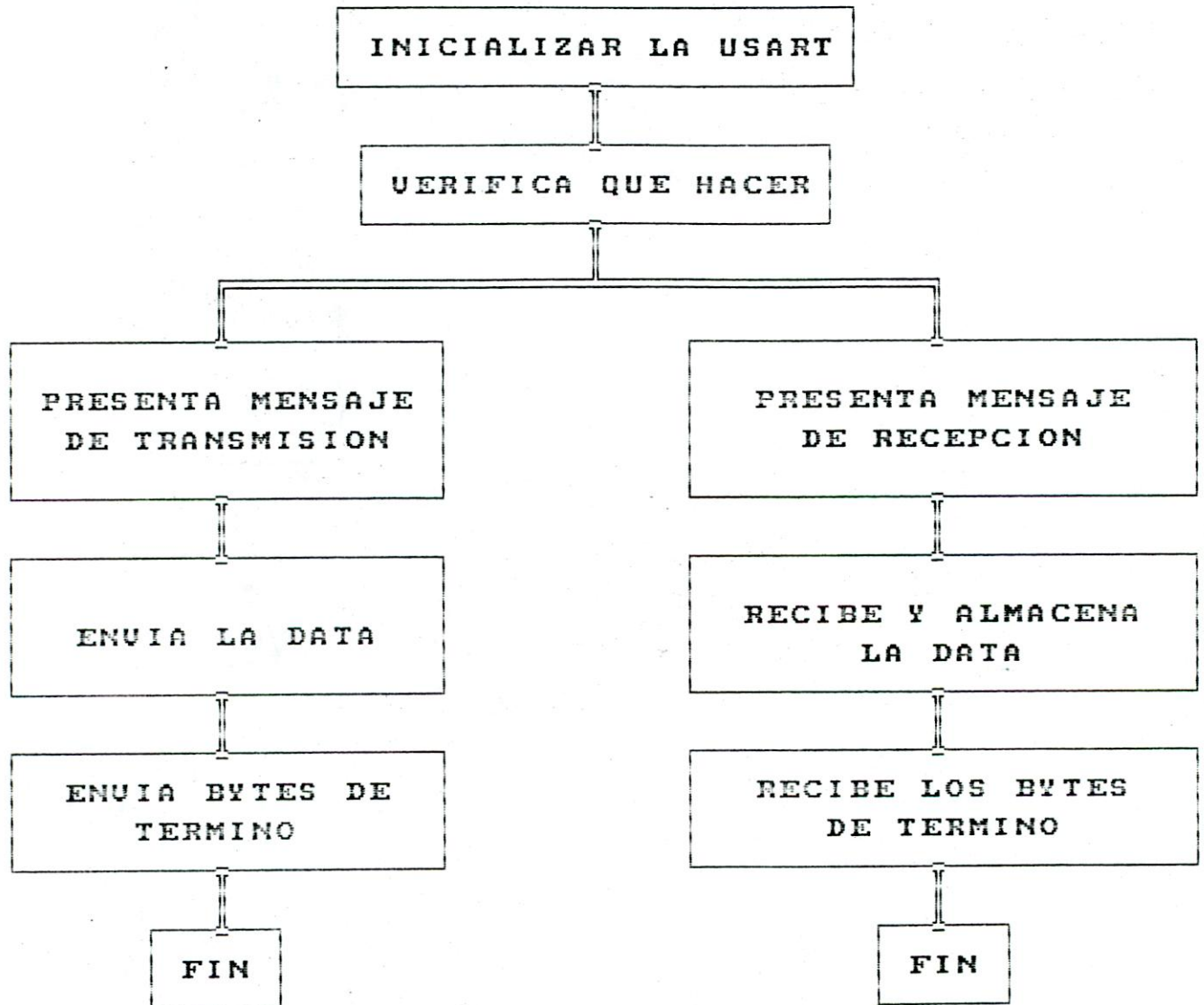
La función responderá con FF Hex si no hay espacio suficiente para un nuevo archivo en el disco y con 0,1,2 o 3 si la operación fue normal.

El formato es el siguiente:

- Cargar el registro C con 16 Hex.
- Cargar el registro pareado DE con la dirección del FCB.
- Llamar la subrutina 05 Hex.

SOFTWARE PARA EL SDK-85

El siguiente es un esquema de bloques del software implementado para el SDK-85. Este esquema se trató de simplificar al máximo para su global comprensión; si alguien quiere más información sobre él, puede remitirse al apéndice A.

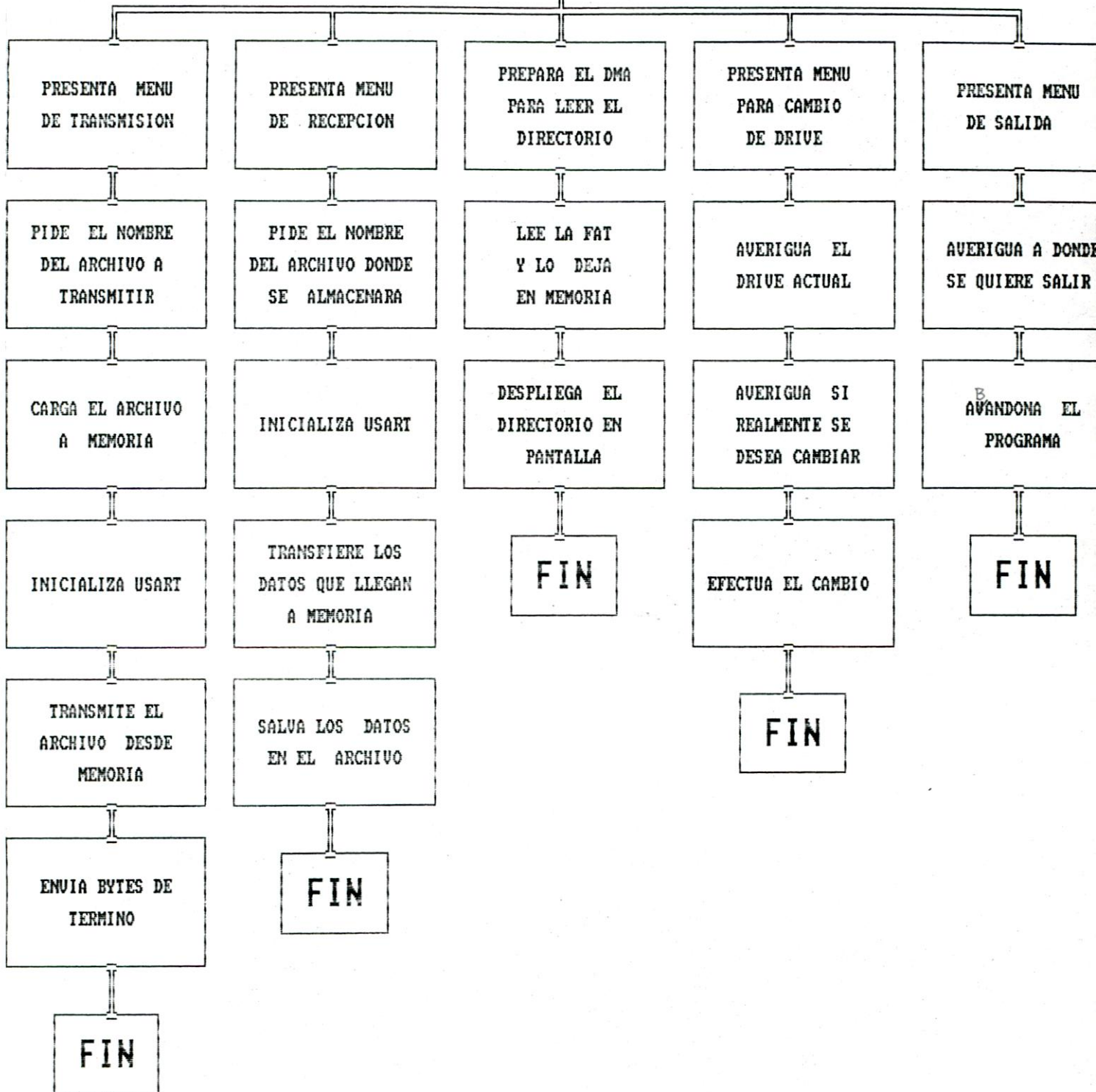


SOFTWARE PARA EL COMPUTADOR ADVANTAGE

El siguiente es un esquema de bloques del software implementado para el computador ADVANTAGE. Este esquema se trató de simplificar al máximo para su global comprensión; si alguien quiere más información sobre él, puede remitirse al apéndice A.

PRESENTA MENU PRICIPAL

AUERIGUAR QUE HACER



CAPITULO VI.- APLICACIONES

EJEMPLO PRACTICO

En este capítulo se tratará de detallar todos los pasos necesarios para la confección de un programa a partir de una idea de aplicación para el kit SDK-85.

Planteemonos el siguiente problema:

Se desea ingresar una palabra de 8 bits (puerta 22 del SDK-85); el complemento de ésta se debe hacer rotar visualizándolo en la puerta 21 de SDK-85, primero a la derecha (un giro completo) y luego a la izquierda (otro giro completo), al cabo de éste, reinicializar el ciclo nuevamente.

Para resolver este problema, podríamos seguir los siguientes pasos:

1'.- Haciendo uso de un procesador de texto disponible en el computador ADVANTAGE (por ejemplo wordstar), se debe hacer el programa fuente que realice lo anteriormente planteado, esto es, utilizando los nemónicos del 8085.

El siguiente programa puede cumplir con lo anteriormente propuesto:

```
                LXI SP,20C2
                MVI A,01                ;DEFINE LAS PUERTAS 21 22
                OUT 20
INTRA:          IN 22                   ;INGRESA UNA PALABRA DEL EXTERIOR
                CMA                     ;COMPLEMENTA LA PALABRA
                MVI C,8                 ;CARGA EL CONTADOR DE ROTACION
                OUT 21                  ;SACA EL DATO A LA PUERTA
ALLA:           CALL TIME              ;DA TIEMPO PARA VISUALIZAR
                RRC                     ;ROTA A LA DERECHA
                DCR C                   ;DECREMENTA CONTADOR
                OUT 21
                JNZ ALLA               ;VERIFICA EL TERMINO DE LA
                MVI C,8                 ;ROTACION A LA DERECHA
OTRA:           CALL TIME
                RLC                     ;ROTA A LA IZQUIERDA
                DCR C
                OUT 21
                JNZ OTRA               ;VERIFICA EL TERMINO DE LA
                JMP INTRA              ;ROTACION A LA IZQUIERDA
                ;REINICIA EL CICLO
```

Subrutina de tiempo.

```
TIME:   PUSH PSW           ; SALVAR EL ACUMULADOR
        LXI D,OFFF        ; DETERMINA EL TIEMPO
        CALL O2B7         ; GENERA EL LOOP
        POP PSW           ; RECUPERAR ACUMULADOR
        RET               ; RETORNO AL PROGRAMA
```

Una vez que se está seguro de que el programa funciona, entonces se debe salvar con extensión ASM.

2'.- Se debe convertir este archivo fuente en un programa con extensión COM o programa ejecutable por medio de un ensamblador, como por ejemplo L80 - M80 o ASM.

3'.- Se debe ejecutar el programa de comunicación MTRABA.COM en el menú principal de éste; aparecerán 5 opciones:

- 1 Transmisión.
- 2 Recepción.
- 3 Directorio.
- 4 Cambio de Driver
- 5 Retorno al CP/M.

Se debe elegir la opción N' 1; a continuación aparecerá el siguiente mensaje:

```
" Transmisión hacia el SDK-85.
  Los programas a trans.....
  .... memoria disponible en este A000,BFFF.
```

Ingrese el nombre del programa: "

Se debe tener cuidado en verificar el drive seleccionado en ese momento por el programa para que coincida con él del archivo a transferir; si esto no es así, se puede usar la opción N' 4.

4'.- A continuación se debe ingresar el nombre del archivo a transmitir.

5'.- Se debe preparar el SDK-85 para recibir la información:

A.- Cargar el registro pareado HL con la dirección de memoria donde se va a dejar la información.

B.- Cargar el registro A con FF.

C.- Empezar la ejecución desde la dirección 8000 Hex.; en este momento aparecerá en el display " RESEP ".

6'.- Una vez ingresado el nombre del archivo a transmitir aparecerá el siguiente mensaje:

"Si el programa a transmitir fue ensamblado por medio del L80 - M80. Sólo en este caso presione RETURN; si no, presione cualquier tecla."

Se debe realizar exactamente lo que se pide en base al ensamblador que se usó.

7'.- Una vez finalizado el traspaso de información en el display del SDK-85 aparecerá " 80-85 ".

8'.- Ahora podemos ejecutar la rutina a partir de la dirección de la cual se cargó el programa; por supuesto, antes de esto se debe definir la palabra a rotar en la puerta de entrada del SDK-85.

De la forma que hemos visto, se maneja el software; cada paso se detalló y además pide ratificar la opción.

IDEAS PARA IMPLEMENTAR

Creo que dentro de las cosas interesantes que se pueden realizar usando el hardware privilegiado del SDK-85 es un prototipo de control de proceso, como por ejemplo usando el servo motor disponible; realizar tareas previamente definidas y a la vez el control completo del motor.

Esto lo planteo por que fue una de las cosas que me habría gustado realizar personalmente y porque, además, la encuentro muy importante para la posterior vida laboral.

CAPITULO VII.- CONCLUSIONES

AUTOCRITICA

Encontré más adecuado llamar a este capítulo **autocritica** en lugar de conclusiones, puesto que aquí quiero hacer mención a varios puntos que considero importantes dentro del desarrollo de la memoria.

En un principio creo que no había tomado realmente el peso a lo que significaba implementar o tratar de implementar un PROTOTIPO DE UN KIT PARA DESARROLLO y, como lo mencioné anteriormente lo llegué a considerar un título pretencioso, pero a esta altura del trabajo creo que no lo es tanto.

En cuanto al HARDWARE creo que se implementó lo necesario para un buen funcionamiento y referente al SOFTWARE me hubiera gustado lograr algunas cosas más, como por ejemplo haber implementado el programa para que corriera en cualquier equipo CP/M, esto es, usando completamente las funciones de CP/M y, es más, haber implementado este SOFTWARE en un equipo DOS y ojalá que éste hubiera sido un PC compatible.

Bueno, además de estas inquietudes personales, me gustaría en lo personal, que de alguna forma esto se lograra.

Espero que este trabajo sirva, ya que en cuanto a mi persona me ha ayudado a desarrollarme profesionalmente.

