

T  
001.0404  
D542



**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

Facultad de Ingeniería en Electricidad

**“Diseño y Construcción de un Equipo Emulador  
de Memorias Eprom - 2716 - 2732 - 2764 Apoyado  
por un Computador”**

**Tesis de Grado**

**Previa a la obtención del Título de  
INGENIERO EN ELECTRICIDAD**

Especialización Electrónica

Presentado por:  
**BOLIVAR DIAZ CHANG**

Guayaquil - Ecuador  
1991



## AGRADECIMIENTO

Al Inq. PEDRO VARGAS G. Director de Tesis, por su ayuda y colaboración para la realización de este trabajo.

Y a todos aquellos que hicieron posible mi formación:

a mis profesores,

a mis amigos,

a mi familia.

DEDICATORIA

A MIS PADRES

A MIS HERMANOS

A MIS TIOS



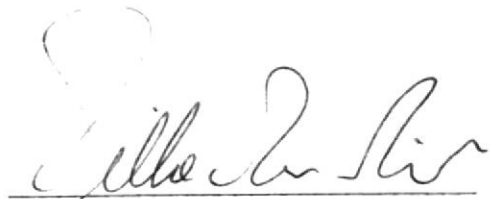
ING. JORGE FLORES  
SUB-DECANO FACULTAD DE  
INGENIERIA ELECTRICA



ING. PEDRO VARGAS G.  
DIRECTOR DE TESIS



ING. EDGAR IZQUIERDO  
MIEMBRO DEL TRIBUNAL



ING. HUGO VILLAVICENCIO  
MIEMBRO DEL TRIBUNAL

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, me corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL".

(Reglamento de Exámenes y Títulos profesionales de la ESPOL).



.....  
BOLIVAR AGUSTIN DIAZ CHANG

## R E S U M E N

El trabajo a desarrollarse tiene como objetivo principal el estudio y construcción de un Circuito que sea capaz de aliviar en algo el trabajo tedioso que tienen que realizar aquellas personas que desarrollan programas con aplicación en microprocesadores.

El interés del desarrollo de este trabajo es eliminar la acción repetida de grabación y borrado de dicho programa; lo que podría ser realizado tantas veces como sea necesario. Este objetivo fijado puede ser logrado por medio de un circuito como el mencionado anteriormente y al cual se lo ha denominado EMULADOR que es capaz de permitir la escritura consecutiva de información, razón por la cual operará con el soporte de un computador que le permitirá la escritura, corrección y transmisión de los programas en mención.

El aspecto más sobresaliente del trabajo a realizarse es el de poder establecer la comunicación Computador-Emulador, por medio de un UART, el cual recibirá los bytes transmitidos en formato serie, de acuerdo al protocolo RS-232C, para convertirlos a formato paralelo y almacenarlos en una memoria tipo ram, de donde serán leídos por el circuito de aplicación como si estos estuvieran grabados en una EPROM.

# INDICE GENERAL

	PAGINA
RESUMEN	VI
INDICE GENERAL	VII
INDICE DE FIGURAS	X
INDICE DE TABLAS	XII
INDICE DE ABREVIATURAS	XIII
INTRODUCCION	15
I. CIRCUITO EMULADOR: DISEÑO Y CONSTRUCCION	17
1.1 MEMORIA EPROM: DEFINICION, CARACTERISTICAS Y APLICACION	17
1.2 DISEÑO DEL CIRCUITO EMULADOR	26
1.2.1 OBJETIVO Y APLICACION	26
1.2.2 ESTUDIO Y DESARROLLO DEL EMULADOR	28
1.2.3 PROTOCOLO DE TRANSMISION ASINCRONO SERIE	30
1.2.4 DESCRIPCION FUNCIONAL DETALLADA DEL CIRCUITO EMULADOR	35
1.2.4.1 CIRCUITO DE INTERFACE SERIE	35
1.2.4.2 CIRCUITO DE MEMORIA	42
1.2.4.3 CIRCUITO CONTADOR DE DIRECCION	47
1.2.4.4 CIRCUITO PULSADOR DE UN PASO	51
1.2.4.5 CIRCUITO DE DISPLAY	54
1.2.4.6 CIRCUITO DE LA FUENTE DE PODER	56
1.2.4.6.1 REQUERIMIENTOS PARA EL DISEÑO	56
1.2.4.6.2 RECTIFICACION Y FILTRADO	56

1.2.4.6.3	REGULACION DE VOLTAGE	58
II.	PROGRAMAS DE CONTROL	59
2.1	ALGORITMO DEL PROGRAMA PRINCIPAL	59
2.1.1	ALGORITMO DEL PROGRAMA MOSTRAR INDICACIONES DE OPERACION DEL EMULADOR	63
2.1.2	ALGORITMO DEL PROGRAMA CARGAR INFORMACION DESDE EL TECLADO	66
2.1.3	ALGORITMO DEL PROGRAMA TRANSMITIR INFORMA- CION DESDE EL COMPUTADOR AL EMULADOR	66
2.1.4	ALGORITMO DEL PROGRAMA REALIZAR CORRECCIO- NES EN LA INFORMACION TRANSMITIDA AL EMULA DOR	69
2.1.4.1	ALGORITMO DEL SUB-PROGRAMA INTRO- DUCIR BYTES	71
2.1.4.2	ALGORITMO DEL SUB-PROGRAMA BORRAR BYTES	80
2.1.4.3	ALGORITMO DEL SUB-PROGRAMA CAM- BIAR BYTES	85
2.2	PUERTO DE SALIDA SERIE RS-232C	88
2.2.1	CONFIGURACIONES COMUNMENTE UTILIZADAS	99
III.	PROCEDIMIENTO TECNICO EN LA CONSTRUCCION DEL CIRCUITO Y RESULTADOS OBTENIDOS	103
3.1	LISTA DE LOS ELEMENTOS UTILIZADOS	103
3.2	LISTA DE DESIGNACION DE LOS ELEMENTOS	105
3.3	TECNICA EMPLEADA EN EL MONTAJE E INTERCONEXION	107



DE LOS ELEMENTOS	
3.4 DIAGRAMAS	108
3.4.1 DIAGRAMA DE UBICACION DE LOS ELEMENTOS	108
3.4.2 DIAGRAMAS ESQUEMATICOS	108
3.5 CONFIGURACION DEL PUERTO SERIE	111
CONCLUSIONES Y RECOMENDACIONES	123
APENDICES	
A. DATOS TECNICOS DEL CIRCUITO INTEGRADO CMOS/LSI UNIVER <u>SAL</u> ASYNCHRONOUS RECEIVER TRANSMITTER IM6402IPL	127
B. DATOS TECNICOS DEL CIRCUITO INTEGRADO RANDOM ACCESS MEMORY HM6264LP-12	131
C. DATOS TECNICOS DEL CIRCUITO INTEGRADO BIT RATE GENERA <u>TOR</u> MC14411P	136
D. PROGRAMAS DE CONTROL	139
E. MANUAL PARA EL USUARIO	147
F. INSTRUCCIONES PARA EL MANEJO DEL COMPUTADOR ATARI 1040ST	150
BIBLIOGRAFIA	153

## I N D I C E   D E   F I G U R A S

FIGURA	TITULO	PAGINA
1.1	Modelo funcional de una ROM de $2^n \times m$	18
1.2	Modelo de una ROM a diodos con fusibles de enlace	19
1.3	Modelo básico de una ROM a transistores	20
1.4	Modelo básico de una MOS-ROM de puerta controlada	21
1.5	Celda MOS-UV borrable	22
1.6	MOS-ROM Estática, 2048 palabras de 8 bits	24
1.7	Esquema básico de un sistema digital con micropro cesadores	27
1.8	Diagrama de bloques general del EMULADOR	29
1.9	Protocolo de transmisión serie asíncrono	31
1.10	Diagrama del Circuito de Interface Serie	37
1.11	Diagrama de tiempo de señales en el UART	38
1.12	Diagrama del Circuito de Memoria	43
1.13	Diagrama del Circuito Contador de Dirección	48
1.14	Diagrama de tiempo de generación del bit más significativo de la dirección	50
1.15	Diagrama del Circuito Pulsador de un Paso	53
1.16	Diagrama del Circuito de Display	55
1.17	Circuito de la Fuente de Poder	
2.1	Diagrama de flujo del Programa Principal	60
2.2	Diagrama de flujo del Programa MIOE	64

2.3	Diagrama de flujo del Programa CIT	67
2.4	Diagrama de flujo del Programa TICE	70
2.5	Diagrama de flujo del Programa RCITE	72
2.6	Diagrama de flujo del Sub-Programa IB	75
2.7	Diagrama de flujo del Sub-Programa BB	82
2.8	Diagrama de flujo del Sub-Programa CB	86
2.9	Handshake entre el DTE y el DCE	94
2.10	Tiempo cuando RTS puede ser verdadero	95
3.1	Diagrama de ubicación de elementos	109
3.2	Circuito de Interface Serie	113
3.3	Circuito de Memoria	115
3.4	Circuito Contador de Dirección	117
3.5	Circuito Pulsador de Un Paso	119
3.6	Circuito de Display	120

## I N D I C E   D E   T A B L A S

TABLA	REFERENCIA	PAGINA
I	Tabla para selección del protocolo de transmisión	40
II	Tabla de selección del factor de velocidad	41
III	Tabla de frecuencias generadas en el circuito integrado U12	42
IV	Circuitos RS-232C clasificados por su categoría	89
V	Configuraciones RS-232C Standard	101

## INDICE DE ABREVIATURAS

BB : Programa Borrar Bytes

BYTEN: Byte Nuevo

CB : Programa Corregir Bytes

CIT : Programa Cargar Información desde teclado

CNTA : Registro Cuenta Anterior

CNTN : Registro Cuenta Nueva

CPU : Unidad de Procesamiento Central

CRS : Matris Caracteres

DMA : Acceso Directo de Memoria

EEPRON: Electrically EPROM

EPRON: Erasable PROM

HBLT : Señal de Habilitación

IB : Programa Introducir Bytes

M,M1,M2: Direcciones Hexadecimales

MDCRG: Modo de Carga

MDLCT: Modo de Lectura

MDPRST: Modo de Presentación

MIOE : Programa Mostrar Indicaciones de Operación del Emulador

NUDBY: Archivo Número de Bytes

NUM : Número de Bytes a Ingresarse

PC : Programa Correcciones

PP : Programa Principal

PROM : Programmable ROM

PS : Señal de Paso Simple

RAM : Random Access Memory

RCITE : Programa Realizar Correcciones en Información Transmitida  
al Emulador

RL : Impedancia de Carga

ROM : Read Only Memory

SYSCLK : Señal de Reloj del Emulador

TICE : Programa Transmitir Información desde Computador al Emulador

TTL : Lógica Transistor a Transistor

TW : Ancho de Pulso generado por el circuito integrado U18

UART : Universal Asynchronous Receiver Transmitter

UV : Radiación ULTRA Violeta

VM : Valor Máximo del Voltage de Alimentación.



## I N T R O D U C C I O N

El circuito a construirse está destinado a facilitar el trabajo de aquellas personas que están desarrollando programas para ser grabados en memorias del tipo EPROM; porque elimina el empleo de equipos especiales tales como el programador y el borrador de rayos ultra-violeta para EPROMs. Además, el usuario al realizar correcciones sobre el programa en desarrollo las realizará en un computador y, no en el programador mismo, para luego ser transferidas a la memoria del Emulador evitándose así el inconveniente de iniciar la tarea de reprogramación de dicha memoria otra vez, es decir, se evita el escribir el programa en mención totalmente en el programador cada vez que éste ha sufrido una modificación porque el programa se mantiene almacenado en un archivo, y con la ayuda de programas auxiliares se podrán realizar operaciones tales como: la introducción, el cambio y el borrado de bytes.

Como su nombre lo indica él tiene la característica de emular una EPROM, es decir, que el circuito de aplicación no notará la diferencia de que el programa en proyecto está almacenado en una RAM.

A continuación se tratará el procedimiento seguido en la elaboración del presente trabajo.

En el capítulo uno se desarrolla y se explica detalladamente los fundamentos teóricos bajo los cuales se ha diseñado y construido la circuitería.

ría del Emulador, haciéndose destacar en lo que consiste la transmisión asíncrona serie de información y la forma cómo se conseguirá esta con el Emulador.

En el capítulo dos son desarrollados los algoritmos y programas para el uso de un computador que facilite el manejo del Emulador. También, se estudia la función que desempeña cada pin del conector RS-232C, incluyendo un análisis de las configuraciones más populares.

En el capítulo tres se presentan los listados y designaciones de los elementos empleados, así como también los diagramas eléctricos completos de los diferentes bloques que conforman el Emulador. Además, en base a lo expuesto en el capítulo anterior se configura el puerto serie que tendrá aplicación con el presente trabajo.



## CAPITULO I

### CIRCUITO EMULADOR: DISEÑO Y CONSTRUCCION

#### 1.1 MEMORIA EPROM: DEFINICION, CARACTERISTICAS Y APLICACION

Una ROM (Read Only Memory) es básicamente un "diccionario digital", un dispositivo de memoria cuyo código de salida depende del código de entrada. Es un arreglo o circuito combinatorial de salida múltiple, teniendo como modelo el mostrado en la figura 1.1.

Las memorias ROMs son usadas en sistemas digitales que trabajan con microprocesadores, en las cuales se puede leer códigos almacenados o dato que no está sujeto a cambio. La información es almacenada en la ROM durante su fabricación antes que llegue al usuario. Por lo tanto, ella puede ser sólo leída, y además, la circuitería tiende a ser más sencilla que en las RAMs.

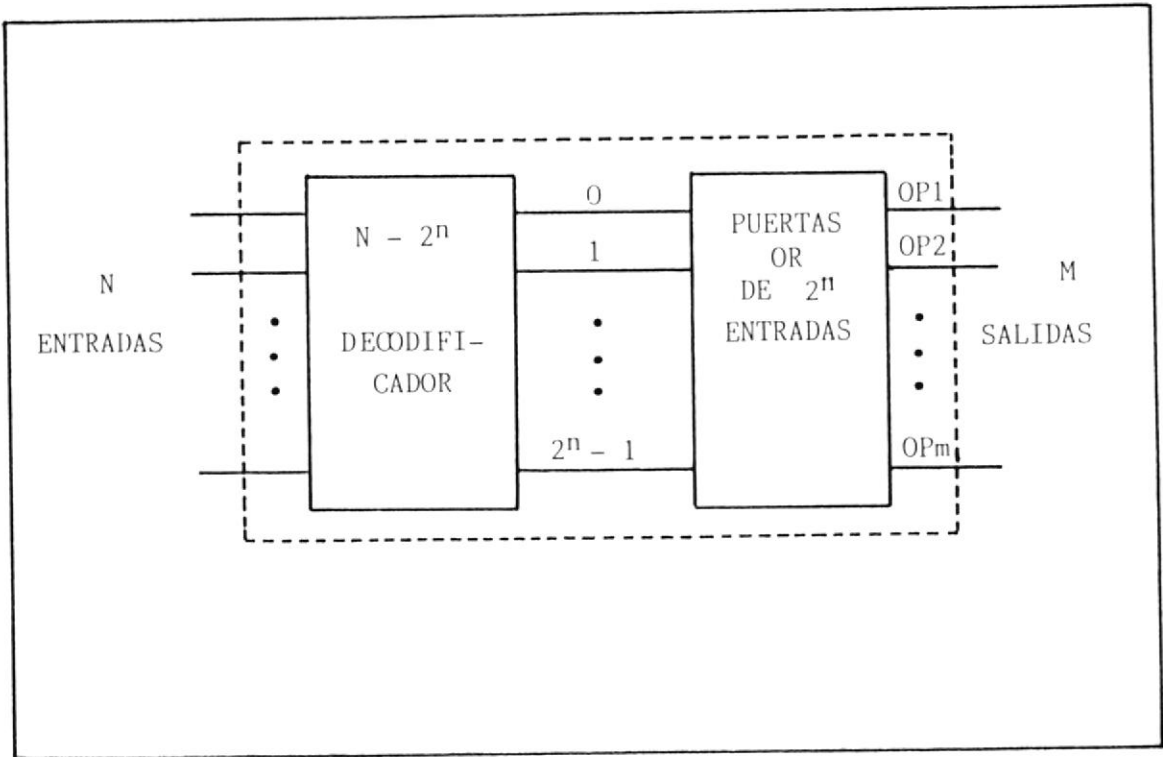


Figura 1.1 Modelo funcional de una ROM de  $2^n \times m$ .

La función de una ROM es la de un codificador cuyo arreglo usualmente es una matriz de líneas conectadas por elementos en determinadas intersecciones para formar el patrón del codificador. Estos elementos conductores pueden ser dejados abiertos o conectados en una determinada intersección para producir un cero o un uno. La figura 1.2 muestra una ROM a diodos con fusibles de enlace. Una línea direccionada transmite una corriente de polarización a través de cada diodo conectado a la misma; las líneas de salida con diodos en las intersecciones recibirán esta corriente, y aquellas para las cuales los diodos fueron eliminados o quemados no la recibirán. ROMs con fusibles de enlace pueden ser progra-

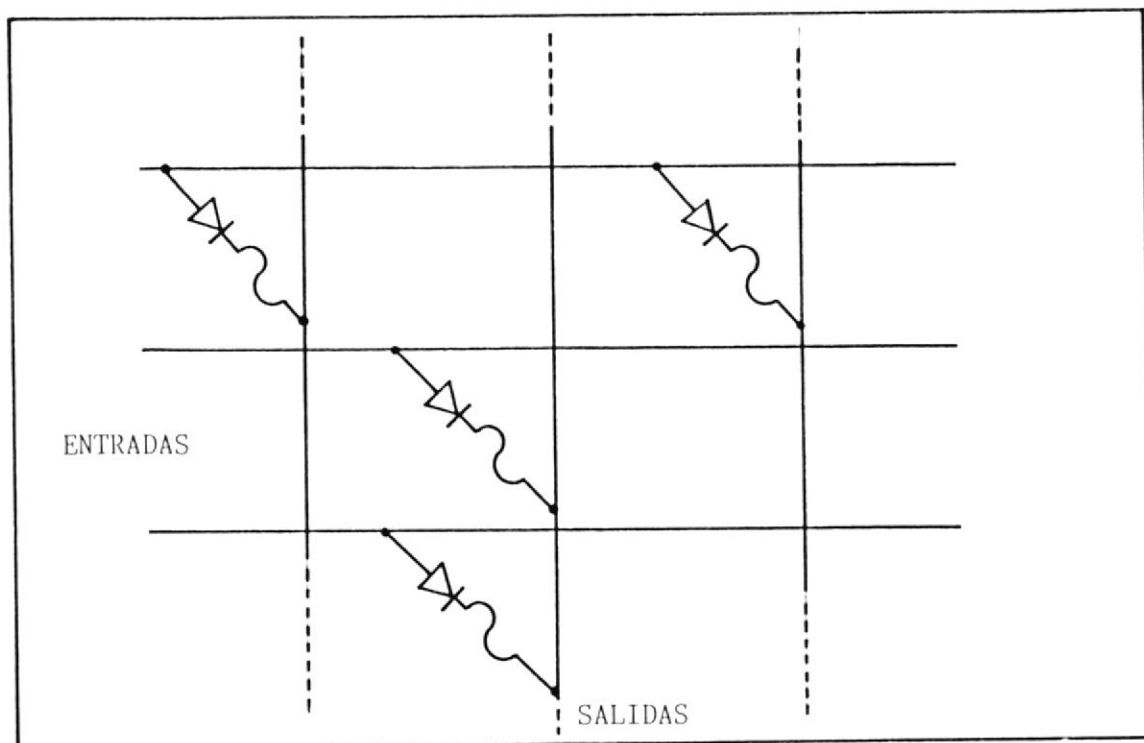


Figura 1.2. Modelo de una ROM a diodos con fusibles de enlace.

mas utilizando un equipo especialmente diseñado para este propósito, donde el programador determina cual localización de memoria tendrá un bit, entonces la rom es colocada en la unidad programadora la cual envía un pulso de corriente a través del fusible de enlace para abrir dicha conexión. El fusible de enlace consiste de una lámina de polycrystalline silicon, silicato de platino, nichrome o titanium tungsten depositado entre los elementos de conexión y las líneas de salida de la memoria. La figura 1.3 muestra una rom que emplea en cambio transistores bipolares como elementos de conexión.

En este tipo de rom los fusibles son quemados igualmente durante la pro

gramación por saturación del transistor (seguidor de emisor).

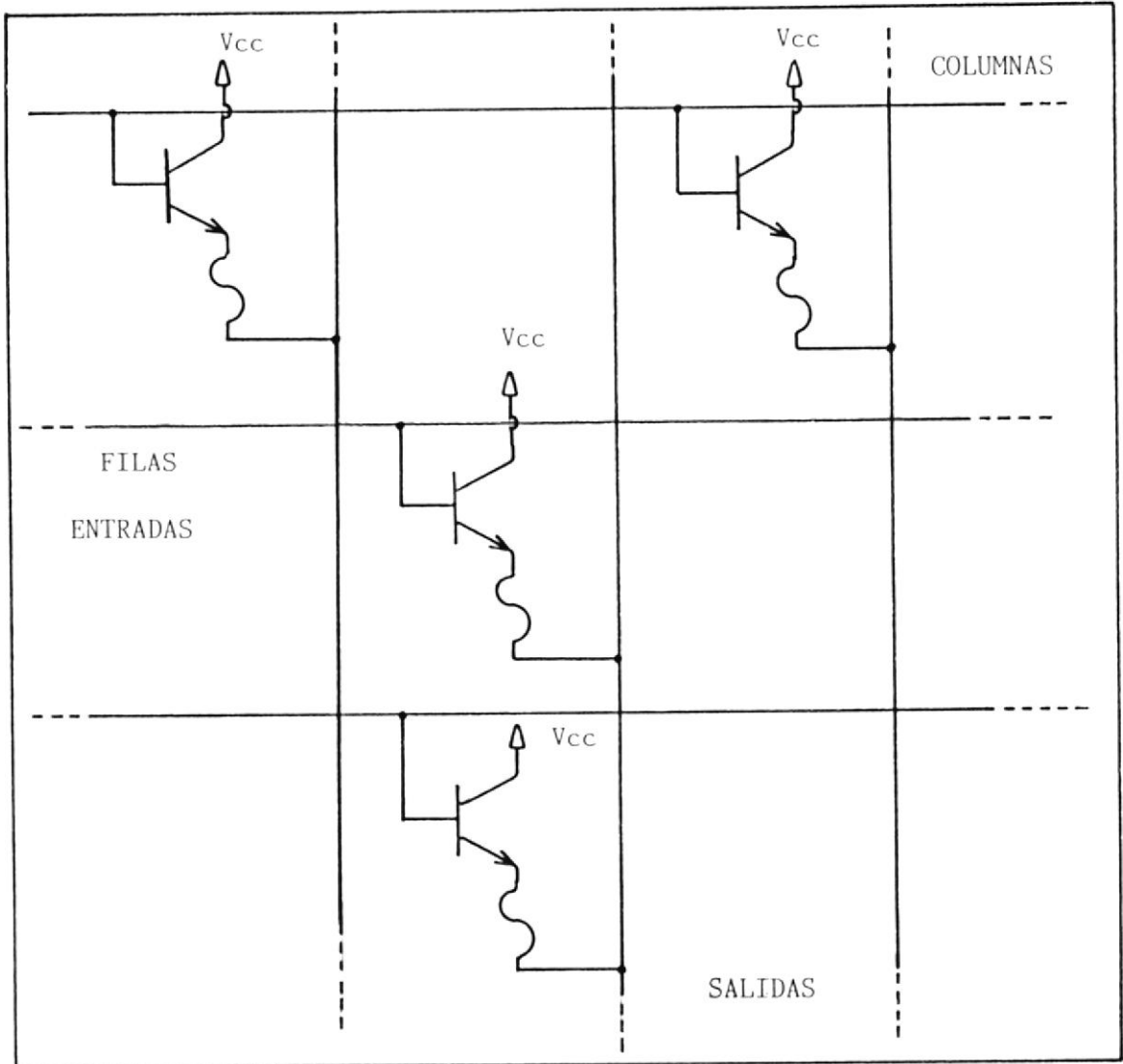


Figura 1.3. Modelo básico de una ROM a transistores.

Con un nivel de tensión alto en la base del transistor y un nivel cercano al potencial de tierra en la línea de salida, una gran corriente circulará a través del transistor y el fusible; la cual abrirá el contacto

del emisor resultando de esta forma en la programación de esa localidad.

Una MOS-ROM de puerta controlada es mostrada en la figura 1.4 donde la puerta cortocircuita la salida a tierra para producir un cero.

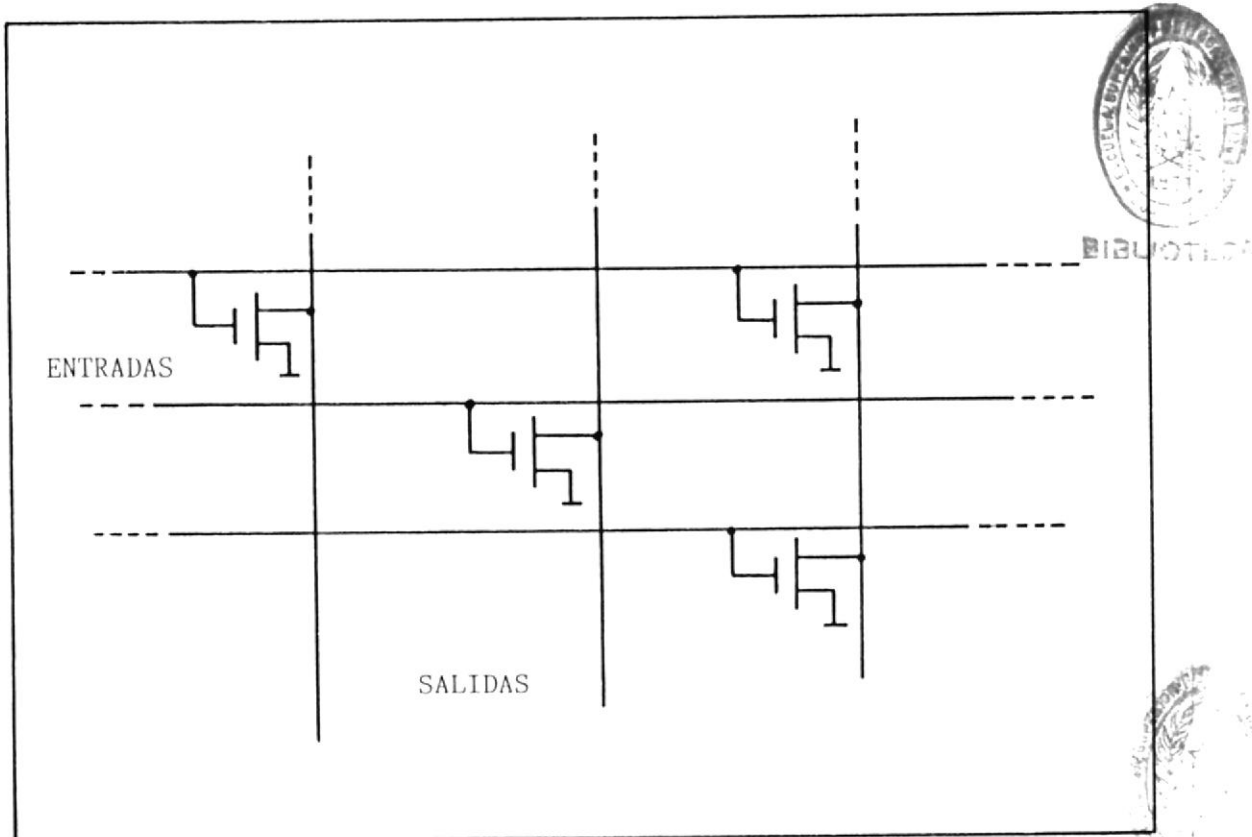


Figura 1.4. Modelo básico de una MOS ROM de puerta controlada.

La permanencia del fusible de enlace no es práctico si el programa de la rom debe ser cambiado un determinado número de veces antes que él esté totalmente depurado.

ROMs borrables o PROMs (programmable read only memory), son útiles enton-

ces para este propósito, y un tipo de estas memorias son las denominadas EPROM, las cuales contienen en cada celda un transistor MOS y dos compuertas o electrodos, como se ilustra en la figura 1.5.

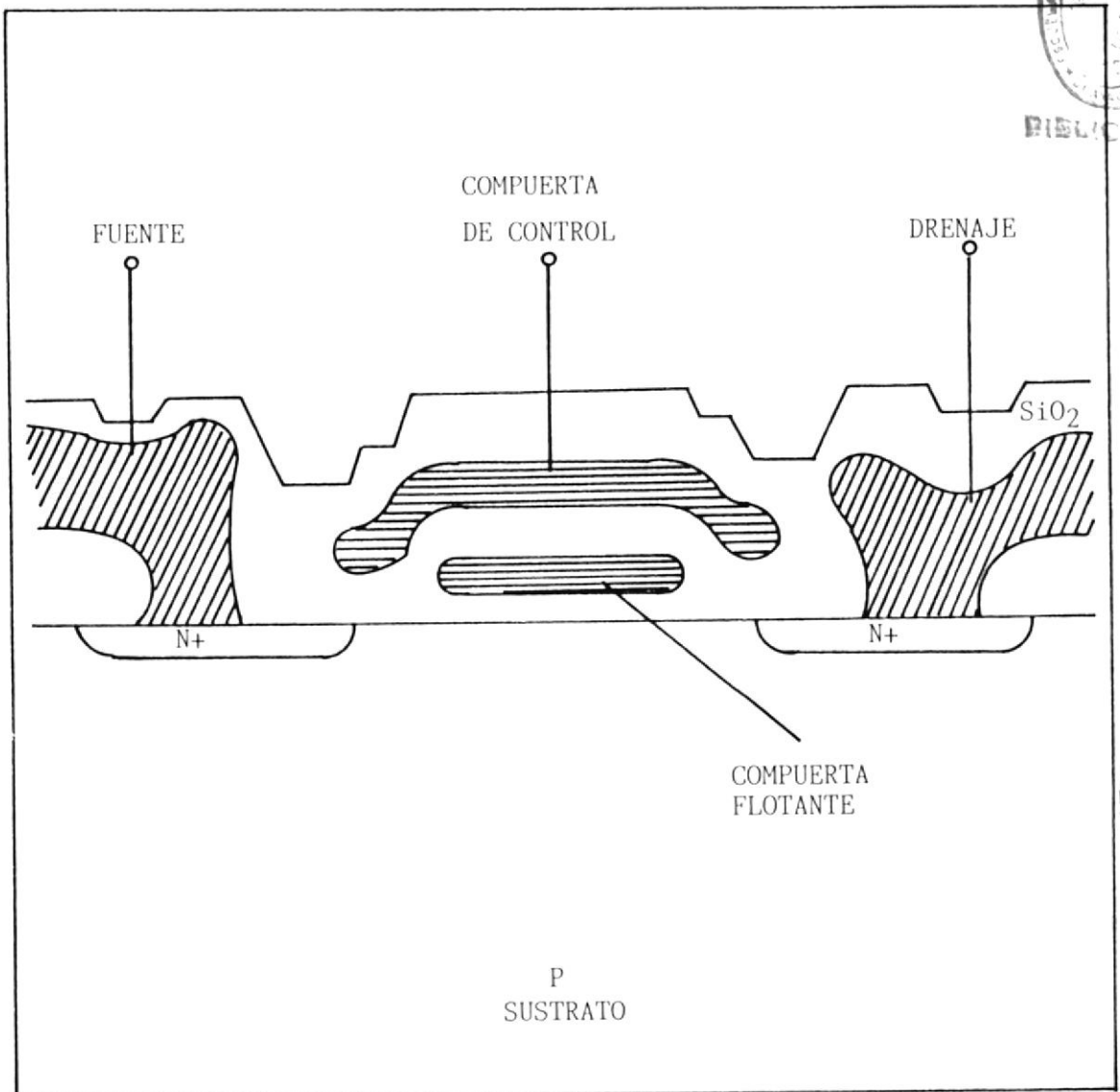


Figura 1.5. Celda MOS UV borrable.

La compuerta superior es la compuerta de control y está conectada al circuito

cuito del decodificador de fila, y la más baja es la compuerta flotante la cual se encuentra sin conexión, sumergida en dióxido de silicio.

Durante la programación un nivel de voltaje elevado es aplicado a la compuerta de control y al drenaje; el cual fuerza la unión de la celda deseada a una condición de avalancha. Esta diferencia de potencial causa la inyección de electrones hacia el área de la compuerta flotante a través de la barrera de dióxido de silicio y quedar atrapados en ella, después que este voltaje es eliminado la compuerta flotante retiene esta carga, puesto que se encuentra eléctricamente aislada. Esta carga sobre la compuerta flotante resulta en la formación de una región de inversión conductora en el sustrato, la cual provee un canal entre la fuente y el drenaje; y es la presencia o ausencia de este canal conductivo que determina si un "uno" o un "cero" es almacenado en la celda de memoria.

Una tapa de cuarzo transparente es colocado sobre el paquete sellado del chip para permitir el borrado por medio de Radiación Ultra-Violeta, la cual foto-energéticamente eleva la conductividad del dióxido de silicio y permite a la compuerta flotante descargarse. El diagrama completo de una MOS-ROM estática se muestra en la figura 1.6.

Esta memoria tiene una organización de 2048 palabras de 8 bits cada una. Los niveles de entrada-salida son compatibles con los niveles TTL (Transistor Transistor Logic) y las tres líneas de selección permiten a la rom ser seleccionada sin necesidad de decodificación externa.

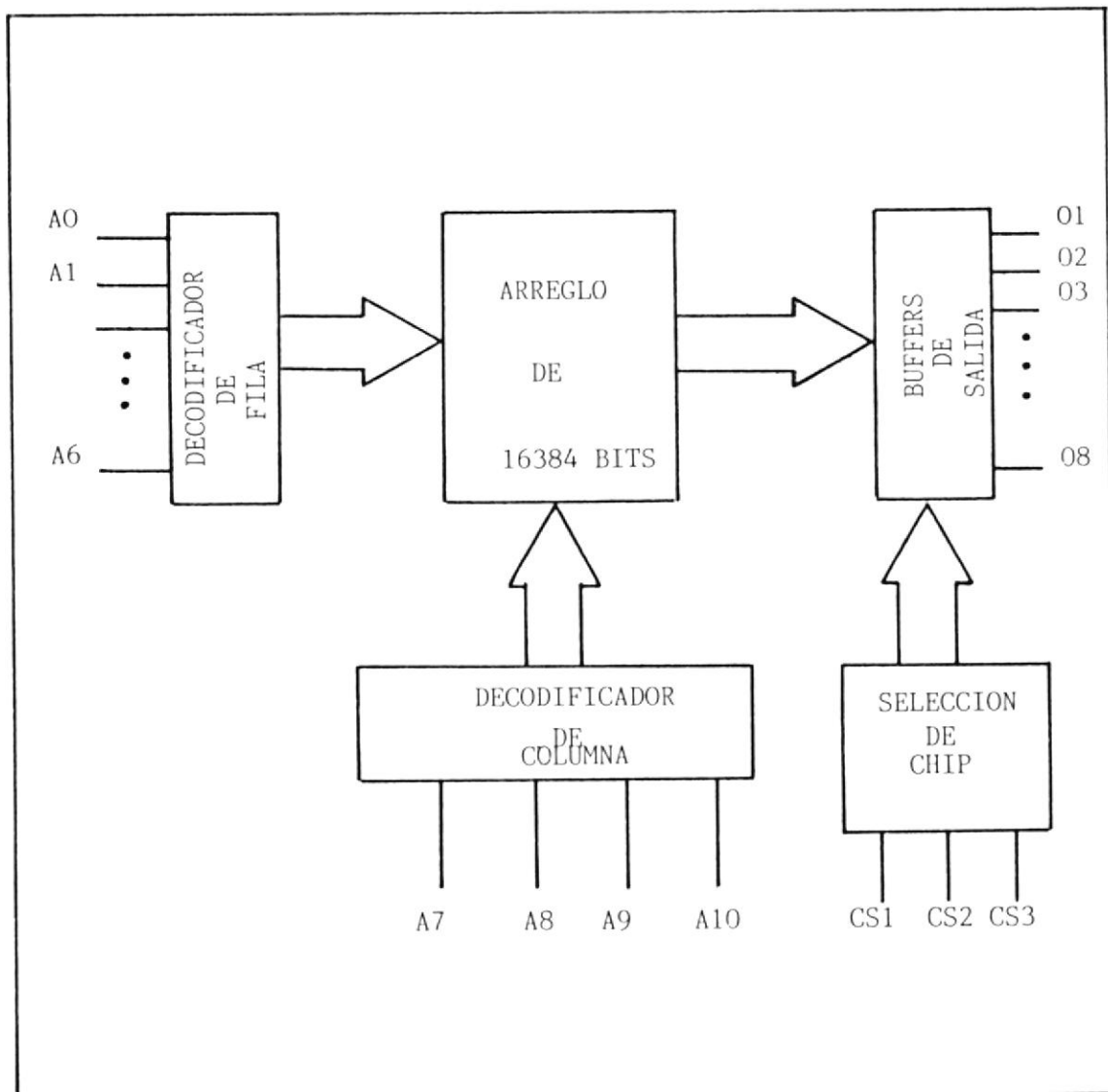


Figura 1.6. MOS ROM Estática, 2048 palabras de 8 bits.

