

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERIA

CONSTRUCCIÓN DE UN EMULADOR DE MEMORIAS EPROM

**PROYECTO PREVIO A LA OBTENCIÓN DEL TITULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES**

JORGE ADRIÁN BENITEZ ESTRELLA

DIRECTOR: ING. RAMIRO MOREJON

Quito, marzo del 2002

DECLARACIÓN

Yo Jorge A. Benítez E. , declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

A handwritten signature in black ink, consisting of a stylized, cursive script that appears to read 'J. Benítez E.' with a horizontal line extending to the left.

Jorge A. Benítez E.

CERTIFICACION

Certifico que el presente trabajo ha sido realizado en su totalidad por el señor Jorge Adrián Benítez Estrella



ING. RAMIRO MOREJON
DIRECTOR DE TESIS

AGRADECIMIENTOS

Al Sr. Ing. Ramiro Morejon por su valiosa colaboración en el desarrollo del presente trabajo, como también al personal de Laboratorio de Sistemas Digitales que contribuyeron en el mismo.

DEDICATORIA

A MIS PADRES,
ESPOSA
E HIJOS

CONTENIDO

RESUMEN	1
PRESENTACIÓN	2
CAPITULO 1 : GENERALIDADES	3
1.1. Introducción	4
1.2. Justificación del presente trabajo	4
1.3. Análisis del microcontrolador escogido	5
1.4. Descripción de los pines del microcontrolador	6
1.5. Temporizadores del microcontrolador	9
CAPITULO 2 : HARDWARE	11
2.1. Análisis global del sistema y requerimientos generales	12
2.2. Acceso a memoria externa	14
2.2.1 Lectura en la memoria externa de programa	14
2.2.2 Lectura en la memoria externa de datos	16
2.2.3 Escritura en la memoria externa de datos	17
2.3. Hardware para comunicaciones	19
2.4. Descripción del sistema	22
2.5. Diagrama de las tarjetas	24
CAPITULO 3 : SOFTWARE	27
3.1. Interface serial de la familia MCS-51	28
3.2. Protocolo de comunicaciones	29
3.3. Programa de control en el MCS-8031	31
3.3.1 Diagramas de flujo del programa PROG	32
3.3.2 Listado del programa PROG	38

3.4.	Programa de control en la computadora	47
3.4.1	Diagramas de flujo del programa EMULA	48
3.4.2	Listado del programa EMULA	67
CAPITULO 4 : CONSTRUCCION Y PRUEBAS		75
4.1.	Construcción del prototipo	76
4.2.	Construcción de un sistema microprocesado	77
4.3.	Pruebas	77
CAPITULO 5 : CONCLUSIONES Y RECOMENDACIONES		84
5.1.	Conclusiones y recomendaciones	85
BIBLIOGRAFIA		87
ANEXOS		88
Anexo 1 :	Manual del usuario del emulador de memorias	89
	Descripción general	89
	Operación	90
Anexo 2 :	Descripción de los elementos utilizados	91
Anexo 3 :	Características de los elementos utilizados	92
Anexo 4 :	Manual de la familia MCS-51/52	128
Anexo 5 :	Distribución de los componentes en las Tarjetas 1 y 2	144
Anexo 6 :	CD con los archivos PROG y EMULA	151

RESUMEN

El propósito del siguiente trabajo es: construir un prototipo, basado en el microprocesador MCS-8031 de la familia MCS-51/52, que permita desarrollar un programa sin el uso de los programadores de memoria Eprom existentes en el mercado; mediante la emulación a una memoria Eprom.

Para su funcionamiento el prototipo debe conectarse a una computadora personal, donde reside el programa depurado para su descarga en el emulador, y posteriormente con el sistema microprocesado en desarrollo.

El prototipo se comunica con la computadora personal a través de la transmisión serial, verifica los datos grabados y emula a una memoria Eprom en diversos formatos como 2, 4 o 8 kilobytes.

La transferencia de datos está sujeta a las normas del formato INTEL de ocho bits para la transmisión/recepción de archivos.

La aplicación total se divide en dos partes que se complementan. La primera está almacenada en la memoria Eprom externa del prototipo basado en el microprocesador MCS-8031 y la segunda en la computadora personal.