BIBLOGRAFIA

- MANUAL DEL USUARIO DEL KIT SDK-85 DE INTEL
- MANUAL DE MEMORIAS DE INTEL
- MICROPROCESADORES Y PERIFERICOS (EDITORIAL MARCOMBO)
- MANUAL TTL DATA BOOCK

APENDICE A

PROGRAMA FUENTE DEL KIT SDK-85

```
                                ; NECESITA EL A Y HL (DE)
ORG 8000H                       ; A=FF-RX-HL=DIREC INICIAL A GUARDAR DATOS
PUSH PSW                        ; A=00-TX-HL=DIREC INICIAL DE=DIREC FINAL
MVI A,80H                       ; DE DATOS
OUT 81H
OUT 81H                          ; RESET POR SOFTWARE Y PALABRAS DE MODO
MVI A,40H                       ; Y COMANDO
OUT 81H
MVI A,0CEH                      ; /16-8 BIT-SIN PARIDAD-2 STOP BIT
OUT 81H
MVI A,17H                      ; TX-DTR=0-RX-RESET FLAGS
OUT 81H
IN 80H                          ; LECTURA FALSA
POP PSW
CPI 00H                        ; REvisa SI EL DATO DEL ACUMULADOR
JZ TXA                          ; ES PARA TX O RX
CPI 0FFH
JZ RXA
RST 1H                          ; SI EL BYTE NO ES IDENTIFICADO TERMINA

TXA: CALL ROTX
TX:  IN 81H
      ANI 01H                   ; REvisa SI EL TRANSMISOR ESTA DESOCUPADO
      JZ TX
      MOV A,M
      OUT 80H
      INX H
      MOV A,L                   ; DETERMINA SI LLEGO AL TERMINO DE LO
      CMP E                     ; QUE HAY QUE TRANSMITIR
      JNZ TX
      MOV A,H
      CMP D
      JNZ TX
TTT: IN 81H
      ANI 01H
      JNZ TTT
      MOV A,M                   ; ENVIA EL ULTIMO DATO
      OUT 80
      MVI B,10H
```

```

TXF:    IN 81H
        ANI 01H
        JZ TXF
        MVI A,OFFH
        OUT 80H
        DCR B                ; ENVIA 15 BYTE PARA DETECTAR EL
        JNZ TXF             ; TERMINO DE LA TX
        RST 1H

RXA:    CALL RORX
RX:     MVI B,00H           ; CONTADOR DE BYTE DE TERMINO
        CALL VER           ; REvisa SI LLEGO UN NUEVO DATO
        IN 80H            ; INGRESO DATO
        CALL DESAC        ; DESACTIVA EL OTRO TX
        CPI OFFH         ; COMPARA CON BYTE DE TERMINO
        JZ CONTA         ; VA A REALIZAR CUENTA
        MOV M,A          ; INGRESA EL DATO A MEMORIA
        INX H            ; INCREMENTA EL PUNTERO
        CALL ACTIV       ; ACTIVA EL OTRO TX
        JMP RX

CONTA:  INR B              ; INCREMENTA EL CONTADOR DE BYTE DE TERMINO
        CALL ACTIV       ; ACTIVA EL OTRO TX
        CALL VER         ; VERIFICA SI LLEGO UN NUEVO DATO
        IN 80H          ; INGRESA EL PROXIMO DATO
        CALL DESAC      ; DESACTIVA EL OTRO TX
        CPI OFFH        ; VERIFICA SI ES UN BYTE DE TERMINO
        JZ ALLA

OTRA:   MVI A,OFFH       ; SI NO -INGRESA A LA MEMORIA LOS BYTES
        MOV M,A          ; FF DETECTADOS Y EL ULTIMO LEIDO POR
        INX H            ; EL CUAL SE DEDUJO QUE NO CORRESPONDIA
        DCR B            ; A LA SECUENCIA DE TERMINO
        JNZ OTRA
        POP PSW
        MOV M,A
        INX H
        CALL ACTIV      ; ACTIVA EL OTRO TX
        JMP RX

ALLA:   MOV A,B          ; VERIFICA SI YA HAN LLEGADO TODOS LOS
        CPI 0EH         ; BYTE CORRESPONDIENTES AL TERMINO DE
        JNZ CONTA      ; LA DATA
        CALL ACTIV
        RST 1H
        NOP
        NOP
        NOP
        NOP
        NOP

```

DESAC: PUSH PSW
MVI A,05H
OUT 81H ;DESACTIVA EL TX DEL OTRO EQUIPO
POP PSW
RET

ACTIV: PUSH PSW
MVI A,07H
OUT 81H ;ACTIVA EL TX DEL OTRO EQUIPO
POP PSW
RET

VER: IN 81H
ANI 02H ;REVISA SI LLEGO UN NUEVO DATO
JZ VER
RET

ROTX: PUSH PSW
PUSH B
PUSH D
PUSH H
MVI A,01H
MVI B,0H
LXI H,DATA
CALL 02B7H ;IMPRIME MENSAJE DE TRANSMISION
MVI A,0H
MVI B,0H
LXI H,ADTX
CALL 02B7H
POP H
POP D
POP B
POP PSW
RET

RORX: PUSH PSW
PUSH B
PUSH D
PUSH H
MVI A,01H
MVI B,0H
LXI H,DATA
CALL 02B7H ;IMPRIME MENSAJE DE RECEPCION

```
MVI A, 0H  
MVI B, 0H  
LXI H, ADRX  
CALL 02B7H  
POP H  
POP D  
POP B  
POP PSW  
RET
```

```
DATA: DB 15H, 15H, 15H, 15H, 15H
```

```
ADTX: DB 07H, 54H, 50H, 0AH, 15H
```

```
ADRX: DB 50H, 0EH, 0CH, 0EH, 15H
```

```
END
```

PROGRAMA FUENTE DEL COMPUTADOR ADVANTAGE

```

PARTE: LXI SP,0C000H ; HAY QUE MOVER LA PILA QUE ORIGINALMENTE SE
MENUP: MVI C,9H ; ENCUENTRA SOBRE EL DMA
      LXI D,ROTUL ; PRESENTA ROTULO
      CALL 5H
      CALL CADRE ; VERIFICA EL DRIVE A USAR Y ALMACENA EL DATO
      MVI C,9H ; EN MEMORIA
      LXI D,TEXME ; PRESENTA MENU PRINCIPAL
      CALL 5H
      CALL LEER ; AVERIGUA QUE TECLA FUE PRESIONADA
      CPI 31H
      JZ TRANS
      CPI 32H ; AVERIGUA QUE HACER
      JZ RECEP
      CPI 33H
      JZ DIREC
      CPI 34H
      JZ CAMDR
      CPI 35H
      JZ FINN
      JMP MENUP

TRANS: MVI C,9H
      LXI D,ROTUL ; PRESENTA ROTULO
      CALL 5H
      MVI C,9H
      LXI D,TEXTX ; PRESENTA MENU
      CALL 5H
      CALL CADRE ; VERIFICA EL DRIVE A USAR Y LO ALMACENA EN
      MVI C,9H ; MEMORIA
      LXI D,DRIVE ; PRESENTA DRIVE A USAR
      CALL 5H
      CALL NOMBRE ; PERMITE INGRESAR DESDE TECLADO EL NOMBRE DEL
      CPI 00H ; ARCHIVO A USAR Y LO DEJA EN EL FCB
      JNZ PERTO ; DETECTA SI SE APRETO RETURN EN VEZ DEL NOM-
      MVI C,0FH ; BRE Y PERMITE EL RETORNO AL MENU PRINCIPAL
      LXI D,005CH ; ABRE EL ARCHIVO INDICADO POR EL USUARIO
      CALL 5H ; PARA LECTURA
      CPI 0FFH ; VERIFICA SI EXISTE EL ARCHIVO SI NO PUBLI-
      JZ MNEFIN ; CA MENSAJE
      LXI H,2000H ; INICIALIZA EL PUNTERO DE BUFFER DE MEMORIA
TLEER: MVI C,14H
      LXI D,005CH ; LEE SECUENCIALMENTE DESDE DISCO Y LO DEJA
      ; EN EL DMA
      PUSH H ; SALVA EL PUNTERO PUESTO QUE LA SUBRRUTINA
      CALL 5H ; LO PUEDE ALTERAR
      POP H

```

```

INR A ; FORMA DE ALTERAR EL FLAG DE CERO
DCR A
JNZ TTERMI
CALL SACAR ; MUEVE EL DMA AL BUFFER DE MEMORIA
JMP TLEER

TTERMI: MVI C,10H
LXI D,005CH ; CIERRA EL ARCHIVO
PUSH H ; SE SALVA EL PUNTERO DEL BUFFER EL CUAL CON-
CALL 5H ; TIENE LA ULTIMA POSICION OCUPADA
MVI A,80H
OUT 51H
OUT 51H ; RESET POR SOFTWARE Y PALABRAS DE MODO
MVI A,40H ; Y COMANDO
OUT 51H
MVI A,0CEH ; /16-8 BIT-SIN PARIDAD-2 STOP BIT
OUT 51H
MVI A,17H ; TX-DTR=0-RX-RESET FLAGS
OUT 51H
IN 50H ; LECTURA FALSA DESEABLE PARA NO COMETER
POP H ; ERRORES
MOV D,H ; SE MUEVE EL CONTENIDO DE HL A DE
MOV E,L
LXI H,2000H ; SE REINICIALIZA EL PUNTERO DEL BUFFER

CALL DECIS ; AVERIGUA EL ORIGEN DEL PROGRAMA A TRAN-
; MITIR PARA ADECUAR EL PUNTERO DEL BUFFER

TX: IN 51H
ANI 01H ; REvisa si el transmisor esta desocupado
JZ TX
MOV A,M
OUT 50H
INX H
MOV A,L ; DETERMINA SI LLEGO AL TERMINO DE LO
CMP E ; QUE HAY QUE TRANSMITIR
JNZ TX
MOV A,H
CMP D
JNZ TX

TTT: IN 51H
ANI 01H
JNZ TTT
MOV A,M ; ENVIA EL ULTIMO DATO
OUT 50H
MVI B,10H

TXF: IN 51H
ANI 01H
JZ TXF
MVI A,0FFH

```

```
OUT 50H
DCR B ; ENVIA 15 BYTE PARA DETECTAR EL
JNZ TXF ; TERMINO DE LA TX
JMP MENUP
```

```
MNEFIN: PUSH PSW
        PUSH B
        PUSH D
        PUSH H
        MVI C, 9H
        LXI D, TEXMNE ; PRESENTA MENSAJE DE ARCHIVO NO EXISTENTE
        CALL 5H
        MVI C, 9H
        LXI D, TEXAVT ; DA TIEMPO PARA LEER MENSAJE
        CALL 5H
        CALL LEER
        POP H
        POP D
        POP B
        POP PSW
        JMP MENUP
```

```
RECEP: MVI C, 9H
        LXI D, ROTUL ; PRESENTA ROTULO
        CALL 5H
        MVI C, 9H
        LXI D, TRECEP ; PRESENTA MENU
        CALL 5H
        CALL CADRE ; AVERIGUA EL DRIVE
        MVI C, 9H
        LXI D, DRIVE ; IMPRIME EL DRIVE
        CALL 5H
        CALL NOMBRE ; INGRESA EL NOMBRE DEL ARCHIVO
        CPI 00H ; A GUARDAR LA DATA TRANSFERIDA
        JNZ PERTO
        LXI H, 2000H ; INICIALIZA EL PUNTERO

        MVI A, 80H ; DE DATOS
        OUT 51H
        OUT 51H ; RESET POR SOFTWARE Y PALABRAS DE MODO
        MVI A, 40H ; Y COMANDO
        OUT 51H
        MVI A, 0CEH ; /16-8 BIT-SIN PARIDAD-2 STOP BIT
        OUT 51H
        MVI A, 17H ; TX-DTR=0-RX-RESET FLAGS
        OUT 51H
        IN 50H ; LECTURA FALSA
```

```

RX:      MVI B,00H      ; CONTADOR DE BYTE DE TERMINO
        CALL VER      ; REvisa si llego un nuevo dato
        IN 50H      ; INGRESO DATO
        CALL DESAC    ; DESACTIVA EL OTRO TX
        CPI OFFH     ; COMPARA CON BYTE DE TERMINO
        JZ CONTA     ; VA A REALIZAR CUENTA
        MOV M,A      ; INGRESA EL DATO A MEMORIA
        INX H      ; INCREMENTA EL PUNTERO
        CALL ACTIV    ; ACTIVA EL OTRO TX
        JMP RX

CONTA:   INR B      ; INCREMENTA EL CONTADOR DE BYTE DE TERMINO
        CALL ACTIV    ; ACTIVA EL OTRO TX
        CALL VER      ; VERIFICA SI LLEGO UN NUEVO DATO
        IN 50H      ; INGRESA EL PROXIMO DATO
        CALL DESAC    ; DESACTIVA EL OTRO TX
        CPI OFFH     ; VERIFICA SI ES UN BYTE DE TERMINO
        JZ RSALLA

RSOTRA:  MOV M,A      ; SI NO -INGRESA A LA MEMORIA LOS BYTES
        INX H      ; FF DETECTADOS Y EL ULTIMO LEIDO POR
        DCR B      ; EL CUAL SE DEDUJO QUE NO CORRESPONDIA
        JNZ RSOTRA  ; A LA SECUENCIA DE TERMINO
        POP PSW
        MOV M,A
        INX H
        CALL ACTIV    ; ACTIVA EL OTRO TX
        JMP RX

RSALLA:  MOV A,B      ; VERIFICA SI YA HAN LLEGADO TODOS LOS
        CPI 0EH     ; BYTE CORRESPONDIENTES AL TERMINO DE
        JNZ CONTA   ; LA DATA
        CALL ACTIV
        INX H
        LXI D,RDIREC ; GUARDA EL PUNTERO EN MEMORIA
        MOV A,L
        STAX D
        INX D
        MOV A,H
        STAX D

RCONUE:  MVI C,16H
        LXI D,005CH ; CREA UN NUEVO ARCHIVO
        CALL 5H
        CPI OFFH
        JZ RMENSA

ROTRA:   LXI H,2000H ; REINICIALIZA EL PUNTERO
        LXI D,0080H
        CALL LIMPIE
        MVI B,80H

RALLA:   MOV A,M
        STAX D
        INX D      ; TRANSFIERE UN BLOQUE AL DMA

```

```

      INX H
      CALL RCOMP
      CPI 00H
      JNZ RMASA
      JMP RALFIN
RMASA: DCR B
      JNZ RALLA
      CALL RESCRI      ; ESCRIBE UN BLOQUE EN EL DISCO
      CPI 01H
      JZ RCONU
      CPI 05H
      JZ RMMME
      JMP ROTRA

RALFIN: CALL RESCRI      ; ESCRIBE UN BLOQUE EN EL DISCO
      CPI 01H
      JZ RCONU
RMMME: MVI C, 10H
      LXI D, 005CH      ; CIERRA EL ARCHIVO
      CALL 5H
      JMP MENUP

RMENSA: MVI C, 9H
      LXI D, RNOESP
      CALL 5H           ; IMPRIME MENSAJE QUE NO HAY ESPACIO
      CALL LEER         ; EN EL DISCO
      CPI 0DH
      JZ RCONUE
PERTO: JMP MENUP

RCONU: MVI C, 10H
      LXI D, 005CH      ; CIERRA EL ARCHIVO
      CALL 5H
      JMP RCONUE

DIREC: MVI B, 0BH
      LXI D, 005DH
      MVI A, 3FH
ALLA:  STAX D           ; ESCRIBE LOS COMODINES EN EL FCB
      INX D
      DCR B
      JNZ ALLA
      CALL CADRE        ; AVERIGUA EL DRIVE
      MVI C, 11H
      LXI D, 5CH        ; LEE POR PRIMERA VEZ
      CALL 5H
      CPI 0FFH          ; VE SI NO HAY ARCHIVOS
      JZ UNFIN