Otra clase de memoria borrable es la EEPROM (Electrically Erasable PROM), que utiliza tecnología MOS con celdas estructuradas en forma similar a las celdas ultra-violeta mostradas en la figura 1.5. La EEPROM puede ser borrada con un pulso de voltage más bien que con radiación ultravioleta. Al



gunas versiones permiten el borrado del arreglo entero, en tanto que otras sólo el borrado individual de ciertos bytes.

Ahora se mencionará algunas características generales de las memorias PROM, las cuales se las puede clasificar en BIPOLARES y MOS. Las primeras construídas a base del fusible de enlace y aunque no pueden ser borradas y reprogramadas tienen una amplia aceptación debido a su velocidad, bajo costo y baja susceptibilidad a la radiación. Sus configuraciones más comunes son 32x8, 256x4, 256x8, 512x4, 512x8, 1024x4, 1024x8, 2048x4, 2048x8, 4096x4 y 4096x8; con tiempos de acceso típicos de 40ns a 80ns para proms bipolares convencionales y de 10ns a 20ns para proms en tecnología ECL. (Emitter Coupled Logic). Las segundas a menudo denominadas como EPROMs son más lentas que las anteriores, pero la ventaja que tienen es su reprogramabilidad y sus arreglos de gran tamaño. Sus configuraciones más comunes son las siguientes: 256x8, 1048x8, 4096x8 y 8192x8, con tiempos de acceso típicos que van desde los 250ns para los más nuevos modelos hasta los 1000ns para los modelos antiguos tal como la 1702A.

Las EEPROM tienen como característica principal, que pueden ser programadas y borradas en el mismo circuito impreso. Las configuraciones más comunes de EEPROM son: 32x16, 256x4, 1024x4, 2048x4 y 2048x8, con tiempos de acceso de lectura variando desde los 250ns a 10ns/palabra y tiempos de escritura desde 1ms a 50ms/palabra y con tiempos de borrado de 10ms a 100ms/palabra.

## 1.2 DISEÑO DEL CIRCUITO EMULADOR

Una vez que se ha descrito las características más relevantes de una ROM, así como las diferencias entre las variedades existentes en el mercado; se entrará a tratar ahora el objetivo y aplicación del EMULADOR.

### 1.2.1 OBJETIVO Y APLICACION

El diagrama de bloques básico de un sistema digital con microprocesador se muestra en la figura 1.7, donde el bloque denominado ROM contiene una secuencia de instrucciones definidas que indicarán al microprocesador o a los periféricos las tareas que deberán realizar. Pero, para llegar hasta este nivel, el programa existente debió someterse a un proceso de depuramiento hasta alcanzar su grado de efectividad, es decir, debió ser escrito y corregido muchas veces.

Este trabajo tedioso puede ser resuelto en alguna medida utilizando dispositivos electrónicos más apropiados para este propósito tales como las memorias borrables UV-EPROM o EEPROM. Pero aún este tipo de memorias presentan el inconveniente de programarlas y borrarlas con equipos especiales continuamente hasta que el programa en proyecto sea desarrollado en su totalidad. Además, algo muy importante de con-

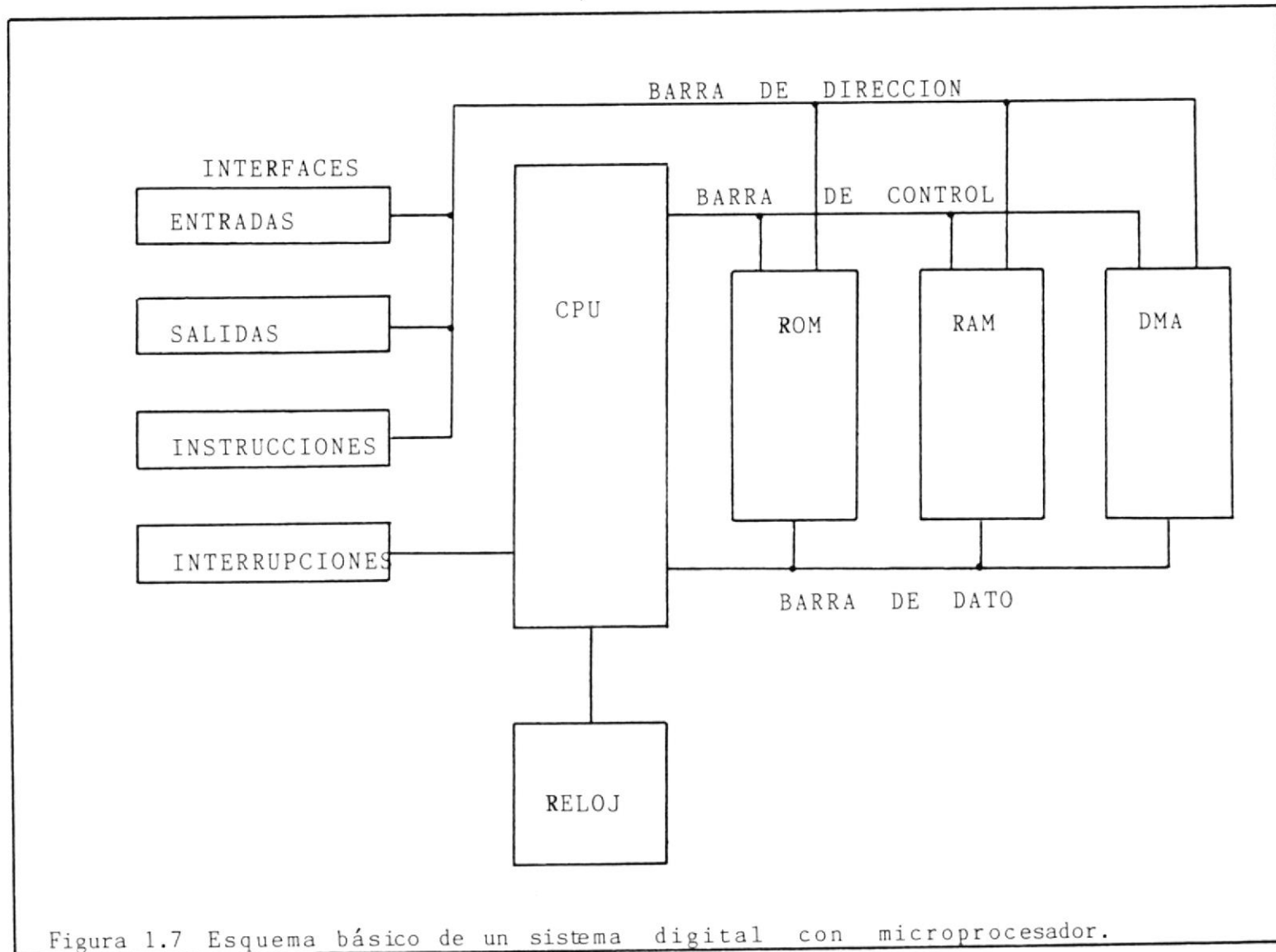


Figura 1.7 Esquema básico de un sistema digital con microprocesador.

siderarse es que estas memorias sólo pueden ser borradas y reprogramadas un limitado número de veces.

Con el EMULADOR se resuelven estas dificultades, porque es básicamente un emulador de EPROMs que se puede interconectar a cualquier computador que tenga una puerta serie RS-232C. El software cargado en el Emulador es ejecutado en el sistema deseado como si él estuviera en una EPROM. Por lo tanto, es ideal para quienes desarrollan programas que trabajan con microprocesadores.

#### 1.2.2 ESTUDIO Y DESARROLLO DEL EMULADOR

Ahora se entrará a tratar en detalle el diseño del circuito del Emulador, el cual para mayor facilidad se lo ha dividido en diferentes bloques que serán considerados oportunamente a medida que se avance con el desarrollo del circuito.

En la figura 1.8 se puede observar el diagrama de bloques general del circuito del Emulador. Como aquí se aprecia, en el Modo de Carga el dato en forma serie se recibe a través de un UART (Universal Asynchronous Receiver Transmitter) siendo este último completamente programable de tal forma que pueda ajustarse a las exigencias de cualquier computador. Después de recibir y procesar esta información, ella

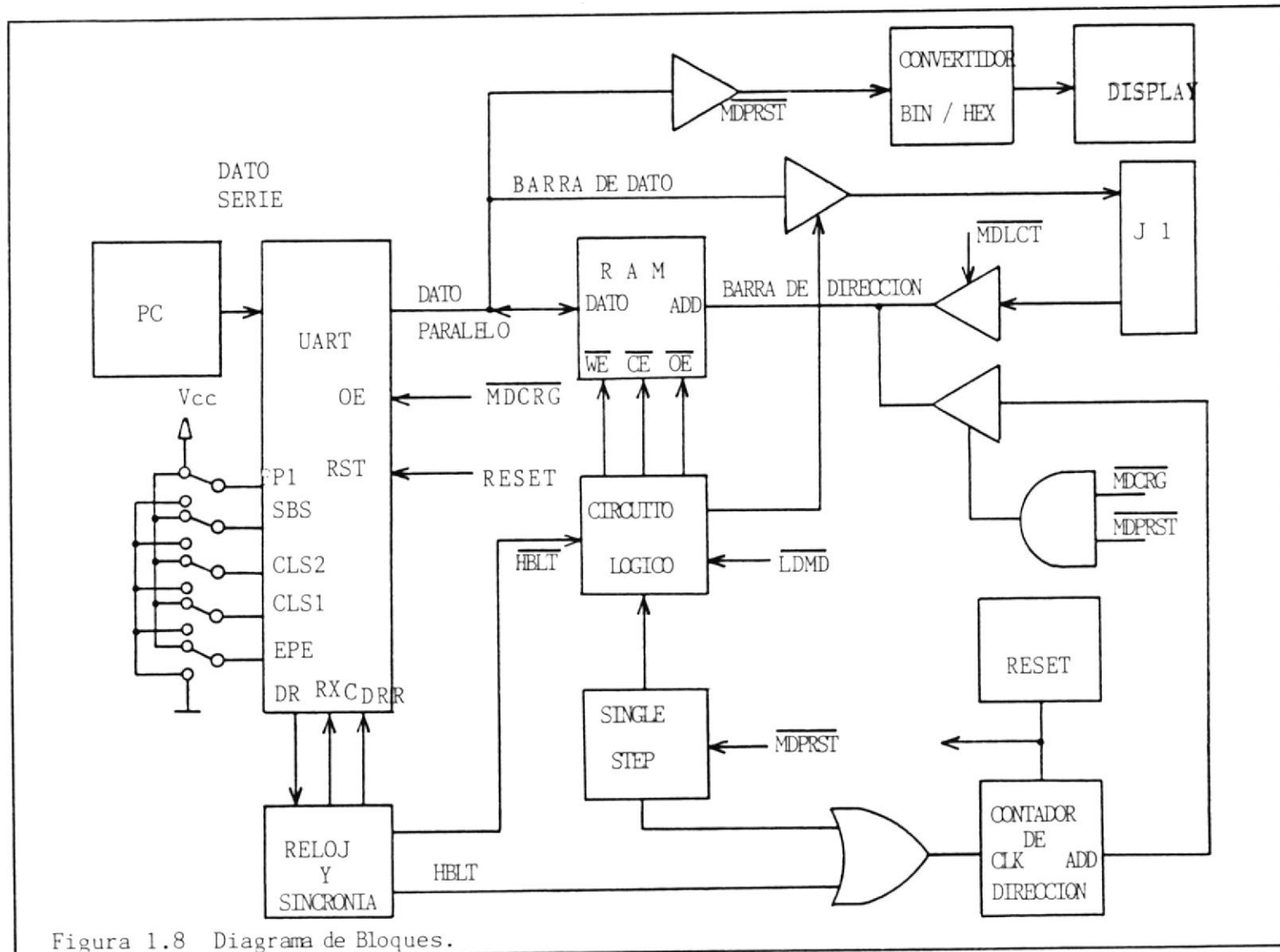


Figura 1.8 Diagrama de Bloques.

es entregada en la barra de dato interna en forma paralela para ser almacenada en la RAM. La señal de control para la escritura de la información es generada en el mismo UART, y sincronizada por medio de la señal de reloj del circuito. En el bloque denominado Reloj y Sincronía se genera el pulso de incremento del Contador de Dirección, el cual apunta la localización de memoria donde se almacenará el byte para su posterior lectura. En el Modo de Lectura los buffers que controlan la dirección del flujo de los bits, cambian del estado de alta impedancia al estado de conducción habilitado de esta forma la barra de dato y la barra de dirección del circuito del Emulador para que puedan ser accesadas por un circuito externo, que leerá la información almacenada en la RAM, como si ésta estuviese en una EPROM. Existe también un tercer modo de operación denominado Modo de Presentación con el que se puede leer el contenido de la RAM y presentarlo en el display paso a paso para verificar que toda la información que se envió desde el computador fue recibida por el UART, convertida a paralelo y almacenada correctamente en la memoria.

### 1.2.3 PROTOCOLO DE TRANSMISION ASINCRONO SERIE

Antes de realizar el análisis detallado de la configuración interna del Emulador, se procederá a mencionar algunas par-

particularidades del protocolo de transmisión.

Los bits transmitidos en forma serie y asíncrona están sujetos a un problema común causado por la inexactitud de las señales de reloj entre el transmisor y el receptor, dando origen de esta forma a los errores de transmisión. Esta incompatibilidad entre las señales de reloj puede ser compensada por medio de una RESINCRONIZACION PERIODICA, la que se realiza carácter por carácter. Dicha resincronización está coordinada por el uso del Bit de Inicio (start bit) y los Bits de Alto (stop bits) que colectivamente son denominados como Bits de Estructura (framing bits).

En la figura 1.9 se muestra el protocolo de transmisión serie de un canal de comunicación. Cuando el canal de comunicación se encuentra desocupado debe estar en un "1" lógico o un estado de marca (marking state). Si el transmisor desea enviar un carácter, primero envía un bit de inicio co-

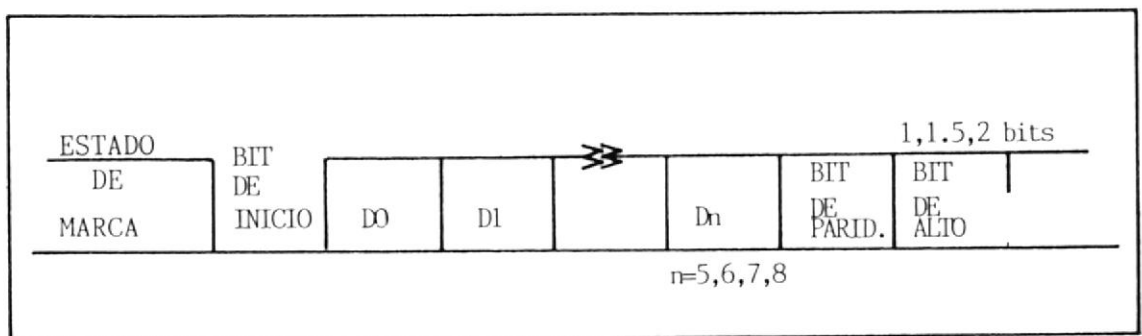


Figura 1.9. Protocolo de Transmisión Serie Asíncrono

locando un "0" lógico sobre la línea por un BIT TIME (el tiempo de duración de un simple bit, previamente acordado entre ambos), de esta forma el receptor detecta la transición del canal de desocupado a activo cuando la línea va a un nivel bajo de tensión por un bit time.

También se produce un problema potencial causado por las oscilaciones intermitentes de una línea desocupada, las que ocurren por un tiempo muy corto creando la impresión de ser un bit de inicio sobre el receptor. Por tal razón, estos equipos son diseñados con una lógica especial para detectar dichas oscilaciones. Esta lógica muestrea la línea desocupada muchas veces en la velocidad de transmisión del dato; es decir, el receptor puede muestrear un canal desocupado 2, 4, 16 o más veces durante un bit time.

La lógica detectora de oscilaciones asencialmente ayuda a asegurar la validez de un bit de inicio chequeando que el nivel bajo en la línea permanezca en ese estado hasta el final de un bit time.

Una vez que el bit de inicio es validado por el receptor, este fija su registro de desplazamiento para empezar a aceptar los bits de dato. Los Bits de Dato pueden ser 5, 6, 7 u 8; siendo usualmente 7 u 8 bits los transmitidos.



Siguiendo a los bits de dato un Bit de Paridad puede ser transmitido. La convención del bit de paridad es un mecanismo para implementar un esquema limitado de detección de error. Varios mecanismos de chequeo de error pueden ser implementados para reducir la probabilidad que un error de bits no detectado exista entre el dato que el destino recibió y el dato que la fuente intentó que el destino reciba. Uno de los métodos para detectar errores es que la fuente provea al destino con información redundante, y el método obvio de proveer información redundante es enviar el mismo dato dos veces, así el destino puede compararlos y si los dos concuerdan, él puede asumir que la información es correcta o de lo contrario él podrá concluir que algo ocurrió en el proceso de transmisión. Pero esto no es económico, además disminuye la velocidad de transmisión del canal a la mitad o requiere aumentar a dos veces el número de líneas de señal dependiendo de cómo sea él implementado.

Un método más económico de transmitir información redundante es entonces necesario, donde el dato mismo no sea repetido, sino una característica de él, tal como su paridad. La Paridad es una característica del dato determinada por el número de unos lógicos que son contenidos en él mismo, más el bit de paridad inclusive. Existen dos selecciones, donde el bit de paridad puede ser determinado tal que el núme-

ro de unos lógicos sobre el dato sea par o impar. Paridad Par significa que un número par de unos existirá en la suma de la paridad más los bits de dato y Paridad Impar es cuando existe un número impar de unos entre la paridad más los bits de dato. Esto se ilustra con el siguiente ejemplo:

Byte de 8 bits	Bit de Paridad	
	Paridad Par	Paridad Impar
0000 0000	0	1
0000 0010	1	0
1111 1111	0	1
1010 0110	0	1

Hay que tener en cuenta que este método tiene un inconveniente y es que si un número par de bits inválidos son recibidos, el destino no va a notar la inconsistencia de paridad. Lo descrito anteriormente es el esquema de detección de error utilizado con el protocolo asíncrono serie de Inicio-Alto. El cálculo del bit de paridad a transmitirse por la fuente y el cálculo y comparación de la paridad en el destino pueden ser realizados por hardware o software. Típicamente, existen interfaces series asíncronas tales como los SIO, UART, USART, etc. que rea-

lizan estas funciones. Siguiendo a los bits de dato y al bit de paridad (si existe alguno), el transmisor envía 1, 1.5 o 2 Bits de Alto. Estos bits de alto son simplemente unos lógicos con una duración de 1, 1.5 o 2 bit time. Los bits de alto obligan a la línea asumir la condición de marca por al menos un bit time antes del próximo carácter. Esto significa que un bit de inicio empezará con una transición de nivel desde un uno lógico a un cero lógico. Hay que notar, que al menos un bit de alto es necesario para asegurar que cada carácter empiece con una transición de 1 a 0, y esta característica es importante en la detección de errores de estructura.

#### 1.2.4 DESCRIPCION FUNCIONAL DETALLADA DEL CIRCUITO EMULADOR

Para facilitar la descripción del circuito del Emulador, este se lo ha dividido en varios bloques los cuales serán analizados a continuación:

##### 1.2.4.1 CIRCUITO DE INTERFACE SERIE

Este primer bloque a considerarse es el más importante porque tiene que ver con la recepción de la información enviada en forma serie desde el computador, y se lo ha denominado CIRCUITO DE INTERFACE

SERIE.

En la figura 1.10 se muestra el diagrama esquemático de este bloque, el cual trabaja exclusivamente cuando el Emulador se encuentra en el Modo de Carga.

Los parámetros mencionados en la sección anterior son fijados en el UART mismo utilizando los switches S9, S8, S7, S6 y S5, que están conectados a los pines PI (PARITY INHIBIT), SBS (STOP BIT SELECT), CLS2 y CLS1 (CHAR LENGHT SELECTED) y EPE (EVEN PARITY ENABLE) respectivamente. En la tabla I se muestran las diferentes combinaciones posibles del protocolo de transmisión el cual deberá ser seleccionado y fijado antes de iniciar la transmisión. La sincronización del dato que llega al Emulador se realiza con el circuito integrado U11, el cual consiste en un juego de cuatro flip-flops tipo D conectados en cascada.

En la figura 1.11, se puede observar que la señal  $\overline{DR}$  se hace falsa aproximadamente  $8 \frac{1}{2}$  pulsos de reloj después que el UART detectó el primer bit de alto. El primer flip-flop lee este nivel

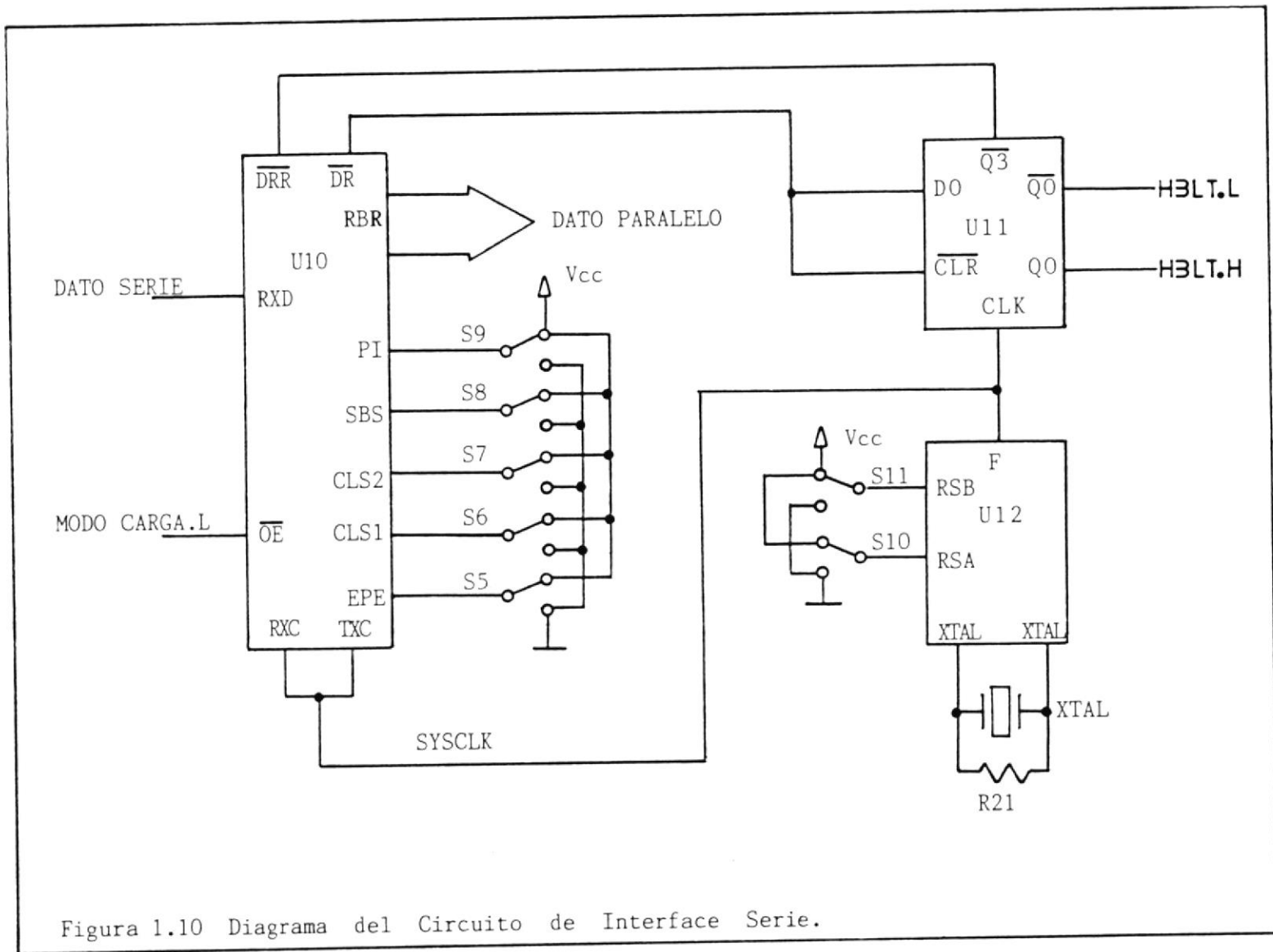


Figura 1.10 Diagrama del Circuito de Interface Serie.

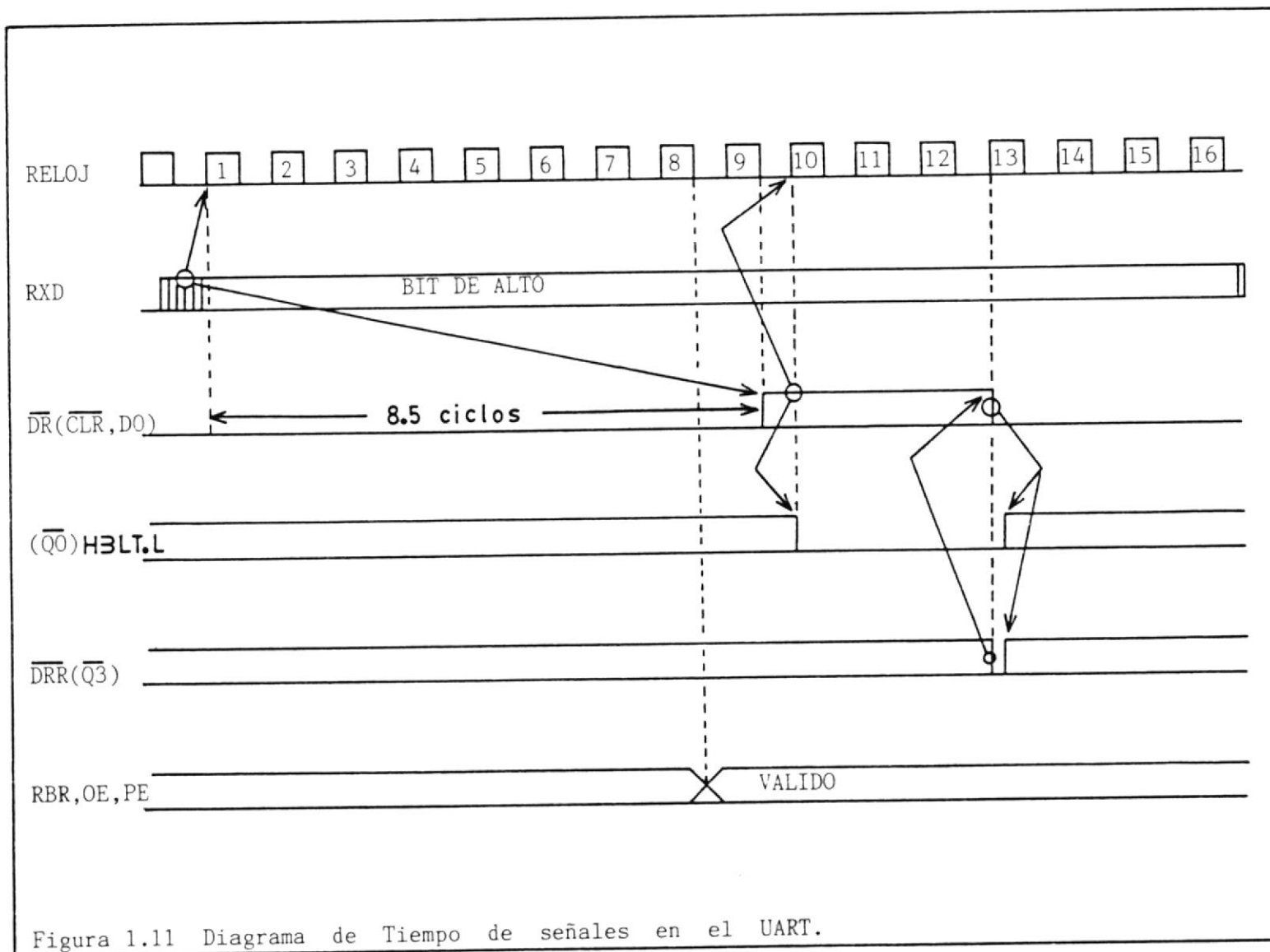


Figura 1.11 Diagrama de Tiempo de señales en el UART.

con el pulso de reloj que llega y origina así en sus terminales Q0 y  $\overline{Q0}$  las señales HBLT.H HBLT.L que son las señales de incremento de la dirección y de habilitación-lectura (CE.L, WB.L) de la RAM. Con la pendiente positiva del ~~décimo-~~ tercer pulso de reloj, Q3.L se hace verdadero obligando a cambiar de nivel a DR.L y por lo tanto fijar a cero a U11. Esta señal Q3.L también denominada DRR.L indica al UART que ha pasado el tiempo prudencial para que el byte haya sido escrito en la RAM.

El circuito integrado U12, el cual es un BAUD RATE GENERATOR, es el designado a generar los pulsos de reloj necesarios para que opere el Emulador. El trabaja con un cristal de 1.8432 MHz siendo capaz de generar todas las frecuencias necesarias para igualar cualquier velocidad de transmisión que varíe desde los 75 Hz hasta los 614.4 kHz. Aunque, 19.2 kbps es la velocidad de transmisión más alta usualmente utilizada en aplicaciones con transmisión RS-232C. Los switches S11 y S10 permiten seleccionar el factor de la velocidad de transmisión, tal como se puede observar en la tabla II. La señal de reloj es

TABLA I

## TABLA PARA SELECCION DE PROTOCOLO DE TRANSMISION

S7	S6	S9	S5	S8	BIT DE INICIO	BIT DE DATOS	BIT DE PARIDAD	BIT DE ALTO
0	0	0	0	0	1	5	impar	1
0	0	0	0	1	1	5	impar	1.5
0	0	0	1	0	1	5	par	1
0	0	0	1	1	1	5	par	1.5
0	0	1	X	0	1	5	ninguno	1
0	0	1	X	1	1	5	ninguno	1.5
0	1	0	0	0	1	6	impar	1
0	1	0	0	1	1	6	impar	2
0	1	0	1	0	1	6	par	1
0	1	0	1	1	1	6	par	2
0	1	1	X	0	1	6	ninguno	1
0	1	1	X	1	1	6	ninguno	2
1	0	0	0	0	1	7	impar	1
1	0	0	0	1	1	7	impar	2
1	0	0	1	0	1	7	par	1
1	0	0	1	1	1	7	par	2
1	0	1	X	0	1	7	ninguno	1
1	0	1	X	1	1	7	ninguno	2
1	1	0	0	0	1	8	impar	1
1	1	0	0	1	1	8	impar	2
1	1	0	1	0	1	8	par	1
1	1	0	1	0	1	8	par	2



BIBLIOTECA





1	1	1	X	0	1	8	none	1
1	1	1	X	1	1	8	none	1

tomada desde el zócalo J2, y esto se lo consigue uniendo el pin denominado SYSCLK con cualquiera de los otros pines que van desde F1 a F16 para obtener la frecuencia deseada de acuerdo a la tabla III. Este circuito integrado es deshabilitado por medio del transistor Q2 cuando el Emulador pasa a trabajar en otro modo de operación.

TABLA II

TABLA DE SELECCION DEL FACTOR DE VELOCIDAD

SELECCION DEL FACTOR DE VELOCIDAD		FACTOR DE VELOCIDAD
S11	S10	
0	0	X1 - no utilizado
0	1	X8 - no utilizado
1	0	X16-
1	1	X64- no utilizado

TABLA III

TABLA DE FRECUENCIAS GENERADAS EN EL CIRCUITO INTEGRADO U12

PIN DE SALIDA	FRECUENCIA DE RELOJ X16	bps (BAUDIOS) X1
F1	153.6k	9600
F2	115.2k	7200
F3	76.8k	4800
F4	57.6k	3600
F5	38.4k	2400
F6	28.8k	1800
F7	19.2k	1200
F8	9600	600
F9	4800	300
F10	3200	200
F11	2400	150
F12	2153.3	134.5
F13	1.758.3	109.9
F14	1200	75
F15	921.6k	921.6k
F16	1.8M	1.8M

#### 1.2.4.2 CIRCUITO DE MEMORIA

En la figura 1.12, se muestra el diagrama de bloques del circuito a tratarse ahora, el cual se

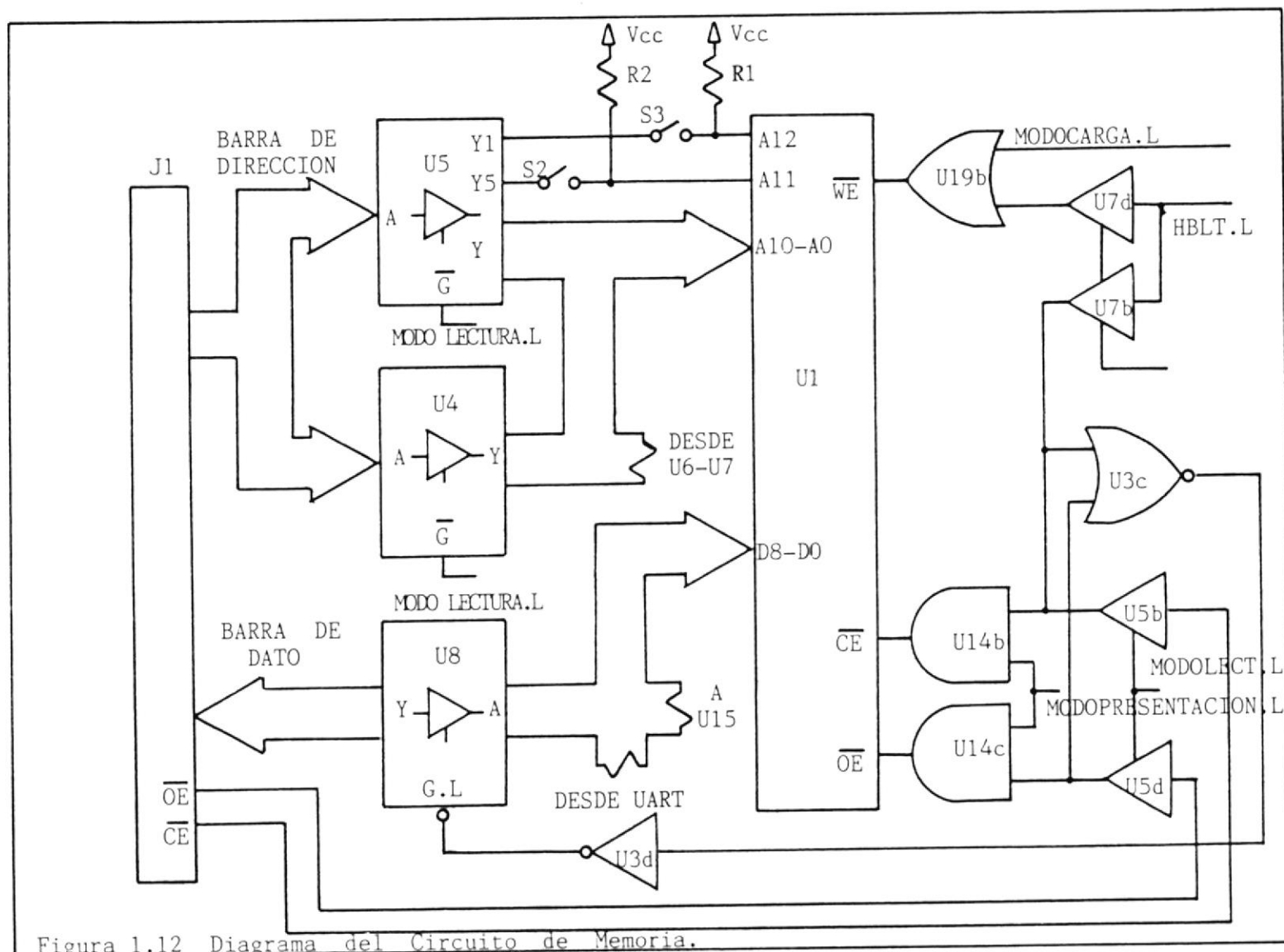


Figura 1.12 Diagrama del Circuito de Memoria.

lo ha denominado CIRCUITO DE MEMORIA. El está formado principalmente por una memoria tipo RAM de 8192 palabras de 8 bits, que va a ser la receptora de la información enviada desde el computador a través del UART.