El programa de control (PROG) en la memoria Eprom externa al MCS-8031 se desarrolló usando los utilitarios AVMAC51 y el AVSIM51, por ser éstos utilizados con mayor frecuencia para el desarrollo de programas para los microprocesadores de la familia MCS-51/52. El programa complemento (EMULA) que se localiza en la computadora personal, se desarrollo en QBASIC 45.

P R E S E N T A C I Ó N

El siguiente texto se ha dividido en cinco capítulos, el primero es una introducción que justifica la realización del mismo; indica como se escogió el microprocesador, la distribución de pines y descripción de los mismos, los temporizadores de que dispone y su operación.

El segundo capítulo explica el hardware, sus requerimientos, elementos a usarse, los tipos de accesos a memoria externa con su respectivo diagrama de tiempos y circuitos; las consideraciones para realizar la comunicación serial entre la computadora personal y el prototipo, como por ejemplo niveles de voltaje y estados de operación por hardware o software. Se da una descripción total del prototipo y termina indicando los diagramas circuitales.

El tercer capítulo explica el uso del interface serial del microprocesador, el protocolo de comunicaciones a usarse para la transferencia de datos; los diagramas de flujo de los programas de control que se almacenan en la memoria Eprom externa al microprocesador y el correspondiente en la computadora personal, con el listado respectivo.

El cuarto capítulo detalla el proceso para la construcción del prototipo y pruebas realizadas, explica el funcionamiento del hardware, lista el programa usado para la emulación de prueba.

El quinto capítulo incluye conclusiones y recomendaciones.

Luego están las referencias bibliográficas en que se basó este trabajo.

Finalmente se presentan los siguientes anexos: manual del usuario, descripción y características de los elementos utilizados, distribución de los componentes, etc.

CAPITULO 1

GENERALIDADES

- 1.1.- Introducción.
- 1.2.- Justificación del presente trabajo.
- 1.3.- Análisis del microcontrolador escogido
- 1.4.- Descripción de los pines del microcontrolador
- 1.5.- Temporizadores del microcontrolador

1.1 INTRODUCCIÓN

Durante los años de universidad y posteriormente en la vida profesional es necesario desarrollar sistemas basados en microprocesadores, el hardware y software tendrán algún fin en particular, sean estos didácticos o industriales. El programa se lo realiza y ejecuta en alguno de los microcontroladores, microprocesadores, memorias EPROM y programadores de memoria EPROM existentes en el mercado.

Como sabemos no siempre el programa es escrito correctamente cien por ciento, por lo que es necesario depurarlo; esto es, hacer correcciones, por cada una de estas correcciones hay que volverlo a grabar en la memoria EPROM y probarlo, lo cual es molesto, costoso y a la larga termina con la vida útil de la memoria EPROM.

El presente trabajo consiste en tener un prototipo que permita:

- Desarrollar un programa y grabarlo en una memoria RAM, que sea fácil de hacer cualquier cambio y volver a grabar, todo esto en el menor tiempo y a un costo razonable.
- Emular a una memoria EPROM en tiempo real.

Una vez que el programa desarrollado es definitivo se procederá a grabarlo una sola vez en la memoria EPROM, usando un grabador de memorias EPROM.

1.2 JUSTIFICACIÓN DEL PRESENTE TRABAJO

Se sobre entiende que el estudiante o la persona que desarrolla programas tiene un dominio alto en lo que a programación se refiere. Debe tener un conocimiento de electrónica, sistemas digitales y sistemas microprocesados.

Se da por entendido que la persona que programa tiene un conocimiento de los microcontroladores y microprocesadores de la familia MCS-51/52 de la casa INTEL. Sabe como usarlos en un sistema dado, como se los puede programar,

cual es el set de instrucciones y cuales son las operaciones que se realizan dentro del microcontrolador.

Al desarrollar este prototipo denominado "Emulador de Memorias EPROM" se pondrán en práctica los conocimientos de electrónica, sistemas digitales y sistemas microprocesados; para diseñar el hardware y software, construirlo y realizar las pruebas necesarias hasta conseguir el objetivo propuesto.

Todo esto con la finalidad de reducir el tiempo de desarrollo de programas en assembler.