```

```

LXI B,ALFIN
MVI E,04H
CALL IMPRI ; LEE DEL FCB Y LO DEJA EN MEMORIA
OTRA:  PUSH B
      PUSH D
      PUSH H
      MVI C,12H
      LXI D,005CH ; SIGUE LEYENDO
      CALL 5H
      POP H
      POP D
      POP B
      CPI OFFH ; DETECTA EL TERMINO
      JZ FIN
      CALL IMPRI ; LEE DEL FCB Y LO DEJA EN MEMORIA
      JMP OTRA

UNFIN: MVI C,9H
      LXI D,NOFIL ; MENSAJE DE NO EXISTENCIA DE ARCHIVOS
      CALL 5H
      JMP TFIN
FIN:   MVI A,24H
      STAX B
      MVI C,9H
      LXI D,TEXPOR ; IMPRIME 'A:' Y DIRECTORIO
      CALL 5H
      MVI C,9H
      LXI D,ALFIN ; CARGA PUNTERO DE DIRECTORIO
      CALL 5H
TFIN:  MVI C,9H
      LXI D,TEXAVT ; IMPRIME A:
      CALL 5H
      CALL LEER
      JMP MENUP

CAMDR: MVI C,9H
      LXI D,ROTUL ; PRESENTA ROTULO
      CALL 5H
      MVI C,9H
      LXI D,TEXCAD ; PRESENTA MENU
      CALL 5H
      MVI C,9H
      LXI D,DRIVE ; AVERIGUA EL DRIVE
      CALL 5H
      MVI C,9H
      LXI D,TEXCAS ; AVERIGUA SI SE DESEA CAMBIAR
      CALL 5H

```

```

CALL LEER
CPI 53H
JZ CAMSY
JMP MENUP
CAMSY: LXI H,DRIVEE      ;CARGA EL DRIVE
MOV A,M
CPI 01H
JZ CANSY
CPI 00H
JZ CANSY
DCR A
MOV M,A                ;CAMBIA EL DRIVE
JMP CANS
CANSY: INR A
MOV M,A
CANS:  JMP MENUP

FINN:  MVI C,9H
LXI D,ROTUL           ;PRESENTA ROTULO
CALL 5H
MVI C,9H
LXI D,TEXFIN         ;PRESENTA MENU
CALL 5H
CALL LEER
CPI 41H
JZ GFIN              ;AVERIGUA QUE HACER
CPI 42H
JZ FGGT
JMP MENUP
GFIN:  JMP 00H
FGGT:  RST 7

```

MEMORIA RESERVADA PARA VALORES Y TEXTO

DRIVE: DB 41H, 3EH, 20H, 24H

DRIVEE: DB 01H, 00H

BORR: DB 20H, 08H, 24H

NOFIL: DB 0DH, 0AH, 0AH, 09H, 09H
DB 'NO HAY ARCHIVOS EN EL DISCO'
DB 24H

TEXPOR: DB 0DH, 0AH, 0AH, 41H, 3AH, 20H, 24H

TEXAVT: DB 0DH, 0AH, 0AH
DB ' PRESIONE UNA TECLA PARA CONTINUAR ', 24H

ROTUL: DB 04H, 0AH, 0AH, 0AH, 0AH, 09H, 09H, 09H, 09H, 09H
DB ' USM ', 0DH, 0AH, 09H, 09H, 09H, 09H, 09H
DB 'SEDE REY BALDUINO DE BELGICA', 0DH, 0AH, 09H, 09H, 09H, 09H, 09H
DB ' DEPARTAMENTO DE ELECTRONICA', 0DH, 0AH, 09H, 09H, 09H, 09H, 09H
DB ' TALCAHUANO', 24H

TEXME: DB 0DH, 0AH, 0AH, 0AH, 0AH, 09H, ' PROGRAMA PARA TRANSMISION '
DB ' DE DATOS ENTRE '
DB ' EL COMPUTADOR ADVANTAGE', 0DH, 0AH, 09H, 09H, 09H, 09H
DB ' Y EL KIT SDK-85'
DB 0DH, 0AH, 0AH, 0AH, 09H
DB ' 1- TRANSMISION', 0DH, 0AH, 0AH, 09H
DB ' 2- RECEPCION', 0DH, 0AH, 0AH, 09H
DB ' 3- DIRECTORIO', 0DH, 0AH, 0AH, 09H
DB ' 4- CAMBIO DE DRIVE', 0DH, 0AH, 0AH, 09H
DB ' 5- RETORNO AL CP/M', 0DH, 0AH, 0AH, 0AH, 09H, 09H, 09H
DB ' QUE DESEA HACER :', 24H

TEXTX: DB 0DH, 0AH, 0AH, 0AH, 0AH, 0AH, 09H, 09H, 09H
DB ' TRANSMISION HACIA EL SDK-85', 0DH, 0AH, 0AH, 0AH, 09H
DB ' LOS PROGRAMAS A TRANSMITIR DEBEN SER PROGRAMAS YA EN'
DB ' SAMBLADOS DE', 0DH, 0AH, 09H
DB ' MANERA QUE SEAN EJECUTABLES EN EL SDK - 85 Y CON LAS'
DB ' DIRECCIONES', 0DH, 0AH, 09H
DB ' CORRESPONDIENTES A LA MEMORIA DISPONIBLE EN ESTE (A'
DB ' 000, BFFF).', 0DH, 0AH, 0AH, 0AH, 0AH, 0AH, 09H, 09H
DB ' INGRESE EL NOMBRE DEL PROGRAMA ', 24H

TEXTMNE: DB ODH, OAH, OAH, O9H, O9H, 'ARCHIVO NO EXISTENTE EN EL DISCO', 24H

TEXCAD: DB ODH, OAH, OAH, OAH, OAH, OAH, OAH, O9H
DB 'USTED SE ENCUENTRA TRABAJANDO EN EL DRIVE ', 24H

TEXCAS: DB ODH, OAH, OAH, OAH, OAH, OAH, O9H, O9H
DB 'SE DESEA CAMBIAR (S) o (N) :', 24H

TEXFIN: DB ODH, OAH, OAH, OAH, OAH, OAH, OAH, O9H
DB 'EXISTE TAMBIEN LA POSIBILIDAD DE RETORNAR AL DDT SIEMPRE'
DB 'QUE SE HAYA', ODH, OAH, O9H
DB 'CORRIDO ESTE PROGRAMA SOBRE EL DDT.', ODH, OAH, OAH, OAH, O9H
DB 'A) RETORNO AL CP/M.', ODH, OAH, OAH, O9H
DB 'B) RETORNO AL DDT.', ODH, OAH, OAH, OAH, O9H, O9H
DB 'QUE DESEA HACER :', 24H

RDIREC: DB OOH, OOH, OOH

TRECEP: DB ODH, OAH, OAH, OAH, OAH, O9H, O9H
DB 'TRANSMISION DESDE EL SDK-85', ODH, OAH, O9H
DB 'LOS PROGRAMAS RECIBIDOS PUEDEN SER EJECUTADOS PERO'
DB ' CUIDADOSAMENTE', ODH, OAH, O9H
DB 'HAY QUE TENER PRESENTE QUE ESTAN DIRECCIONADOS A PARTIR'
DB ' DE LA DIRECCI-', ODH, OAH, O9H
DB 'ON AOOO HASTA BFFF', ODH, OAH, OAH, OAH, O9H
DB 'INGRESE EL NOMBRE DEL ARCHIVO EN EL CUAL SE ALMACENARA'
DB ' LA INFORMA-', ODH, OAH, O9H, 'CION RECIBIDA DESDE EL SDK-85'
DB ' .', 24H

RNOESP: DB ODH, OAH, OAH, O9H
DB 'NO HAY ESPACIO EN EL DISCO PARA EL ARCHIVO'
DB ODH, OAH, OAH, O9H
DB 'SI DESEA GRABAR CAMBIE EL DISCO DEL DRIVE EN QUE ESTA'
DB ' TRABAJANDO', ODH, OAH, O9H
DB 'Y APRIETE RETURN, SI NO APRETE CUALQUIER TECLA PARA AB'
DB 'ANDONAR', 24H

TEDESI: DB ODH, OAH, OAH, O9H
DB 'SI EL PROGRAMA A TRANSMITIR FUE ENSAMBLADO POR MEDIO DE'
DB '"M80" Y "L80"', ODH, OAH, O9H
DB 'SOLO EN ESTE CASO PRESIONE RETURN SI NO PRESIONE CUALQU'
DB 'IER TECLA', 24H

SOUND: DB O5H, 24H

SUBROUTINAS

```

LEER:   PUSH B
        PUSH D
        PUSH H
        MVI C, 9H
        LXI D, SOUND      ; PONE SONIDO A LAS TECLAS
        CALL 5H
        MVI C, 1H
        CALL 5H           ; LEE UN DATO DESDE EL TECLADO
        PUSH PSW
        MVI C, 9H
        LXI D, NOUND     ; SUPRIME SONIDO A LAS TECLAS
        CALL 5H
        POP PSW
        POP H
        POP D
        POP B
        RET

NOMBRE: LXI H, 005DH      ; CARGA PUNTERO AL INICIO DEL FCB
        MVI C, 0BH
        MVI A, 20H
MOON:   MOV M, A
        INX H             ; LIMPIA EL FCB
        DCR C
        JNZ MOON
        LXI H, NOMB      ; CARGA PUNTERO CON LA DIRECCION DEL
        MVI B, 10H      ; BUFFER DE MEMORIA
        MVI A, 20H
NOM:    MOV M, A
        INX H             ; LIMPIA EL BUFFER
        DCR B
        JNZ NOM
        LXI H, NOMB     ; CARGA PUNTERO CON LA DIRECCION DEL
        ; BUFFER DE MEMORIA
        CALL LEER       ; LEE TECLADO
        CPI 0DH
        JZ PERRMI
        JMP MONBR
NOMBR:  CALL LEER       ; LEE TECLADO
MONBR:  CPI 0BH         ; AVERIGUA SI SE QUIERE BORRAR
        JNZ MOMB
        CALL BORRA     ; BORRAR UN CARACTER
        DCX H
        JMP NOMBR
MOMB:   MOV M, A
        INX H
        CPI 0DH         ; DETECTA EL TERMINO
    
```

```

JNZ NOMBR
LXI D,NOMB ;CARGA PUNTERO CON BUFFER DE MEMORIA
LXI H,005DH ;CARGA PUNTERO DEL FCB
MVI B,0H
CAMBIO: LDAX D
CPI 2EH
JZ RELLE ;DETECTA EL PUNTO
CPI ODH
JZ RELLE
MOV M,A
INR B
INX H ;ACTUALIZA PUNTEROS
INX D
MOV A,B
CPI 9H ;DETECTA EL TERMINO DEL NOMBRE PRIMARIO
JNZ CAMBIO
CAMBI: LDAX D
INX D
CPI ODH
JZ TERMI
CPI 2EH ;DETECTA EL PUNTO
JNZ CAMBI
MVI B,0H
JMP SECUM
RELLE: MOV A,B
CPI 8H
JNC SECUN ;RELLENA EL NOMBRE PRIMARIO
MVI A,20H
MOV M,A
INX H
INR B
JMP RELLE
SECUN: MVI B,0H
SEKUN: INX D
SECUM: LDAX D
CPI 2EH ;DETECTA EL PUNTO
JZ SEKUN
CPI ODH
JZ TERMI ;DETECTA EL TERMINO
MOV M,A
INX H
INX D
INR B
MOV A,B
CPI 3H ;DETECTA EL TERMINO DEL NOMBRE SECUNDARIO
JZ TERMI
JMP SECUM
TERMI: LXI H,007CH ;ASEGURA REALIZAR LA PROXIMA OPERACION
MVI A,0H ;DE DISCO DESDE EL PRINCIPIO
MOV M,A
PERRMI: RET

```

BORRA: PUSH B
 PUSH D
 PUSH H
 MVI C,9H
 LXI D,BORR ; BORRA UN CARACTER DE PANTALLA
 CALL 5
 POP H
 POP D
 POP B
 RET

SACAR: PUSH PSW
 PUSH B
 PUSH D
 LXI D,0100H ; INICIALIZA PUNTEROS
 LXI B,0080H

HHH: LDAX B
 MOV M,A
 INX H
 INX B
 MOV A,C
 CMP E ; MUEVE DE MEMORIA A UN BUFFER DE MEMORIA
 JNZ HHH
 MOV A,B
 CMP D
 JNZ HHH
 POP D
 POP B
 POP PSW
 RET

IMPRI: RLC
 RLC
 RLC
 RLC
 RLC
 ADI 80H ; DETECTA DONDE LEER
 MOV L,A
 MVI H,00H
 MOV A,M
 INX H
 CPI 3FH ; DETECTA SI EL ARCHIVO ESTA BORRADO
 JZ RETOR
 MVI D,0BH
OTTA: MOV A,M
 ANI 7FH
 STAX B

```

    INX B
    INX H
    DCR D
    JNZ OTTA           ; LEE DEL FCB Y LO DEJA EN MEMORIA
    MVI A,20H
    STAX B
    INX B
    STAX B
    INX B
    STAX B
    INX B
    STAX B
    INX B
    STAX B
    INX B
    DCR E
    JNZ RETOR
    CALL PORTA
    MVI E,04H
    RET
RETOR:

```

```

PORTA:  MVI A,0DH           ; ORDENA PRESENTACION EN PANTALLA
        STAX B
        INX B
        MVI A,0AH
        STAX B
        INX B
        PUSH B             ; DEJA ENTRE NOMBRES EL DRIVE
        PUSH H
        LXI H,DRIVE       ; AVERIGUA EL DRIVE
        MOV A,M
        POP H
        POP B
        STAX B
        INX B
        MVI A,3AH        ; DEJA AL FINAL 'A:'
        STAX B
        INX B
        MVI A,20H
        STAX B
        INX B
        RET

```

```

CADRE:  LXI H,DRIVEE     ; AVERIGUA EL DRIVE ACTUAL
        MOV A,M
        LXI H,005CH     ; CAMBIA EL DRIVE EN EL FCB
        MOV M,A
        ADI 40H

```

```
LXI H,DRIVE      ;ALMACENA EL DRIVE PARA PRESENTARLO
MOV M,A          ;EN PANTALLA
LXI H,TEXPOR + 3
MOV M,A
RET
```

```
VER:  IN 51H
      ANI 02H      ;REVISA SI LLEGO UN NUEVO DATO
      INR A
      DCR A
      JZ VER
      RET
```

```
DESAC: PUSH PSW
      MVI A,05H
      OUT 51H      ;DESACTIVA EL TX DEL OTRO EQUIPO
      POP PSW
      RET
```

```
ACTIV: PUSH PSW
      MVI A,07H
      OUT 51H      ;ACTIVA EL TX DEL OTRO EQUIPO
      POP PSW
      RET
```

```
LIMPIE: PUSH PSW
      PUSH B
      PUSH D
      PUSH H
      LXI H,0080H
      MVI B,80H    ;CARGA EN EL DMA CON CF HEX
      MVI A,0CFH
```

```
LALLA: MOV M,A
      INX H
      DCR B
      JNZ LALLA
      POP H
      POP D
      POP B
      POP PSW
      RET
```

```

RCOMPA:  PUSH B
          PUSH D
          LXI D, RDIREC      ; INICIALIZA PUNTERO
          LDAX D
          CMP L
          JNZ RCMAS
          INX D              ; AVERIGUA EL TERMINO DE LA DATA
          LDAX D            ; ALMACENADA EN MEMORIA
          CMP H
          JNZ RCMAS
          MVI A, 00H
          JMP RRETOR
RCMAS:   MVI A, 01H
RRETOR:  POP D
          POP B
          RET

```

```

RESCRI:  PUSH B
          PUSH D
          PUSH H
          MVI A, 00H
          MVI C, 15H
          LXI D, 005CH      ; ESCRIBE EN EL DISCO
          CALL 5H
          CPI 00H
          JZ RESCR
          CALL RESMEN
RESCR:   POP H
          POP D
          POP B
          RET