Este bloque puede trabajar en los tres modos de operación, que son: Modo de Carga, Modo de Lectura y Modo de Presentación.

En el Modo de Carga, seleccionando adecuadamente la posición de los switches S3 y S2 se puede emular una memoria 2716, 2732 o 2764. La función de estos es conectar o desconectar las líneas de dirección A12 y A11 de la RAM, de tal forma que cuando se emula una 2716 la información sea almacenada en la sección de memoria que va desde la dirección 1800 (hex) hasta 1FFF (hex), para una 2732 lo será desde 1000 (hex) a 1FFF (hex) y finalmente para una 2764 lo será desde 0000 (hex) a 1FFF (hex), es decir, ocupará la memoria entera.

Los circuitos U7 y U5 son buffers con salidas de tres estados los cuales gobiernan la barra de control cuando el Emulador opera en el modo de car

ga o de lectura. En el modo de carga el circuito U7 se encuentra en el estado de baja impedancia, permitiendo el paso de la señal HBLT.L hacia la puerta OR (U19b) la cual da la autorización para escritura de información en la memoria; el otro terminal de la puerta está conectado directamente a la señal MODO-CARGA.L con la intención de permitir únicamente en este modo el acceso del pin WE.L de la memoria, que por encontrarse conectado con la salida de la puerta lógica, en él habrá un nivel bajo (indicación de escritura) cada vez que HBLT.L sea verdadero.

Esta señal HBLT.L también se utiliza para la habilitación de la memoria a través de una puerta AND (U14b), la cual tiene en su otro terminal de entrada la señal MODO PRESENTACION.L, que para este caso se encuentra en un estado de falso, y su salida por consiguiente, estará conectada al pin CE.L de la memoria. En definitiva se puede decir que el ciclo de escritura de la memoria es controlado completamente por el UART. La barra de dato de la memoria está conectada directamente al UART, en cambio que la barra de dirección se une a la del resto del circuito (contador de di-

rección) a través de los buffers con salidas de tres estados U6 y U7.

Cuando el Emulador va a ser accesado por un circuito externo, debe ser fijado en el modo de operación de lectura, utilizando para ello el switch S1. Las conexiones eléctricas necesarias para llevar a cabo esto se lo hace a través del conector J1. En este modo la barra de dirección se conecta desde el exterior a la memoria por medio de los buffers U5 y U4 que pasan del estado de alta impedancia al de baja. Además, los buffers U6 y U7 pasan al estado de alta impedancia, aislando al contador de dirección U2.

El flujo de dato hacia el exterior desde la memoria se realiza a través del buffer U8, cuya habilitación es controlada por las señales OE.L y CE.L que vienen igualmente desde el exterior, pasan por U5 y se aplican a la puerta NOR (U3c), luego su salida es invertida y aplicada a los pines 1G.L y 2G.L de U8.

El tercer modo de operación es el de Presentación del dato previamente almacenado; por medio de un

display. La selección de este modo y la habilitación de la memoria al igual que sus salidas se realiza con el switch S1, es decir, es aplicado un nivel bajo a los pines de entrada de las puertas AND U14b y U14c, originando así un estado verdadero en los pines CE.L y OE.L de la memoria. El contador de dirección es puesto a cero con el switch S4 e incrementado manualmente bit a bit con el switch S12. De esta forma es accesada la memoria y leído su contenido, el cual pasa a través del buffer U15 que se encuentra en un estado de baja impedancia, y llega al latch U20 que es utilizado para agarrar el dato y entregarlo a los decodificadores binario a hexadecimal U16 y U17 para ser presentados en el display.

#### 1.2.4.3 CIRCUITO CONTADOR DE DIRECCION

El circuito a ser analizado ahora es el denominado CIRCUITO CONTADOR DE DIRECCION, el mismo que se ilustra en la figura 1.13. Su función principal es generar dos bytes de dirección para ser utilizados en el direccionamiento del dato que transmite el UART hacia la memoria para su almacenamiento cuando se está en el modo de carga, y

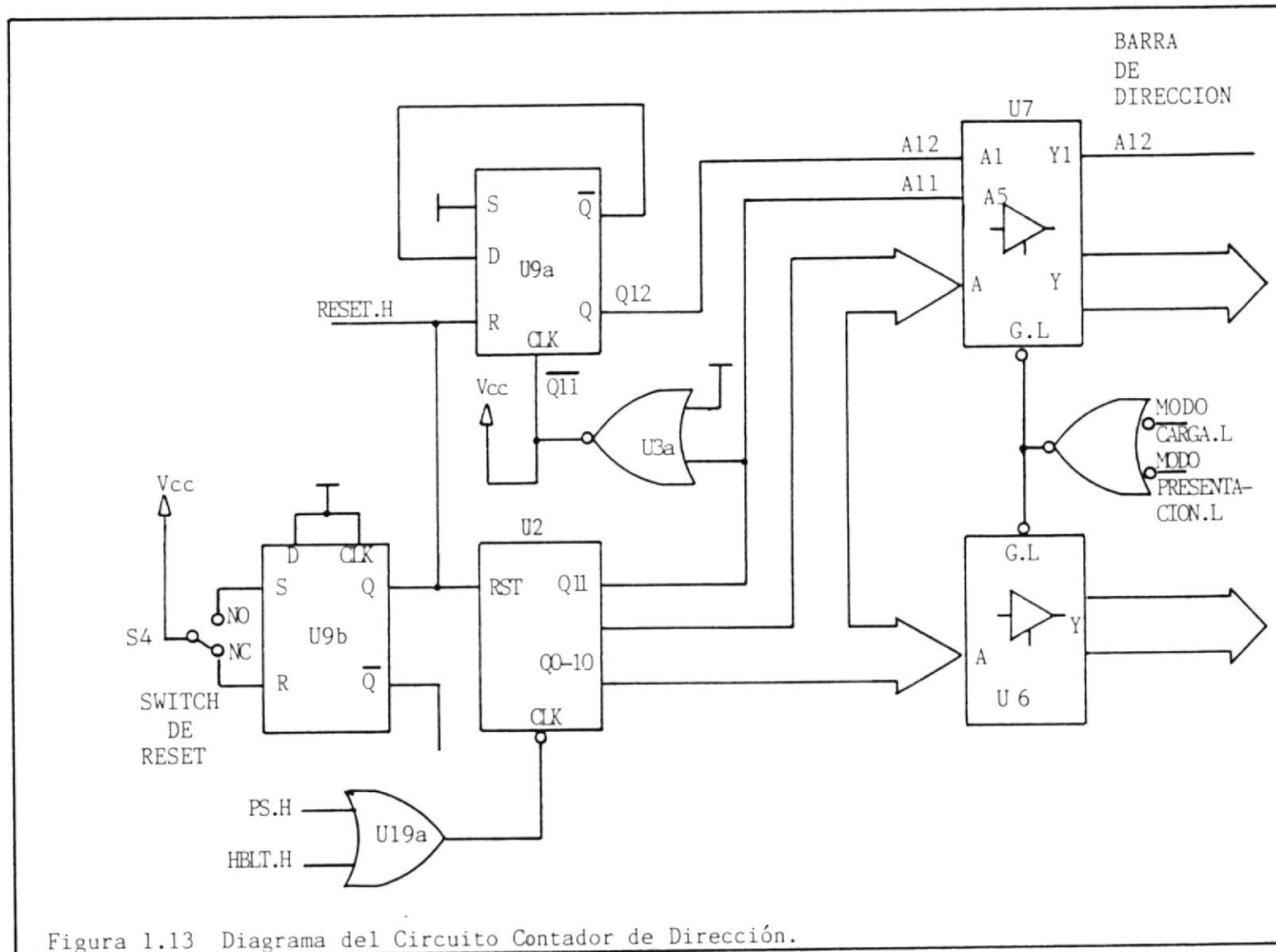


Figura 1.13 Diagrama del Circuito Contador de Dirección.



luego para ser leído nuevamente desde la memoria y presentarlo a través del display.

Este circuito está conformado básicamente por un contador binario de 12 etapas, el cual trabaja en dos modos de operación: el modo de carga y el modo de presentación.

En el modo de carga, los pulsos generados por el circuito integrado U11 (circuito de interface serie), llegan hasta el contador a través de una puerta OR (U19a), que actúa como selector, debido a que también pueden llegar los pulsos de incremento de dirección desde U18 (circuito de paso simple), cuando el Emulador opera en el modo de presentación.

La dirección es incrementada con la pendiente negativa de los pulsos PS.H o HBLT.H y conducida al resto de la circuitería a través de U7 y U6 (buffers de tres estados), los cuales se encuentran habilitados desde U14a.

El bit más significativo de la dirección A12 es generado a partir de A11, como se puede apreciar

en la figura 1.14.

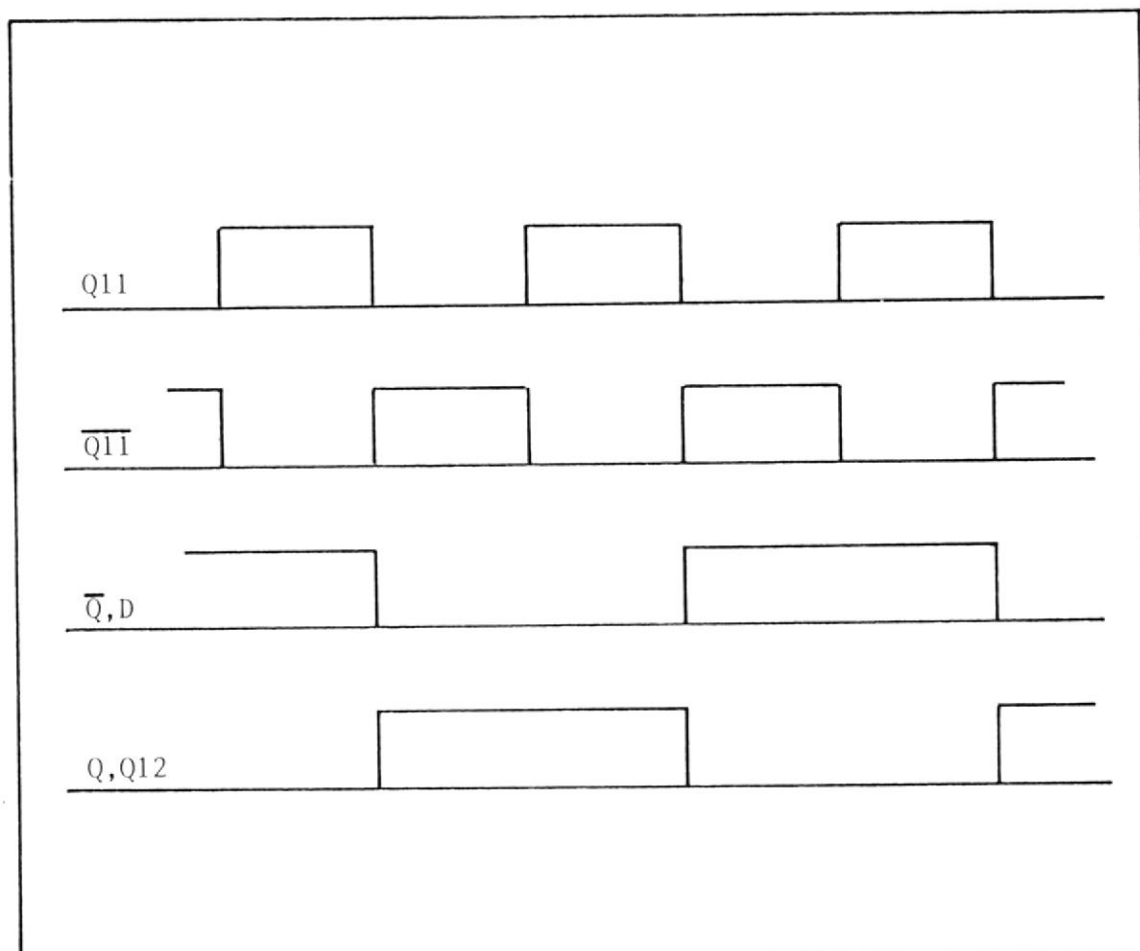


Figura 1.14. Diagrama de tiempo de generación del bit más significativo de la dirección.

La señal  $\overline{Q11}$  generada desde U2 es negada y utilizada como señal de reloj en el flip-flop U9a. Después del estado inicial que originalmente es cero ocurre el cambio a uno, entonces con la primera pendiente positiva de  $\overline{Q11}$  el valor de  $\overline{Q}$ , que es un

nivel alto en ese momento es leído en el pin D y transmitido al pin Q.H, esto cambiará el estado de Q.L a un nivel bajo. Com la próxima pendiente positiva de  $\overline{Q_{11}}$ , el valor de  $\overline{Q}$  (nivel bajo ahora) es nuevamente leído y transmitido al pin Q.H; esto cambiará otra vez el estado de  $\overline{Q}$  el cual será leído en la próxima pendiente positiva de  $\overline{Q_{11}}$ , y así sucesivamente.

En este bloque está incluido también el circuito de RESET de todo el Emulador, el cual está integrado por un flip-flop tipo D y un switch momentáneo normalmente cerrado. Cada vez que el switch S4 (Switch de Reset) es presionado, se aplica un nivel alto en el terminal SET del flip-flop, lo que generará un nivel alto en el pin de salida Q, siendo este el pulso de reset que será utilizado para poner a cero la dirección y se lo ha denominado RESET.H.

#### 1.2.4.4 CIRCUITO PULSADOR DE UN PASO

Continuando con el estudio del diseño del Emulador ahora se entrará a tratar el circuito denominado PULSADOR DE UN PASO el cual se muestra en la figu-



ra 1.15.

Su función es generar las señales necesarias para el incremento de la dirección y para autorizar al circuito U20 el agarre de la información. El diseño de este bloque está centrado sobre el circuito integrado U18, el cual es un multivibrador mono-estable con entradas schmitt-trigger, que es habilitado a través del pin 5 cuando es seleccionado el Modo de Presentación. Su utilización se hace necesaria debido a que el ancho de pulso del pulso producido en S12 (switch de paso simple) presenta una duración demasiado extensa la cual se hace necesario reducirla en algún grado.

Como se mencionó anteriormente, el propósito de utilizar un multivibrador en este bloque, fue con el objeto de reducir el ancho de pulso del pulso entrante, donde el flanco negativo del pulso producido en el switch S12 causa el disparo del multivibrador generando así un pulso más angosto eliminando de esta forma la excesiva duración que podría producir un re-disparo del mismo.

El ancho del pulso de salida está determinado por

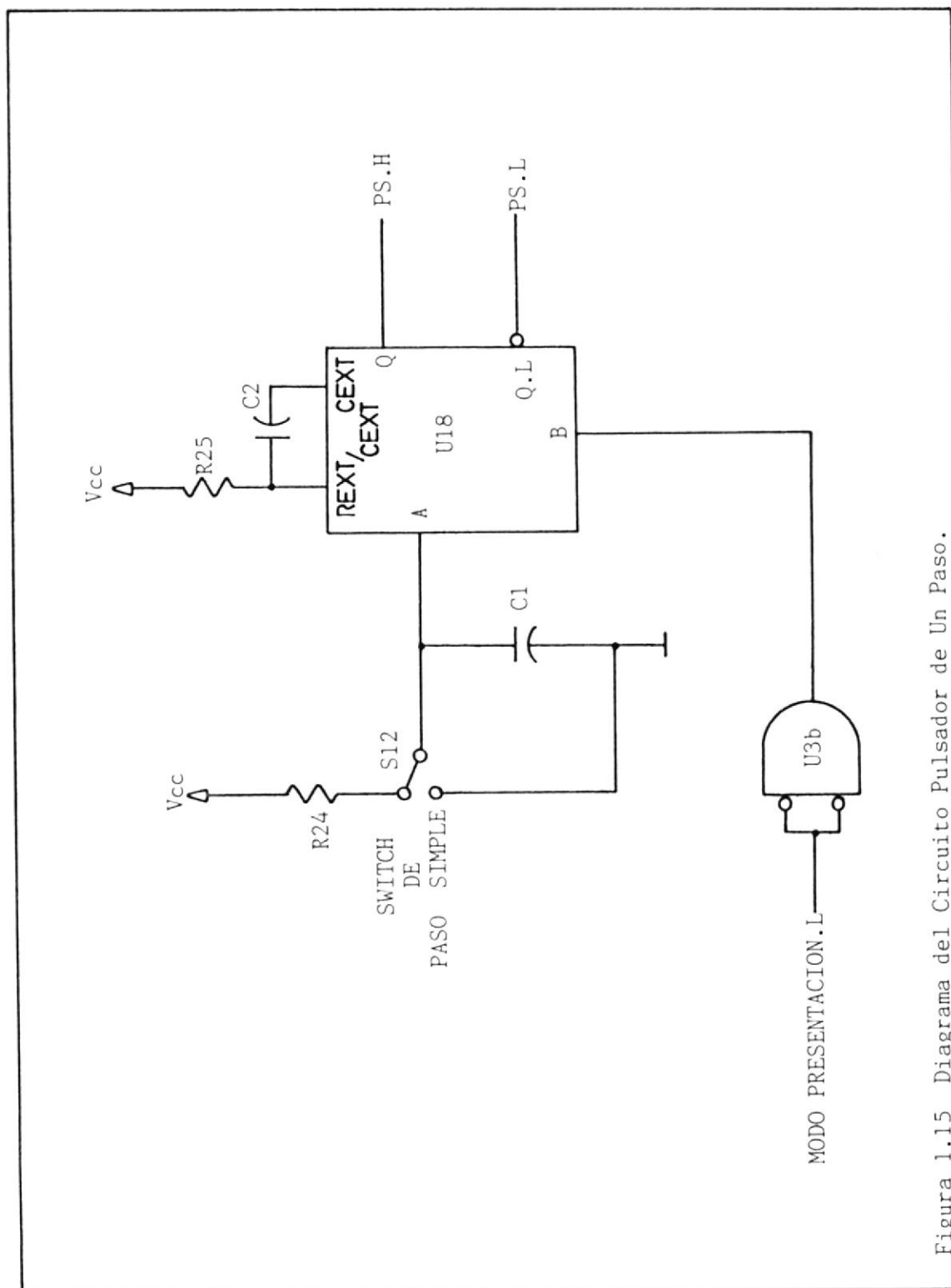


Figura 1.15 Diagrama del Circuito Pulsador de Un Paso.

el valor de la resistencia y el capacitor externos que se conectan al circuito integrado, cuyos valores pueden ser seleccionados a conveniencia utilizando la fórmula siguiente:

$T_w = 0.7 C_{ext} R_{ext}$  según referencia 14, Cap. 6 pp 64. El agarre del byte por U20 se produce con el flanco positivo de PS.H y el incremento de la dirección se produce en cambio con su flanco negativo, de tal forma que ella siempre apuntará al próximo byte a ser leído.

Este circuito es usado exclusivamente en el modo de presentación para leer bit a bit el contenido de la memoria y poder chequear visualmente que fue almacenada toda la información enviada desde el UART correctamente.

#### 1.2.4.5 CIRCUITO DE DISPLAY

El objetivo de este circuito es el de poder chequear visualmente el contenido de la memoria que fue previamente almacenado. Su diagrama se muestra en la figura 1.16.

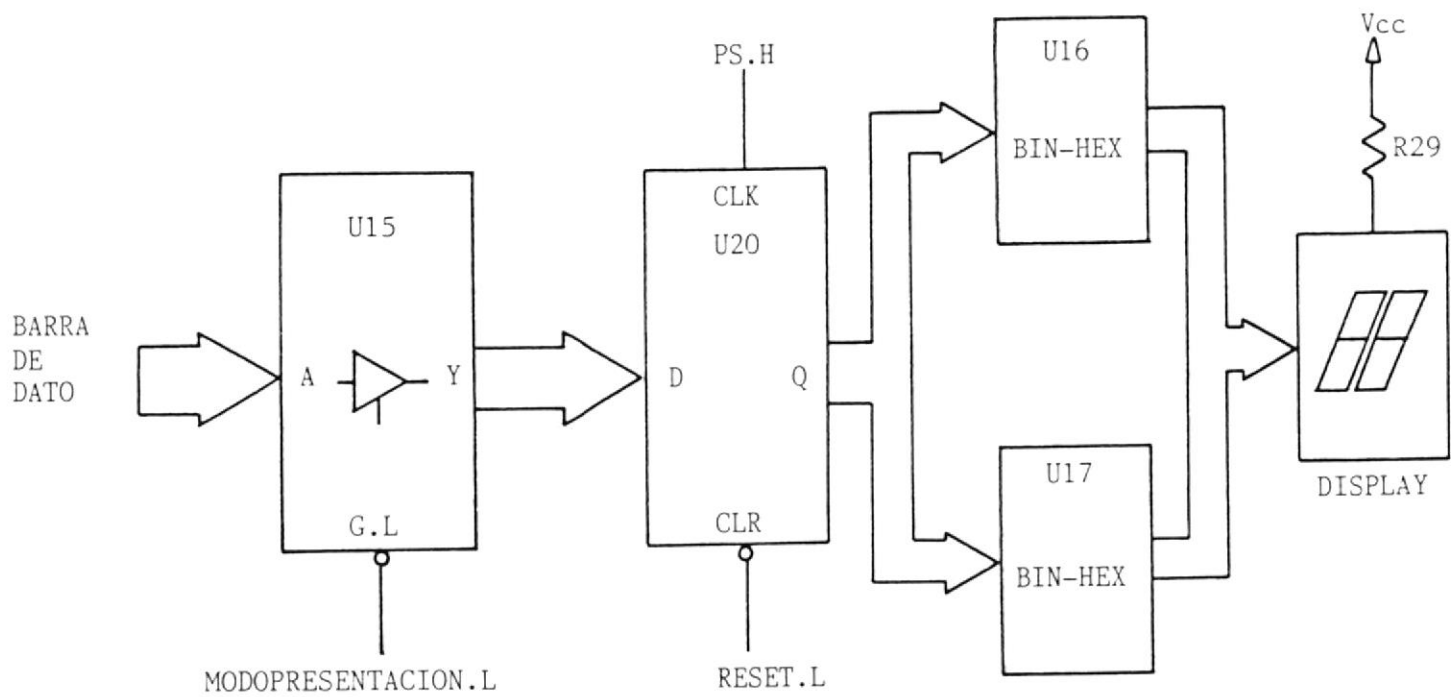


Figura 1.16 Diagrama del Circuito de Display.

El dato una vez que ha sido direccionado y se encuentra en la barra de dato, es conducido a través de U15 (buffer de 3 estados) hasta el latch U20 el cual agarrará esta información con la pendiente positiva de P.S.H señal que es aplicada al terminal del reloj del integrado. Este dato después que ha sido leído es aplicado a dos convertidores binario a hexadecimal y llevado luego hasta el display (ánodo común) para su presentación. Este circuito trabaja en conjunto con el circuito Pulsador de Un Paso únicamente en el modo de Presentación.

#### 1.2.4.6 CIRCUITO DE LA FUENTE DE PODER

##### 1.2.4.6.1 REQUERIMIENTOS PARA EL DISEÑO

El circuito del Emulador está conformado en su totalidad con circuitos integrados TTL, razón por la cual será requerido una tensión regulada de +5 voltios de para la polarización con una capacidad de corriente de 800. mA.

##### 1.2.4.6.2 RECTIFICACION Y FILTRADO



En la figura 3.7 se muestra el diagrama de la fuente de poder. Como se ha mencionado anteriormente se requiere obtener 10 voltios dc no regulados, con una capacidad de corriente de un amperio aproximadamente y con un rizado menor al 10%.

Utilizando un puente rectificador de onda completa y de acuerdo a la referencia 2, apéndice B figura B.3 se obtiene:

Para un rizado del 10%

$$\frac{V_{dc}}{V_m} = 0.852$$

$$V_m = \frac{V_{dc}}{0.852} = \frac{10}{0.852} = 11.74 \text{ voltios}$$

$$V_{rms} = \frac{V_m}{2^{1/2}} = \frac{11.74}{2^{1/2}} = 8.30 \text{ V}$$

El valor de la carga se obtiene de la siguiente manera:

$$R_l = \frac{V_{dc}}{I_{dc}} = \frac{10}{0.8} = 16.7 \text{ ohms}$$

Con la siguiente ecuación se procederá al cálculo del Capacitor de Filtro:

$$\%r = \frac{2.4}{R_L} \times 100$$

donde C está en microfaradios y  $R_L$  en kilohms.

$$C = \frac{2.4 \times 100}{\% r \times R_L} = \frac{2.4 \times 100}{10 \times 0.0167} = 1440 \mu\text{F}$$

$$C = 1440 \mu\text{F}$$

#### 1.2.4.6.3 REGULACION DE VOLTAGE

La regulación de voltage se hace necesaria debido a la utilización de circuitos integrados TTL, la misma que es realizada por medio del Regulador de Voltage TL 780-05C. A continuación se presenta algunas características eléctricas de este bloque:

TIPO	POT. MAX.	VOLT. ENT.	VOLT. SAL.	CORR. SAL.
TL 780-05C	8 W	+ 10 V	+ 5 V	1 Amp.

## CAPITULO II

### PROGRAMAS DE CONTROL: DISEÑO

#### 2.1 ALGORITMO DEL PROGRAMA PRINCIPAL

El Programa Principal, es aquel que permitirá el llamado a los demás programas con los que se podrán realizar diferentes operaciones tales como el ingreso, la transmisión y la corrección de la información que se almacenará en el Emulador, para su posterior utilización por un circuito externo.

En la figura 2.1 se muestra el diagrama de flujo del Programa Principal, donde se puede ver que la primera acción a realizarse es la presentación del Menú de Operación, lo cual tiene como finalidad mostrar las operaciones que se pueden realizar, y estas son:

- 1.- Mostrar indicaciones para operación del Emulador.

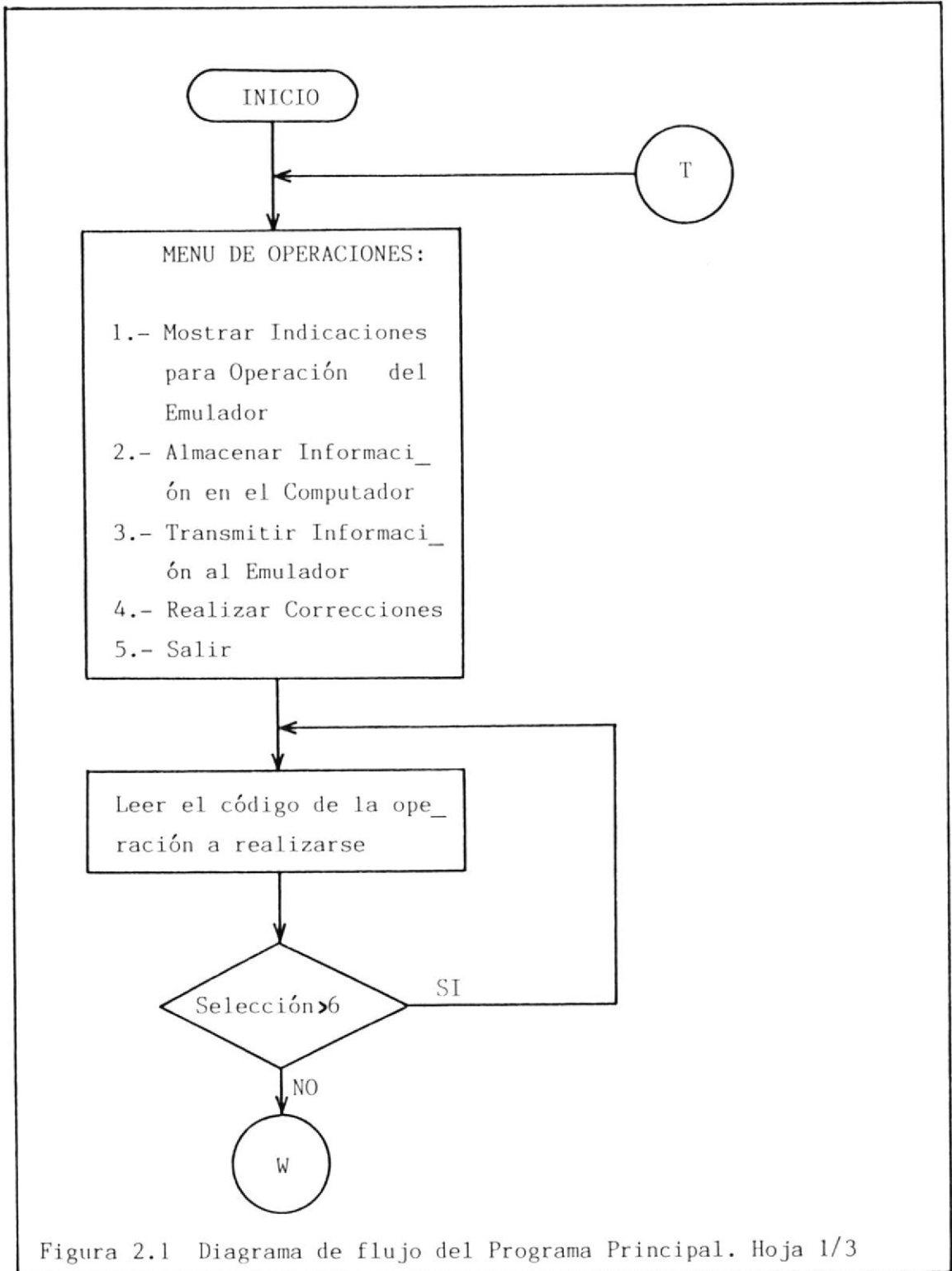
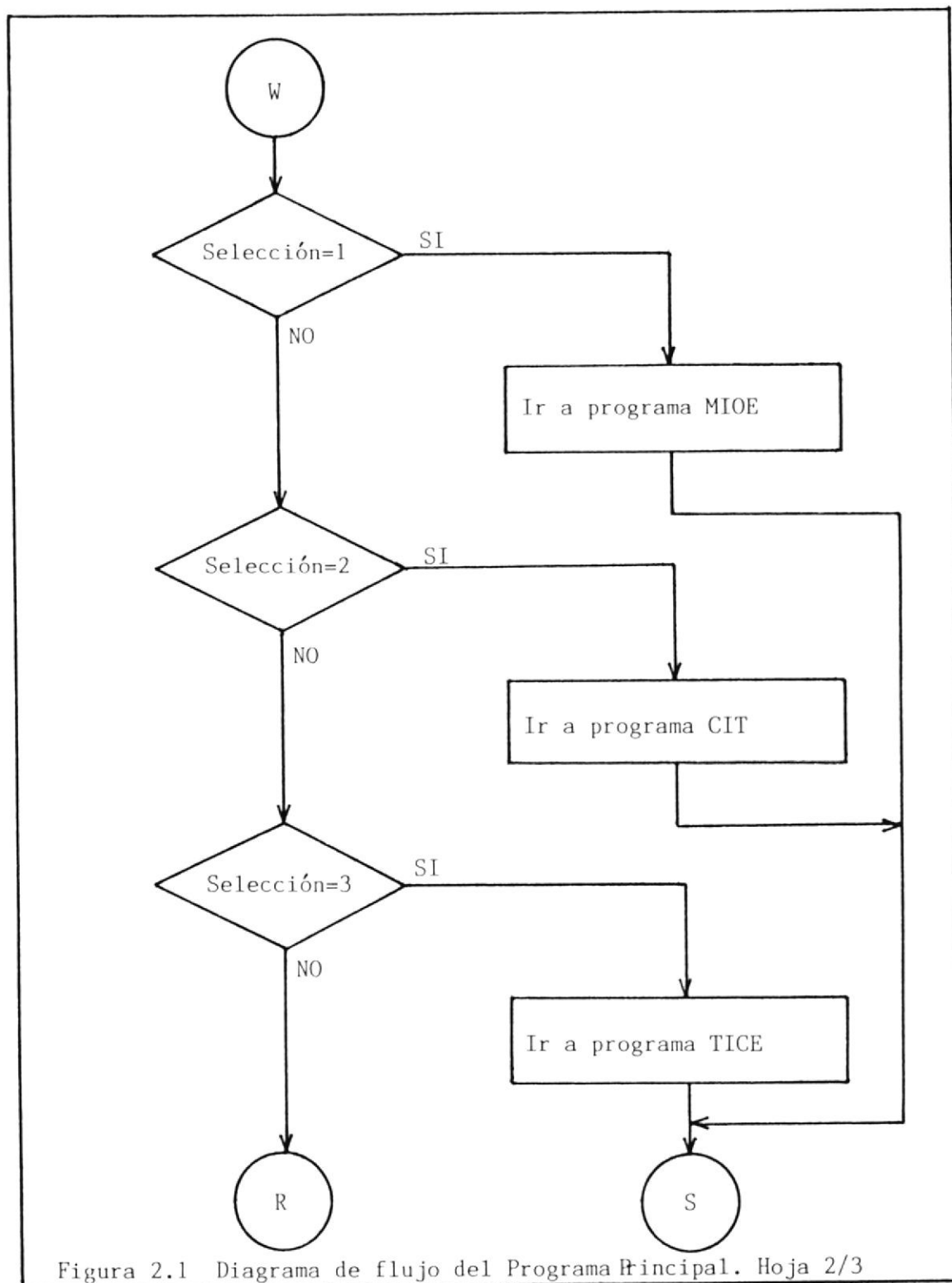


Figura 2.1 Diagrama de flujo del Programa Principal. Hoja 1/3



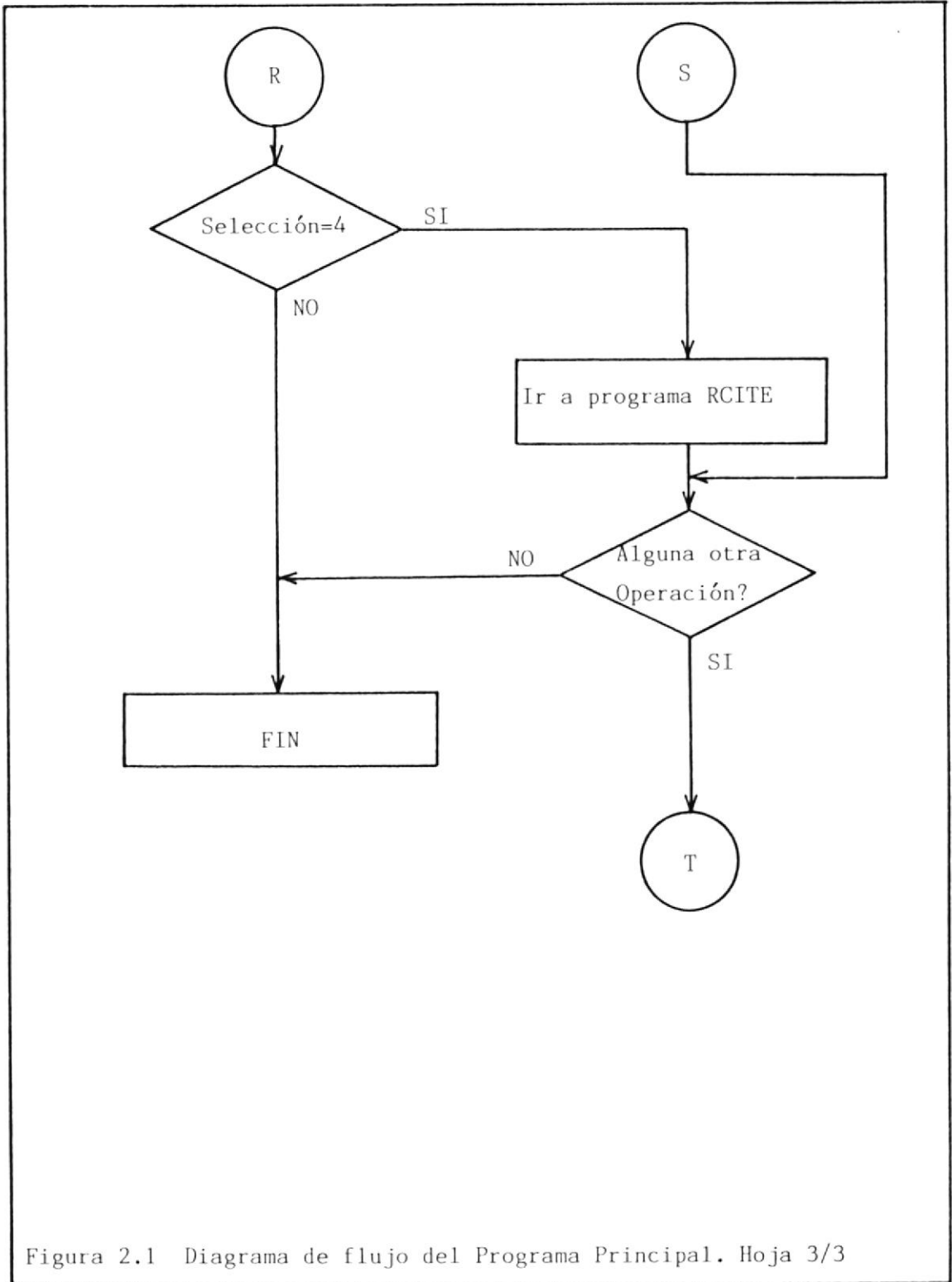


Figura 2.1 Diagrama de flujo del Programa Principal. Hoja 3/3

- 2.- Almacenar información (bytes) en el computador.
- 3.- Transmitir información (bytes) al Emulador.
- 4.- Realizar correcciones sobre la información.
- 5.- Salir

Después, de la realización de esta acción, se debe ingresar el código de la operación deseada y el programa la discriminará transfiriendo el control al programa seleccionado. Después, que la operación seleccionada es ejecutada en su totalidad, se regresará al programa principal para preguntar si se desea realizar alguna otra operación. Si la respuesta es negativa, entonces se terminará la sesión; de lo contrario, se regresará al inicio, es decir, a la presentación del menú de operación, para seleccionar otra operación y dar inicio así a una nueva sesión. A continuación se procederá a tratar con detalle el propósito de los diferentes programas que forman parte del programa principal.