Se han empleado los utilitarios AVMAC51 y el AVSIM51, por ser estos utilizados con mayor frecuencia para el desarrollo de programas para los microcontroladores de la familia MCS-51/52 .

Este prototipo, permitirá emular en tiempo real a una memoria EPROM de 2 , 4 y 8 kilobytes.

1.3 ANALISIS DEL MICROCONTROLADOR ESCOGIDO

Entre los diferentes tipos de familias de microcontroladores existentes en el mercado, se ha optado por la familia de los microcontroladores MCS-51/52, por estar más familiarizado con su uso.

Dentro de esta familia se pudo escoger el microcontrolador MCS-8751, el cual tiene incluido la memoria de programa (EPROM) dentro del mismo chip, pero dado que se va a realizar un prototipo, durante la etapa de pruebas y optimización del programa se tendrá que borrar, insertar instrucciones dentro del programa, por lo cual se tendrá que borrar y grabar nuevos datos dentro de la memoria de programa del microcontrolador. Las continuas borradas y grabadas pueden causar que el chip se deteriore, y tomando en cuenta que este tipo de microcontrolador es costoso y difícil de conseguir en nuestro medio, se ha optado por tener separado físicamente a la memoria de programas y al microcontrolador. Reponer la memoria de programas externa EPROM es más fácil y menos costosa. Por tal razón se ha optado en utilizar el microcontrolador MCS-8031, cuyas características principales son:

- CPU de 8 bits optimizado para aplicaciones de control y transmisión full duplex de datos.
- Capacidad de procesamiento booleano (lógica bit a bit).
- Espacio direccionable de 64 Kbytes para memoria de programa externa.
- Espacio direccionable de 64 Kbytes para memoria de datos externa.
- **No tiene memoria de programas en el mismo chip.**
- 128 bytes de memoria de datos en el mismo chip.
- 32 líneas bidireccionables de I/O, direccionables individualmente.
- 2 temporizadores/contadores.
- Canal full duplex para transmisión serial.
- Una estructura de interrupciones de 6 fuentes/5 vectores, con 2 niveles de prioridades.
- Oscilador construido en el mismo chip.

1.4 DESCRIPCION DE LOS PINES DEL MICROCONTROLADOR

En la figura 1.1 se indica la distribución de los pines del microcontrolador. A continuación se describen los pines utilizados en este trabajo.

Vcc : Alimentación positiva de +5 voltios DC.

Vss : Conexión a tierra (cero voltios).

Puertos : Existen 4 puertos de 8 bits bidireccionables (P0, P1, P2 y P3), esto quiere decir que pueden programarse como entrada o salida (E/S) para enviar o recibir información de los periféricos. Y cada uno es direccionable "bit a bit".

Adicionalmente están capacitados para realizar las siguientes funciones:

Puerto 0 (P0) : Multiplexa en el tiempo por sus 8 líneas (P0.1 hasta P0.7) la parte baja del bus de direcciones (A0 hasta A7) durante el acceso a la memoria externa de programas y datos, y el bus de datos (D0 hasta D7),

Esto es, primero se tiene la parte baja del bus de direcciones y luego el bus de datos.

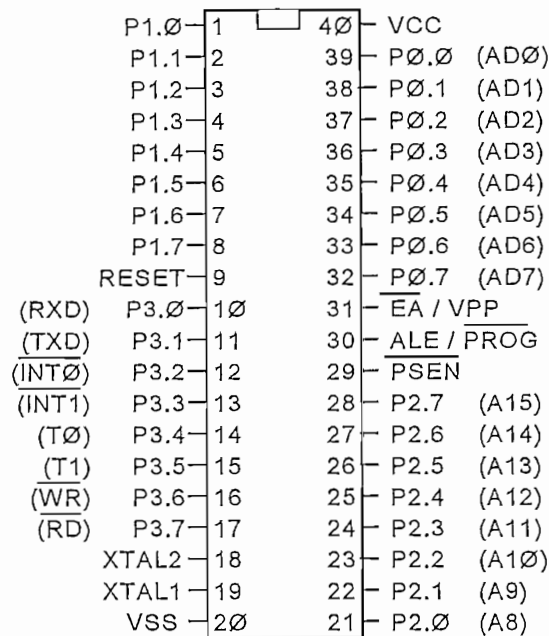


Figura 1.1 Distribución de pines del microcontrolador

Puerto 2 (P2) : Emite la parte alta del bus de direcciones (A8 hasta A15) en los accesos de memoria externa (memoria de programa) cuando utilizan 16 bits de dirección y en los accesos a la memoria de datos que usa también 16 bits de dirección (MOVX @DPTR).