```

```

RESMEN:  PUSH B
          PUSH D
          PUSH H
          MVI C, 9H
          LXI D, RNOESP     ; MENSAJE DE NO ESPACIO EN EL DISCO
          CALL 5H
          CALL LEER
          CPI 0DH
          JZ RESALL
          MVI A, 05H
          JMP RESTOO
RESALL:  MVI A, 01H
RESTOO:  POP H
          POP D
          POP B
          RET

```

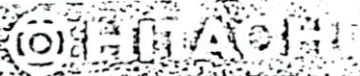
```
DECIS:  PUSH PSW
        PUSH B
        PUSH D
        PUSH H
        MVI C, 9H
        LXI D, TEDESI ; PRESENTA RECOMENDACIONES
        CALL 5H
        CALL LEER
        CPI ODH ; AVERIGUA COMO TRANSMITIR LA DATA
        JNZ DEALLA
        POP H
        INX H
        INX H
        INX H
        PUSH H
DEALLA: POP H
        POP D
        POP B
        POP PSW
        RET

        END PARTE
```

ORIGINAL

HITACHI IC MEMORIES

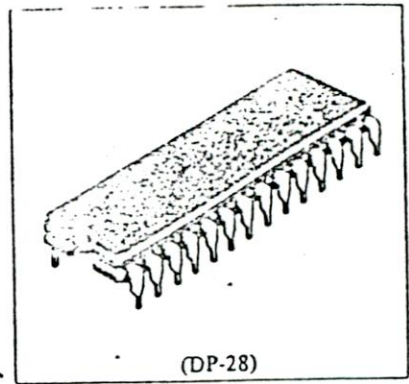
HM6264LP-10, HM6264LP-12, HM6264LP-15



3192 word x 8-bit High Speed Static CMOS RAM

■ FEATURES

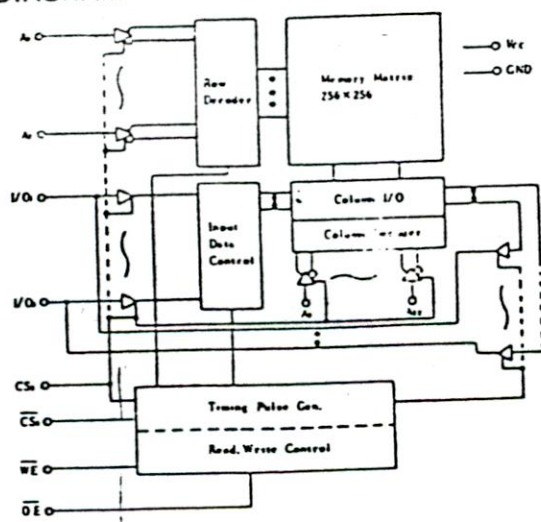
- Fast access Time 100ns/120ns/150ns (max.)
- Low Power Standby Standby: 0.01mW (typ.)
- Low Power Operation Operating: 200mW (typ.)
- Capability of Battery Back-up Operation
- Single +5V Supply
- Completely Static Memory. . . . No clock or Timing Strobe Required
- Equal Access and Cycle Time
- Common Data Input and Output, Three State Output
- Directly TTL Compatible: All Input and Output
- Standard 28pin Package Configuration
- Pin Out Compatible with 64K EPROM HN482764



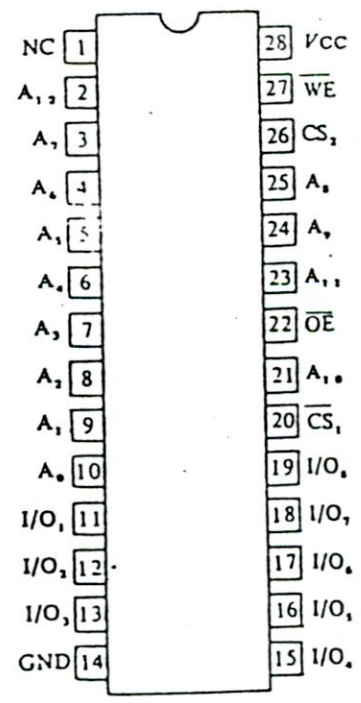
(DP-28)

6264

■ BLOCK DIAGRAM



■ PIN APPRANGEMENT



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Rating	Unit
Terminal Voltage *	V_T	-0.5 ** to +7.0	V
Power Dissipation	P_T	1.0	W
Operating Temperature	T_{opr}	0 to +70	°C
Storage Temperature	T_{stg}	-55 to +125	°C
Storage Temperature (Under Bias)	T_{bias}	-10 to +85	°C

* With respect to GND. ** Pulse width 50ns: -3.0V

■ TRUTH TABLE

WE	CS1	CS2	OE	Mode	I/O Pin	VCC Current	Note
X	H	X	X	Not Selected (Power Down)	High Z	ISB, ISB1	
X	X	L	X		High Z	ISB, ISB2	
H	L	H	H	Output Disabled	High Z	ICC, ICC1	
H	L	H	L	Read	Dout	ICC, ICC1	
L	L	H	H	Write	Din	ICC, ICC1	Write Cycle (1)
L	L	H	L		Din	ICC, ICC1	Write Cycle (2)

■ RECOMMENDED DC OPERATING CONDITIONS ($T_a = 0$ to $+70^\circ\text{C}$)

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}	4.5	5.0	5.5	V
	GND	0	0	0	V
Input Voltage	V_{IH}	2.2	-	5.0	V
	V_{IL}	-0.3*	-	0.8	V

* Pulse Width 50ns: -3.0V

■ DC AND OPERATING CHARACTERISTICS ($V_{CC} = 5V \pm 10\%$, GND = 0V, $T_a = 0$ to $+70^\circ\text{C}$)

Item	Symbol	Test Condition	min	typ*	max	Unit
Input Leakage Current	I_{LI}	$V_{in} = \text{GND to } V_{CC}$	-	-	2	μA
Output Leakage Current	I_{LOI}	$\overline{\text{CS1}} = V_{IH}$ or $\text{CS2} = V_{IL}$ or $\overline{\text{OE}} = V_{IH}$, $V_{I/O} = \text{GND to } V_{CC}$	-	-	2	μA
Operating Power Supply Current	I_{CC}	$\overline{\text{CS1}} = V_{IL}$, $\text{CS2} = V_{IH}$, $I_{I/O} = 0\text{mA}$	-	40	80	mA
Average Operating Current	I_{CC1}	Min. cycle, duty=100%, $\overline{\text{CS1}} = V_{IL}$, $\text{CS2} = V_{IH}$	-	60	110	mA
Standby Power Supply Current	I_{SB}	$\overline{\text{CS1}} = V_{IH}$ or $\text{CS2} = V_{IL}$, $I_{I/O} = 0\text{mA}$	-	1	3	mA
	I_{SB1}^{**}	$\overline{\text{CS1}} \geq V_{CC} - 0.2\text{V}$, $\text{CS2} \geq V_{CC} - 0.2\text{V}$ or $\text{CS2} \leq 0.2\text{V}$	-	2	100	μA
	I_{SB2}^{**}	$\text{CS2} \leq 0.2\text{V}$	-	2	100	μA
Output Voltage	V_{OL}	$I_{OL} = 2.1\text{mA}$	-	-	0.4	V
	V_{OH}	$I_{OH} = -1.0\text{mA}$	2.4	-	-	V

* Typical limits are at $V_{CC} = 5.0\text{V}$, $T_a = 25^\circ\text{C}$ and specified loading.
 ** V_{IL} min = -0.3V

■ CAPACITANCE ($f = 1\text{MHz}$, $T_a = 25^\circ\text{C}$)

Item	Symbol	Test Condition	typ	max	Unit
Input Capacitance	C_{in}	$V_{in} = 0\text{V}$	-	6	pF
Input/Output Capacitance	$C_{I/O}$	$V_{I/O} = 0\text{V}$	-	8	pF

Note) This parameter is sampled and not 100% tested.

■ AC CHARACTERISTICS ($V_{CC} = 5V \pm 10\%$, $T_a = 0$ to $+70^\circ\text{C}$)

• AC TEST CONDITIONS

Input Pulse Levels: 0.8 to 2.4V

Input Rise and Fall Times: 10ns

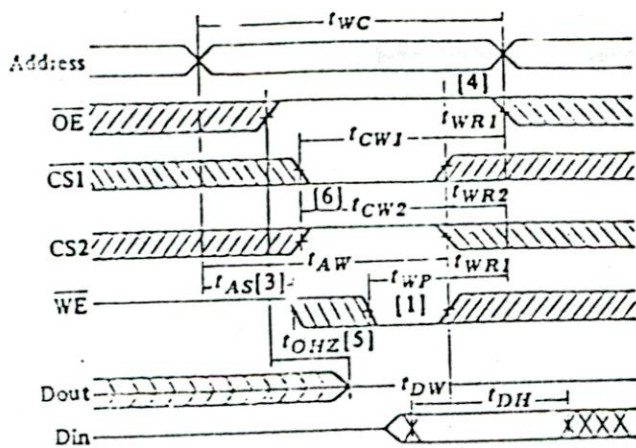
Input and Output Timing Reference Level: 1.5V

Output Load: 1TTL Gate and $C_L = 100\text{pF}$ (including scope and jig)

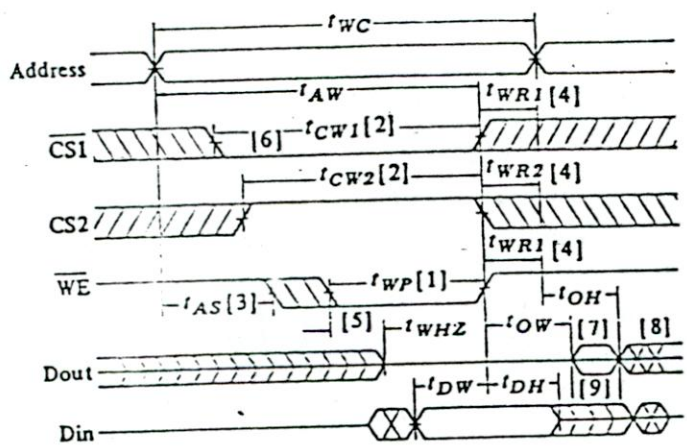
• READ CYCLE

Item	Symbol	HM6264LP-10		HM6264LP-12		HM6264LP-15		Unit	
		min	max	min	max	min	max		
Read Cycle Time	t_{RC}	100	-	120	-	150	-	ns	
Address Access Time	t_{AA}	-	100	-	120	-	150	ns	
Chip Selection to Output	$\overline{\text{CS1}}$	t_{CO1}	-	100	-	120	-	150	ns
	CS2	t_{CO2}	-	100	-	120	-	150	ns
Output Enable to Output Valid	t_{OE}	-	50	-	60	-	70	ns	
Chip Selection to Output in Low Z	$\overline{\text{CS1}}$	t_{LZ1}	10	-	10	-	15	-	ns
	CS2	t_{LZ2}	10	-	10	-	15	-	ns
Output Enable to Output in Low Z	t_{OLZ}	5	-	5	-	5	-	ns	
Chip Deselection to Output in High Z	$\overline{\text{CS1}}$	t_{HZ1}	0	35	0	40	0	50	ns
	CS2	t_{HZ2}	0	35	0	40	0	50	ns
Output Disable to Output in High Z	t_{OHZ}	0	35	0	40	0	50	ns	
Output Hold from Address Change	t_{OH}	10	-	10	-	15	-	ns	

NOTES: 1 t_{HZ} and t_{OHZ} are defined as the time at which the outputs achieve the open circuit condition and are not returned to output voltage levels.
 2 At any given temperature and voltage condition, t_{HZ} max is less than t_{LZ} min both for a given device and from device to device.



• WRITE CYCLE (2) (OE Low Fix)



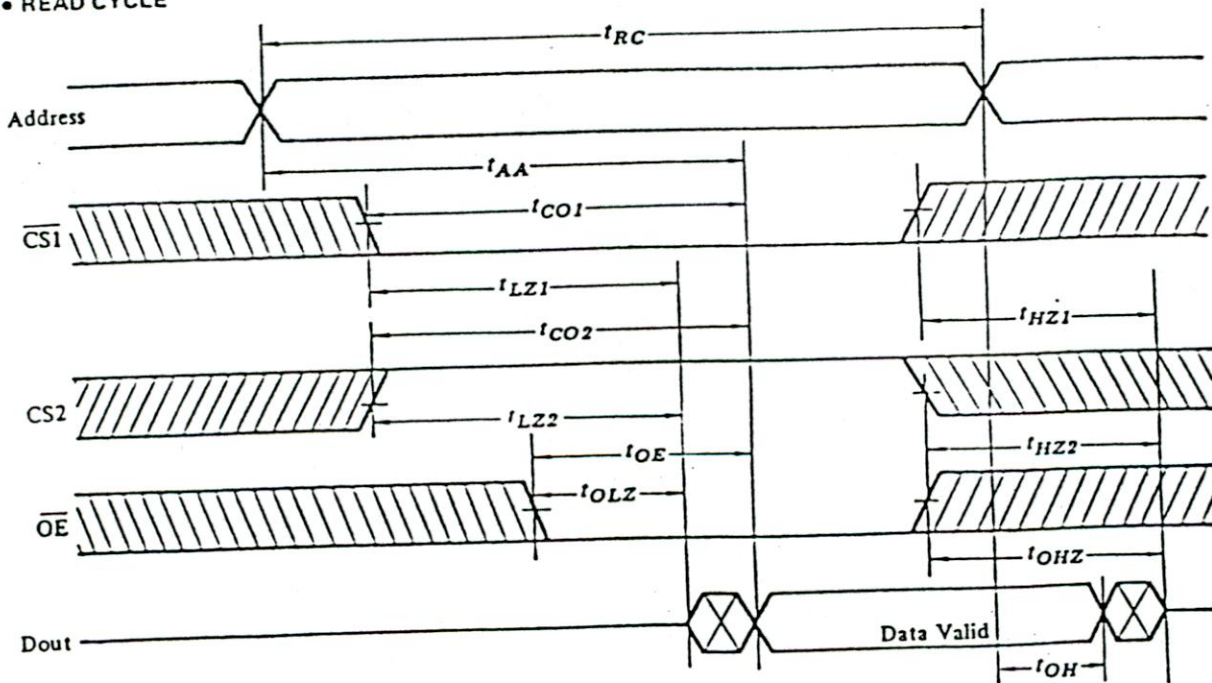
- NOTES:
- 1) A write occurs during the overlap of a low $\overline{CS1}$, a high $CS2$ and a low \overline{WE} . A write begins at the latest transition among $\overline{CS1}$ going low, $CS2$ going high and \overline{WE} going low. A write ends at the earliest transition among $\overline{CS1}$ going high, $CS2$ going low and \overline{WE} going high. t_{WP} is measured from the beginning of write to the end of write.
 - 2) t_{CW} is measured from the later of $\overline{CS1}$ going low or $CS2$ going high to the end of write.
 - 3) t_{AS} is measured from the address valid to the beginning of write.
 - 4) t_{WR} is measured from the end of write to the address change.
 t_{WR1} applies in case a write ends at $\overline{CS1}$ or \overline{WE} going high.
 t_{WR2} applies in case a write ends at $CS2$ going low.
 - 5) During this period, I/O pins are in the output state, therefore the input signals of opposite phase to the outputs must not be applied.
 - 6) If $\overline{CS1}$ goes low simultaneously with \overline{WE} going low or after \overline{WE} going low, the outputs remain in high impedance state.
 - 7) $Dout$ is in the same phase of written data of this cycle.
 - 8) $Dout$ is the read data of the new address.
 - 9) If $\overline{CS1}$ is low and $CS2$ is high during this period, I/O pins are in the output state. Therefore, the input signals of opposite phase to the outputs must not be applied to them.