#### 2.1.1 ALGORITMO DEL PROGRAMA MOSTRAR INDICACIONES DE OPERACION DEL EMULADOR.

Este programa denominado MIOE (Mostrar Indicaciones para Operación del Emulador) y cuyo diagrama de flujo se muestra en la figura 2.2, tiene como finalidad presentar por medio de la pantalla del computador las indicaciones necesarias para que el usuario pueda realizar los pasos y las conexiones

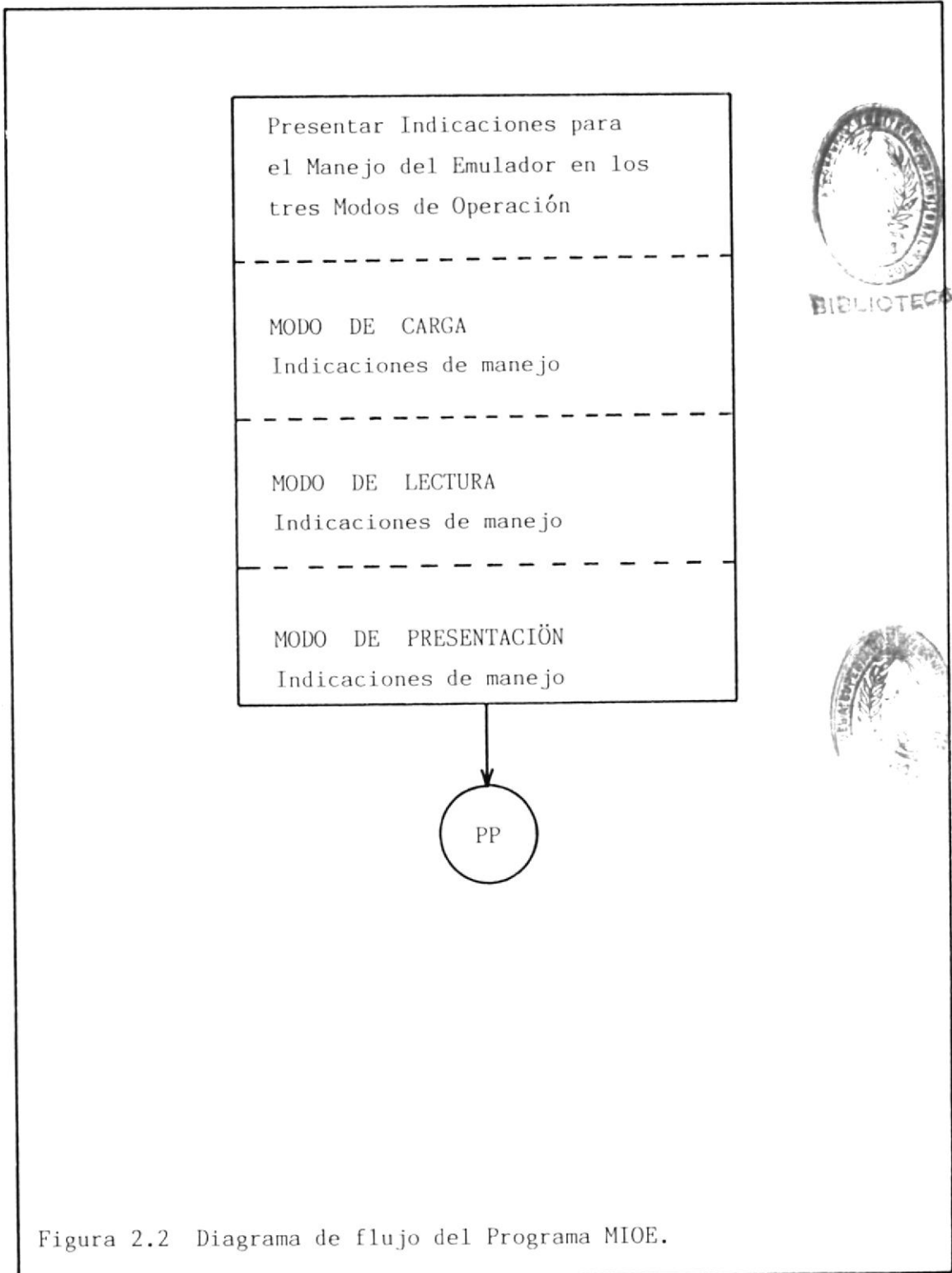


Figura 2.2 Diagrama de flujo del Programa MIOE.



necesarias para alistar el Emulador y este pueda trabajar adecuadamente en los tres modos de operación. Estos modos son los siguientes:

- Modo de Carga o Load Mode
- Modo de Lectura o Read Mode
- Modo de Presentación

Para el Modo de Carga se mostrará el procedimiento a seguir se para la interconexión del Emulador con el computador a través del puerto serie RS-232C. Luego, se indicará también la forma de ajustar el Emulador para satisfacer las exigencias del computador, es decir, igualar el protocolo y la velocidad de transmisión de datos.

Para el Modo de Lectura se mostrará de igual forma el procedimiento de conectar el Emulador con el circuito externo que hará uso de él, es decir, la forma de preparar el Emulador para que pueda ser leída la información (previamente almacenada) desde la memoria RAM, como si ésta estuviese en una EPROM.

El programa muestra también las indicaciones necesarias, para poder chequear el contenido de la memoria en el Modo de Presentación y verificar de esta forma que dicha información fue almacenada correctamente.

### 2.1.2 ALGORITMO DEL PROGRAMA CARGAR INFORMACION DESDE EL TECLADO

Este programa denominado CIT (Cargar Información desde el Teclado) tiene como finalidad el permitir el ingreso de información en formato hexadecimal a través del teclado y almacenarla para su posterior recuperación en un diskette. El diagrama de flujo se muestra en la figura 2.3.

El programa inicia solicitando el número de bytes a ser ingresados para ser asignado este valor a la variable numérica denominada CONTADOR. A continuación procederá a leer un byte de información a través del teclado para ser almacenado en el archivo previamente abierto denominado DATOS. Esta acción será ejecutada tantas veces como el número contenido en el CONTADOR. Luego, el programa procederá a almacenar el CONTADOR en el archivo NUDBY, y por consiguiente dar por terminada la sesión regresando al programa principal.

### 2.1.3 ALGORITMO DEL PROGRAMA TRANSMITIR INFORMACION DESDE EL COMPUTADOR AL EMULADOR

Este programa denominado TICE (Transmitir Información desde el Computador al Emulador), tiene como finalidad realizar la transmisión de la información previamente almacenada en

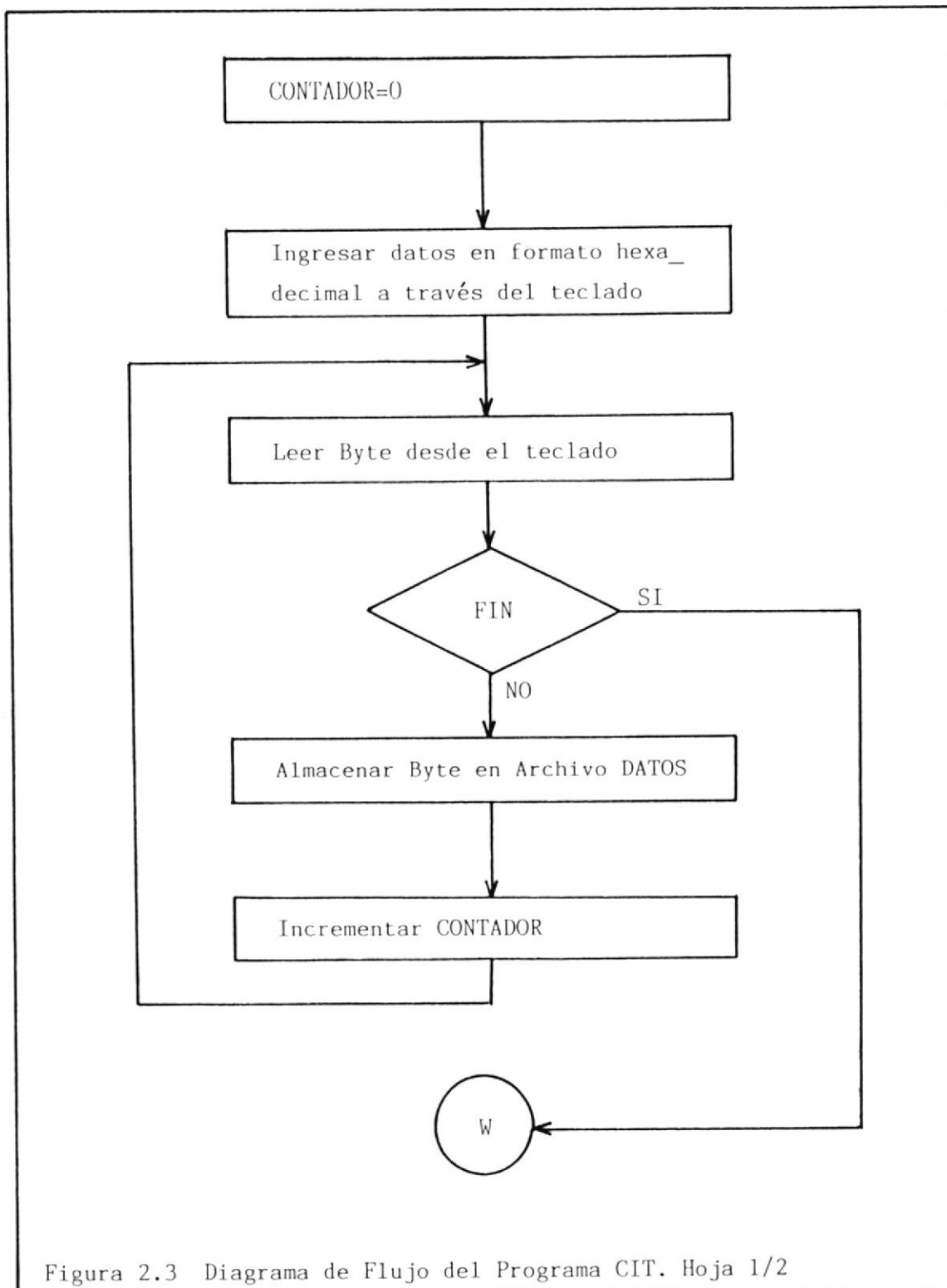


Figura 2.3 Diagrama de Flujo del Programa CIT. Hoja 1/2

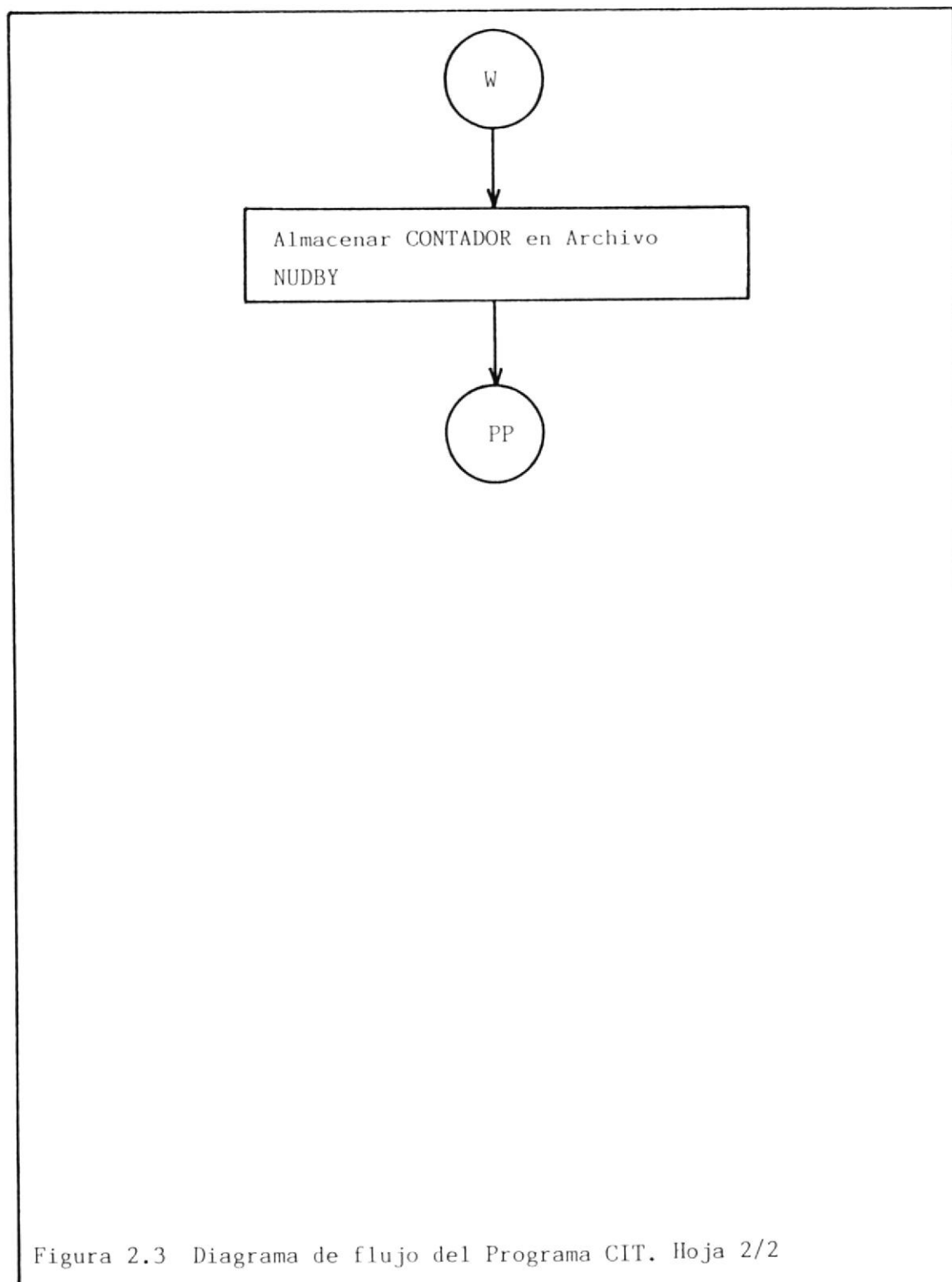


Figura 2.3 Diagrama de flujo del Programa CIT. Hoja 2/2

un archivo hacia el Emulador. Su diagrama de flujo se lo puede observar en la figura 2.4.

El programa empieza recuperando el número de bytes almacenados desde el archivo NUDBY para asignarlo al registro numérico CONTADOR. Luego, lee desde el archivo DATOS el número de bytes que le indica el CONTADOR y los transmite uno a uno a través del puerto serie RS-232C ubicado en el mismo computador hacia el Emulador.

Una vez que el archivo DATOS se ha vaciado y han sido transmitidos todos los bytes, entonces el programa transfiere el control al programa principal para preguntar si se va a continuar con alguna otra operación.

Además, el protocolo de transmisión del computador debe ser igualado en el Emulador antes del inicio de la transmisión.

#### 2.1.4 ALGORITMO DEL PROGRAMA REALIZAR CORRECCIONES EN LA INFORMACION TRANSMITIDA AL EMULADOR

Este programa denominado RCITE (Realizar Correcciones en la Información Transmitida al Emulador), tiene como finalidad el permitir la realización de determinadas modificaciones sobre la información que fue ingresada. Como se ilustra en

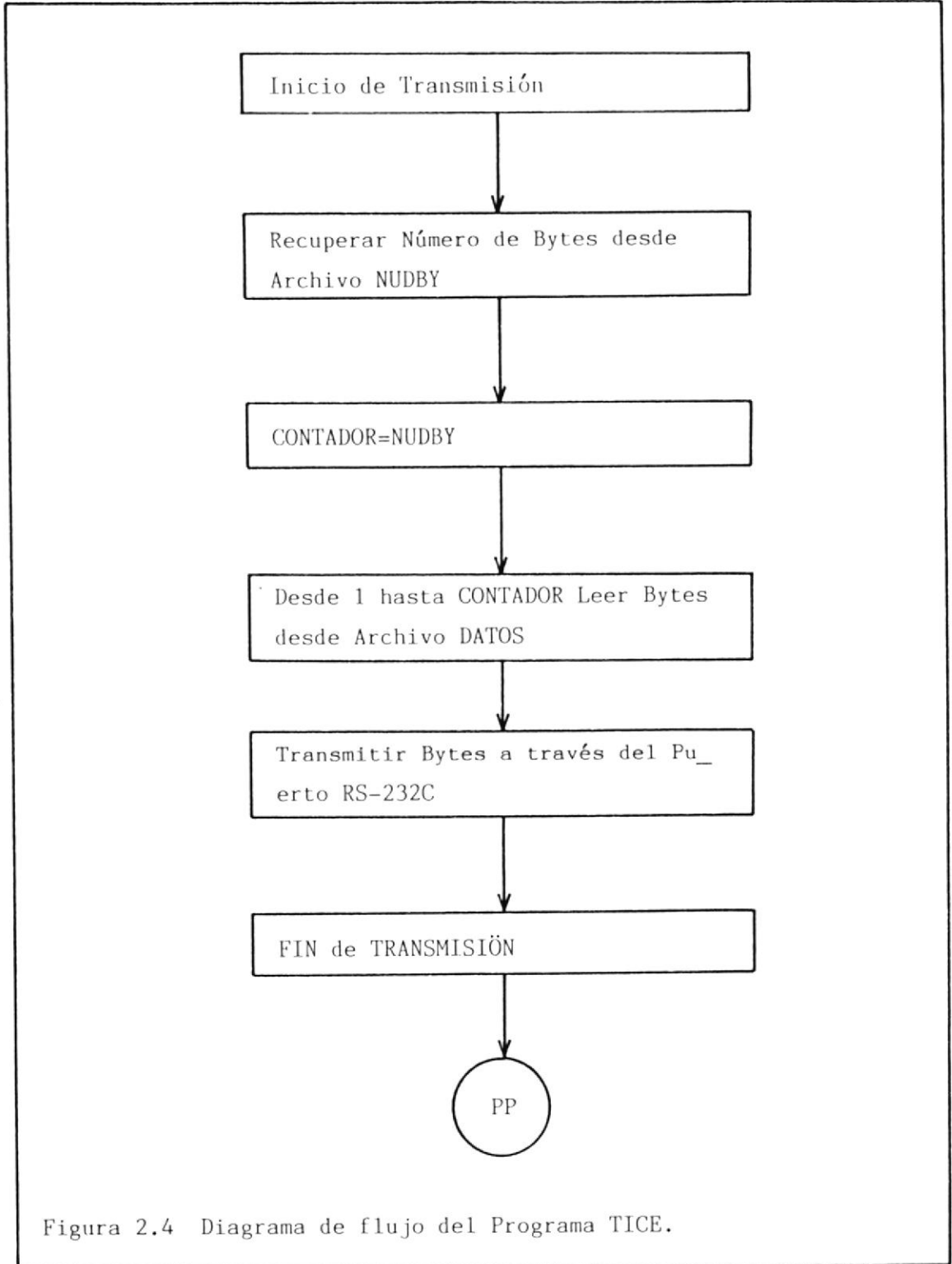


Figura 2.4 Diagrama de flujo del Programa TICE.

la figura 2.5 el programa inicia presentando a través de la pantalla un Menú donde constan los tipos de correcciones que pueden realizarse, y estas son:

- Introducir Bytes
- Borrar Bytes
- Cambiar Bytes

Una vez seleccionada la corrección deseada el programa transfiere el control al subprograma indicado, el cual realizará dicha tarea. Terminada la corrección sobre la información se regresa al programa RCITE para preguntar si se va a realizar alguna otra corrección, de ser afirmativa la respuesta se retorna al inicio, es decir, a presentar nuevamente el menú de correcciones. De lo contrario se regresará al programa principal.

Los sub-programas mencionados anteriormente serán considerados con detalle en las siguientes secciones.

#### 2.1.4.1 ALGORITMO DEL SUB-PROGRAMA INTRODUCIR BYTES

Este programa denominado IB (Introducir Bytes), tiene como objetivo el permitir la introducción de bytes en cualquier localidad dentro de una se

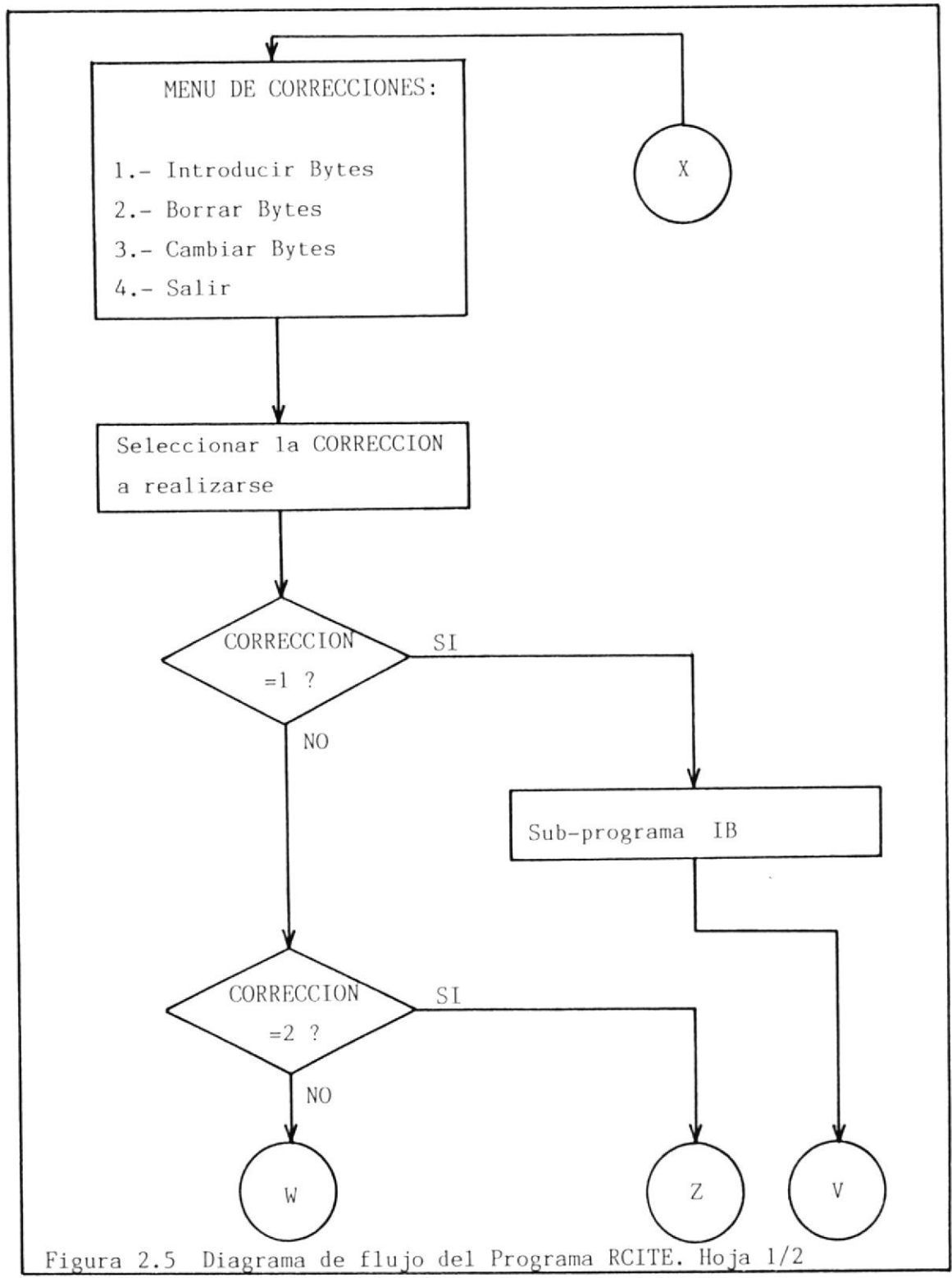
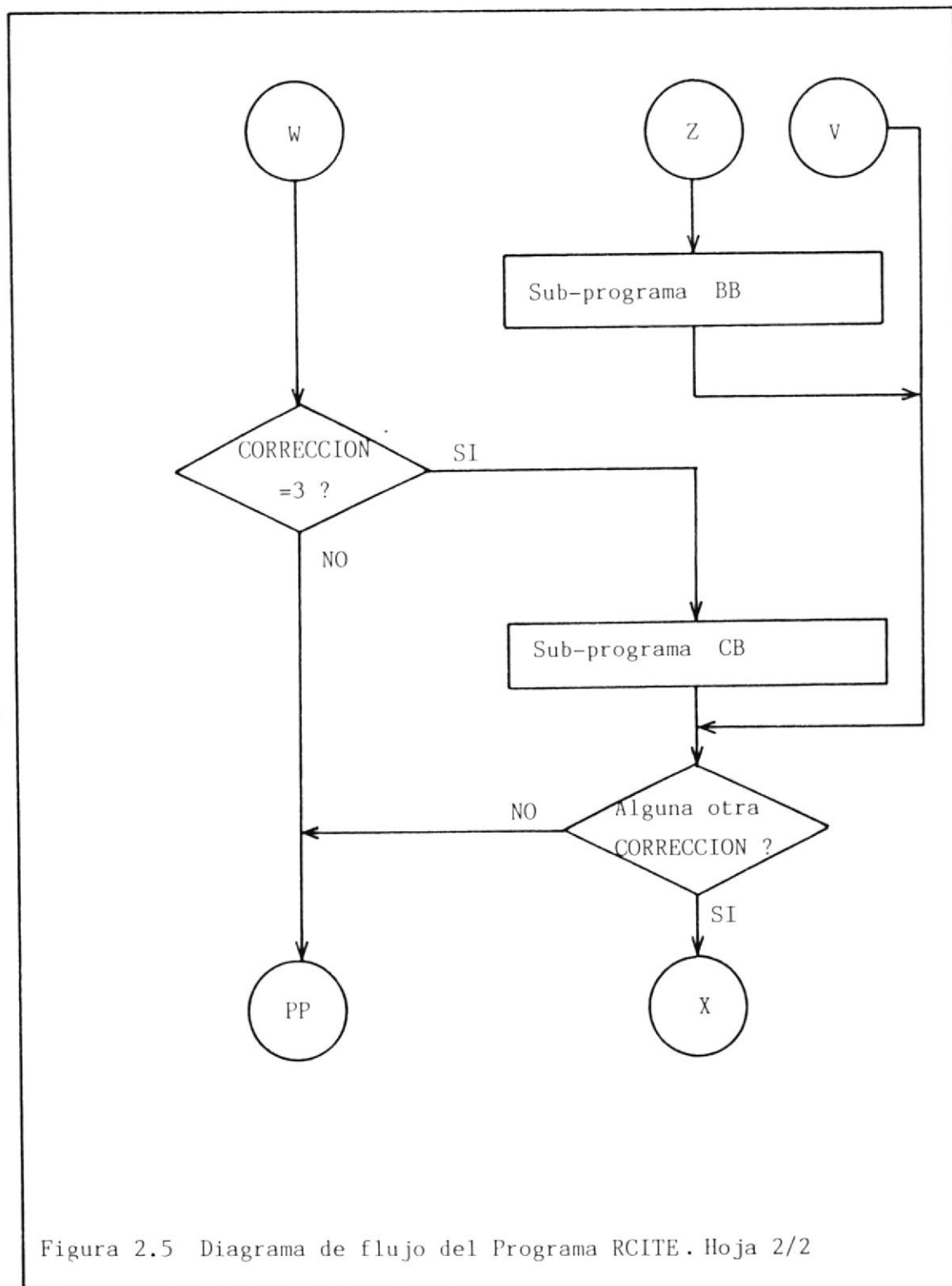


Figura 2.5 Diagrama de flujo del Programa RCITE. Hoja 1/2

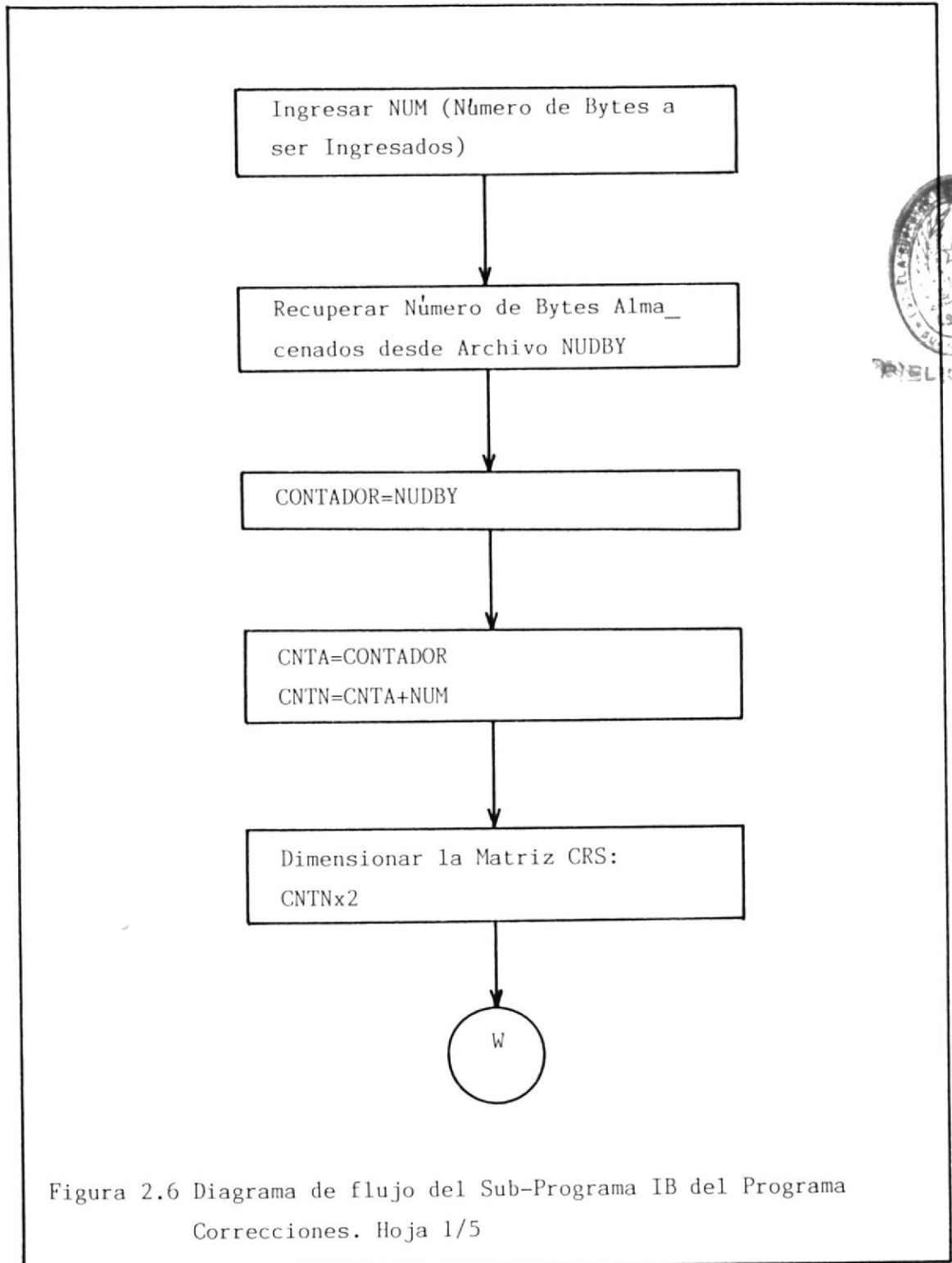




cuencia ya existente. En la figura 2.6 se puede observar el diagrama de flujo de este programa, el cual empieza leyendo a través del teclado el número de bytes que van a ser introducidos. Luego, recupera desde el archivo NUDBY el número de bytes almacenados y suma estas dos cantidades, para dimensionar una matriz de dos columnas y por un número de filas equivalente a la cantidad mencionada anteriormente. Esta matriz se la ha denominado CRS y es utilizada para almacenar en forma ordenada los bytes contenidos en el archivo DATOS.