ALE : ALE (Address Latch Enable) es un pulso que emite el microcontrolador para indicar que en el puerto 0 se encuentra la parte baja del bus de direcciones en el acceso a la memoria externa. Generalmente se la usa como señal de habilitación a un LATCH externo de ocho bits con lo cual se retiene esta parte del bus de direcciones. ALE se emite con una frecuencia de 1/6 de la frecuencia de emisión del reloj.

PSEN : (Program Store Enable) esta señal autoriza leer en la memoria de programa externo. PSEN no se activa cuando se esta ejecutando el programa de la ROM o EPROM interna.

\overline{EA} : (External Access), cuando se mantiene a nivel alto, se ejecuta sólo el programa de la ROM interna. Si EA se mantiene a nivel bajo, se ejecuta el programa de la memoria externa.

Es decir : $\overline{EA}=1$ actúa como microcontrolador.

$\overline{EA}=0$ actúa como microprocesador.

RXD : Comúnmente (según modos) entrada del puerto serie.

TXD : Comúnmente (según modos) salida del puerto serie.

$\overline{INT0}$: Entrada de la interrupción externa 0.

$\overline{INT1}$: Entrada de la interrupción externa 1.

T0 : Entrada externa del temporizador/contador (Timer 0).

T1 : Entrada externa del temporizador/contador (Timer 1).

WR : Señal que, cuando está en nivel bajo, autoriza la escritura en dispositivos externos, por ejemplo la memoria externa de datos.

RD : Señal que, cuando está en nivel bajo, autoriza la lectura en dispositivos externos, por ejemplo la memoria externa de datos.

XTAL1 y XTAL2 : son la entrada y salida, respectivamente, de un amplificador inversor que puede ser configurado para su uso como chip oscilador. Se puede usar indistintamente un cristal de cuarzo o un resonador cerámico.

RESET : Manteniendo este pin a un nivel alto durante cierto tiempo, el CPU responde generando un RESET interno y ejecuta el algoritmo del RESET interno, el cual inicializa el sistema, esto es escribe ceros o unos lógicos en algunos registros.

1.5 TEMPORIZADORES DEL MICROCONTROLADOR

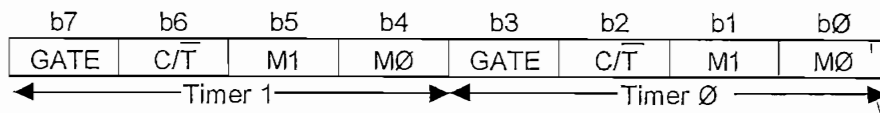
El microcontrolador MCS-8031 tiene dos registros temporizadores-contadores denominados Timer 0 (T0) y Timer 1 (T1), los cuales pueden ser configurados para que operen como temporizadores o contadores.

Los Timers actúan como temporizadores cuando el registro contador se incrementa cada ciclo de máquina; mientras que, si actúan como contadores, el registro es incrementado en respuesta a la transición del nivel alto de tensión al nivel bajo de la señal externa aplicada al pin 14 para el contador T0 y al pin 15 para el contador T1.

El registro contador son pares de registros de 16 bits, se denominan TH1 y TL1 para el Timer 1 y TH0 y TL0 para el Timer 0.

Para trabajar con los Timers se usa dos registros: TMOD y TCON.

Registro TMOD : Selecciona el Timer 0 o 1, no es direccionable bit a bit .



Si GATE=1 el control es por hardware y si GATE=0 el control es por software.

Si C/T=1 el Timer actúa como contador y si C/T=0 actúa como temporizador.

M1 y M0 determinan el modo de operación del Timer de acuerdo a la siguiente tabla:

MODO	M1	M0	MODO DE OPERACIÓN
0	0	0	Temporizador de 13 bits
1	0	1	Temporizador/contador de 16 bits
2	1	0	Temporizador/contador de 8 bits con auto-recarga
3	1	1	Contadores múltiples específicos

Registro TCON : es direccionable bit a bit y controla el modo de operación de los Timers 0 y 1 en relación con las interrupciones y flancos de activación de las mismas.

b7	b6	b5	b4	b3	b2	b1	b0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

El bit IT controla la interrupción externa, si es igual a 0 es activada por nivel bajo y si es igual a 1 es activada por flanco de bajada.