• WRITE CYCLE

Item	Symbol	HM6264LP-10		HM6264LP-12		HM6264LP-15		Unit	
		min	max	min	max	min	max		
Write Cycle Time	t_{WC}	100	—	120	—	150	—	ns	
Chip Selection to End of Write	t_{CW}	80	—	85	—	100	—	ns	
Address Setup Time	t_{AS}	0	—	0	—	0	—	ns	
Address Valid to End of Write	t_{AW}	80	—	85	—	100	—	ns	
Write Pulse Width	t_{WP}	60	—	70	—	90	—	ns	
Write Recovery Time	CS1, WE	t_{WR1}	5	—	5	—	10	—	ns
	CS2	t_{WR2}	15	—	15	—	15	—	ns
Write to Output in High Z	t_{WHZ}	0	35	0	40	0	50	ns	
Data to Write Time Overlap	t_{DW}	40	—	50	—	60	—	ns	
Data Hold from Write Time	t_{DH}	0	—	0	—	0	—	ns	
OE to Output in High Z	t_{OHZ}	0	35	0	40	0	50	ns	
Output Active from End of Write	t_{OW}	5	—	5	—	10	—	ns	

■ TIMING WAVEFORM

• READ CYCLE



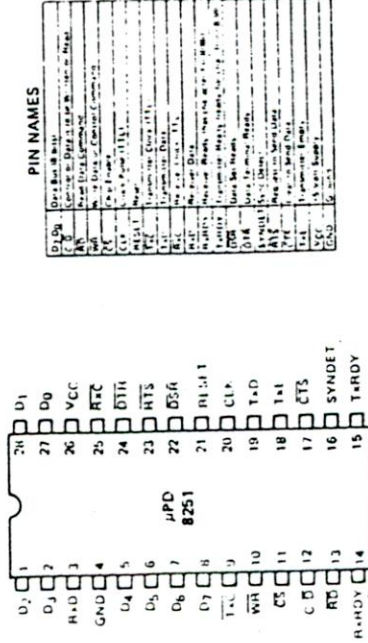
- NOTES: 1) \overline{WE} is high for Read Cycle
 2) When $\overline{CS1}$ is Low and CS2 is High the address input must not be in the high impedance state.

PROGRAMMABLE COMMUNICATION INTERFACE

DESCRIPTION The μPD8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is designed for microcomputer systems data communications. The USART is used as a peripheral and is programmed by the μPD8080 or other processor to communicate in commonly used serial data transmission techniques including IBM Bi-Sync. The USART receives serial data streams and converts them into parallel data characters for the processor. While receiving serial data, the USART will also accept data characters from the processor in parallel format, convert them to serial format and transmit. The USART will signal the processor when it has completely received or transmitted a character and requires service. Complete USART STATUS including data format errors and control signals such as TXE and SYNDET is available to the processor at any time.

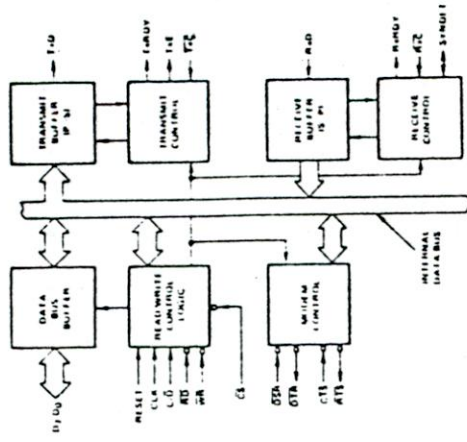
- FEATURES**
- Asynchronous or Synchronous Operation
 - Asynchronous
 - 5.8 Bit Characters
 - Clock Rate - 1, 16 or 64 x Baud Rate
 - Break Character Generation
 - Select 1, 1-1/2, or 2 Stop Bits
 - False Start Bit Detector
 - Synchronous:
 - 5.8 Bit Characters
 - Internal or External Character Synchronization
 - Automatic Sync Insertion
 - Single or Double Sync Characters
 - Baud Rate - Synchronous - DC to 56K Baud
 - Asynchronous - DC to 9.6K Baud
 - Full Duplex, Double Buffered Transmitter and Receiver
 - Parity, Overrun and Framing Flags
 - Fully Compatible with 8080
 - All Inputs and Outputs are TTL Compatible
 - Single +5 Volt Supply
 - Separate Device, Receive and Transmit TTL Clocks
 - 28 Pin Plastic DIP Package
 - N Channel MOS Technology

PIN CONFIGURATION



In the receive mode, a communication interface device must convert incoming serial format data into parallel data and make certain format checks on the data. And in the transmit mode, the device must format data into serial data. The device must also supply or remove characters or bits that are unique to the communication format in use. By performing conversion and formatting services automatically, the USART appears to the processor as a simple or "transparent" input or output of byte oriented parallel data.

BLOCK DIAGRAM



BASIC OPERATION

C/D	RD	WR	CS
0	0	1	0
0	1	0	0
1	0	1	0
1	1	0	0
X	X	X	1
X	1	1	0

0 8251 → Data Bus
Data Bus → 8251
Status → Data Bus
Data Bus → Control
Data Bus → 3 State

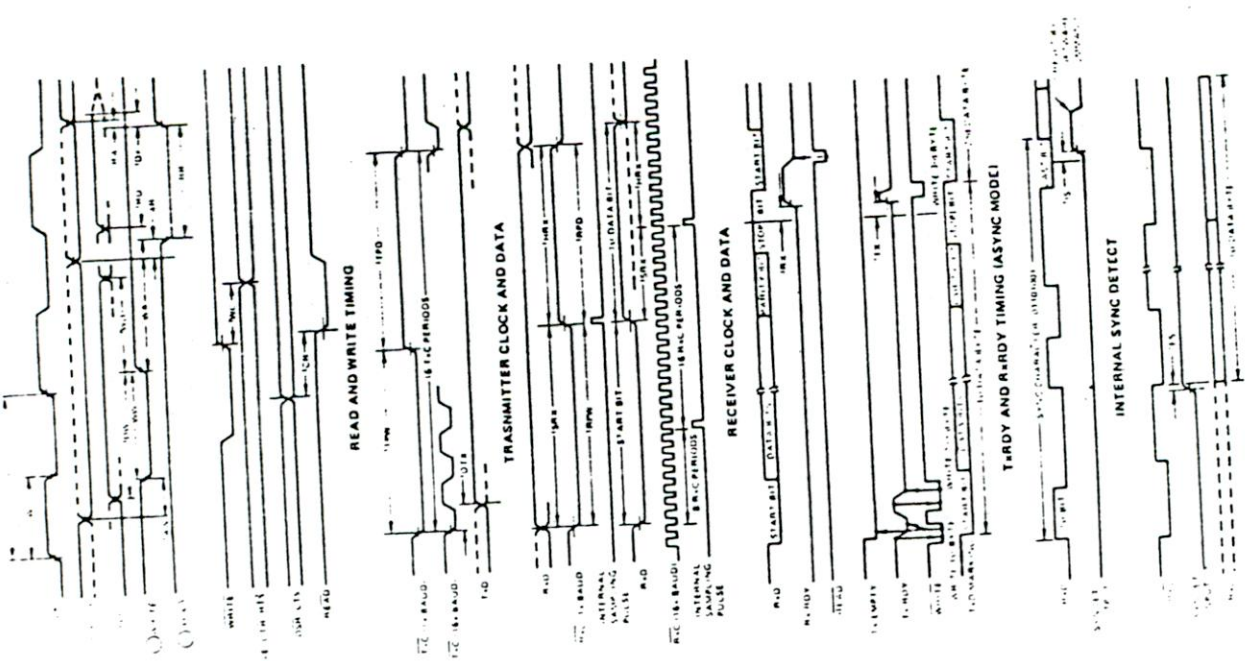
ABSOLUTE MAXIMUM RATINGS*

- Operating Temperature: 0°C to +70°C
- Storage Temperature: 65°C to +125°C
- All Output Voltages: 0.5 to +7 Volts
- All Input Voltages: 0.5 to +7 Volts
- Supply Voltages: 0.5 to +7 Volts

COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*T_a = 25°C

NO.	PIN		FUNCTION
	SYMBOL	NAME	
1, 2, 27, 28, 5, 8	D ₇ - D ₀	Data Bus Buffer	An 8 bit, 3 state bi-directional buffer used to interface the 8251 to the processor data bus. Data is transmitted or received by the buffer in response to output/output or Read/Write strobe from the processor. The Data Bus Buffer also transfers Command word, Command status, and Status.
16, 4	V _{CC} GND	V _{CC} Supply Voltage Ground	V _{CC} supply Ground
Read/Write Control Logic			
21	RESET	Reset	This logic block accepts inputs from the processor Control Bus and generates control signals for overall USART operation. The Mode Instruction and Command Instruction registers that store the control format for device logic (total definition are located in the Read/Write Control Logic.
20	CLK	Clock Pulse	A "one" on this input forces the USART into the "idle" mode where it will remain until reinitiated with a new set of control words. Minimum RESET pulse width is 100 ns.
10	WR	Write Data	The CLK input provides for internal device timing and is usually connected to the Phase 2 (TTL) output of the 8255 Clock Generator. External inputs and outputs are not referenced to CLK, but the CLK frequency must be 30 times the Receiver or Transmitter clocks in the asynchronous mode and 4.5 times for the synchronous mode.
13	RD	Read Data	A "zero" on this input instructs the 8251 to accept the data or control word which the processor is writing out to the USART via the data bus.
12	C/D	Control/Data	A "zero" on this input instructs the 8251 to place the data or status information onto the Data Bus for the processor to read.
11	CS	Chip Select	The Control Data input, in conjunction with the WR and RD inputs, informs USART to accept or provide either a data character, control word or status information via the Data Bus. 0 = Data, 1 = Control
Modem Control			
22	DSR	Data Set Ready	A "zero" on this input enables the USART for reading and writing to the processor.
24	DTR	Data Terminal Ready	The Data Set Ready input can be tested by the processor via Status information. The DSR input is normally used to test Modem Data Set Ready condition.
23	RTS	Request to Send	The Data Terminal Ready output can be controlled via the Command word. The DTR output is normally used to drive Modem Data Terminal Ready or Line Select lines.
17	CTS	Clear to Send	The Request to Send output can be controlled via the Command word. The CTS output is normally used to drive the Modem Request to Send line.
<p>A "zero" on the Clear to Send input enables the USART to transmit serial data if the 1-CLK bit in the Command Instruction register is enabled (one).</p>			



Note: Write and Read pulses have no timing limitation with respect to CLK.

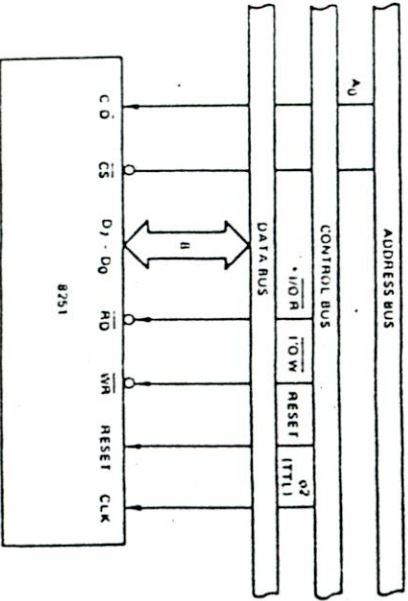
TRANSMIT BUFFER
CONVERTER

The Transmitt Buffer/Converter receives parallel data from the Data Bus Buffer via the internal data bus, converts parallel to serial data, inserts the necessary characters or bits needed for the programmed communication format and outputs composite serial data on the TxD output.

PIN IDENTIFICATION
(CONT 1)

NO	SYMBOL	PIN NAME	FUNCTION
Transmit Control Logic			
15	TARDY	Transmitter Ready	Transmitter Ready signal the processor that the Transmitter is ready to send a data character. TARDY can be used as an interrupt or may be tested through the Status Information for Poll operation. Latching a character from the processor automatically resets TARDY.
18	TxE	Transmitter Empty	The Transmitter Empty output signals the processor that the USART has no further characters to transmit. TxE is automatically reset upon receiving a data character from the processor. In half duplex, TxE can be used to signal end of a transmission. The TxE bit is used to turn the Transmitter off. The TxE bit's latching operation does not affect TxE. In the Synchronous mode, a "one" on this output indicates that a Sync character or characters are about to be automatically transmitted as "filler" because the next data character has not been loaded.
9	TxC	Transmitter Clock	The Transmitter Clock controls the serial clock for transmission rate. In the Asynchronous mode, the TxC frequency is a multiple of the actual Baud Rate. Two bits of the Mode Instruction select the multiple to be 1x, 16x, or 64x the Baud Rate. In the Synchronous mode, the TxC frequency is automatically selected to equal the actual Baud Rate. Note that for both Synchronous and Asynchronous modes, serial data is shifted out of the USART by the falling edge of TxC.
19	TxD	Transmitter Data	The Transmitt Control Logic outputs the composite serial data stream on this pin.

8251 INTERFACE TO 8080
STANDARD SYSTEM BUS



RECEIVER BUFFER

The Receiver Buffer accepts serial data input at the RxD pin and converts the data from serial to parallel format. Bits or characters required for the specific communication technique in use are checked and then an eight bit "assembled" character is created for the processor. For communication techniques which require less than eight bits, the μ PD8251 sets the extra bits to "zero".

PIN IDENTIFICATION
(CONT 1)

NO	SYMBOL	PIN NAME	FUNCTION
Receiver Control Logic			
14	RxDY	Receiver Ready	The Receiver Ready output indicates that the Receiver Buffer is ready with an "assembled" character for input to the processor. For Poll operation, the processor can check RxDY using a Status Read or RxDYV can be connected to the processor interrupt structure. Note that reading the character to the processor automatically resets RxDY.
25	RxC	Receiver Clock	The Receiver Clock is the rate at which the incoming character is received. In the Asynchronous mode, the RxC frequency may be 1, 16 or 64 times the actual Baud Rate but in the Synchronous mode the RxC frequency must equal the Baud Rate. Two bits on the mode instruction select Asynchronous at 1x, 16x or 64x or Synchronous operation at 1x the Baud Rate. Units TxC, data is sampled by the μ PD8251 on the rising edge of RxC.
3	RxD	Receiver Data	A composite serial data stream is received by the Receiver Control Logic on this pin.
16	SYNDET	Sync Detect	The SYNC Detect pin is only used in the Synchronous mode. The μ PD8251 may be programmed to either the internal or external Sync mode and SYNDET then functions as an output or input, respectively. In the internal Sync mode, the SYNDET output will go to a "one" when the μ PD8251 has located the SYNC character on the Receiver module. If double SYNC character (by typed operation has been programmed), SYNDET will go to "one" in the middle of the last bit of the second SYNC character. SYNDET is automatically reset to zero on a Status Read or RESET. In the external Sync mode, a "zero" to "one" transition on the SYNDET input will cause the μ PD8251 to start assembling data character on the next falling edge of RxC. The length of the SYNDET input should be at least one period, but may be removed once the μ PD8251 is in SYNC.

Note: Since the μ PD8251 will frequently be handling both the reception and transmission for a given link, the Receiver and Transmitt Baud Rates will be same. RxC and TxC then require the same frequency and may be tied together and connected to a single clock source or Baud Rate Generator.