Una vez dimensionada y cargada la matriz, el programa procede a leer desde el teclado un byte y la dirección M (decimal) donde va a ser escrito. Si la dirección ingresada es diferente a uno (01), entonces la matriz CRS1 que contiene la información original es copiada en la matriz CRS2 desde la dirección 01 hasta la dirección M-1. Luego, escribirá el byte ingresado en la matriz CRS2 en la posición indicada. En el caso que la dirección M sea igual a 01, entonces el programa pasará a realizar esta última acción directamente.

A continuación, se verifica si la dirección ingre



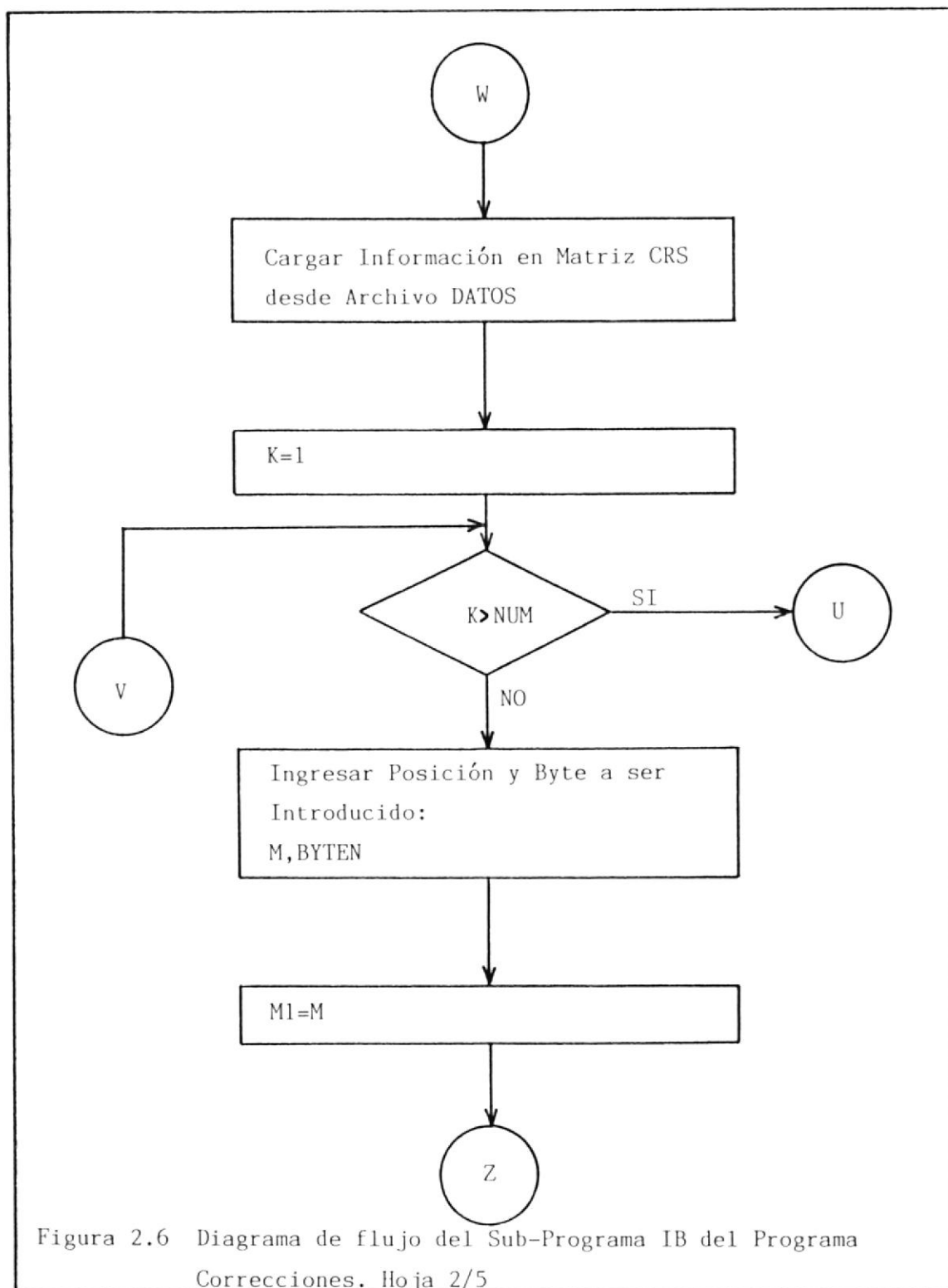
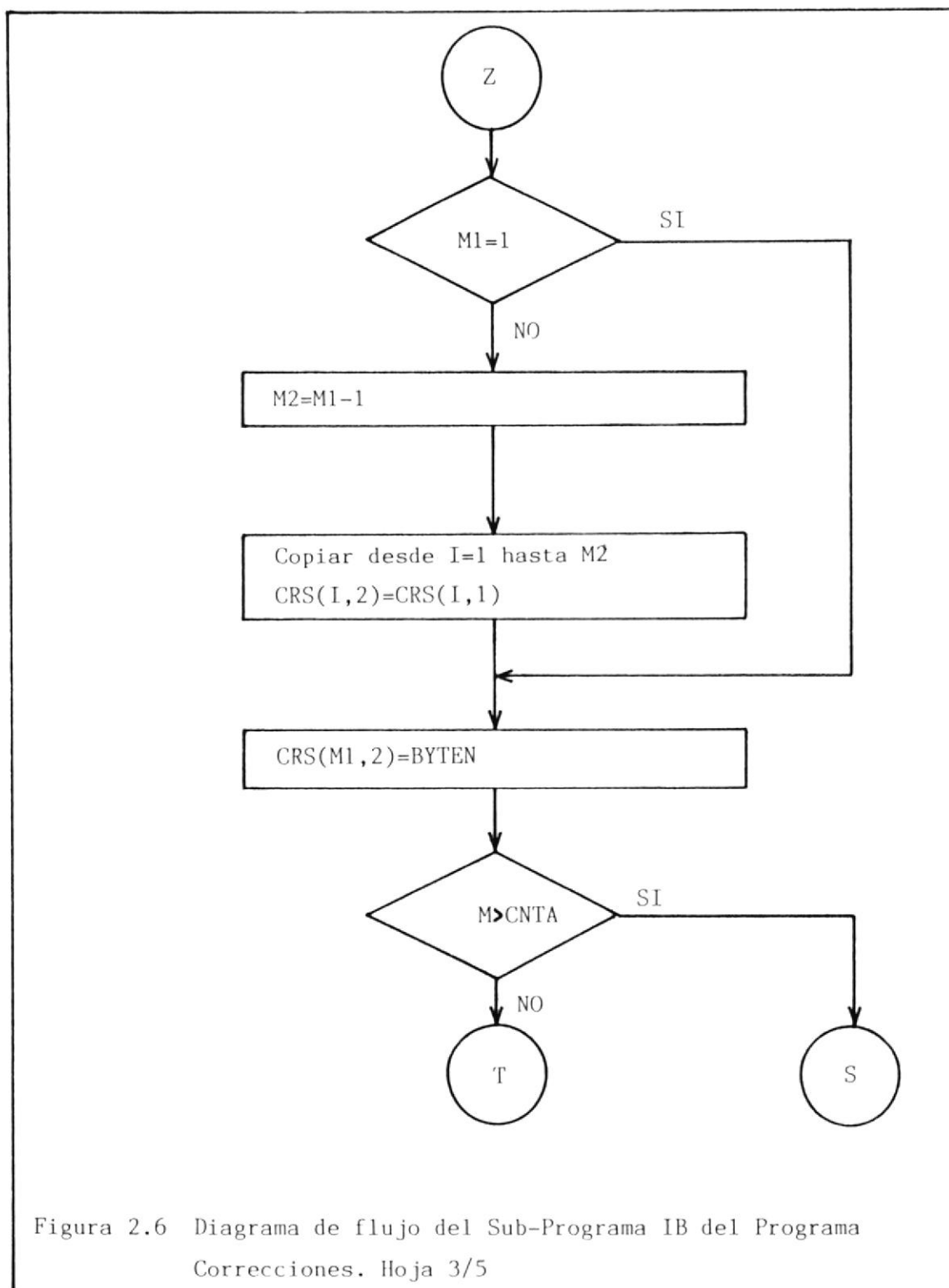
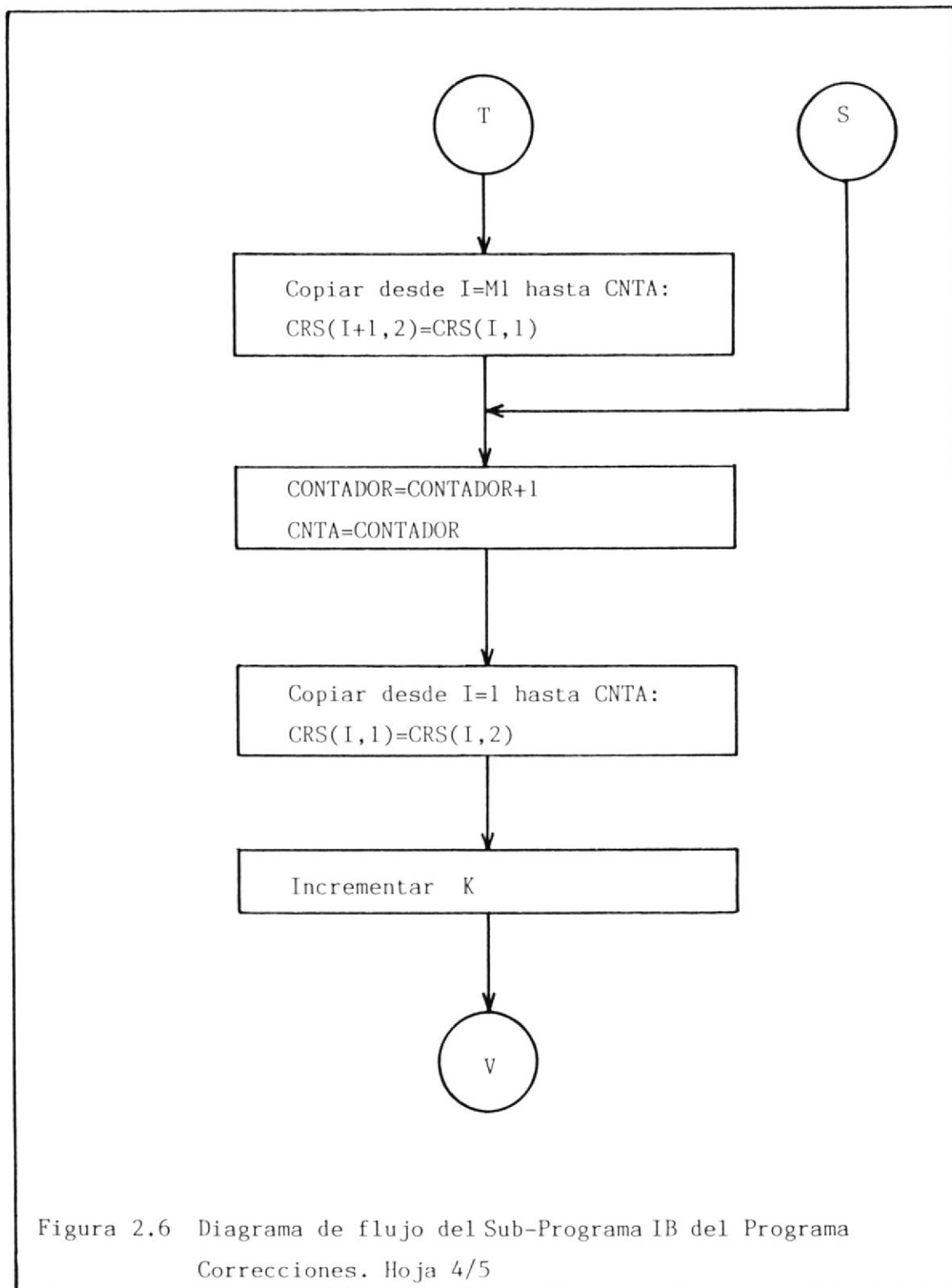


Figura 2.6 Diagrama de flujo del Sub-Programa IB del Programa Correcciones. Hoja 2/5





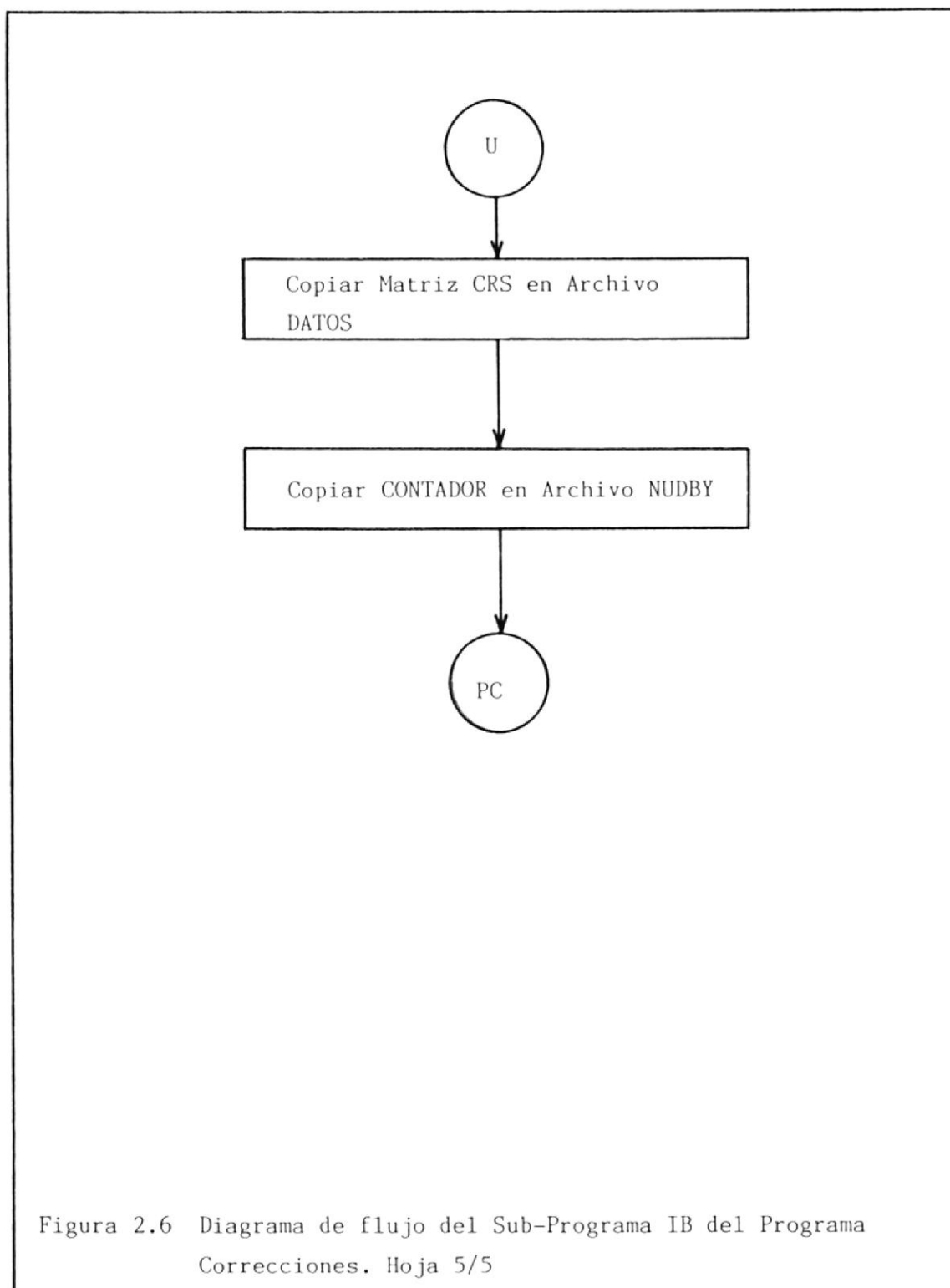


Figura 2.6 Diagrama de flujo del Sub-Programa IB del Programa Correcciones. Hoja 5/5

sada es mayor que la cantidad existente de bytes en la matriz CRS1; con el objeto de que si esta verificación es falsa, entonces se termina de copiar CRS1 en CRS2 a partir de la dirección  $M+1$ ; y en caso de ser verdadera se procede entonces a incrementar el registro CONTADOR y a actualizar el número de bytes almacenados en la matriz.

Después, se copia íntegramente la matriz CRS2 en CRS1 con el objeto de poder realizar esta gestión tantas veces como el número de bytes que fue ingresado para ser adicionados a los ya existentes.

Terminada la gestión, se realiza el almacenamiento de la matriz CRS y el registro CONTADOR actualizado en los archivos DATOS y NUDBY respectivamente. El paso siguiente es devolver el control al programa RCITE y terminar con la ejecución del sub-programa.

#### 2.1.4.2 ALGORITMO DEL SUB-PROGRAMA BORRAR BYTES

Este programa denominado BB (Borrar Bytes), tiene como finalidad el permitir el borrado de bytes en la secuencia previamente almacenada. El diagrama de flujo de este programa se muestra en la figura



## 2.7

El programa inicia recuperando el valor del número de bytes almacenados en el archivo NUDBY y cargándolo en el registro numérico denominado CONTADOR. Este dato es utilizado para dimensionar la matriz CRS, la cual será de dos columnas por un número de filas igual al valor del CONTADOR. A continuación se pasa a llenar esta matriz con los bytes almacenados en el archivo DATOS. Después, se procede a ingresar la dirección M a partir de la cual serán borrados los bytes. De acuerdo al valor de esta dirección se realizará lo siguiente:

Copiar CRS1 en CRS2 desde 01 hasta M-1 si M fuese igual a 01; ó preguntar si la suma de la dirección M más el número de bytes a ser borrados NUM es mayor que el número actual de bytes almacenados en CNTA. Para el caso de que esta suma sea falsa terminar de copiar CRS1 en CRS2 desde M+NUM hasta CNTA y si fuese verdadera pasar directamente a actualizar el contador restándole el valor NUM para luego igualar CNTA a esta cantidad. De esta forma se evita cualquier problema causado por la posición de los bytes que van a ser borrados.

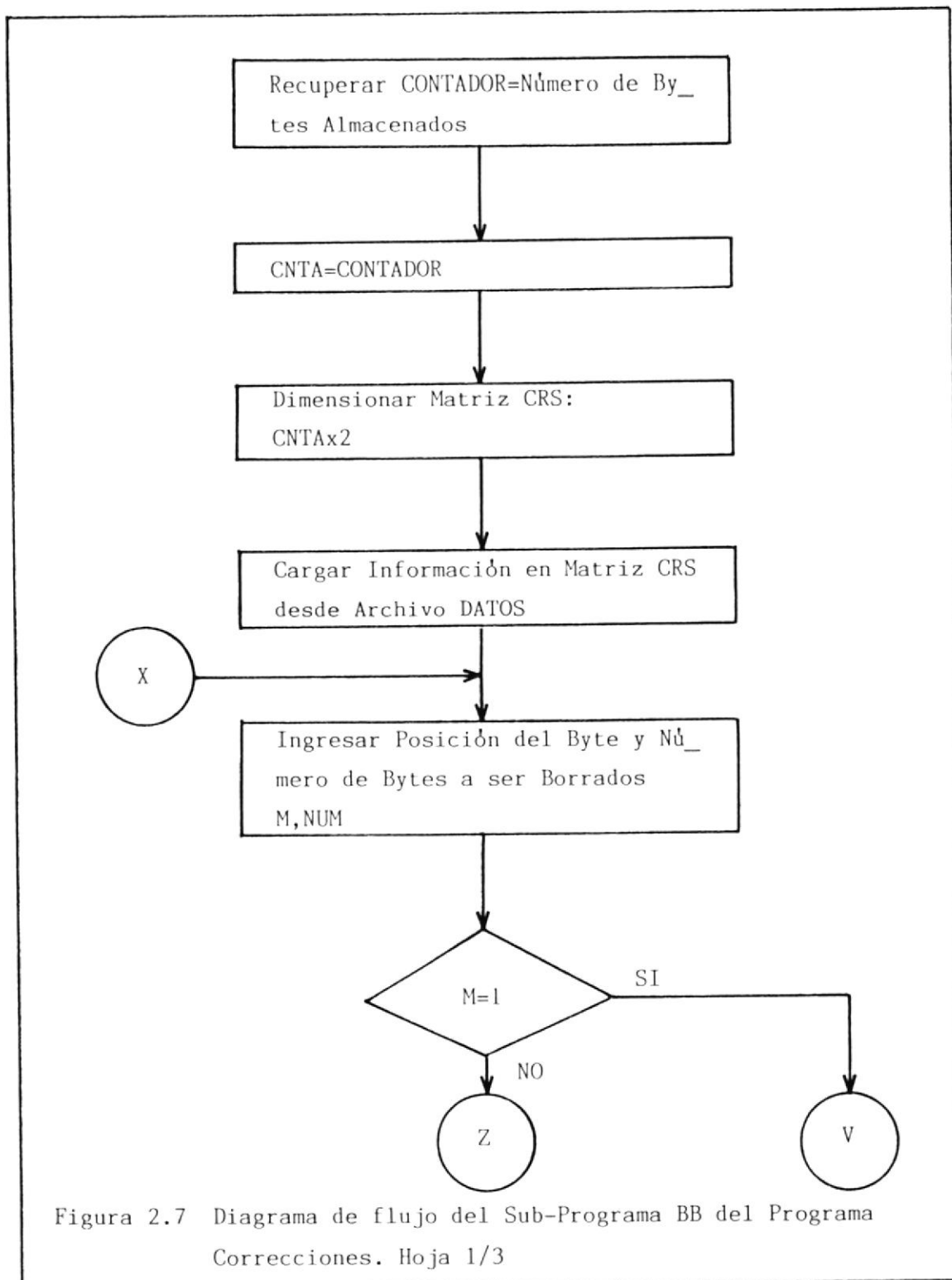


Figura 2.7 Diagrama de flujo del Sub-Programa BB del Programa Correcciones. Hoja 1/3

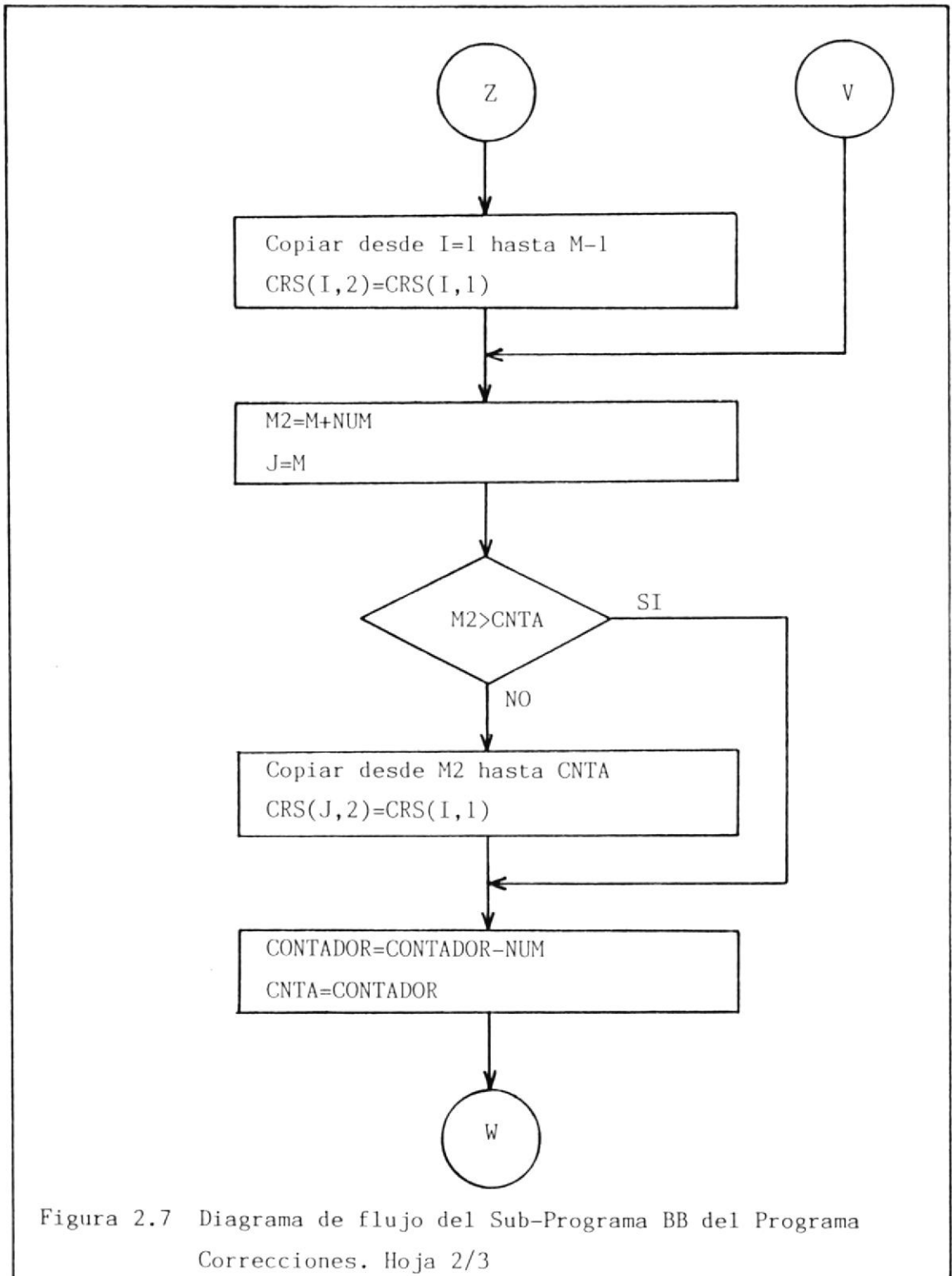


Figura 2.7 Diagrama de flujo del Sub-Programa BB del Programa Correcciones. Hoja 2/3

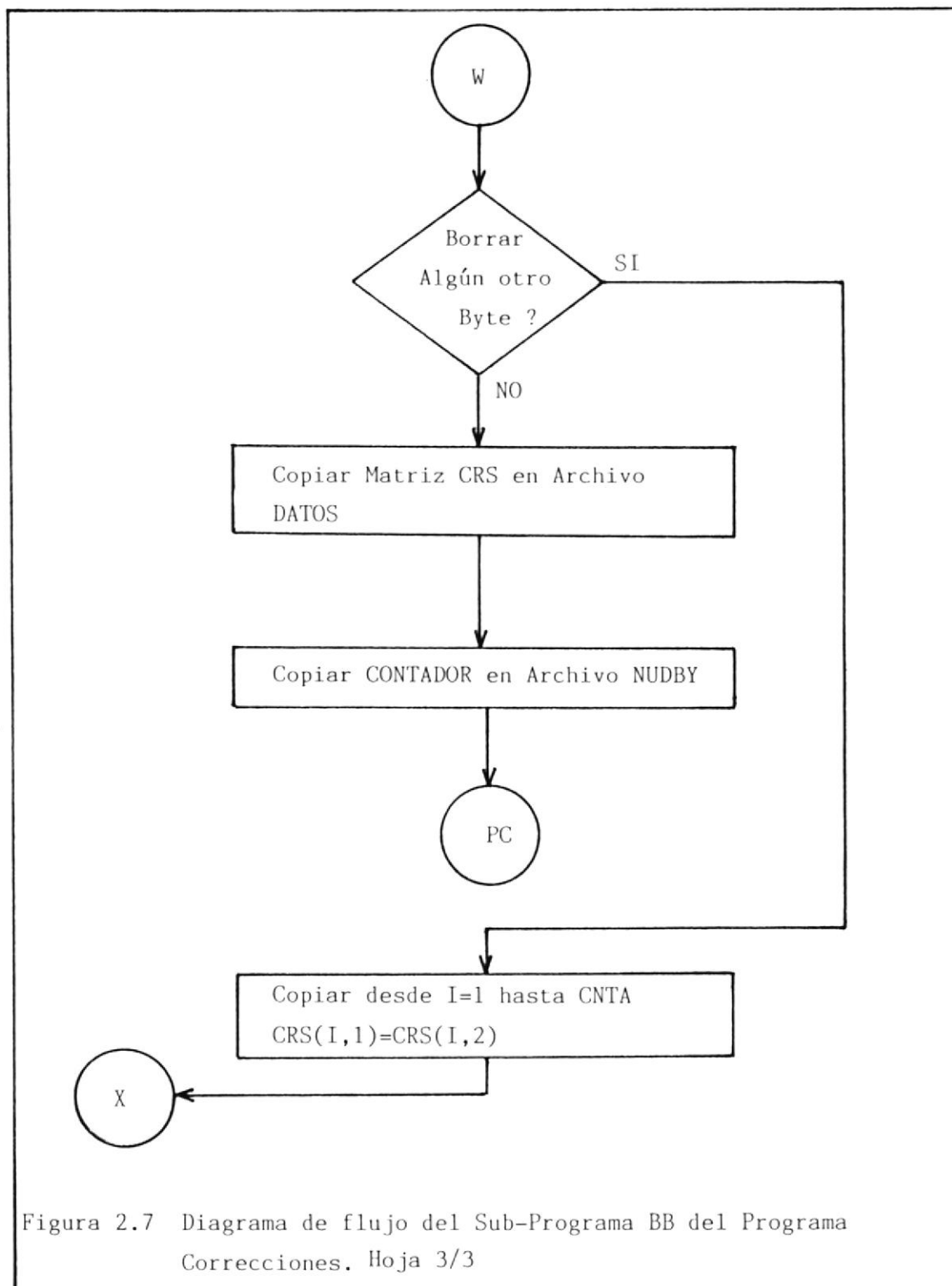


Figura 2.7 Diagrama de flujo del Sub-Programa BB del Programa Correcciones. Hoja 3/3

Después, de realizar los pasos anteriores se pregunta si se va a borrar algún otro byte, para que de acuerdo a la respuesta en caso de ser afirmativa pasar a copiar la matriz CRS2 en CRS1 (íntegramente) e ir a ejecutar el programa nuevamente a partir de la instrucción donde se ingresa la posición del byte y número de ellos que van a ser borrados. Si la respuesta es negativa, entonces esto significa que han sido borrados los bytes indicados y por lo tanto se almacenarán ordenadamente los bytes restantes en el archivo DATOS y el número de ellos en el archivo NUDBY, antes de regresar al programa correcciones.

#### 2.1.4.3 ALGORITMO DEL SUB-PROGRAMA CAMBIAR BYTES

Este programa denominado CB (Cambiar Bytes), tiene como finalidad permitir el cambio de bytes, es decir, borrar un byte y escribir otro en esa misma dirección. Su diagrama de flujo se muestra en la figura 2.8. Como se puede apreciar el programa inicia recuperando el número de bytes almacenados, luego asigna esta cantidad al registro CONTADOR. Paso seguido es dimensionar una matriz denominada CRS de una columna por un número de filas igual al

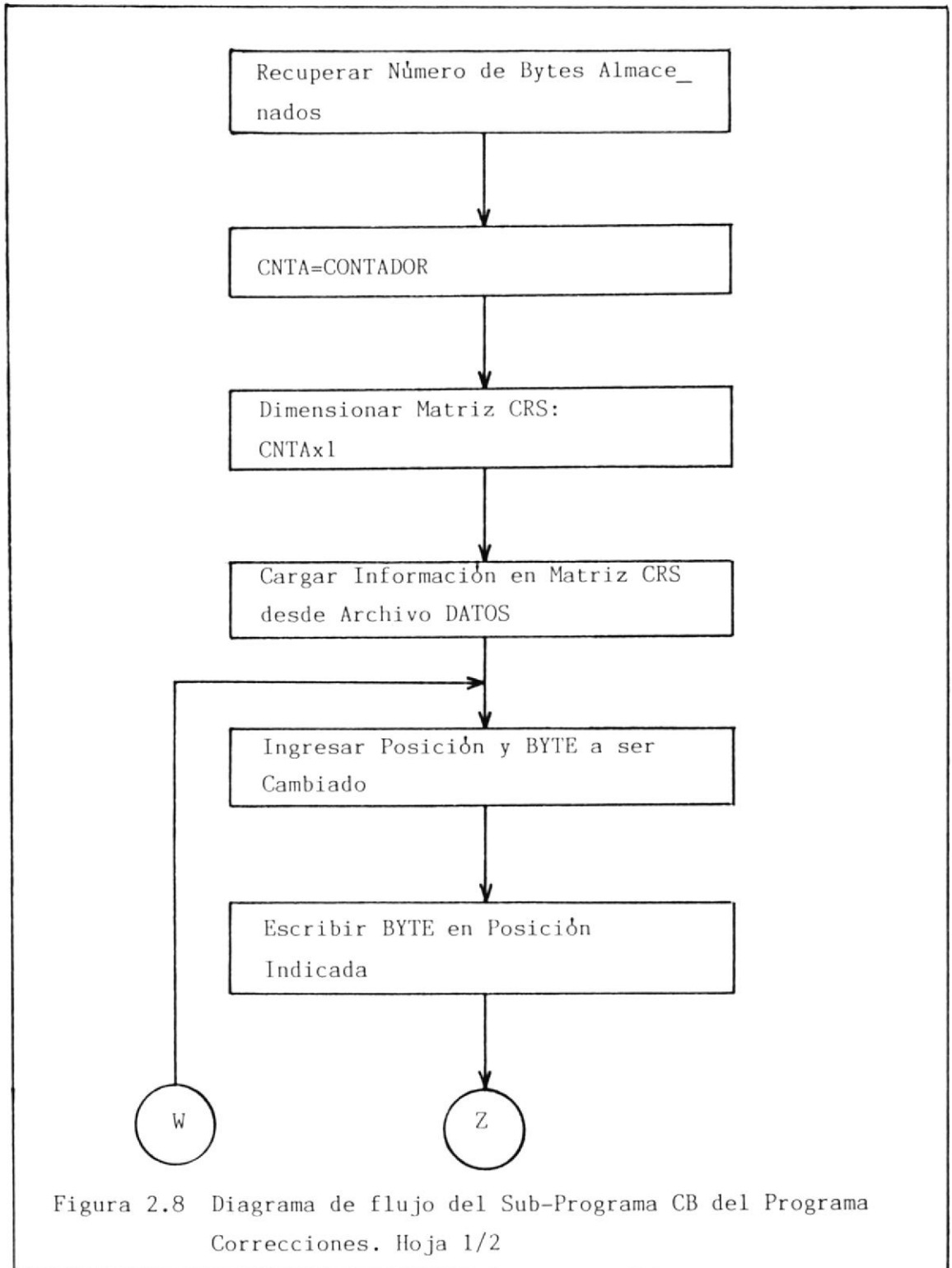


Figura 2.8 Diagrama de flujo del Sub-Programa CB del Programa Correcciones. Hoja 1/2

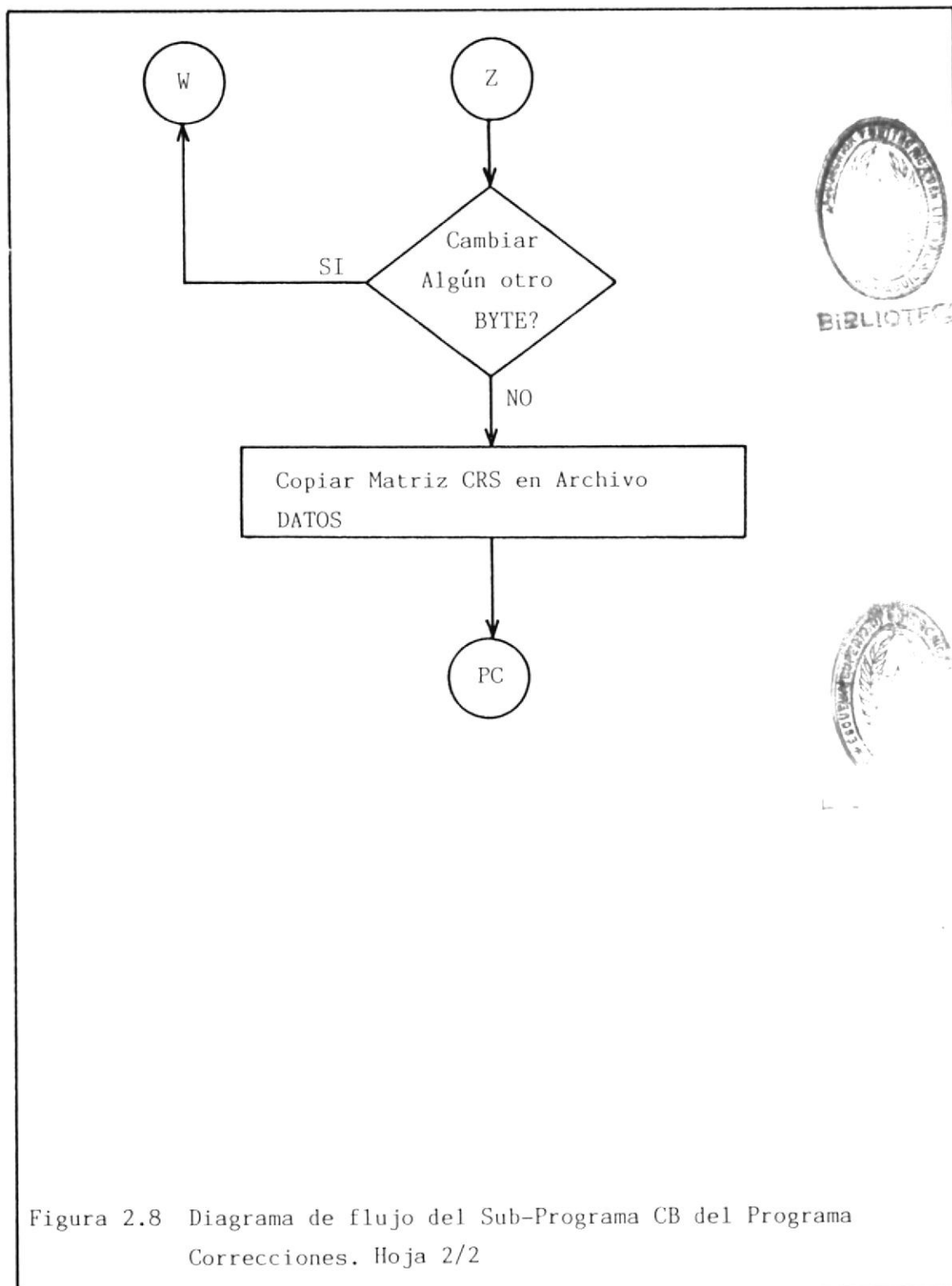


Figura 2.8 Diagrama de flujo del Sub-Programa CB del Programa Correcciones. Hoja 2/2

valor del CONTADOR.