El bit IE es la bandera de interrupción para la interrupción externa, se pone a 1 cuando se detecta interrupción externa y se repone automáticamente al atender la interrupción por flanco.

TR habilita o deshabilita al Timer ya sea si es 1 o 0, respectivamente.

TF es la bandera de sobrepasamiento del Timer, se repone automáticamente al atender la interrupción.

CAPITULO 2

H A R D W A R E

- 2.1 Análisis global del sistema y requerimientos generales.
- 2.2 Acceso a memoria externa.
 - 2.2.1 Lectura en la memoria externa de programa.
 - 2.2.2 Lectura en la memoria externa de datos.
 - 2.2.3 Escritura en la memoria externa de datos.
- 2.3 Hardware para comunicaciones.
- 2.4 Descripción del sistema.
- 2.5 Diagrama de las tarjetas.

2.1 ANÁLISIS GLOBAL DEL SISTEMA Y REQUERIMIENTOS GENERALES

El emulador de memorias EPROM propuesto está conformado por 2 tarjetas. Ver figura 2.1

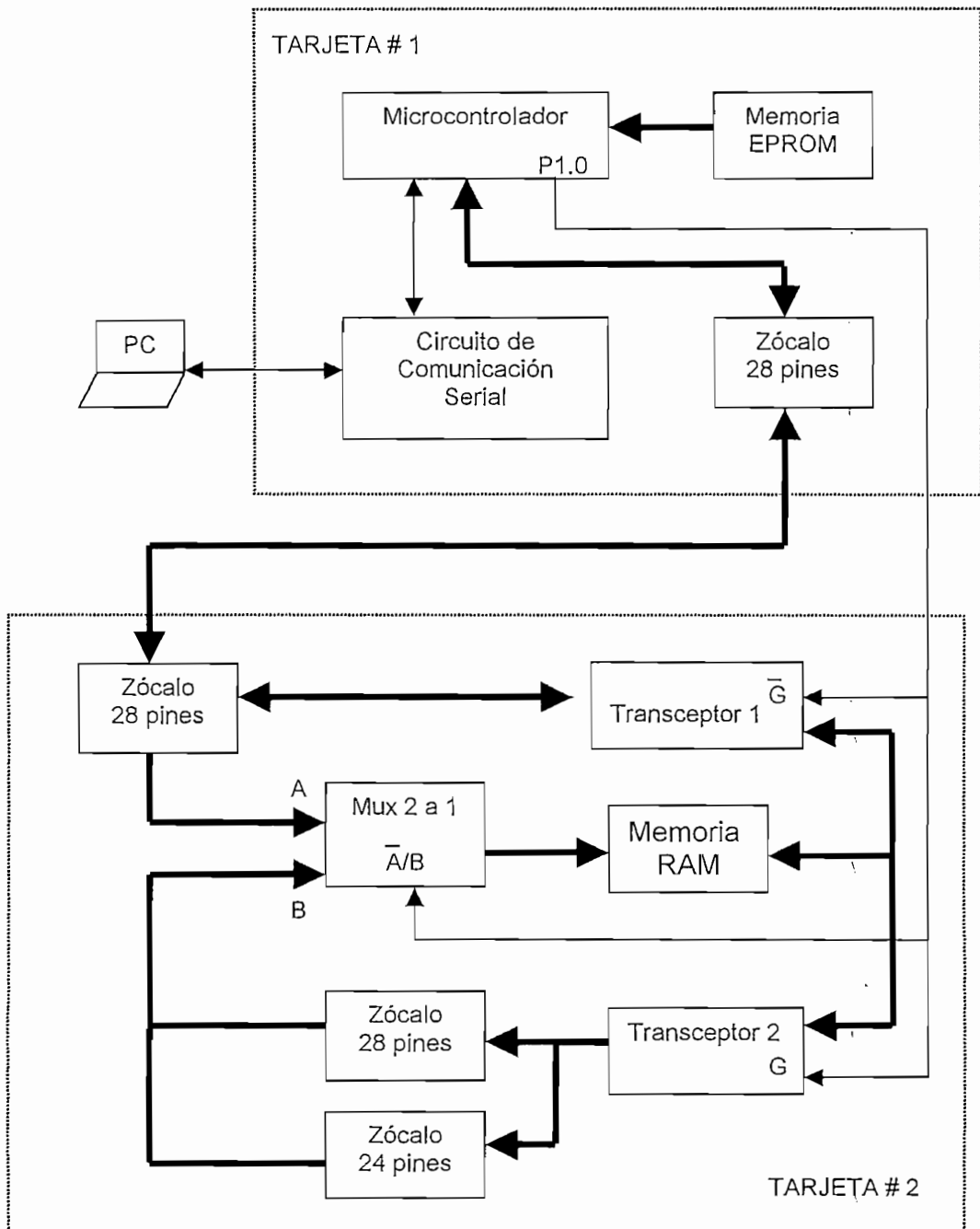


Figura 2.1 Circuito emulador

La Tarjeta # 1, que se enlaza directamente con el PC, está conformada por un microcontrolador, una memoria EPROM, un circuito de comunicación serial y un zócalo.

La memoria EPROM contendrá el programa que permitirá:

- Revisar si todas y cada una de las celdas de la memoria RAM están en buen estado.
- Escoger el tipo de memoria EPROM a emular.
- Verificar datos y grabarlos en la memoria RAM de la Tarjeta # 2, todo esto con ayuda de un microcontrolador.

El circuito de comunicación serial a más de permitir el intercambio de información, sirve como interfaz entre la computadora personal y la Tarjeta # 1, ya que los niveles de voltaje que manejan son diferentes. La comunicación serial entre la computadora personal y el circuito emulador debe ser en ambos sentidos, por cuanto es necesario confirmar si la información recibida es correcta.