- Examples:
- If the Baud Rate equals 110 (Asynch)
 - RxC or TxC equals 110 Hz (1x)
 - RxC or TxC equals 176 KHz (16x)
 - RxC or TxC equals 704 KHz (64x)
 - If the Baud Rate equals 300
 - RxC or TxC equals 300 Hz (1x)
 - RxC or TxC equals 4800 Hz (16x)
 - RxC or TxC equals 19200 Hz (64x)

OPERATIONAL DESCRIPTION

A set of control words must be sent to the μ PD8251 to define the desired mode and command format. The control words will specify the BAUD RATE FACTOR (1x, 16x, 64x), CHARACTER LENGTH (5 to 8), NUMBER OF STOP BITS (1, 1.5, 2), ASYNCHRONOUS or SYNCHRONOUS MODE, SYNDET (IN or OUT), PARITY, etc.

After receiving the control words, the μ PD8251 is ready to communicate. TxRDY is raised to signal the processor that the USART is ready to receive a character for transmission. When the processor writes a character to the USART, TxRDY is automatically reset.

Concurrently, the μ PD8251 may receive serial data; and after receiving an entire character, the RxRDY output is raised to indicate a completed character is ready for the processor. The processor fetch will automatically reset RxRDY.

Note: The μ PD8251 may provide faulty RxRDY for the first read after power-on or for the first read after receive is re-enabled by a command instruction (RxIE). A dummy read is recommended to clear faulty RxRDY. But this is not the case for the first read after hardware or software reset after the device operation has once been established.

The μ PD8251 cannot transmit until the TxEN (Transmitter Enable) bit has been set by a Command Instruction and until the CTS (Clear to Send) input is a "zero". TxD is held in the "marking" state after Reset awaiting new Command Words.

The USART must be loaded with a group of two to four control words provided by the processor before data reception and transmission can begin. A Reset (internal or external) must immediately proceed the control words which are used to program the complete operational description of the communications interface. If an external RESET is not available, three successive 00 Hex or two successive 80 Hex command instructions (C/D = 1) followed by a software reset command instruction (40 Hex) can be used to initialize the 8251.

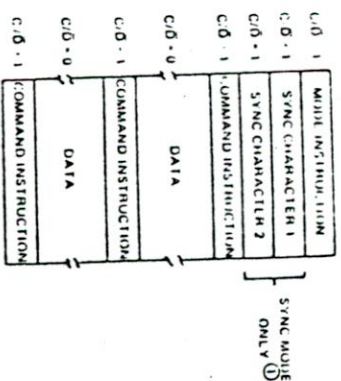
There are two control word formats:

1. Mode Instruction
2. Command Instruction

MODE INSTRUCTION
This control word specifies the general characteristics of the interface regarding the SYNCHRONOUS or ASYNCHRONOUS MODE, BAUD RATE FACTOR, CHARACTER LENGTH, PARITY, and NUMBER OF STOP BITS. Once the Mode Instruction has been received, SYNC characters or Command Instructions may be inserted depending on the Mode Instruction content.

COMMAND INSTRUCTION
This control word will be interpreted as a SYNC character definition if immediately preceded by a Mode Instruction which specified a Synchronous format. After the SYNC character(s) are specified or after an Asynchronous Mode Instruction, all subsequent control words will be interpreted as an update to the Command Instruction. Command Instruction updates may occur at any time during the data block. To modify the Mode Instruction, a bit may be set in the Command Instruction which causes an internal Reset which allows a new Mode Instruction to be accepted.

TYPICAL DATA BLOCK



NOTE: The second SYNC character is skipped if MODE instruction has programmed the 8251 to single character internal SYNC mode. Both SYNC characters are skipped if MODE instruction has programmed the 8251 to ASYNC mode.

The μ PD8251 can operate in either Asynchronous or Synchronous communication modes. Understanding how the Mode Instruction controls the functional operation of the USART is easiest when the device is considered to be two separate circuits: one asynchronous and the other synchronous, which share the same support circuits and package. Although the format definition can be changed at will or "on the fly", the two modes will be explained separately for clarity.

When a data character is written into the μ PD8251, the USART automatically adds a START bit (low level or "space") and the number of STOP bit (high level or "mark") specified by the Mode Instruction. If Parity has been enabled, an odd or even Parity bit is inserted just before the STOP bit(s), as specified by the Mode Instruction. Then, depending on CTS and TxEN, the character may be transmitted as a serial data stream at the TxD output. Data is shifted out by the falling edge of $\overline{\text{TxC}}$ at $\overline{\text{TxC}}/16$ or $\overline{\text{TxC}}/64$, as defined by the Mode Instruction.

If no data characters have been loaded into the μ PD8251, or if all available characters have been transmitted, the TxD output remains "high" (marking) in preparation for sending the START bit of the next character provided by the processor. TxD may be forced to send a BREAK (continuously low) by setting the correct bit in the Command Instruction.

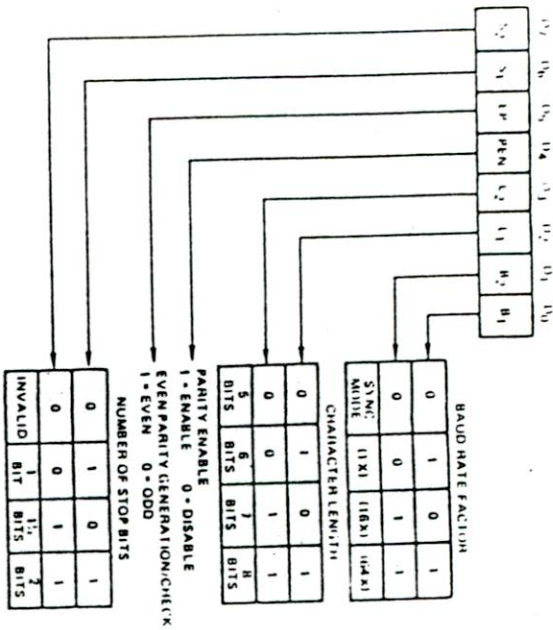
The RxD input line is normally held "high" (marking) by the transmitting device. A falling edge at RxD signals the possible beginning of a START bit and a new character. The START bit is checked by looking for a "low" at its normal center character. The START bit is detected again, it is considered valid, as specified by the BAUD HATE II. A "low" is detected again, it is considered valid, and the bit assembling counter starts counting. The bit counter locates the adjacent mate center of the data, parity (if specified), and STOP bit. The parity error flag (PE) is set, if a parity error occurs. Input bits are sampled at the RxD pin with the rising edge of $\overline{\text{RxC}}$. If a high is not detected for the STOP bit, which normally signals the end of an input character, a framing error (FE) will be set. After a valid STOP bit, the input character is loaded into the parallel Data Bus Buffer of the μ PD8251 and the RxRDY signal is raised to indicate to the processor that a character is ready to be fetched. If the processor has failed to fetch the previous character, the new character replaces the old and the overrun flag (OE) is set. All the error flags can be reset by setting a bit in the Command Instruction. Error flag conditions will not stop subsequent USART operation.

MODE INSTRUCTION DEFINITION

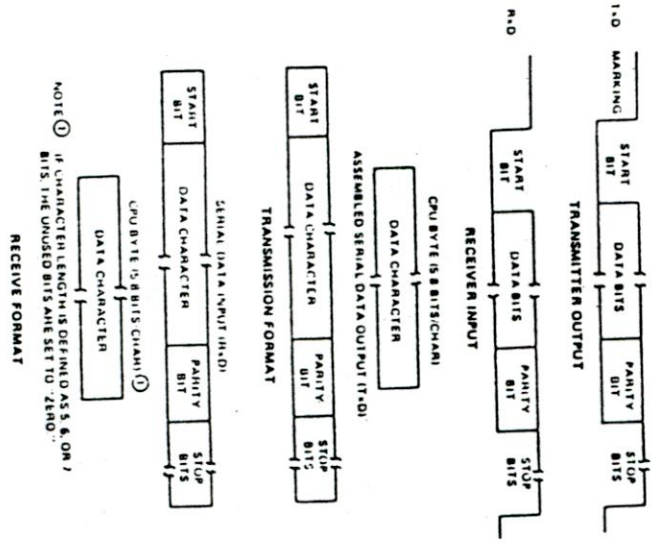
ASYNCHRONOUS TRANSMISSION

ASYNCHRONOUS RECEIVE

MODE INSTRUCTION FORMAT ASYNCHRONOUS MODE



TRANSMIT/RECEIVE FORMAT ASYNCHRONOUS MODE

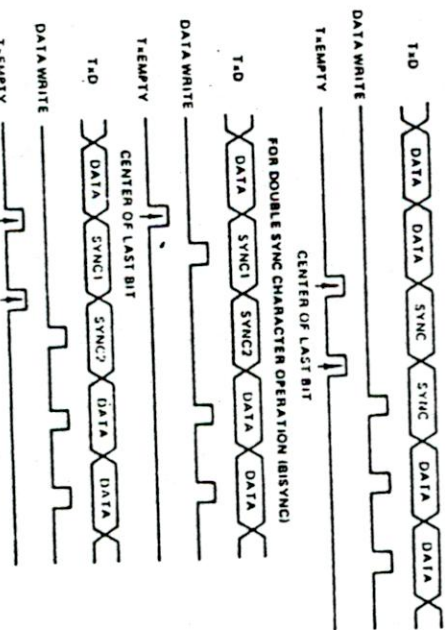


SYNCHRONOUS TRANSMISSION

As an Asynchronous transmission, the TxD output remains "high" (marking) until the jPDB251 receives the first character from the processor which is usually a SYNC character. After a Command Instruction has set TxEH and after Clear to Send (CTS) goes low, the first character is serially transmitted. Data is shifted out on the falling edge of TxC and the same rate as TxC.

Once transmission has started, Synchronous Mode format requires that the serial data stream at TxD continue at the TxC rate or SYNC will be lost. If a data character is not provided by the processor before the jPDB251 Transmitter Buffer becomes empty, the SYNC character(s) flushed directly following the Mode Instruction will be automatically inserted in the TxD data stream. The SYNC character(s) are inserted to fill the line and maintain synchronization until new data characters are available for transmission. If the jPDB251 becomes empty, and must send the SYNC character(s), the TxEEMPTY output is raised to signal the processor that the Transmitter Buffer is empty and SYNC characters are being transmitted. TxEEMPTY is automatically reset by the next character from the processor.

TxEEMPTY goes high at the middle of the last data bit when the Transmit Register is EMPTY. TxEEMPTY goes low again as sync characters are transmitted. See figure below.

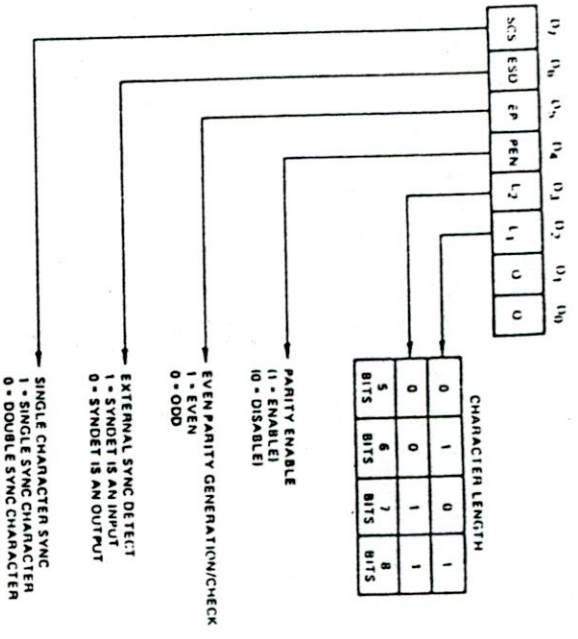


SYNCHRONOUS RECEIVE

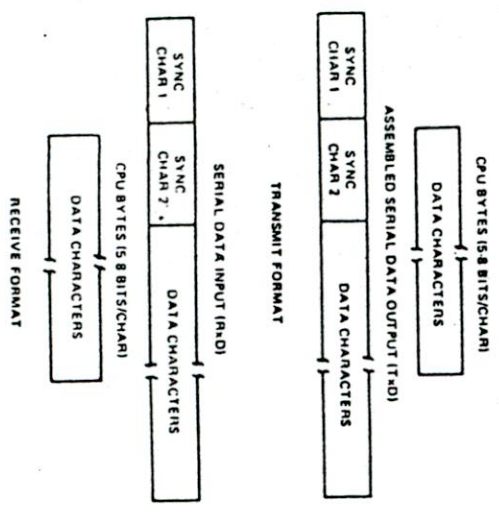
Incoming data on the RxD input is sampled on the rising edge of TxC and the Receiver Buffer is compared with the first SYNC character after each bit has been loaded until a match is found. If two SYNC characters have been programmed, the next received character is also compared. When the SYNC character(s) programmed have been detected, the jPDB251 leaves the HUNT mode and is in character synchronization. At this time, the SYNDET (output) is set high. SYNDET is automatically reset by a STATUS READ.

If external SYNC has been specified in the Mode Instruction, a "one" applied to the SYNDET (input) for at least one RxC cycle will synchronize the USART. Parity and Overrun Errors are treated the same in the Synchronous as in the Asynchronous Mode. Framing errors do not apply in the Synchronous format. The processor may command the receiver to enter the HUNT mode with a Command Instruction which sets Enter HUNT (EH). If synchronization is lost.

MODE INSTRUCTION
FORMAT
SYNCHRONOUS MODE



TRANSMIT/RECEIVE
FORMAT
SYNCHRONOUS MODE



COMMAND INSTRUCTION
FORMAT

After the functional definition of the μPD8251 has been specified by the Mode Instruction and the SYNC character(s) have been entered, if in SYNC mode, the USART is ready to receive Command Instructions and begin communication. A Command Instruction is used to control the specific operation of the format selected by the Mode Instruction. Enable Transmit, Enable Receive, Error Reset and Modern Controls are controlled by the Command Instruction.

After the Mode Instruction and the SYNC character(s), as needed, are loaded, all subsequent "control writes" ($C/D = 1$) will load or overwrite the Command Instruction register. A Reset operation (internal via CMD IR or external via the RESET input) will cause the μPD8251 to interpret the next "control write", which must immediately follow the reset, as a Mode Instruction.

STATUS READ FORMAT

It is frequently necessary for the processor to examine the "status" of an active interface device to determine if errors have occurred or to notice other conditions which require a response from the processor. The μPD8251 has features which allow the processor to "read" the device status at any time. A data fetch is issued by the processor while holding the C/D input "high" to obtain device Status Information. Many of the bits in the status register are copies of external pins. This dual status arrangement allows the μPD8251 to be used in both Polled and interrupt driven environments. Status updates can have a maximum delay of a 16 clock period.

PARITY ERROR

When a parity error is detected, the PE flag is set, and is cleared by setting the ER bit in a subsequent Command Instruction. PE being set does not inhibit USART operation.

OVERRUN ERROR

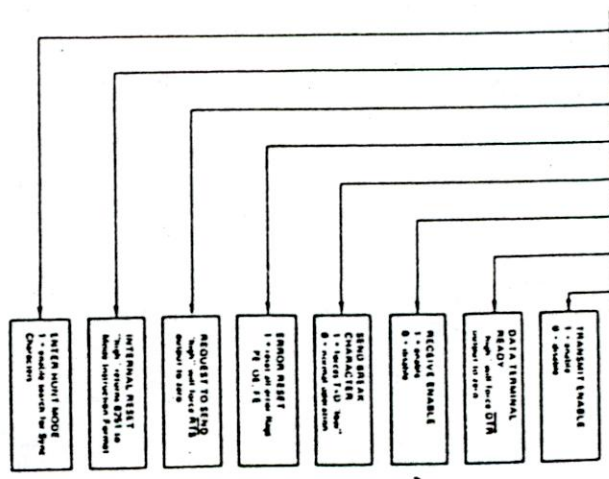
If the processor fails to read a data character before the one following is available, the OE flag is set, and is cleared by setting the ER bit in a subsequent Command Instruction. Although OE being set does not inhibit USART operation, the previously received character is overwritten and lost.