A continuación se procede a llenar esta matriz con la información contenida en el archivo DATOS. Luego se ingresa a través del teclado un byte y la posición del byte que va a ser sustituido. Después, se procede a escribir el byte ingresado en la posición indicada y paso seguido preguntar si va a ser cambiado otro byte. De ser positiva la respuesta se retorna a ejecutar nuevamente el programa desde la instrucción donde se ingresa la posición y el nuevo byte. Pero, si la respuesta es negativa, entonces se procede a copiar la matriz CRS corregida en el archivo DATOS para finalmente regresar al programa CORRECCIONES.

## 2.2 PUERTO DE SALIDA SERIE RS-232C

En la presente sección se tratará en detalle las características y funciones de cada uno de los circuitos asociados con el protocolo de transmisión serie RS-232C. Este protocolo utiliza un conector tipo "D" de 25 pines con las denominadas siguientes: DB 25P para el macho y DB 25S para la hembra. La tabla IV agrupa los circuitos por su categoría, mostrando la nomenclatura de designación, así como la dirección del dato y la asignación de cada pin en el cable.



TABLA IV

CIRCUITOS RS-232C CLASIFICADOS POR SU CATEGORIA						
Circuito de Intercambio	Asignación del pin en el conector	DESCRIPCION	TIERRA	DATO	CONTROL	TIEMPO
				desde a DCE DCE	desde a DCE DCE	desde a DCE DCE
AA	1	Tierra de protección	X			
AB	7	Señal de Tierra o Común de Retorno	X			
BA	2	Dato Transmitido		X		
BB	3	Dato Recibido		X		
CA	4	Peticion de Envio			X	
CB	5	Despejado para Enviar			X	
CC	6	Data Set Ready			X	
CD	20	Data Terminal Ready				X
CE	22	Ring Indicator			X	
CF	8	Received Line Detector			X	
	21	Signal Quality Detector			X	
	23	Data Signal Rate Selector (DTE)				X
	23	Data Signal Rate Selector (DCE)			X	
DA	24	Transmitter Signal Element Timing (DTE)				X
DB	15	Transmitter Signal Element Timing (DCE)				X
DD	17	Receiver Signal Elem.				X

Circuito de Intercambio	Asignación del pin en el conector	DESCRIPCION	TIERRA	DATO	CONTROL	TIEMPO
				desde a DCE DCE	desde a DCE DCE	desde a DCE DCE
		Timing (DCE)				
SBA	14	Sec.Transm.Data		X		
SBB	16	Sec.Received Data		X		
SCA	19	Secondary RTS			X	
SCB	13	Secondary CTS			X	
SCF	12	Secondary Received Line Signal Detector			X	

Antes de describir la función de cada pin se definirá el significado de DTE y DCE.

DTE (DATA TERMINAL EQUIPMENT) es el equipo que puede abarcar la fuente de datos, el disipador (transmisor de datos), o ambos. También comprende equipos que cumplen cualquiera de las siguientes funciones: Control Lógico, Buffers de Almacenamiento, Dispositivos de Entrada/Salida o Computadores. El también puede contener Control de Error, Sincronización y Estaciones con capacidad de Identificación.

DCE (DATA COMUNICACION EQUIPMENT) es el equipo que provee las funciones necesarias para establecer, mantener y terminar una comunicación y realizar además la conversión y codificación de la señal requerida para la comunicación entre el terminal de datos y el circui-

to de datos. El DCE puede o no ser parte integral de un computador.

Los circuitos mencionados a continuación pueden ser divididos en cinco categorías:

- .- Referencia o Retorno Común ( A )
- .- Circuitos de Dato ( B )
- .- Circuitos de Control ( C )
- .- Circuitos de Tiempo ( D )
- .- Circuitos de Canal Secundario ( S )

A continuación se procederá a la descripción detallada de cada uno de los circuitos utilizados en el protocolo RS-232C.

TIERRA DE PROTECCION (PROTECTIVE GROUND) O CIRCUITO AA: Este conductor está electricamente soldado a la estructura del equipo, y su conexión es opcional puesto que él no representa el común o la referencia de la señal, por lo tanto es bastante común dejarlo desconectado, lo cual no significa que el cable no cumpla con el protocolo RS-232C.

SEÑAL DE TIERRA O COMUN DE RETORNO (SIGNAL GROUND O COMMON RETURN) O CIRCUITO AB: Este conductor provee el punto de referencia con respecto al cual los otros circuitos RS-232C (excepto la Tierra

de Protección), son medidos. Este circuito es absolutamente requerido, y no importa que aplicación vaya a tener el cable.

**DATO TRANSMITIDO (TRANSMITTED DATA) O CIRCUITO BA:** La información sobre este circuito es transmitida desde el DTE hacia el DCE, donde el DTE mantiene el circuito BA en un "uno" lógico (condición de marca o partida) todo el tiempo en que ningún dato es transmitido.

En todos los sistemas que se rigen bajo el protocolo o patrón RS-232C el DTE no puede transmitir a menos que un "cero" lógico (condición de ON) esté en cada uno de los siguientes circuitos:

- 1.- Petición de Envío (Request to Send)
- 2.- Despejado para Enviar (Clear to Send)
- 3.- Set de Dato Listo (Data Set Ready)
- 4.- Terminal de Dato Listo (Data Terminal Ready)

**DATO RECIBIDO (RECEIVED DATA) O CIRCUITO BB:** Las señales sobre este circuito son transmitidas desde el DCE hacia el DTE. El circuito es mantenido en un uno lógico durante intervalos entre la transmisión de dato y el tiempo en que ningún dato está siendo transmitido.

En un canal Half-Duplex (Aquel canal de comunicación que puede o perar en ambas direcciones pero no simultáneamente, es decir, la dirección de transmisión es reversible. Ambas direcciones sopor- tan la misma variación de velocidad de transmisión), el conductor Dato Recibido (circuito BB) es mantenido en una condición de OFF cuando Petición de Envío (Request to Send) está en la condición de ON. También, él es mantenido en OFF por un breve intervalo des- pués de una transición de ON a OFF de la línea Petición de Envío (RTS) para permitir la conclusión de la transmisión.

PETICION DE ENVIO (REQUEST TO SEND) O CIRCUITO CA: Este circui- to transporta una petición para transmitir desde el DTE hacia el DCE. En las figuras 2.9 y 2.10 se ilustra cómo esta señal traba- ja en conjunto con otra señal denominada Despejado para Enviar (CTS o circuito CB) para coordinar la transmisión entre el DTE y el DCE. Para canales Simplex y Full-Duplex (Simplex es un canal capaz de operar en una sola dirección y Full-Duplex o Duplex es un canal de comunicación el cual es capaz de operar en ambas direccio- nes simultáneamente), un "cero" lógico sobre la línea RTS mantiene el DCE en el modo de transmisión. Esto significa que un DCE en mo do de transmisión aceptará dato desde el DTE y lo pasará al lazo de comunicación al cual está unido. Inversamente, para canales de comunicación Simplex y Half-Duplex, si la señal RTS es "uno" ló gico (eso es una condición de OFF), el DCE local es mantenido en un modo no transmisión (esto significa que el DCE no pasará el da-

to que ha recibido desde el DTE al circuito de comunicación).

Para canales de comunicación Half-Duplex, la condición de ON mantiene al DCE en el modo de transmisión, y la condición de OFF lo mantendrá en el modo de recepción (Modo de Recepción significa que el DCE aceptará dato desde el circuito de comunicación y lo pasará al DTE respectivo).

En la figura 2.9 una transición de RTS de la condición de OFF a ON lleva al DCE local a entrar en el modo de transmisión y realizar las acciones necesarias para establecer la comunicación.

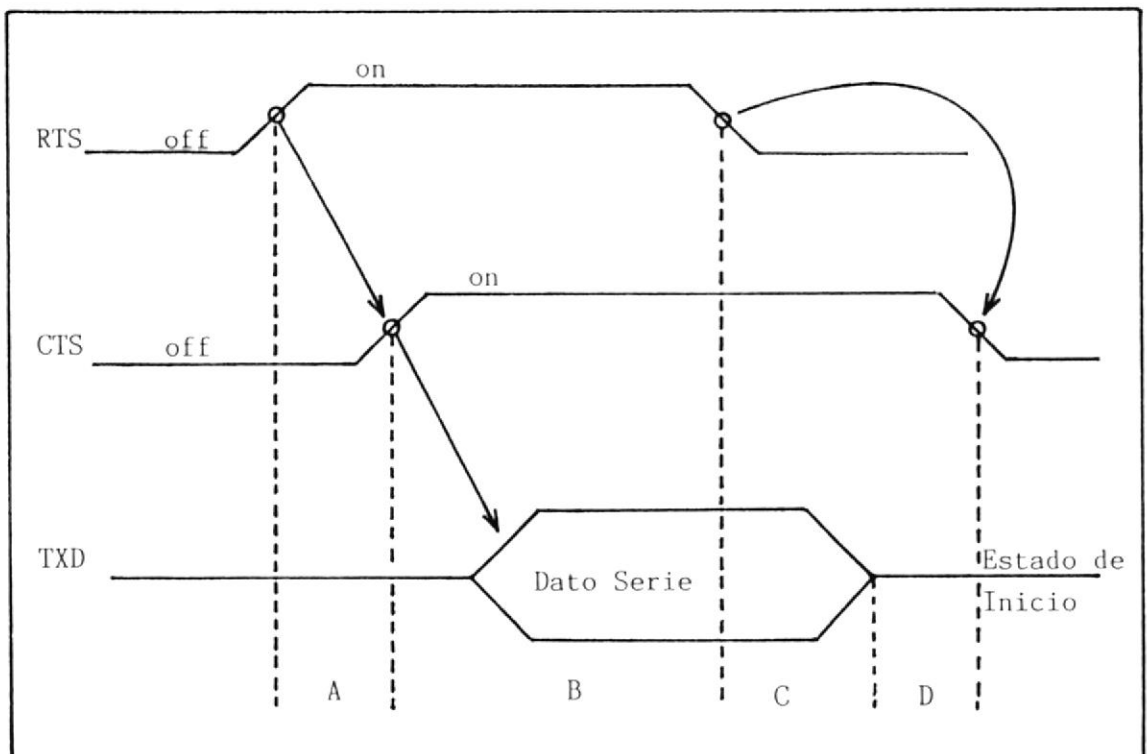


Figura 2.9 Hanshake entre el DTE y el DCE. A) El DTE hace verdadero RTS indicando que él desea transmitir dato; el DCE es activado para establecer la comunicación; el DCE hace verdadero CTS, indicando que el DTE puede empezar la transmisión; TXD está en condición de marca, B). La transmisión de dato ocurre sobre el circuito Dato Transmitido; el DTE desactiva RTS, indicando que él ya no desea transmitir, C). El DCE completa la transmisión del último dato; TXD retorna a la condición de marca, D). El DCE queda libre para responder al próximo cambio de RTS.

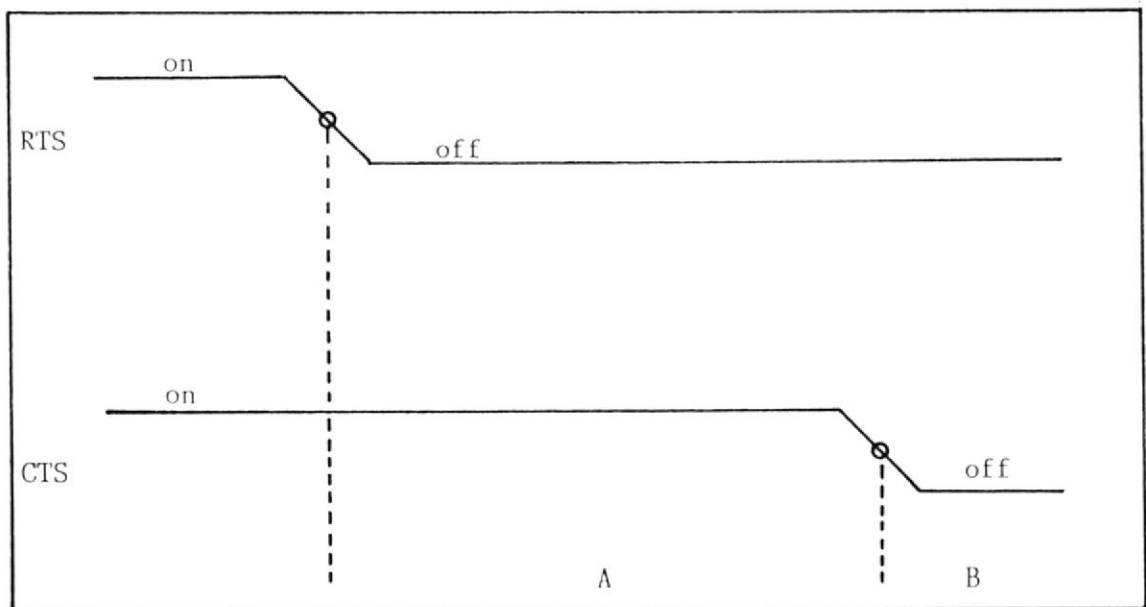


Figura 2.10 Tiempo cuando RTS puede ser verdadero: A) Durante este intervalo el DTE no puede revalidar RTS. B) Aquí el CTS pasa al estado de off, y el DTE está libre para revalidar RTS.

Una vez completado el proceso anterior, el DCE lleva la línea CTS (Circuito CB) a la condición de ON, indicando de esta forma al DTE que el dato puede ser transferido a través del punto de interface sobre el circuito Dato Transmitido. Hay que notar que una transición de la línea RTS desde la condición de ON a OFF indica al DCE la finalización de la transmisión puesto que todos los datos han cruzado el punto de interface (Circuito Dato Transmitido) y entonces asume el modo de no-transmisión (Full-Duplex o Simplex) o modo de recepción (Half-Duplex). El DCE local responde a esta señal llevando a un estado de OFF la línea CTS cuando él está preparado para responder a una subsecuente condición de ON del circuito RTS.

Como se observa en la figura 2.10, una vez que RTS ha ido a la condicción de OFF, él no puede ser llevado a ON otra vez hasta que CTS haya sido llevado a OFF por el DCE local.

**DESPEJADO PARA ENVIAR (CLEAR TO SEND) O CIRCUITO CB:** Esta es una señal de control que es transmitida desde el DCE hacia el DTE para indicar que el DCE está listo para recibir dato desde el DTE sobre el circuito Dato Transmitido. Cuando esta señal está en ON y las señales RTS (Request to Send), DSR (Data Set Ready) y DTR (Data Terminal Ready) están todas en ON también, esto constituye una indicación al DTE que el dato transmitido por medio de él será comunicado a través del DCE a los canales de comunicación respectivos.



Cuando CTS está en la condición de OFF, esto indica que el DCE no está listo de tal forma que el DTE no deberá intentar transmitir dato. Por lo tanto, CTS es activado como respuesta a simultáneas condiciones de ON de RTS y DSR.

Los siguientes circuitos son todos requeridos en el protocolo RS 232C para que se establezca la comunicación entre un DTE y un DCE.

DATA SET READY (Circuito CC)

DATA TERMINAL READY (Circuito CD)

RING INDICATOR (Circuito CE)

RECEIVED LINE SIGNAL DETECTOR (Circuito CF)

DATA SET READY O CIRCUITO CC: La dirección de esta señal de control es desde el DCE hacia el DTE. El indica el estado del set de datos local. Si la señal DATA SET READY está en la condición de ON, entonces esto significa que el DCE está conectado al canal de comunicación.

DATA TERMINAL READY O CIRCUITO CD: La dirección de esta señal de control es desde el DTE hacia el DCE. El DTR debe estar en ON antes que el DCE pueda llevar a ON Data Set Ready, indicando que él ha sido conectado al canal de comunicación. Una vez que el DCE está conectado y el dato está siendo transmitido, una transición desde ON a OFF causa al DCE ser eliminado del canal de comu-

nicación.

Esencialmente DTR y DSR implementan una versión más estática que el protocolo CTS-RTS. Por más estática se debe entender que el protocolo antes mencionado indica disponibilidad de canal, mientras que el protocolo DTR-DSR indica disponibilidad de equipos.

**RING INDICATOR O CIRCUITO CE:** La dirección de esta señal de control es desde el DCE hacia el DTE. Cuando ésta señal está en la condición de ON, aquello indica que el DCE está recibiendo una señal de timbrado. Esta señal es mantenida en la condición de OFF entre timbrados y en todo tiempo que el DCE no está recibiendo una señal de timbrado.

**RECEIVED LINE SIGNAL DETECTOR O CIRCUITO CF:** El DCE envía una condición de ON al DTE sobre este circuito cuando él está recibiendo una señal portadora (CARRIER) que reúne los criterios apropiados desde el DCE remoto. Estos criterios son determinados por el fabricante del DCE. Esta señal también es denominada DATA CARRIER DETECT (DCD), o simplemente CARRIER.

Los circuitos restantes que no han sido tratados en esta sección no tienen mayor significado y son dejados desconectados en la mayoría de las configuraciones con microcomputadores.

### 2.2.1 CONFIGURACIONES COMUNMENTE UTILIZADAS

La EIA (Electronic Industries Association) ha desarrollado un cable patrón para diferentes situaciones. Siete de sus configuraciones de muestra son aplicables a sistemas computarizados, y ellos son:

- .- Sólo Transmisión (Transmit Only)
- .- Sólo Transmisión con RTS (Transmit Only with RTS)
- .- Sólo Recepción (Received Only)
- .- Half Duplex
- .- Full Duplex
- .- Full Duplex con RTS
- .- Casos Especiales

La tabla V muestra cada uno de los tipos de interface mencionados anteriormente y las señales RS-232C que deben estar presentes en un cable patrón. Aquí han sido omitidos los circuitos de tiempo los cuales serían requeridos para transmisión síncrona, la que raramente ocurre en sistemas computarizados.

Como se puede observar en la tabla, hay un solo circuito el cual es "absolutamente" indispensable en cualquier cable RS-232C y este es el denominado Señal de Tierra y se



BIBLIOTECA

localizan en el pin 7. Todas las otras señales pueden o no estar presentes, dependiendo de la aplicación que vaya a tener el cable.

Mientras todos los cables patrón mostrados en la tabla V son aplicables a sistemas computarizados, la gran mayoría de estos microsistemas utilizan alguna variante del cable full duplex RS 232C. Muchos sistemas microcomputarizados incorporan dispositivos de una vía como los impresores que sólo reciben, joysticks que sólo transmiten, y dispositivos similares.

Pero la mayor parte del tiempo, los cables RS-232C usados son configurados para manejar la situación más común denominada Full-Duplex, o Comunicación de Dos Vías. De esta forma, el cable normalmente usado para un impresor que sólo recibe puede ser aplicable a diferentes dispositivos de acuerdo a la necesidad, sin ningún efecto negativo.

La aplicación clásica de comunicación full-duplex en sistemas de microcomputadoras es para terminales de comunicación de transmisión/recepción donde caracteres son transmitidos desde un teclado a un computador y subsecuentemente repetidos otra vez al monitor o el impresor. En esta situación el dato está simultáneamente viajando en ambas direcciones

TABLA V

Configuraciones RS-232C Standard							
Circuito de Intercambio RS-232C	Solo Transmisión	Solo Transmisión con RTS	Solo Recepción con RTS	Half Duplex	Full Duplex	Full Duplex con RTS	Especial
1 Tierra de protección	-	-	-	-	-	-	o
7 Señal de Tierra	X	X	X	X	X	X	X
2 Dato Transmitido	X	X		X	X	X	o
3 Dato Recibido			X	X	X	X	o
4 Petición de Envío		X		X		X	
5 Despejado para Enviar	X	X		X	X	X	o
6 Data Set Ready	X	X	X	X	X	X	o
20 Data Terminal Ready	S	S	S	S	S	S	o
22 Ring Indicator	S	S	S	S	S	S	o
8 Received Line Signal Detector			X	X	X	X	o

- X - Significa requerido para cualquier configuración
- S - Significa requerido para uso en circuitos telefónicos públicos
- o - Significa que será especificado por el diseñador del cable

entre el DTE (Teclado y Monitor) y el DCE (Puerto Serie E/S del computador).

### CAPITULO III

#### PROCEDIMIENTO TECNICO EN LA CONSTRUCCION DEL CIRCUITO EMULADOR Y RESULTA DOS OBTENIDOS

##### 3.1 LISTADO DE ELEMENTOS EMPLEADOS

Los elementos utilizados en la construcción del circuito del Emula-  
dor son los siguientes:

01	Resistencia	15	Mohn,1/4 w
01	Resistencia	33	Kohn,1/4 w
21	Resistencias	10	Kohn,1/4 w
01	Resistencia	6.8	Kohn,1/4 w
02	Resistencias	1	Kohn,1/4 w
01	Resistencia	470	ohn,1/4 w
01	Resistencia	20	ohn,1/4 w
02	Diodos Emisores de Luz		

01 Diodo (1N914)

01 Transistor (2N2222)

01 Puente Rectificador (B80C800)

01 Cristal (1.8432 Mhz)

02 Display 7 Segmentos (IEE1737R)

01 Conector DB-25S

01 Transformador 120V60Hz/9VAC1A

01 Capacitor Tantalio 0.1 uF/35V

01 Capacitor Mylar 0.1 uF/50V

01 Capacitor Electrolítico 1000 uF/35V

01 Portachip 40 patas w/w

02 Portachips 28 patas w/w

01 Portachip 24 patas w/w

07 Portachips 20 patas w/w

01 Portachip 18 patas w/w

04 Portachips 16 patas w/w

05 Portachips 14 patas w/w

01 Porta Fusible

02 Switches SPDT (Momentáneo)

01 Switch SP3T (Slide)



10 Switches SPDT (Toggle)

01 CMOS UART IM6402IPL

01 CMOS RAM 8Kx8 Bits HM6264LP-12

01 BIT RATE GENERATE (MC14411P)

06 OCTAL BUFFERS AND LINE DRIVERS w 3 STAGE OUTPUT (74LS541)





01	OCTAL D F/F w CLEAR (74LS273)
01	12 STAGE RIPPLE CARRY BIT/COUNTER (CD4040)
01	QUAD D TYPE F/F x CLEAR (74LS175)
02	BCD TO 7 SEGMENT DECODER, DRIVES (7446)
01	QUAD 2 INPUT NOR GATES (74LS02)
01	DUAL D TYPE F/F (CD4013)
01	QUAD 2 INPUT AND GATES (74LS08)
01	MONOSTABLE MULTIVIBRATOR w SCHMITT TRIGGER INPUTS (74121)
01	QUAD 2 INPUT OR GATES (74LS32)
01	POSITIVE 5 VOLT REGULATOR (TL780-05C)
02	TARJETAS PERFORADAS

### 3.2 LISTA DE DESIGNACION DE ELEMENTOS

DENOMINACION	SERIE	FUNCION
U1	HM6264	CMOS RAM
U2	CD4040	Contador
U3	74LS02	Puertas NOR
U4, U5, U6, U7, U8	74LS541	Buffers/Drivers
U9	CD4013	F/F tipo D
U10	IM6402	CMOS UART
U11	74LS175	F/F tipo D
U12	MC14411	BIT RATE GENERATOR
U13	TL780-05C	Regulator +5Vdc

U14	74LS08	Puertas AND
U15	54LS244	Buffers/Drivers
U16, U17	7446	Decodificador BCD a 7 Segmentos
U18	74121	Multivibrador Monostable
U19	74LS32	Puertas OR
U20	74LS273	F/F Tipo D
U21	B80C800	Puente Rectificador
LD1, LD2	TIL241-1	Diodos Emisores de Luz
D1	1N914	Diodo Rectificador
Q1	2N2222	Transistor NPN
S1	SP3T	Switch Slide
S2, S3, S5, S6, S7, S8, S9, S10, S11, S13	SPDT	Switches Toggle
S4, S12	SPDT	Switches Momentáneos
R1-R5, R7, R8 R10-R12, R14, R20, R22, R24, R26	10K, 1/4w	Resistencias de Carbón
R6, R27	1K, 1/4w	Resistencias de Carbón
R9	470, 1/4w	Resistencia de Carbón
R13	6.8K, 1/4w	Resistencia de Carbón
R21	15M, 1/4w	Resistencia de Carbón
R25	33K, 1/4w	Resistencia de Carbón

R28	20,1/4w	Resistencia de Carbón
XIAL	1.8432MHz	Cristal
C1	0.1uF	Capacitor Tantalio
C2	0.1uF	Capacitor Mylar
C3	1000uF	Capacitor Electrolítico
T1	120Vac/9Vac	Transformador
F1		Fusible
DP1, Dp2	IEE1737R	Display 7 Segmentos
J1		Zócalo para IC 28 patas
J2		Zócalo para IG 18 patas
J3	DB-25S	Conector RS-232C

### 3.3 TECNICA EMPLEADA EN LA IMPLANTACION E INTERCONEXION DE LOS ELEMENTOS

La técnica empleada para la interconexión de los elementos es la denominada WIRE WRAP, la cual se realiza sobre dos tarjetas con perforaciones aisladas. Estas permiten incrustar los sockets que soportarán los circuitos integrados y tienen una separación entre sí de 0.1 pulgadas, siendo esta la distancia standard entre pines en los circuitos integrados.

El tipo de cable empleado para este propósito es el 30 AWG, el cual es enrollado en las patas de los sockets con un instrumento denominado STRIP WRAP UNWRAP TOOL. Los sockets mencionados ante

riormente son especiales para la utilización de esta técnica debido a la longitud de sus patas (5/8") cuya finalidad es facilitar el enrollamiento del cable.

A causa de la baja frecuencia a que se trabaja en el circuito no se producirán inductancias parásitas de valor considerable en los enrollamientos de alambre, capaces de afectar el normal funcionamiento del EMULADOR; siendo esta la razón principal que permite la utilización de esta técnica.

### 3.4 DIAGRAMAS

En esta sección se presentarán los diagramas de ubicación de los elementos y los esquemáticos, donde se encuentran todos los detalles pertinentes a la construcción del circuito del Emulador.

#### 3.4.1 DIAGRAMA DE UBICACION DE ELEMENTOS

La disposición de los elementos que conforman el circuito del Emulador se muestra en la figura 3.1. Además, se incluye una fotografía mostrando la tarjeta ya elaborada en su totalidad.

#### 3.4.2 DIAGRAMAS ESQUEMATICOS

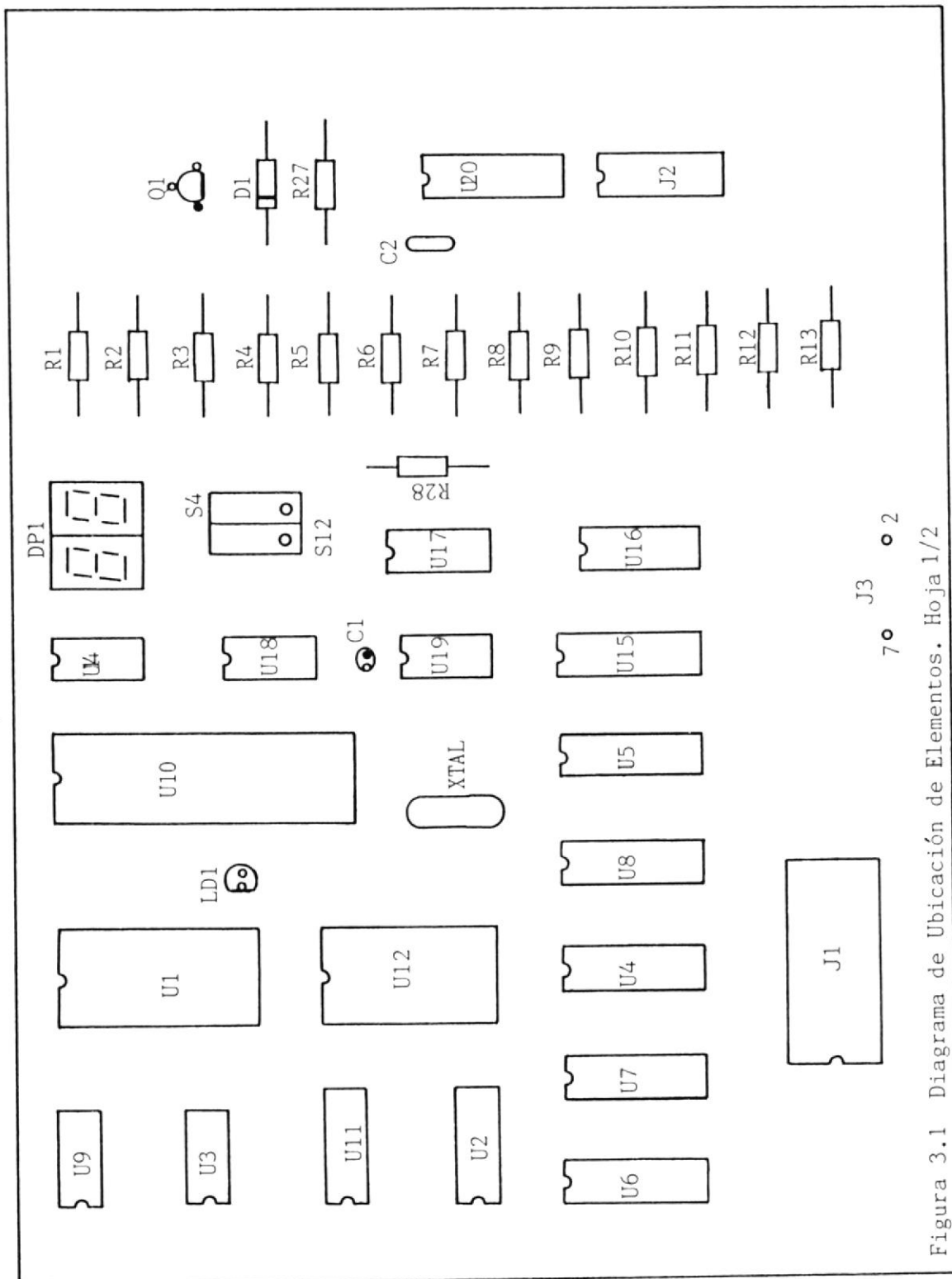


Figura 3.1 Diagrama de Ubicación de Elementos. Hoja 1/2

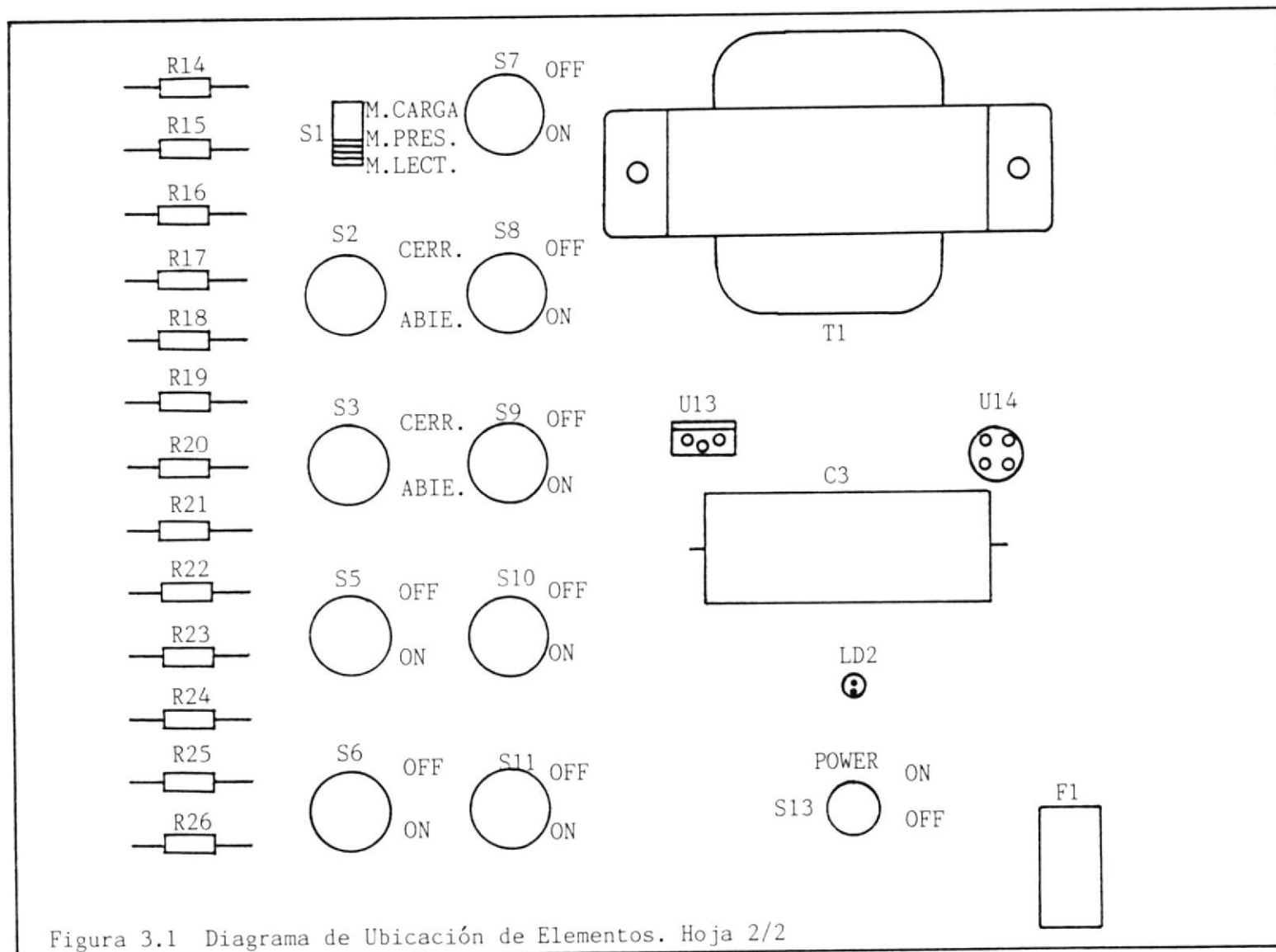


Figura 3.1 Diagrama de Ubicación de Elementos. Hoja 2/2

La figura 3.2 muestra el diagrama esquemático del Circuito de Interface Serie.