FRAMING ERROR

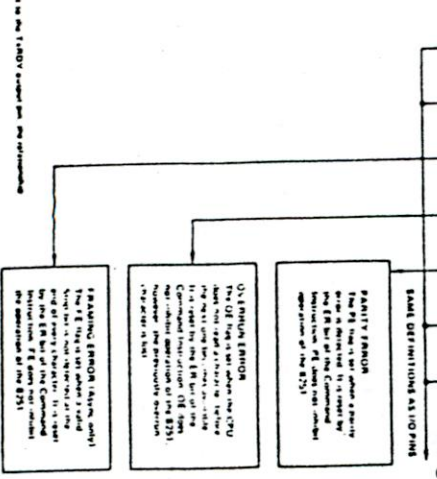
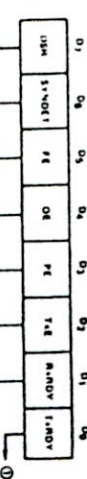
If a valid STOP bit is not detected at the end of a character, the FE flag is set, and is cleared by setting the ER bit in a subsequent Command Instruction. FE being set does not inhibit USART operation.

Note: ① ASYNC mode only.

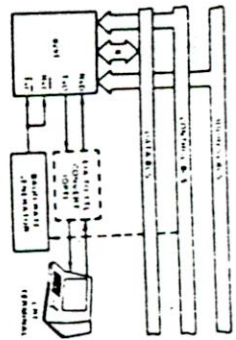
COMMAND INSTRUCTION
FORMAT



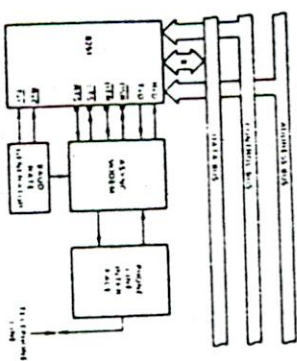
STATUS READ FORMAT



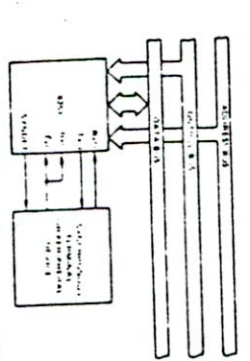
Note: 1. A DTR signal that is not within specification on the 1480V computer can be interpreted as a DTR signal on the 08 Series Computer.
2. A DTR signal that is not within specification on the 1480V computer can be interpreted as a DTR signal on the 08 Series Computer.
3. A DTR signal that is not within specification on the 1480V computer can be interpreted as a DTR signal on the 08 Series Computer.



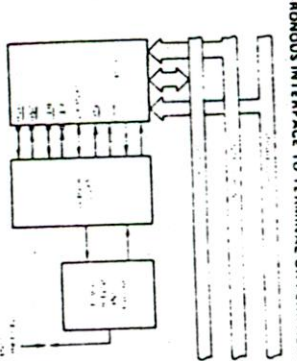
ASYNC SERIAL INTERFACE TO CRT TERMINAL.
DC to 9600 BAUD



ASYNC INTERFACE TO TELEPHONE LINES



SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE



SYNCHRONOUS INTERFACE TO TELEPHONE LINES



MC1489L
MC1489AL

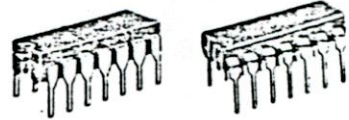
QUAD LINE RECEIVERS

The MC1489 monolithic quad line receivers are designed to interface data terminal equipment with data communications equipment in conformance with the specifications of EIA Standard No. RS-232C.

- Input Resistance — 3.0 k to 7.0 kilohms
- Input Signal Range — ± 30 Volts
- Input Threshold Hysteresis Built In
- Response Control
 - a) Logic Threshold Shifting
 - b) Input Noise Filtering

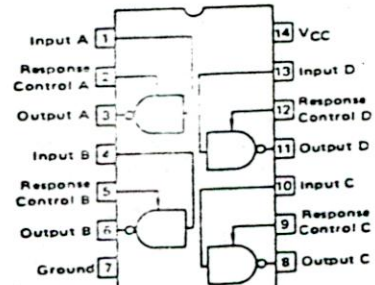
**QUAD MDTL
LINE RECEIVERS
RS-232C**

**SILICON MONOLITHIC
INTEGRATED CIRCUIT**

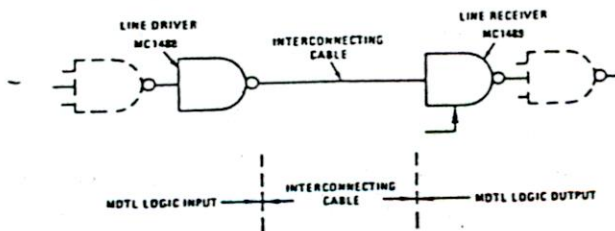


**L SUFFIX
CERAMIC PACKAGE
CASE 632
TO-116**

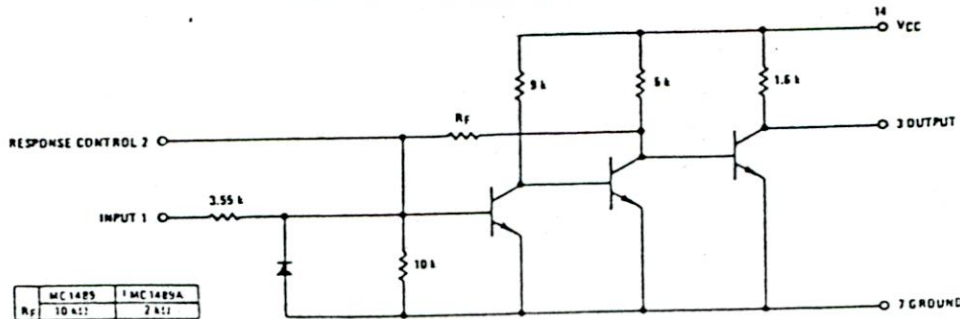
**P SUFFIX
PLASTIC PACKAGE
CASE 646**



TYPICAL APPLICATION



CIRCUIT SCHEMATIC (1/4 OF CIRCUIT SHOWN)





MC1488

QUAD LINE DRIVER

The MC1488 is a monolithic quad line driver designed to interface data terminal equipment with data communications equipment in conformance with the specifications of EIA Standard No. RS-232C.

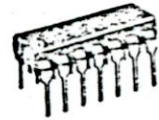
Features:

- Current Limited Output
±10 mA typ
- Power-Off Source Impedance
300 Ohms min
- Simple Slew Rate Control with External Capacitor
- Flexible Operating Supply Range
- Compatible with All Motorola MDTL and M TTL Logic Families

QUAD MDTL LINE DRIVER
RS-232C
SILICON MONOLITHIC
INTEGRATED CIRCUIT

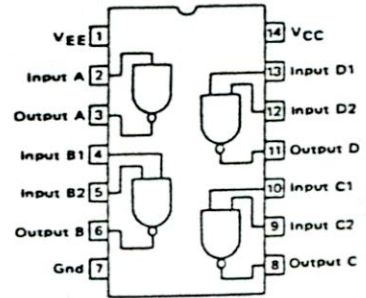


L SUFFIX
CERAMIC PACKAGE
CASE 832
TO-116

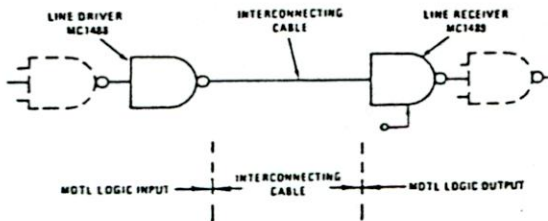


P SUFFIX
PLASTIC PACKAGE
CASE 646

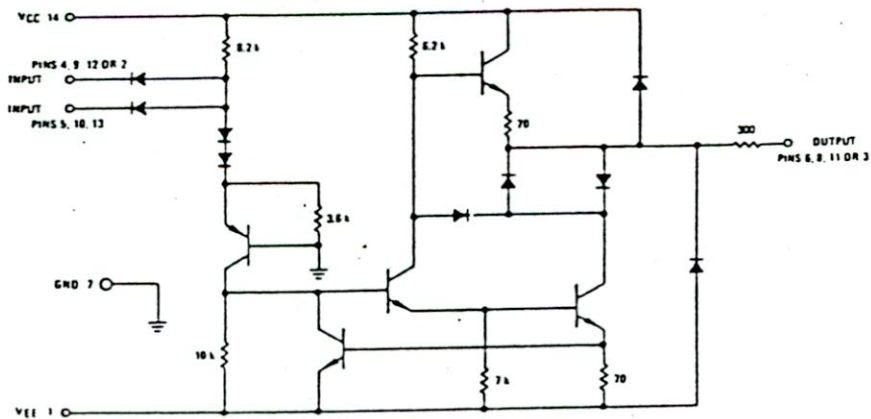
PIN CONNECTIONS



TYPICAL APPLICATION



CIRCUIT SCHEMATIC
(1/4 OF CIRCUIT SHOWN)



EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ± 8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (-40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

READ OPERATION

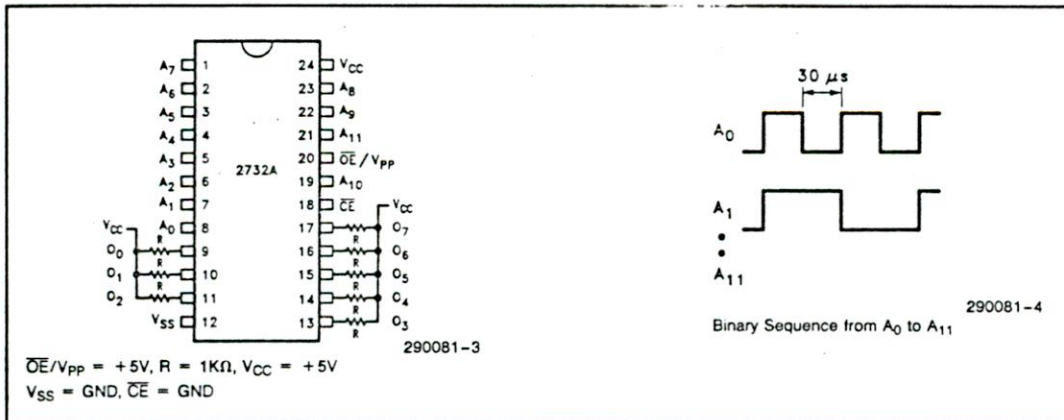
D.C. CHARACTERISTICS

Electrical Parameters of EXPRESS EPROM products are identical to standard EPROM parameters except for:

Sym- bol	Parameter	TD2732A LD2732A		Test Conditions
		Min	Max	
I _{SB}	V _{CC} Standby Current (mA)		45	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I _{CC1} (1)	V _{CC} Active Current (mA)		150	$\overline{OE} = \overline{CE} = V_{IL}$
	V _{CC} Active Current at High Temperature (mA)		125	$\overline{OE} = \overline{CE} = V_{IL}$, V _{PP} = V _{CC} , T _{Ambient} = 85°C

NOTE:

1. Maximum current value is with outputs O₀ to O₇ unloaded.



Burn-In Bias and Timing Diagrams

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITIONS

Type	Operating Temperature	Burn-In 125°C (hr)
Q	0°C to +70°C	168 ± 8
T	-40°C to +85°C	None
L	-40°C to +85°C	168 ± 8

EXPRESS OPTIONS

2732A Versions

Packaging Options	
Speed Versions	Cerdip
-2	Q
STD	Q, T, L
-3	Q
-4	Q, T, L
-20	Q
-25	Q, T, L
-30	Q
-45	Q, T, L

ABSOLUTE MAXIMUM RATINGS*

Operating Temp. During Read 0°C to +70°C
Temperature Under Bias -10°C to +80°C
Storage Temperature -65°C to +125°C
All Input or Output Voltages with Respect to Ground -0.3V to +6V
Voltage on A9 with Respect to Ground -0.3V to +13.5V
V _{PP} Supply Voltage with Respect to Ground During Programming -0.3V to +22V
V _{CC} Supply Voltage with Respect to Ground -0.3V to +7.0V

*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

READ OPERATION
D.C. CHARACTERISTICS 0°C ≤ T_A ≤ +70°C

Symbol	Parameter	Limits			Units	Conditions
		Min	Typ ⁽³⁾	Max		
I _{LI}	Input Load Current			10	μA	V _{IN} = 5.5V
I _{LO}	Output Leakage Current			10	μA	V _{OUT} = 5.5V
I _{SB} ⁽²⁾	V _{CC} Current (Standby)			35	mA	$\overline{CE} = V_{IH}, \overline{OE} = V_{IL}$
I _{CC1} ⁽²⁾	V _{CC} Current (Active)			100	mA	$\overline{OE} = \overline{CE} = V_{IL}$
V _{IL}	Input Low Voltage	-0.1		0.8	V	
V _{IH}	Input High Voltage	2.0		V _{CC} + 1	V	
V _{OL}	Output Low Voltage			0.45	V	I _{OL} = 2.1 mA
V _{OH}	Output High Voltage	2.4			V	I _{OH} = -400 μA

A.C. CHARACTERISTICS 0°C ≤ T_A ≤ 70°C

Versions	V _{CC} ± 5%	2732A-2		2732A		2732A-3		2732A-4		Units	Test Conditions
		V _{CC} ± 10%		2732A-20		2732A-25		2732A-30			
Symbol	Parameter	Min	Max	Min	Max	Min	Max	Min	Max		
t _{ACC}	Address to Output Delay		200		250		300		450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
t _{CE}	\overline{CE} to Output Delay		200		250		300		450	ns	$\overline{OE} = V_{IL}$
t _{OE}	\overline{OE}/V_{PP} to Output Delay		70		100		150		150	ns	$\overline{CE} = V_{IL}$
t _{DF} ⁽⁴⁾	\overline{OE}/V_{PP} High to Output Float	0	60	0	60	0	130	0	130	ns	$\overline{CE} = V_{IL}$
t _{OH} ⁽⁴⁾	Output Hold from Addresses, \overline{CE} or \overline{OE}/V_{PP} , Whichever Occurred First	0		0		0		0		ns	$\overline{CE} = \overline{OE} = V_{IL}$

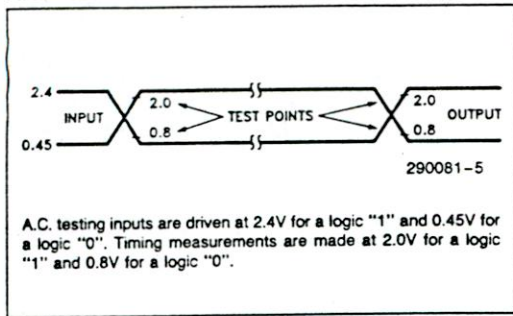
NOTES:

- V_{CC} must be applied simultaneously or before \overline{OE}/V_{PP} and removed simultaneously or after \overline{OE}/V_{PP} .
- The maximum current value is with outputs O₀ to O₇ unloaded.
- Typical values are for T_A = 25°C and nominal supply voltages.
- This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram.

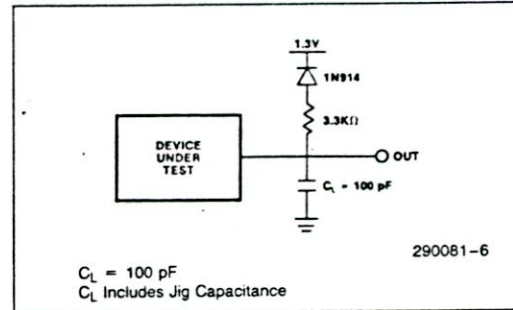
CAPACITANCE (2) $T_A = 25^\circ\text{C}$, $f = 1\text{ MHz}$

Symbol	Parameter	Typ	Max	Unit	Conditions
C_{IN1}	Input Capacitance Except \overline{OE}/V_{PP}	4	6	pF	$V_{IN} = 0V$
C_{IN2}	\overline{OE}/V_{PP} Input Capacitance		20	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT} = 0V$

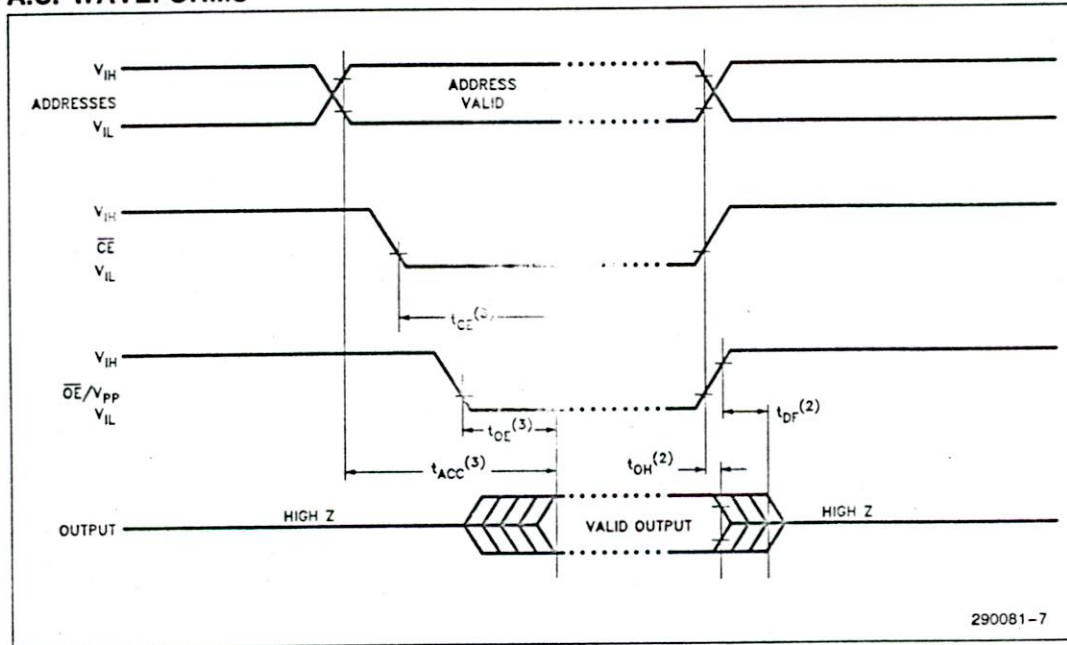
A.C. TESTING INPUT/OUTPUT WAVEFORM



A.C. TESTING LOAD CIRCUIT



A.C. WAVEFORMS



NOTES:

1. Typical values are for $T_A = 25^\circ\text{C}$ and nominal supply voltages.
2. This parameter is only sampled and is not 100% tested. Output float is defined as the point where data is no longer driven—see timing diagram.
3. \overline{OE}/V_{PP} may be delayed up to $t_{ACC} - t_{OE}$ after the falling edge of \overline{CE} without impacting t_{CE} .

DEVICE OPERATION

The modes of operation of the 2732A are listed in Table 1. A single 5V power supply is required in the read mode. All inputs are TTL levels except for \overline{OE}/V_{PP} during programming and 12V on A_9 for the Intelligent Identifier™ mode. In the program mode the \overline{OE}/V_{PP} input is pulsed from a TTL level to 21V.

Table 1. Mode Selection

Mode	Pins					
	\overline{CE}	\overline{OE}/V_{PP}	A_9	A_0	V_{CC}	Outputs
Read/Program Verify	V_{IL}	V_{IL}	X	X	V_{CC}	D_{OUT}
Output Disable	V_{IL}	V_{IH}	X	X	V_{CC}	High Z
Standby	V_{IH}	X	X	X	V_{CC}	High Z
Program	V_{IL}	V_{PP}	X	X	V_{CC}	D_{IN}
Program Inhibit	V_{IH}	V_{PP}	X	X	V_{CC}	High Z
Intelligent Identifier ⁽³⁾						
	—Manufacturer	V_{IL}	V_{IL}	V_H	V_{IL}	V_{CC}
—Device	V_{IL}	V_{IL}	V_H	V_{IH}	V_{CC}	01H

NOTES:

1. X can be V_{IH} or V_{IL} .
2. $V_H = 12V \pm 0.5V$.
3. $A_1-A_8, A_{10}, A_{11} = V_{IL}$.

Read Mode

The 2732A has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable (\overline{CE}) is the power control and should be used for device selection. Output Enable (\overline{OE}/V_{PP}) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from \overline{CE} to output (t_{CE}). Data is available at the outputs after the falling edge of \overline{OE}/V_{PP} , assuming that \overline{CE} has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

EPPOMs can be placed in a standby mode which reduces the maximum active current of the device

by applying a TTL-high signal to the \overline{CE} input. When in standby mode, the outputs are in a high impedance state, independent of the \overline{OE}/V_{PP} input.

Two Line Output Control

Because EPROMs are usually used in larger memory arrays, Intel has provided two control lines which accommodate this multiple memory connection. The two control lines allow for:

- a) The lowest possible memory power dissipation, and
- b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, \overline{CE} should be decoded and used as the primary device selecting function, while \overline{OE}/V_{PP} should be made a common connection to all devices in the array and connected to the \overline{READ} line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

SYSTEM CONSIDERATION

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC} , has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient current peaks is dependent on the output capacitive and inductive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's two-line control and by use of properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor of low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for

every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

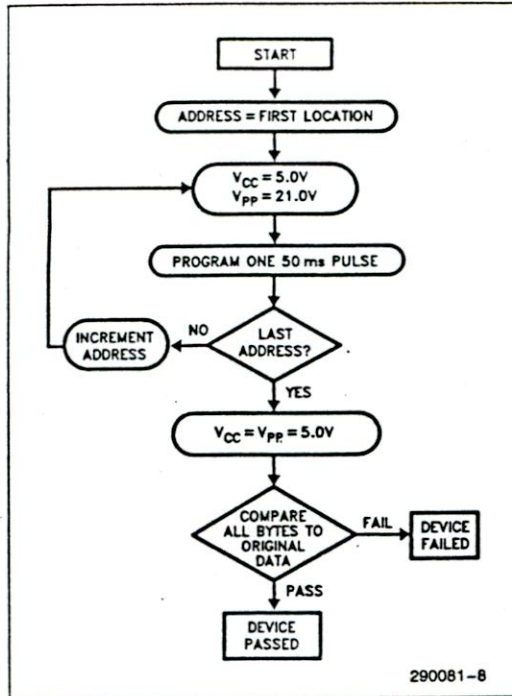


Figure 3. Standard Programming. Flowchart

PROGRAMMING MODE

CAUTION: Exceeding 21V on \overline{OE}/V_{PP} will permanently damage the device.

Initially, and after each erasure (cerdip EPROMs), all bits of the EPROM are in the "1" state. Data is introduced by selectively programming "0s" into the bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" in cerdip EPROMs is by ultraviolet light erasure.

The device is in the programming mode when the \overline{OE}/V_{PP} input is at 21V. It is required that a 0.1 μ F capacitor be placed across \overline{OE}/V_{PP} and ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 20 ms (50 ms typical) active low, TTL program pulse is ap-

plied to the \overline{CE} input. A program pulse must be applied at each address location to be programmed (see Figure 3). Any location can be programmed at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 ms. The EPROM must not be programmed with a DC signal applied to the \overline{CE} input.

Programming of multiple 2732As in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2732As may be connected together when they are programmed with the same data. A low level TTL pulse applied to the \overline{CE} input programs the paralleled 2732As.

Program Inhibit

Programming of multiple EPROMs in parallel with different data is easily accomplished by using the Program Inhibit mode. A high level \overline{CE} input inhibits the other EPROMs from being programmed. Except for \overline{CE} , all like inputs (including \overline{OE}/V_{PP}) of the parallel EPROMs may be common. A TTL low level pulse applied to the \overline{CE} input with \overline{OE}/V_{PP} at 21V will program that selected device.

Program Verify

A verify (Read) should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with \overline{OE}/V_{PP} and \overline{CE} at V_{IL} . Data should be verified t_{DV} after the falling edge of \overline{CE} .

intelligent Identifier™ Mode

The intelligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C \pm 5°C ambient temperature range that is required when programming the device.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 of the EPROM. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 from V_{IL} to V_{IH} . All other address lines must be held at V_{IL} during the intelligent Identifier Mode.

Byte 0 ($A0 = V_{IL}$) represents the manufacturer code and byte 1 ($A0 = V_{IH}$) the device identifier code. These two identifier bytes are given in Table 1.

ERASURE CHARACTERISTICS (FOR Cerdip EPROMS)

The erasure characteristics are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the EPROM in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the device is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Wsec/cm². The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm² power rating. The EPROM should be placed within 1 inch of the lamp tubes during erasure. The maximum integrated dose an EPROM can be exposed to without damage is 7258 Wsec/cm² (1 week @ 12000 μW/cm²). Exposure of the device to high intensity UV light for longer periods may cause permanent damage.

PROGRAMMING

D.C. PROGRAMMING CHARACTERISTICS

T_A = 25°C ± 5°C, V_{CC} = 5V ± 5%, V_{PP} = 21V ± 0.5V

Symbol	Parameter	Limits			Units	Test Conditions (Note 1)
		Min	Typ ⁽³⁾	Max		
I _{LI}	Input Current (All Inputs)			10	μA	V _{IN} = V _{IL} or V _{IH}
V _{IL}	Input Low Level (All Inputs)	-0.1		0.8	V	
V _{IH}	Input High Level (All Inputs Except \overline{OE}/V_{PP})	2.0		V _{CC} + 1	V	
V _{OL}	Output Low Voltage During Verify			0.45	V	I _{OL} = 2.1 mA
V _{OH}	Output High Voltage During Verify	2.4			V	I _{OH} = -400 μA
I _{CC2} ⁽⁴⁾	V _{CC} Supply Current (Program and Verify)		85	100	mA	
I _{PP2} ⁽⁴⁾	V _{PP} Supply Current (Program)			30	mA	$\overline{OE} = \overline{OE}/V_{PP} = V_{PP}$
V _{ID}	A ₉ intelligent Identifier Voltage	11.5		12.5	V	

A.C. PROGRAMMING CHARACTERISTICS

$T_A = 25^\circ\text{C} \pm 5^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 5\%$, $V_{PP} = 21\text{V} \pm 0.5\text{V}$

Symbol	Parameter	Limits			Units	Test Conditions* (Note 1)
		Min	Typ ⁽³⁾	Max		
t _{AS}	Address Setup Time	2			μs	
t _{OES}	$\overline{\text{OE}}/V_{PP}$ Setup Time	2			μs	
t _{DS}	Data Setup Time	2			μs	
t _{AH}	Address Hold Time	0			μs	
t _{DH}	Data Hold Time	2			μs	
t _{DFP}	$\overline{\text{OE}}/V_{PP}$ High to Output Not Driven	0		130	ns	(Note 2)
t _{PW}	$\overline{\text{CE}}$ Pulse Width During Programming	20	50	55	ms	
t _{OEH}	$\overline{\text{OE}}/V_{PP}$ Hold Time	2			μs	
t _{DV}	Data Valid from $\overline{\text{CE}}$			1	μs	$\overline{\text{CE}} = V_{IL}$, $\overline{\text{OE}}/V_{PP} = V_{IL}$
t _{VR}	V _{PP} Recovery Time	2			μs	
t _{PRT}	$\overline{\text{OE}}/V_{PP}$ Pulse Rise Time During Programming	50			ns	

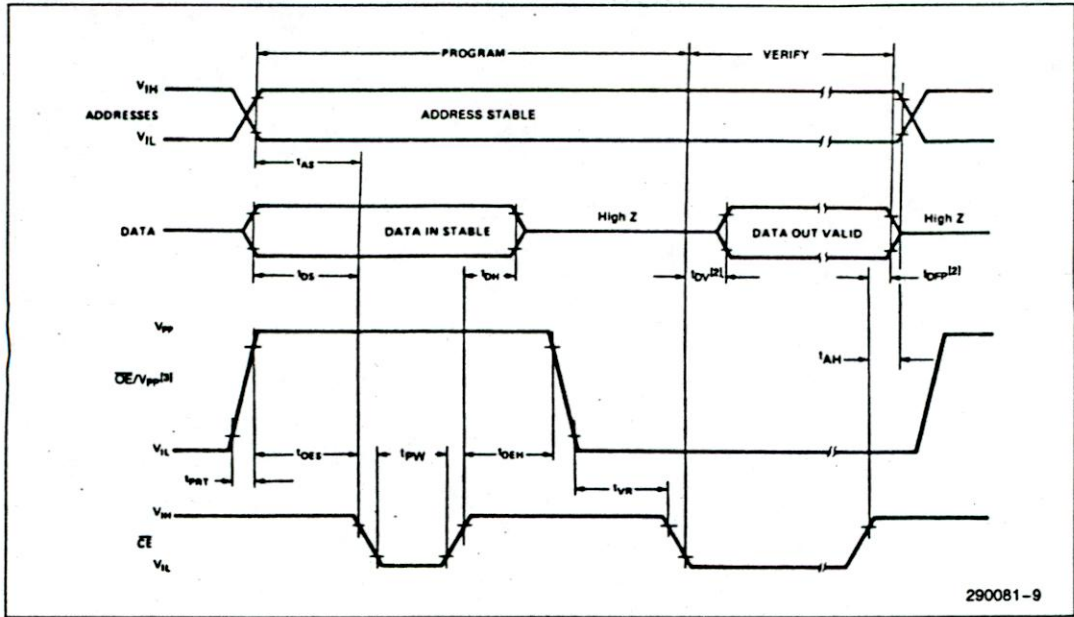
NOTES:

1. V_{CC} must be applied simultaneously or before $\overline{\text{OE}}/V_{PP}$ and removed simultaneously or after $\overline{\text{OE}}/V_{PP}$.
2. This parameter is only sampled and is not 100% tested. Output Float is defined as the point where data is no longer driven—see timing diagram.
3. Typical values are for T_A = 25°C and nominal supply voltages.
4. The maximum current value is with outputs 0₀ to 0₇ unloaded.

***A.C. TEST CONDITIONS**

Input Rise and Fall Time (10% to 90%) ≤ 20 ns
 Input Pulse Levels 0.45V to 2.4V
 Input Timing Reference Level 0.8V and 2.0V
 Output Timing Reference Level 0.8V and 2.0V

PROGRAMMING WAVEFORMS



NOTES:

1. The input timing reference level is 0.8V for a V_{IL} and 2V for a V_{IH} .
2. t_{DV} and t_{DVP} are characteristics of the device but must be accommodated by the programmer.
3. When programming the 2732A, a $0.1\mu F$ capacitor is required across \overline{OE}/V_{PP} and ground to suppress spurious voltage transients which can damage the device.