La figura 3.3 muestra el diagrama esquemático del Circuito de Memoria.

La figura 3.4 muestra el diagrama esquemático del Circuito Contador de Dirección.

La figura 3.5 muestra el diagrama esquemático del Circuito Pulsador de Un Paso.

La figura 3.6 muestra el diagrama esquemático del Circuito de Display.

La figura 3.7 muestra el diagrama esquemático del Circuito de la Fuente de Poder.

### 3.5 CONFIGURACION DEL PUERTO SERIE

Según lo expuesto en el capítulo anterior secciones 2.2 y 2.2.1 ha sido configurado el cable RS-232C que tendrá aplicación en el presente trabajo. A continuación se muestran los pines utilizados y las conexiones realizadas sobre ellos:

2 Transmitted Data

- 4 Request To Send
- 5 Clear To Send
- 8 Received Line Signal Detector
- 7 Signal Ground
- 6 Data Set Ready
- 20 Data Terminal Ready

Esta configuración está acorde con las normas que exige el protocolo mencionado, cumpliendo con el número de señales requeridas que deben estar presentes en el computador utilizado, el ATARI modelo 1040ST, para que se pueda establecer la comunicación (Handshake) entre el computador y el Emulador (Referencia 7 pp. 90 y Referencia 6 pp. 8).



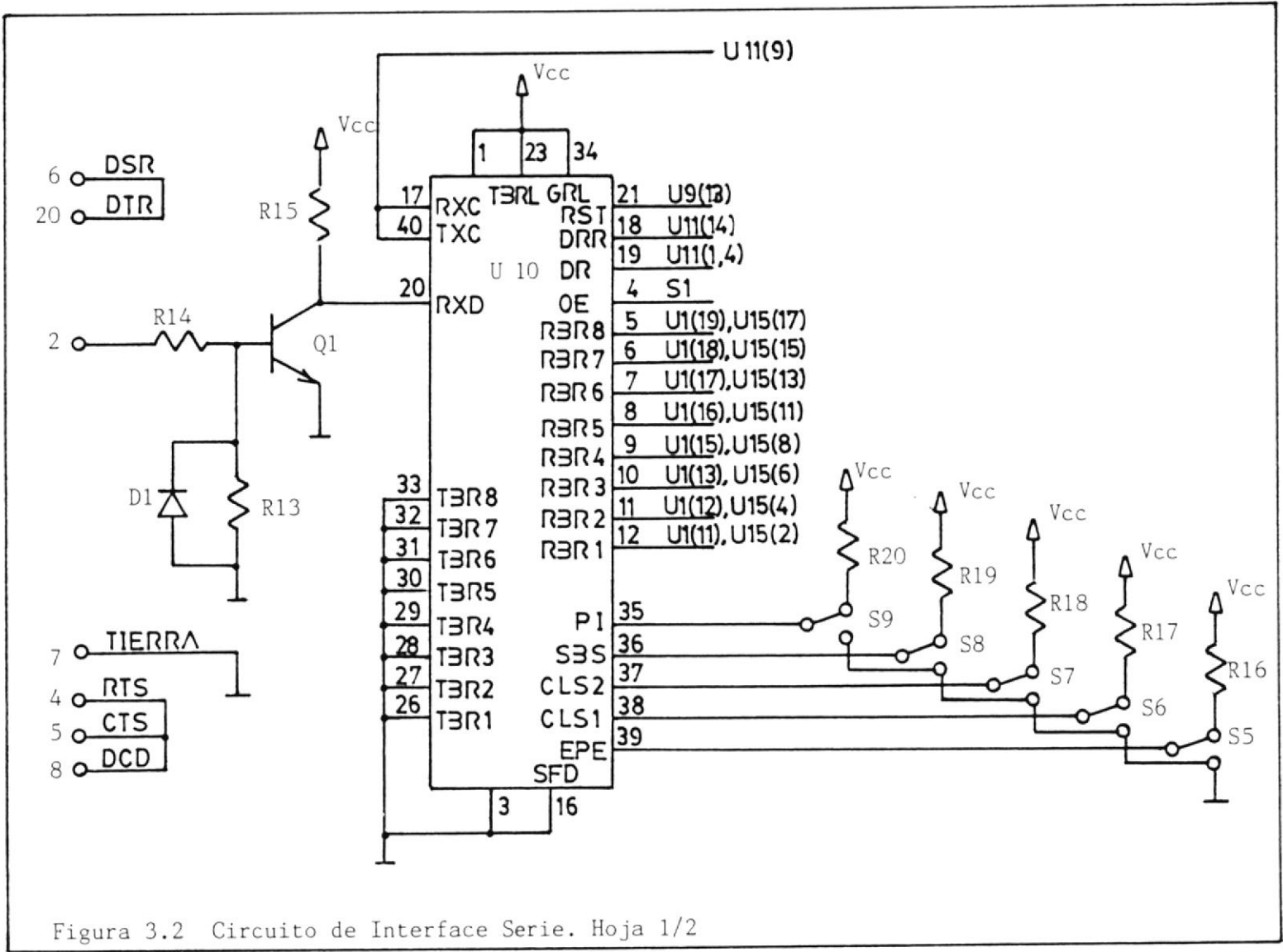


Figura 3.2 Circuito de Interface Serie. Hoja 1/2

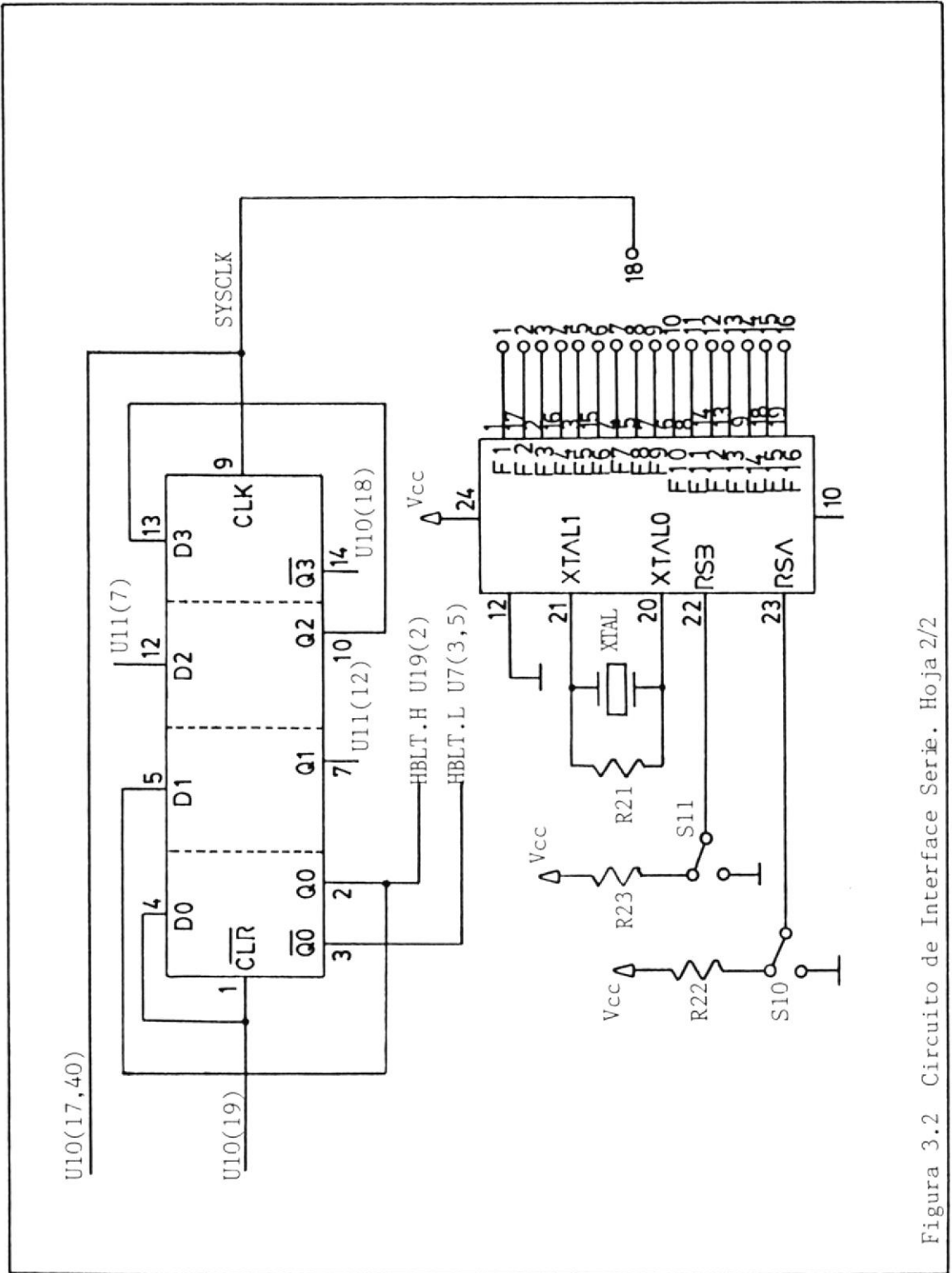


Figura 3.2 Circuito de Interface Serie. Hoja 2/2

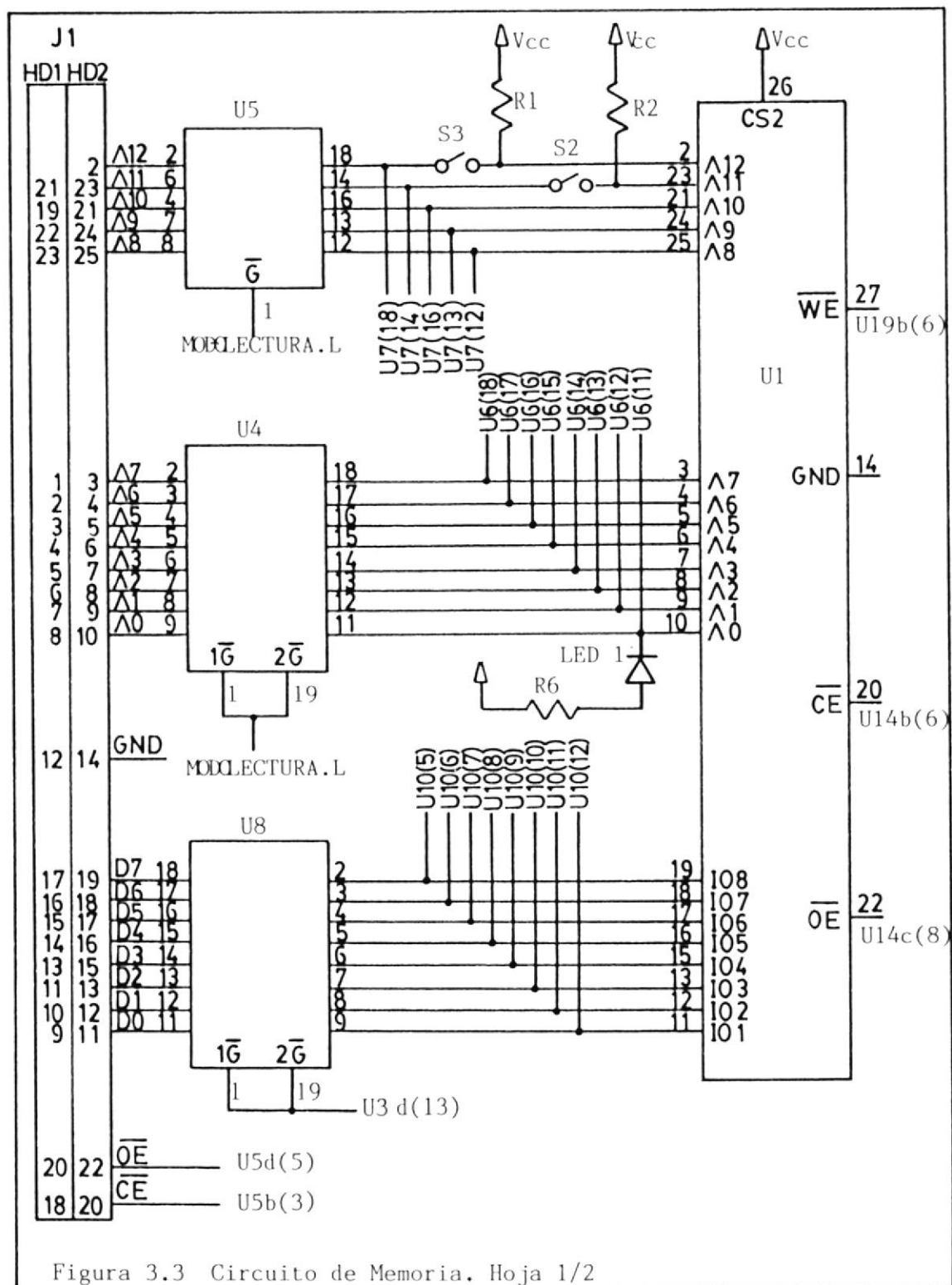
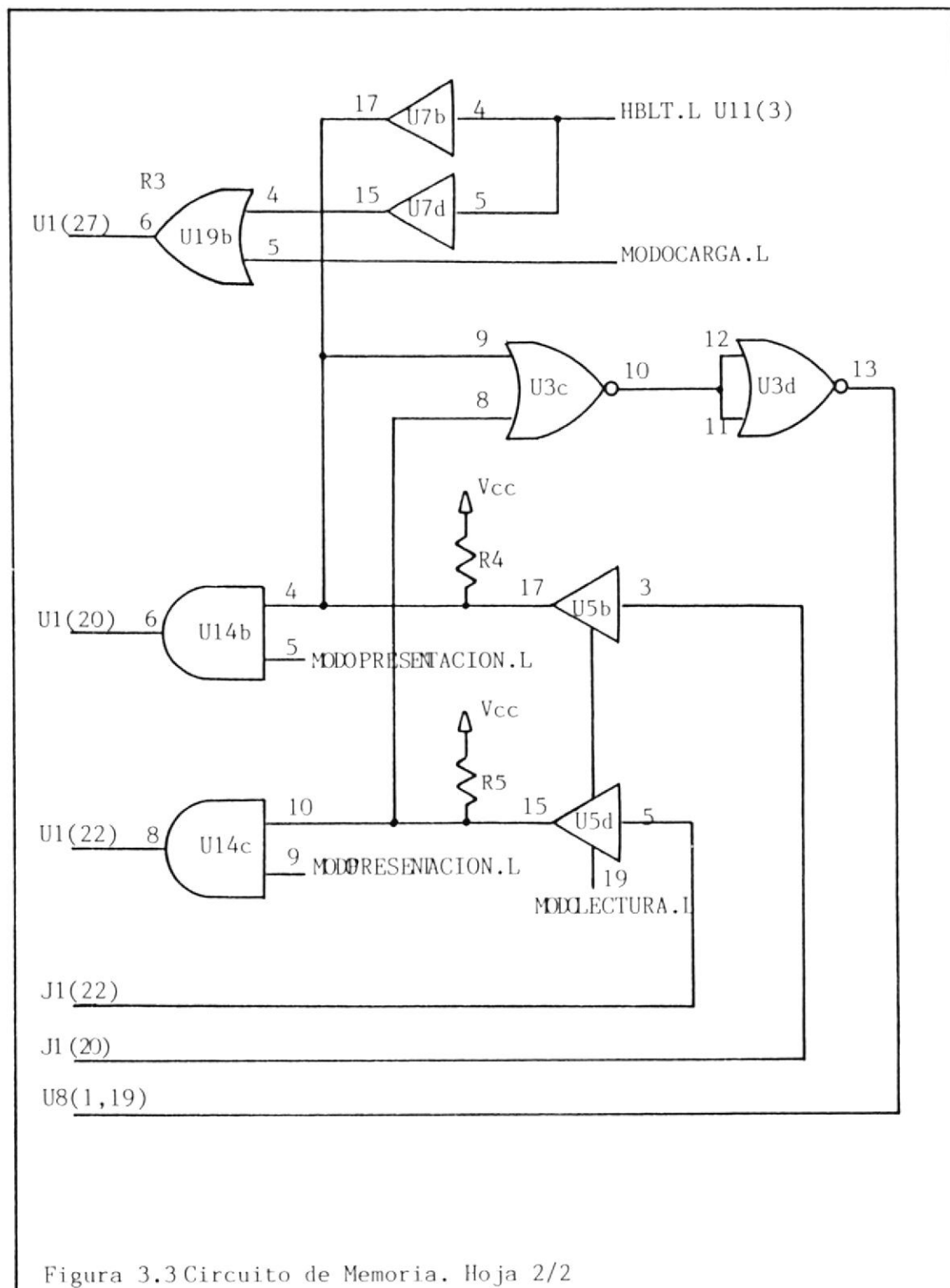


Figura 3.3 Circuito de Memoria. Hoja 1/2



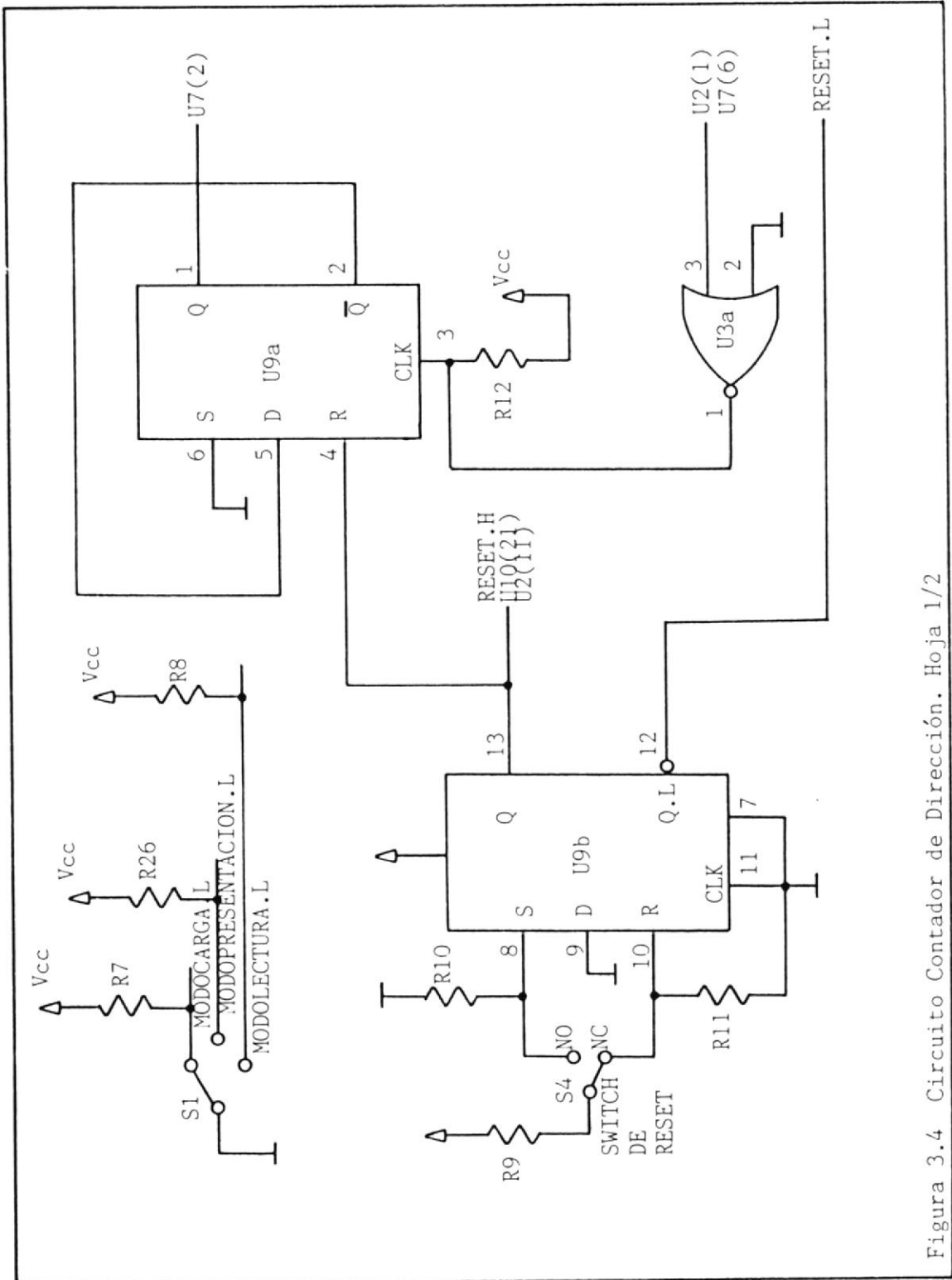


Figura 3.4 Circuito Contador de Dirección. Hoja 1/2

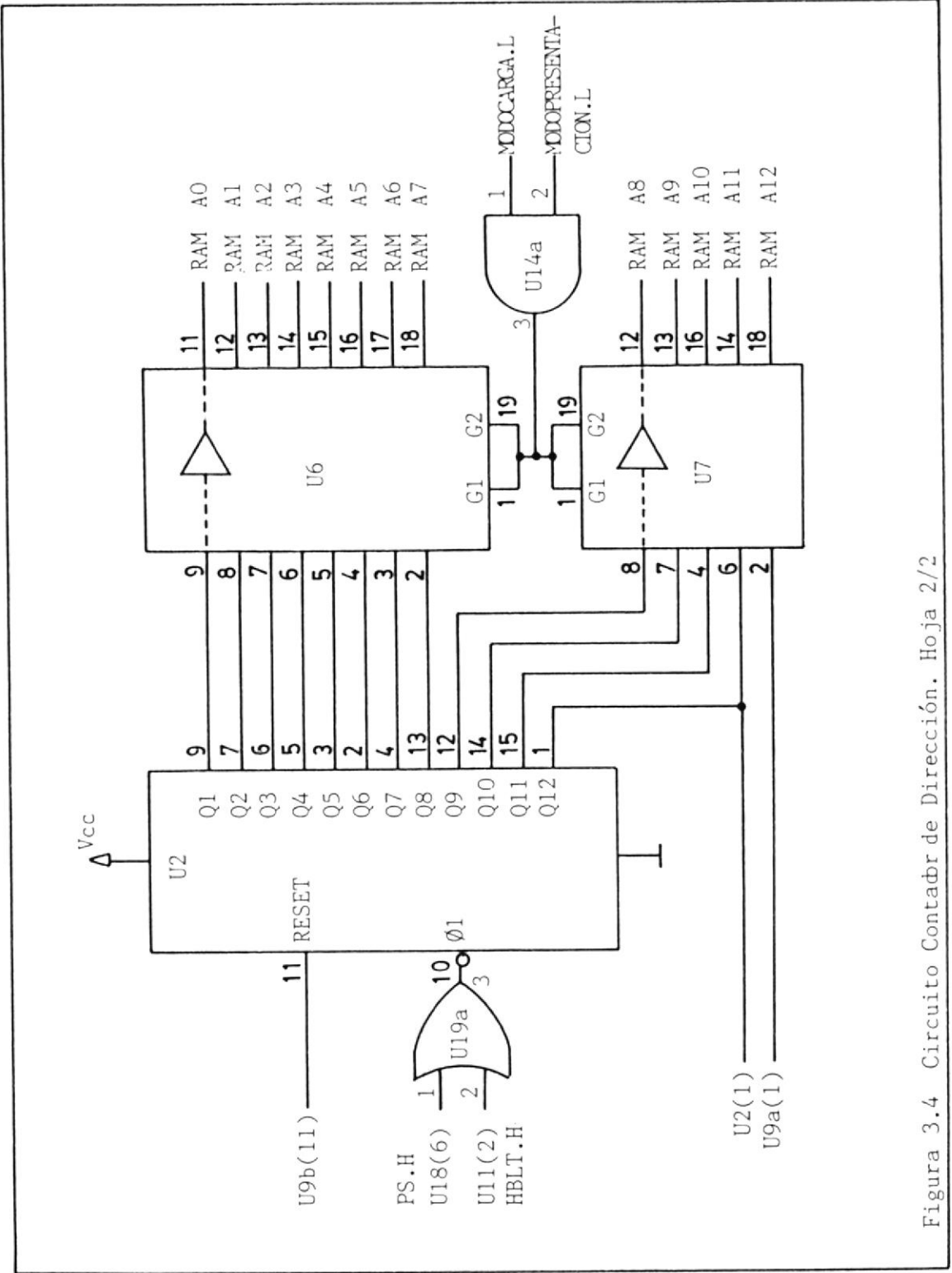


Figura 3.4 Circuito Contador de Dirección. Hoja 2/2

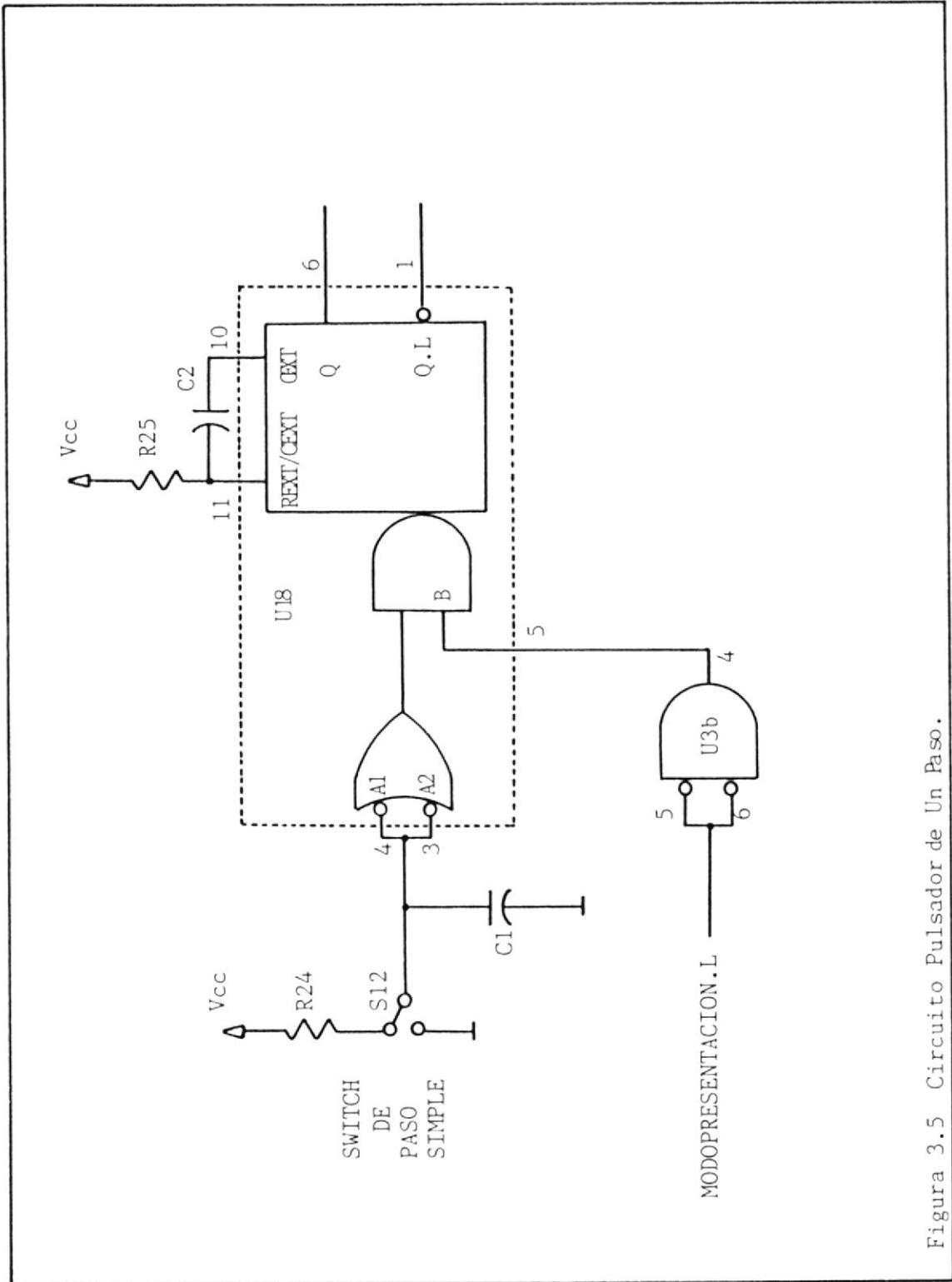


Figura 3.5 Circuito Pulsador de Un Paso.

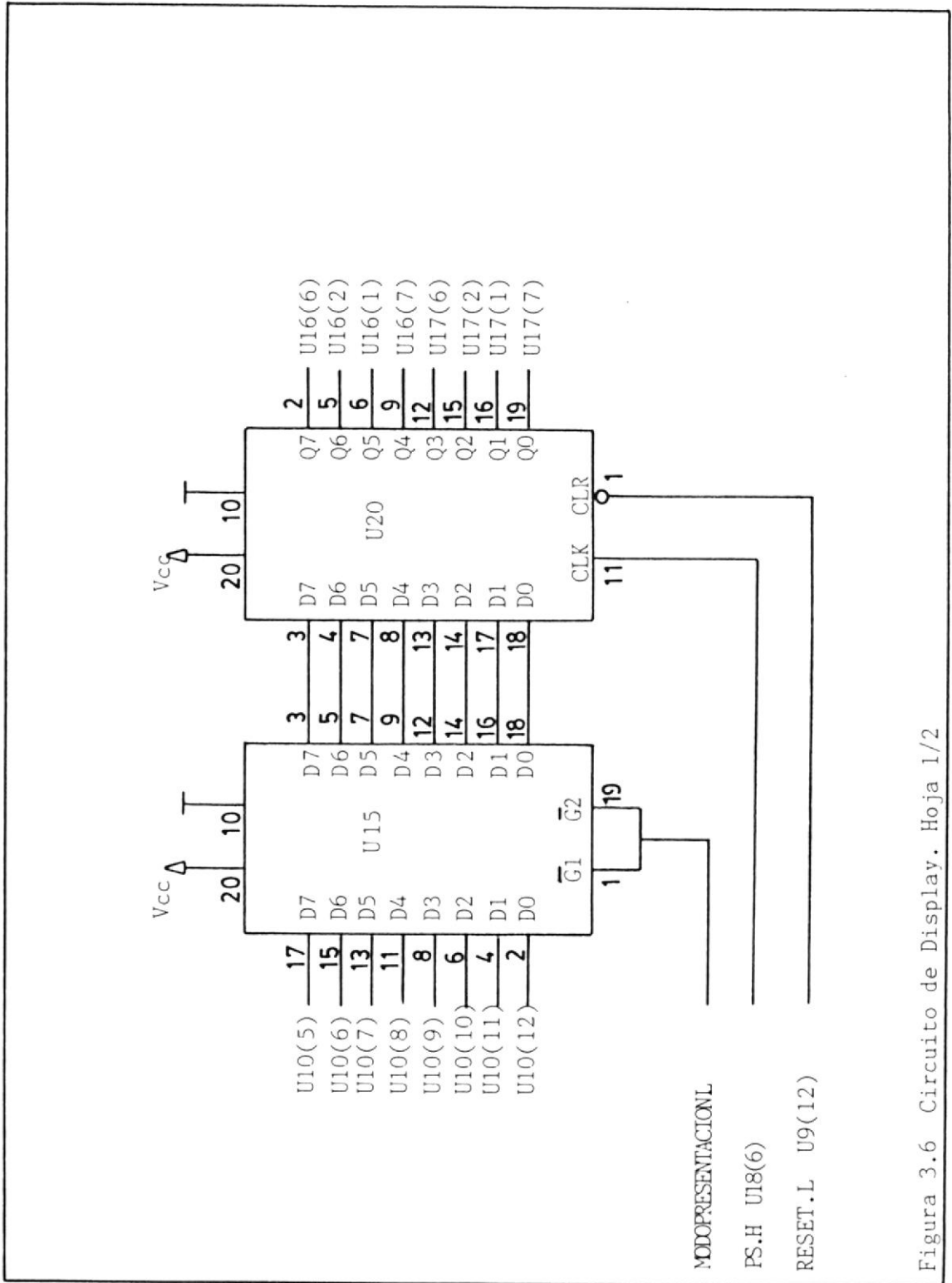


Figura 3.6 Circuito de Display. Hoja 1/2



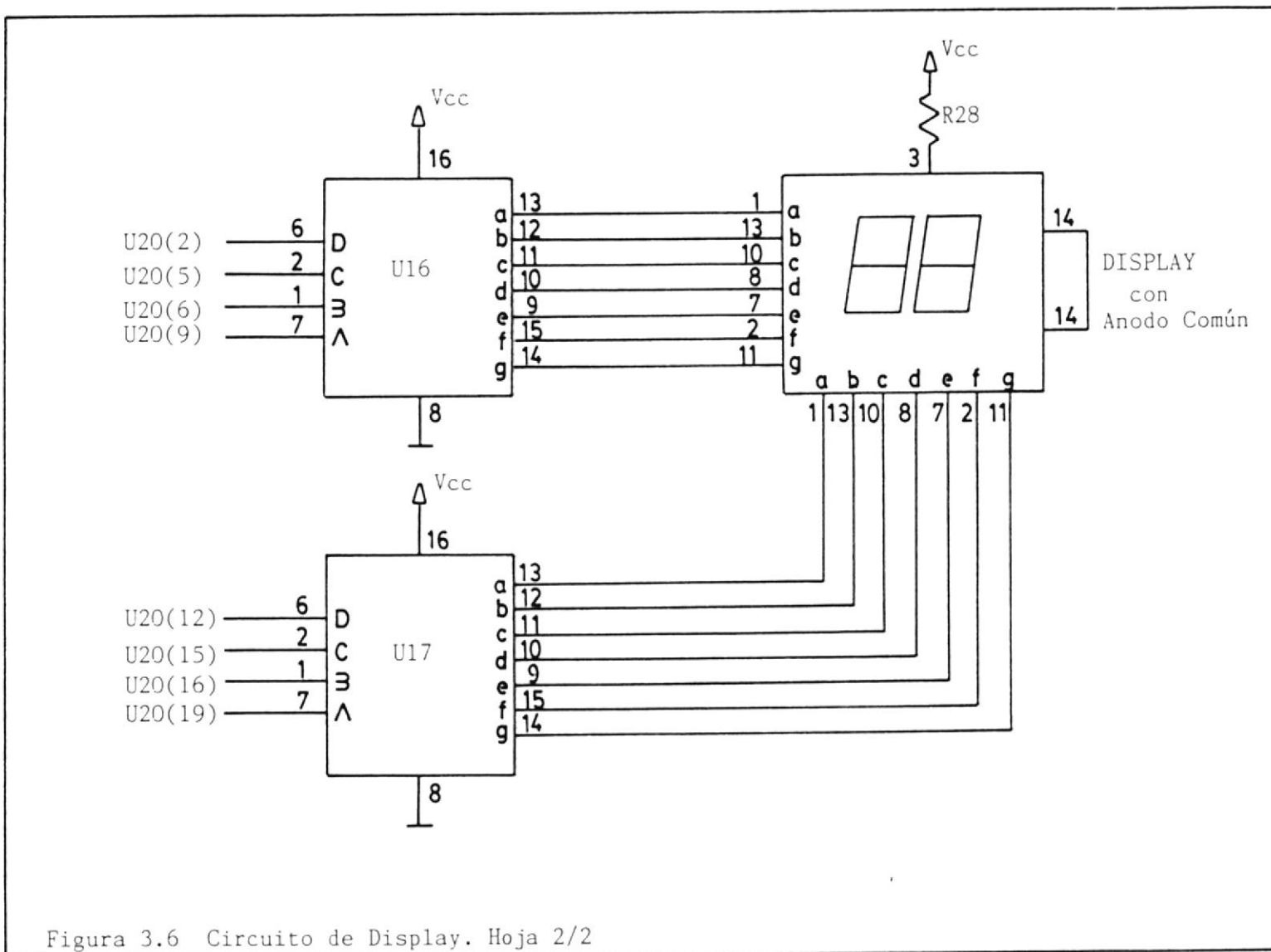


Figura 3.6 Circuito de Display. Hoja 2/2

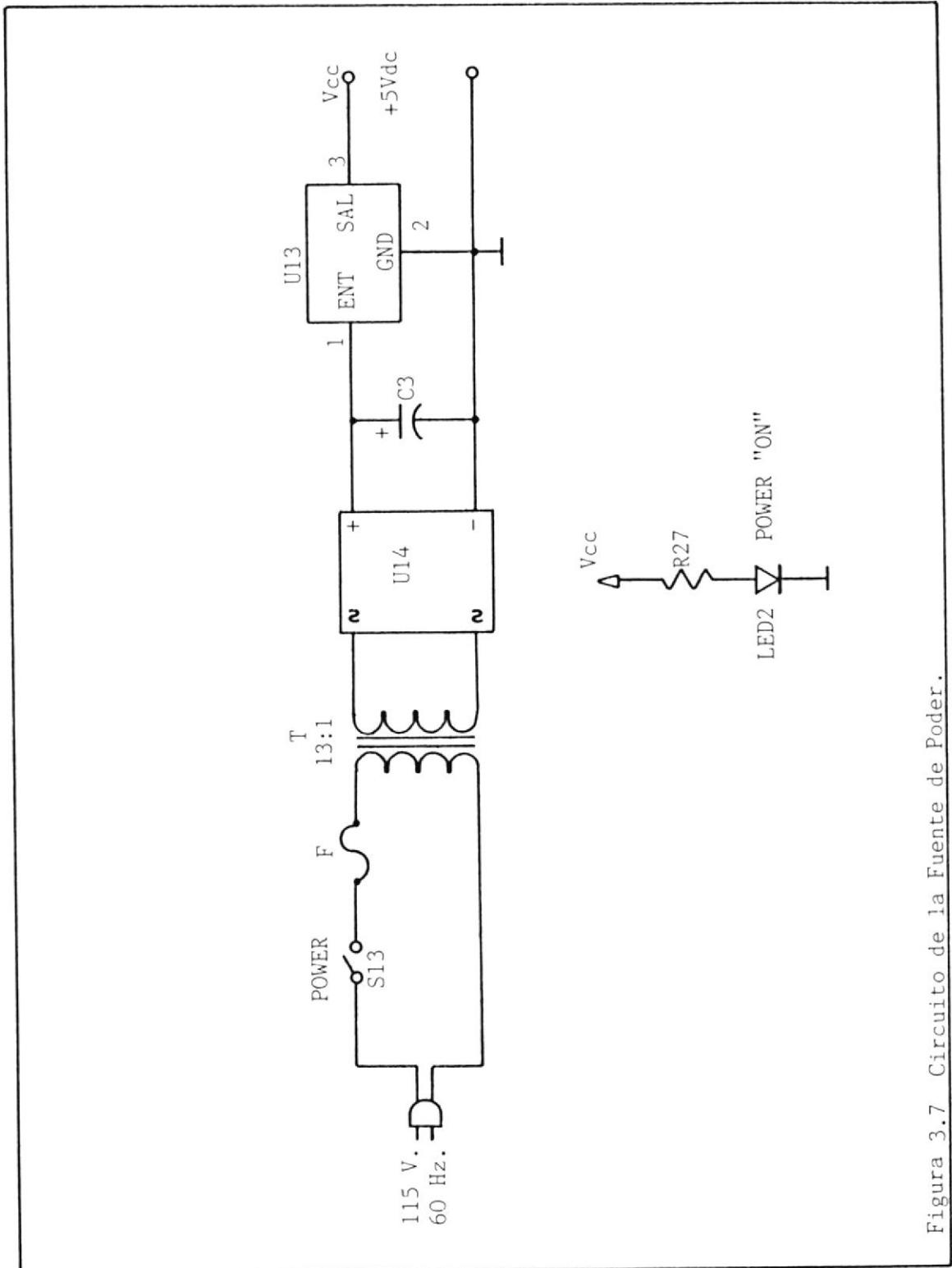


Figura 3.7 Circuito de la Fuente de Poder.

## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES:

- 1.- Debido a la clase de circuitos integrados utilizados se provee al circuito construido de un buen funcionamiento, alta confiabilidad y fácil manejo.
- 2.- Adquirir los convertidores BCD a siete segmentos hexadecimal MC14495, fue dificultoso debido a su alto costo y al desabastecimiento en el mercado nacional; por lo tanto tuvieron que ser reemplazados por dos 7446.
- 3.- La fuente de poder tiene capacidad hasta de un amperio, aunque su consumo es inferior (0.43 Amp. dc).
- 4.- El circuito ha sido diseñado y construido de tal forma que puede ser seleccionado a voluntad, los parámetros del protocolo de trans-

misión; cuya finalidad es el de poder igualar cualquier frecuencia de transmisión impuesta por el computador.

5.- Se ha tratado en lo posible de que los programas desarrollados en base a los diagramas de flujo puedan ser empleados con cualquier otro computador. El lenguaje que nos permite hacer posible este deseo es el lenguaje BASIC por su gran popularidad, su uso generalizado y porque sufre pocas variaciones de una versión a otra. Además, el lenguaje empleado el "BASIC ST" de la Atari es compatible con el BASIC utilizado en las máquinas IBM, cosa que no ocurre por ejemplo con el Ensamblador (debido a que utilizan diferentes microprocesadores) ó el FORTRAN, cuyo aprendizaje es más complicado. Por último, es más probable que el usuario conozca algo de BASIC que de algún otro lenguaje con el cual se puedan realizar las mismas funciones.

6.- Los parámetros seleccionados para probar el Emulador en conjunto con el computador fueron los siguientes: Un bit de inicio, ocho bits de dato, un bit para paridad par y un bit de alto. La frecuencia de transmisión seleccionada fue de 2400 baudios, aunque en el computador se pueden seleccionar tres diferentes frecuencias y estas son de 2400, 4800 y 9600 baudios que son los comunmente utilizados por los modems y otros equipos de esta índole.

7.- Se pudo verificar que toda la información cargada en el computador

y transmitida al Emulador fue recibida y almacenada correctamente en la memoria. Esta verificación se la hizo en el Modo de Presentación, lo cual es una indicación de que cualquier circuito conectado al Emulador podrá de forma similar leer el contenido de la memoria; siendo esto una prueba de que el objetivo propuesto ha sido alcanzado.

### RECOMENDACIONES

- 1.- Los circuitos integrados U16 y U17 (7446) utilizados son decodificadores BCD a siete segmentos por lo cual no presentan la indicación hexadecimal correspondiente a los valores mayores de nueve, es decir, A se verá como "c", B como "o", C como "u", D como "c", E como "b" y, F será representada por la ausencia de iluminación en todos los segmentos. Por lo tanto, según lo expuesto anteriormente hay que tener cuidado al leer el contenido de la memoria en el Modo de Presentación para los valores mencionados.
- 2.- Los programas desarrollados no pueden ser ejecutados en otras máquinas que no sean de la clase ATARI ST. Aunque, el usuario podría realizar las modificaciones necesarias para que el programa original escrito en basic pueda ser utilizado en una máquina IBM o compatible.
- 3.- Antes de iniciar la comunicación Computador-Emulador, se reco-

mienda revisar la opción uno del menú de operación, donde se indican los pasos necesarios que se tienen que seguir para una correcta conexión de los equipos mencionados anteriormente.



APENDICE A

DATOS TECNICOS DEL CIRCUITO INTEGRADO CMOS/LSI UNIVERSAL ASYNCHRONOUS  
RECEIVER TRANSMITTER IM6402IPL



## CARACTERISTICAS

Operación desde DC a 2.0 MHz en +5 Volts.

Baja Potencia 10 mW Típico.

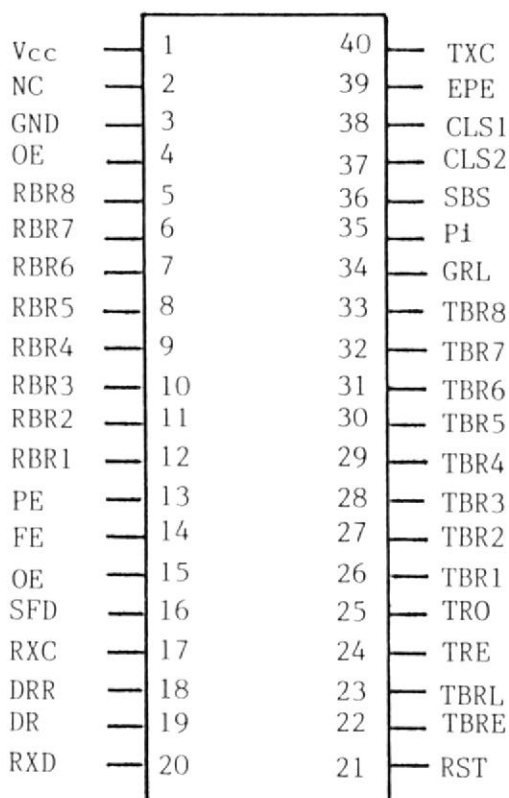
Longitud de Palabra, Bits de Alto y Paridad Programables.

Formateo de Dato y Generación del Status Automático.

Compatible con otros UARTs Comerciales.

Fuente de Poder Unica

## CONFIGURACION EXTERNA





## VALORES MAXIMOS ABSOLUTOS

Voltage de Alimentación	+8.0V
Voltage Aplicado de Entrada o Salida	GND-0.3V a Vcc+0.3V
Almacenaje de Temperatura	-65°C a +150°C
Temperatura de Operación	-40°C a +85°C

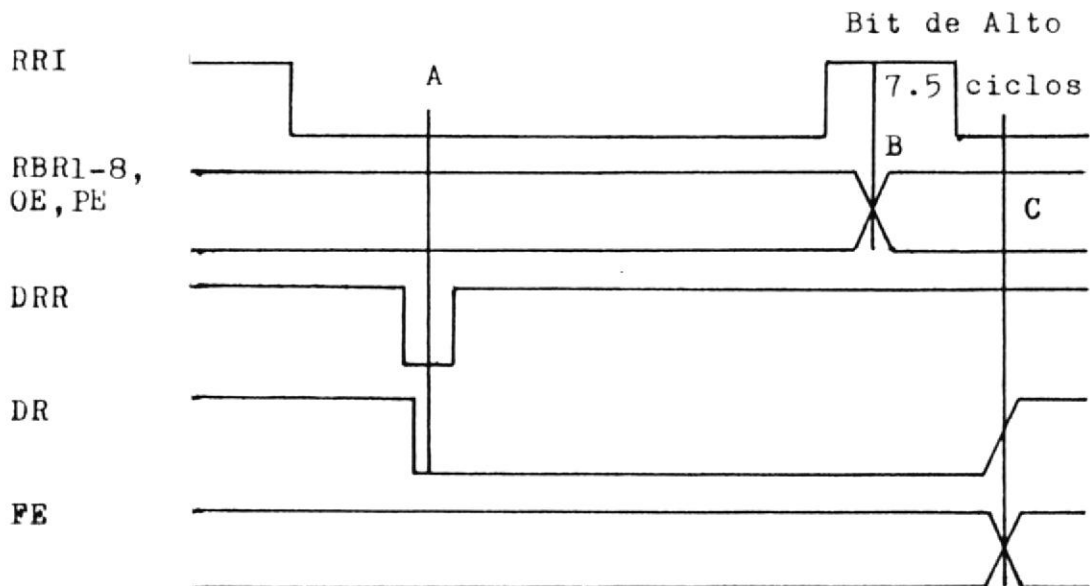
## CARACTERISTICAS ELECTRICAS

Vcc = 5.0V +/- 10%

PARAMETRO	SIMBOLO	MIN	TIP	MAX	UNID
Voltage de Entrada "1" Log "0" Log	Vih	70%Vcc	-	-	V
	Vil	-	-	20%Vcc	V
Voltage de Salida "1" Log "0" Log	Voh	2.4	-	-	V
	Vol	-	-	0.45	V
Corriente de Alimentación	Icc	-	1.0	100	uA
Capacitancia de Entrada	Cin	-	7.0	8.0	pF
Capacitancia de Salida	Co	-	8.0	10.0	pF
Frecuencia de Reloj	Fclock	DC	-	3.0	MHz
Ancho de Pulso CRL, DRR, TBRL	Tpw	150	-	-	nS
Ancho de Pulso MR	Tmr	350	-	-	nS
Tiempo de Fijación de Entradas	Tset	50	-	-	nS
Tiempo de Mantenimiento del Dato de Entrada	Thold	60	-	-	nS
Tiempo de Habilitación de la Salida	Ten	-	-	125	nS

## OPERACION DE RECEPCION

El dato es recibido en forma serie en RInput. Cuando ningún dato está siendo recibido, RInput debe permanecer en un estado de alto. El dato es sincronizado a través de RRClock. La frecuencia del reloj es 16 veces la frecuencia del dato. (A) Un nivel bajo sobre DRReset limpia la línea DReady. (B) Durante el primer Bit de Alto el dato es transferido desde el registro del receptor a el RBRegister. RBRI es el bit menos significativo. Un nivel alto sobre OError indica desbordamiento. Un desbordamiento ocurre cuando DReady no ha sido limpiado antes que el carácter presente fuese transferido al RBRegister. (C) 1 ciclo de reloj más tarde DReady es refijado a un nivel alto, y FError es evaluado. Un nivel alto sobre FError indica que un bit de alto inválido fue recibido, un error de estructura y un nivel alto sobre PError indica un error de paridad.



APENDICE B

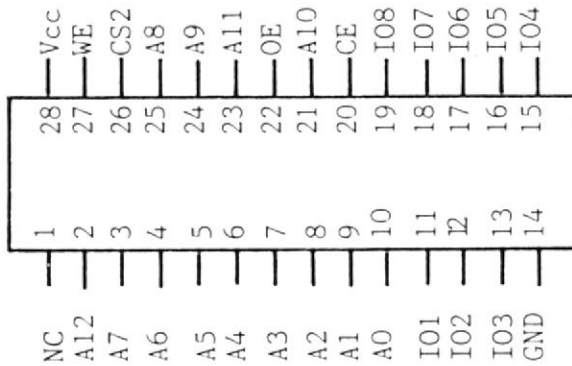
DATOS TECNICOS DEL CIRCUITO INTEGRADO RANDOM ACCESS MEMORY HM6264LP-12

CMOS RAM Estática de Alta Velocidad de 8192 palabras x 8 bits

### CARACTERISTICAS

Rápido Tiempo de Acceso	120ns (max)
Baja Potencia de Standby	0.1mW (tip)
Baja Potencia en Operación	200mW (tip)
Alimentación Unica de +5Vdc	
Memoria Completamente Estática	
Entrada y Salidas de dato Comunes, con Salidas de Tres Estados	
Directamente Compatible con TTL	
Configuración Standard en paquete de 28 Pines	
Compatible pin a pin con la EPROM de 64K HN482764	

### CONFIGURACION EXTERNA



## VALORES MAXIMOS ABSOLUTOS

Voltage en los terminales	-0.5 a 7.0	V
Disipación de Potencia	1.0	W
Temperatura de Operación	0 a+70	°C
Almacenaje de Temperatura	-55 a +125	°C
Almacenaje de Temperatura (bajo polarización)	-10 a +85	°C

## TABLA DE LA VERDAD

WE	CS1	CS2	OE	MODO	PINES	E/S	NOTA
X	H	X	X	No Seleccionada	Alta Z		
X	X	L	X	No Seleccionada	Alta Z		
H	L	H	H	Sal. Desabilit.	Alta Z		
H	L	H	L	Lectura	Salida Dato		
L	L	H	H	Escritura	Entrada Dato		Escr.(1)
L	L	H	L	Escritura	Entrada Dato		Escr.(2)

## CONDICIONES DE OPERACION DC RECOMENDADAS

ITEM	SIMB	MIN	TIP	MAX	UNIDAD
Voltage de Alimentación	Vcc	4.5	5.0	5.5	V
	GND	0	0	0	V
Voltage de Entrada	Vih	2.2	-	6.0	V

	Vi1	-0.3	-	0.8	V
Voltage de Salida	Voh	2.4	-	-	V
	Vol	-	-	0.4	V

## CICLO DE LECTURA

ITEM	SIMB	MIN	MAX	UNID
Tiempo del Ciclo de Lectura	Trc	120	-	ns
Tiempo de Acceso de Dirección	Taa	-	120	ns
Selección del Chip a Salida $\overline{CS1}$	Tco1	-	120	ns
CS2	Tco2	-	120	ns
Salida Habilitada a Salida Validada	Toe	-	60	ns
Selección de Chip a Salida en $\overline{CS1}$	T1z1	10	-	ns
Baja Impedancia CS2	T1z2	10	-	ns
Salida Habilitada a Salida en Baja Impedancia	Tolz	5	-	ns
Deselección del Chip a Salida $\overline{CS1}$	Thz1	0	40	ns
en alta Impedancia CS2	Thz2	0	40	ns
Salida Desabilitada a Salida en Alta Impedancia	Tohz	0	40	ns
Salida Mantenido desde Cambio de la Dirección	Toh	10	-	ns

## CICLO DE ESCRITURA

ITEM

ns	-	120	Twc	Tiempo del Ciclo de Escritura
ns	-	85	Tcw	Selección del Chip a Fin de Escritura
ns	-	0	Tas	Tiempo de Fijación de la Dirección
ns	-	85	Taw	Dirección Válida a Fin de Escritura
ns	-	70	Twp	Ancho del Pulso de Escritura
ns	-	5	Twrl	Tiempo de Recuperación de CS1, <u>WE</u>
ns	-	15	Twr2	Escritura CS2
ns	40	0	TwHz	Escritura a Salida en Alta Z
ns	-	50	Tdw	Dato a Tiempo de Escritura Coincidente
ns	-	0	Tdh	Dato Mantenido desde Tiempo de Escritura
ns	40	0	Tohz	OE a Salida en Alta Impedancia
ns	-	5	Tow	Salida Activa desde Fin de Escritura

APENDICE C

DATOS TECNICOS DEL CIRCUITO INTEGRADO BIT RATE GENERATOR MC 14411P



## CARACTERISTICAS

Alimentación Unica de +5.0 Vdc

Oscilador de Cristal Interno Controlado para Estabilidad (1.8432 MHz)

Dieciseis Diferentes Salidas de Frecuencias de Reloj

Ciclo de Salida del 50%

Tiempo de Bases Programables para Uno de Cuatro Indices de Salidas

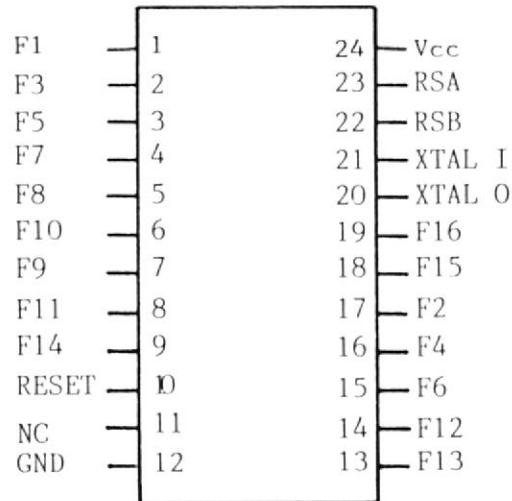
Múltiples

Salidas Compatibles con TTL

Protección a Diodos en Todas las Entradas

Reloj Externo puede ser Aplicado al pin 21

## CONFIGURACION EXTERNA



## VALORES MAXIMOS

PARAMETRO	SIMB	VALOR	UNID
Rango de la Fuente de Poder	Vdd	5.25 a -0.5	Vdc
Voltage en las Entradas	Vin	Vdd+0.5 a Vss-0.5	Vdc
Corriente de Drenaje por Pin	I	10	mAdc
Rango de la Temperatura de Operación	Ta	-40 a +85	°C
Rango de Almacenaje de Temperatura	Tstg	-65 a +150	°C

## CARACTERISTICAS ELECTRICAS

CARACTERISTICAS	SIMB	Vdd	MIN	TIP	MAX	UNID
Voltage de Alimentación	Vdd	-	4.75	5.0	5.25	Vdc
Voltage de "0" Lógico	Vout	5.0	-	0	0.05	Vdc
Salida "1" Lógico		5.0	4.95	5.0	-	Vdc
Voltage de Entrada	Vil	5.0	-	2.25	1.5	Vdc
	Vih	5.0	3.5	2.75		Vdc
Corriente de Entrada	Iin	-	-	$\pm 1 \times 10^{-5}$	0.1	uAdc
Tiempo de la Pendiente Positiva de la Salida	Ttlh	5.0	-	70	200	ns
Tiempo de la Pendiente Negativa de la Salida	Tthl	5.0	-	70	200	ns
Frecuencia del Reloj Ent.	Fcl	5.0	-	-	1.85	MHz

APENDICE D

PROGRAMAS DE CONTROL.

```

10 rem *** PROGRAMA PRINCIPAL ***
20 fullw 2: clearw 2: gotoxy 0,0
30 print: print: print spc (28); " - MENU DE OPERACION - "
40 print: print: "1.- Mostrar indicaciones para operación del EMULADOR"
50 print: print: "2.- Almacenar datos en el minicomputador."
60 print: print: "3.- Transmitir datos al EMULADOR."
70 print: print: "4.- Realizar correcciones."
80 print: print: "5.- SALIR.":print
90 input: "Digitar 1-2-3-4- o 5 según operación a realizar"; selección%
100 if selección % 6 then goto 90
110 on selección % goto 1690, 170; 330; 480, 140
120 print: input "Alguna otra operación"; respuesta $
130 if respuestas $ = "si" then goto 20
140 clearw 2: gotoxy 0,0
150 print: print: print: print: spc (20), "FIN DE SESION DE TRABAJO, GRA-
CIAS....."
160 full 2: end
170 rem *** CARGAR DATOS DESDE EL TECLADO ***
180 clearw 2: gotoxy 0,0
190 open "0", # 1, "datos"
200 input "Digitar el Numero de Bytes a ingresarse", contador %: print
210 print "Ingresar datos en formato hexadecimal a través del teclado

```

```
U. debe
22Ø print "Anteponer - &h - antes de escribir el byte.": print
23Ø for i=1 to contador %
24Ø input "Ingresar byte:" byte %
25Ø write # 1, byte %
26Ø print "Número de bytes almacenados"; i
27Ø next i
28Ø close # 1
29Ø open "O", #2, "nudby"
30Ø write # 2, contador %
31Ø close # 2
32Ø goto 12Ø: rem * REGRESAR A PROGRAMA PRINCIPAL *
33Ø rem *** TRANSMISION DE DATOS DESDE EL COMPUTADOR AL EMULADOR **
34Ø clearw 2: gotoxy Ø,Ø
35Ø print: print spc (25); "Inicio de transmision"
36Ø open "I", # 2, "nudby"
37Ø input # 2, contador %: print spc (22); "Numero de Bytes a Transmitir:
    contador %
38Ø close # 2
39Ø open "I", # 1, "datos"
40Ø for i=1 to contador %
41Ø input # 1, byte %
42Ø bytx%=byte%: print hex$ (bytx%)
43Ø rem out 2, bytx%
44Ø next i
```

```

450 close # 1
460 print spc (25); *Fin de Transmisión"
470 goto 120: rem *REGRESAR A PROGRAMA PRINCIPAL*
480 rem *** REALIZAR CORRECCIONES EN DATOS INGRESADOS ***
490 clearw 2: gotoxy 0,0: option base 1
500 print: print: print spc (28); "Menú de Correcciones"
510 print: print " 1.- Introducir bytes."
520 print: print " 2.- Borrar bytes."
530 print: print " 3.- Cambiar bytes."
540 print: print " 4.- SALIR."
550 print: input "      Digite 1, 2, 3 o 4 según la corrección a rea
      lizar"; corr%
560 on corr% goto 600, 1030, 1450, 590
570 print: input: "Alguna otra corrección"; rep$
580 if rep$= "si" then goto 490
590 goto 120: rem *REGRESAR A PROGRAMA PRINCIPAL*
600 clearw 2: gotoxy 0,0
620 input "INGRESAR NUMERO DE BYTES A SER INTRODUCIDOS: ";num%: print
630 open "I", #2, "nudby": rem * LEER # DE BYTES*
640 input # 2, contador%
650 close #2
660 cnta%=contador %: cntn%=cnta%num%: ref%=0
670 dim crs% (cntn%,2)
680 open "I", #1, "datos":rem *CARGAR DATOS EN MATRIZ CRS *
690 for i=1 to cnta%

```

```
700  input #1, byte%
710  crs% (i,1) = byte%
720  next
730  close # 1
740  for k=1 to num%
750  input "Ingresar posición y byte a ser introducido";m%,byten%
760  m1%=m%
770  if m1%-1 then goto 820
780  m2%=m1%-1
790  for i=1 to m2%
800  crs% (i,2)=crs% (i,1)
810  next
820  crs% (m1%,2) = byten%: if m% chta% then goto 870
830  for i=m1% to chta%
840  j=i+1
850  crs% (j,2)=crs(i,1)
860  next
870  contador%=contador%+1:cnta%=contador%
880  rem if ref%=num% then goto 920
890  for i=1 to cnta%
900  crs% (i,1)=crs% (i,2):next
910  next k
920  print:print "Ud. no puede introducir mas bytes."
930  open "0", #1, "datos"
940  for i=1 to cnta%
```

```
950 byte%=crs%(i,2)
960 write #1, byte%
970 next
980 close # 1
990 open "O", #2, "nubby"
1000 write #2, contador%
1010 close #2: clearw 2: gotoxy 0,0
1020 erase crs%: goto 570:rem *REGRESAR A PROGRAMA CORRECCIONES*
1030 rem *** SUB-PROGRAMA B ***
1040 clearw 2: gotoxy 0,0
1050 open "I", #2, "nubby"
1060 input #2, contador%
1070 close # 2
1080 cnta%=contador%
1090 dim crs% (cnta%,2)
1100 open "I", #1, "datos"
1110 for i=1 to cnta%
1120 input #1, byte%
1130 crs%(i,1)= byte%
1140 next
1150 close # 1
1160 print "Ingresar posición y número de bytes a ser borrados"
1170 input m%, num%
1180 if m%=1 then goto 1230
1190 m1%=m%-1
```





```
1200 for l=1 to ml%
1210 crs% (i,2)=crs% (i,1)
1220 next
1230 m2%=m%+num%: j=m%: if m2% chta% then goto 1280
1240 for i=m2% to chta%
1250 crs% (j,2)=crs% (i,1)
1260 j=j+1
1270 next
1280 contador%=contador%-num%:cnta%=contador%
1290 input " Borrar algún otro byte"; rep$
1300 If rep$= "no" then goto 1350
1310 for i=1 to chta%
1320 crs%(i,1)=crs%(i,2)
1330 next
1340 goto 1160
1350 open "0", #1, "datos"
1360 for i=1 to chta%
1370 byte%=crs% (i,2)
1380 write #1, byte%
1390 next
1400 close # 1
1410 open "0" #2, "nubby"
1420 write # 2, contador %
1430 close # 2, clearw 2: gotoxy 0,0
1440 erase crs%: goto 570: rem *REGRESAR A PROGRAMA CORRECCIONES *
```

```
1450 rem *** SUB-PROGRAMA C ***
1460 clearw 2: gotoxy 0,0
1470 open "I", # 2, "nudby"
1480 input # 2, contador %
1490 close # 2
1500 cnta % = contador %: option base 1
1510 dim crs % (contador%)
1520 open "I", # 1, "datos"
1530 for i=1 to cnta%
1540 input # 1, bute%
1550 crs% (i) = byte%
1560 next
1570 close # 1
1580 print: input "Ingresar posición y byte a ser cambiado"; m%,byten%
1590 crs% (m%) = byten%
1600 input "Cambiar algún otro byte"; rep$
1610 if rep$="si" then goto 1580
1620 open "O", # 1, "datos"
1630 for i=1 to cnta%
1640 byte%=crs% (i)
1650 write # 1, byte%
1660 next
1670 close # 1: clearw 2: gotoxy 0,0
1680 erase crs%: goto 570: rem * Regresar a Programa Correcciones *
1700 print: print spc (15); "INDICACIONES DE OPERACION DEL EMULADOR":
print
```

APENDICE E  
MANUAL PARA EL USUARIO

Los terminales para conexión del puerto serie del Emulador se identifican en la figura 3.2, los cuales están ordenados de acuerdo al conector utilizado (DB-25S). Como el UART empleado opera con una relación entre la frecuencia del reloj y la frecuencia del dato de 16, entonces se hace necesario introducir este valor y esto se obtiene fijando a cero el switch S10 y a uno el switch S11. La señal de reloj se toma desde el conector J2 de acuerdo a la tabla III, es decir, según sea el número de bps requeridos y esto es obtenido cortocircuitando externamente el pin seleccionado con el pin denominado SYSCLK.H.

Los switches S5, S6, S7, S8 y S9 sirven para definir el protocolo de transmisión serie, mostrándose en la tabla I, las indicaciones de la posición en la cual deben estar para elegir las diferentes posibles estructuras.

Además, el tipo de memoria a emularse puede ser elegido a voluntad por medio de los switches S3 y S2 tal como se indica a continuación:

S3	S2	TIPO DE MEMORIA
Abierto	Abierto	2716
Abierto	Cerrado	2732
Cerrado	Cerrado	2764

Considerando todos los detalles anteriores y colocando el switch S1 en el Modo de Carga el Emulador estará listo para recibir información desde

un computador.

Luego, la información transmitida y recibida en el Emulador puede ser chequeada byte a byte solamente colocando el switch S1 al Modo de Presentación y a cero la dirección con el switch S4 e incrementándola con el switch S12 paso a paso para poder leer el contenido de la RAM a través de un display.

Después de lo expuesto anteriormente ahora el Emulador puede ser conectado al circuito de aplicación. Para ello se tiene que colocar el switch S1 en el Modo de Lectura y además conectar la barra de dato y la barra de dirección del circuito en mención al conector J1 del Emulador; respetando el orden asignado a los terminales tal como lo especifica la figura 3.3. Además, dentro del Programa Principal existe un sub-programa denominado MIOE (Mostrar Indicaciones para Operación del Emulador) donde se presentan a través de la pantalla del computador las indicaciones necesarias para el manejo del Emulador, incluyendo los pasos a seguir para su conexión con el computador y con el circuito de aplicación.

APENDICE F

INSTRUCCIONES PARA EL MANEJO DEL COMPUTADOR ATARI 1040ST

Antes de encender el computador, el usuario debe introducir en el disk drive un disco flexible de 3.5 pulgadas previamente formateado conteniendo por lo menos el archivo DESKTOP. INF y el directorio BASIC conteniendo el archivo BASIC. PRG. El primer archivo mencionado contiene to la información que el ST necesita para crear el desktop, es decir, contiene la información necesaria para configurar el puerto RS-232C y con trolar la configuración del impresor principalmente. Además, contiene otros datos los cuales son útiles para controlar el matíz de las presen taciones en la pantalla, la forma de presentación de los archivos existentes, las ventanas, etc.

Luego se procede a cargar el BASIC y a abrir el archivo BASIC. PRG. Al cabo de unos pocos minutos la pantalla cambiará dramáticamente y en el borde superior aparecerán las palabras "DESK", "FILE", "RUN" "EDIT" y "DEBUG". El resto de la pantalla contendrá tres ventanas, donde la ven tana superior izquierda es denominada "LIST", la superior derecha "OUT PUT" y la inferior es llamada "COMMAND". Si se mira cuidadosamente se podrá observar una cuarta ventana escondida detrás de las otras. Esta ventana es denominada "EDIT".

En la ventana Command es donde se puede tipear comandos directos al com putador y, donde el ST responde mostrando mensajes directos. La venta na Edit es usada para escribir y editar programas. La ventana Output es donde los resultados de los programas aparecen y, por último en la venta na List se podrá observar un listado de las instrucciones que con

forman un programa. Una vez que se ha llegado hasta este nivel se está en capacidad de realizar cualquier operación permisible por el BA SIC ST, tal como hacer el llamado de un programa en especial con la instrucción denominada LOAD.



## B I B L I O G R A F I A

- 1.- BALL STUART. Byte; Mc Graw-Hill Inc.; USA: Abril 1986: pp. 105-110.
- 2.- BOYLESTAD R. - NASHELSKY L.. Electronic Devices and Circuit Theory; Prentice-Hall; USA; 1978; pp. 87-102, 679-686.
- 3.- BRUCKMANN R. - ENGLISH L. - WALKOWIAT J. - GERITS K.; Atari S/Tri-cks and Tips; Abacus Software; USA; pp. 3-54.
- 4.- COMPUTE; S ST/Programmers Guide; Compute; Publications; USA; pp. 3-125, 291-298.
- 5.- DISEÑO DIGITAL: Un Método Sistematizado; USA.
- 6.- DITTRICH STEFA; Peeks and Pokes; Abacus Software; USA; pp. 8-10, 34-37, 73-77, 148-150.
- 7.- GERITS K. ENGLISH L. - BRUCKMANN R.; Atari ST/ Internals; Abacus Software; USA; pp. 28-40, 62-63, 90-92, 144, 174-175, 239-241.
- 8.- GUIDE TO PROFITABLE PROM PROGRAMMING; Data I/O; USA: Septiembre 1982 pp. 3-7.
- 9.- HORSEDESKI MICHAEL; Microprocessor Cookbook; Tab Books; USA; Mayo 1979; pp. 39-55.

- 10.- IC MASTER; Hearst Business Communications; USA; 1983; pp. 672-674.
- 11.- LINEAR DATABOOK I; National Semiconductor Corp.; USA; 1988.
- 12.- MOTOROLA CMOS INTEGRATED CIRCUITS; Motorola Inc.; USA; 1978.
- 13.- NICHOLS ELIZABETH - NICHOLS JOSEPH - MUSSON KEITH; Data Communications for Microcomputers; McGraw-Hill Inc.; USA; 1982.
- 14.- THE TTL DATA BOOK Vol. II; Texas Instruments; USA; 1985.



A.F. 141915