Ambas tarjetas poseen un zócalo de 28 pines, que a través de un cable plano quedan conectadas.

En la Tarjeta # 2 se tiene la memoria RAM, denominada emu-eprom, que contendrá el programa a ser emulado.

La información a ser grabada en la memoria RAM, será recibida a través del circuito de comunicación serial. Razón por la cual es necesario que la Tarjeta # 1 se conecte con una computadora personal, para que ésta pueda descargar por su pórtico serial todos los códigos de máquina del programa a ser emulado hacia el emulador de memorias EPROM.

Se tiene un bloque conformado por varios multiplexores 2 a 1 y dos transceptores, que permiten dos opciones: primero grabar en la RAM el programa a emular y posteriormente su emulación. La señal de control que permite escoger una u otra opción, se genera a través del pin P1.0 del microcontrolador. En la primera opción, el transceptor 1 permite leer y grabar en la RAM, mientras que en la segunda opción, el transceptor 2 permite solo leerla. Es importante notar que solo uno de los transceptores se habilita a la vez.

Una vez que el programa a ser emulado ha sido grabado en la memoria RAM, el circuito emulador se conecta con el sistema microprocesado a través de un zócalo de 28 pines si va a emular una memoria Eprom 2764, o de 24 pines si es una Eprom 2732 o 2716.

La comunicación entre el circuito emulador y el sistema microprocesado es en forma paralela y en un solo sentido. Ver figura 2.2.

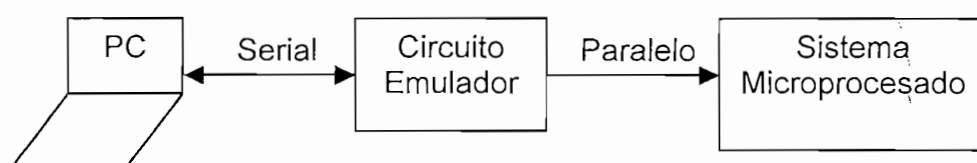


Figura 2.2 Sistema completo

Según la figura 2.1, el microcontrolador debe acceder a dos tipos de memoria: EPROM y RAM, razón por la cual a continuación se explicará con diagrama de tiempos y circuitos como se puede acceder a estos dos tipos de memoria.

2.2 ACCESO A MEMORIA EXTERNA

Para acceder a la memoria externa, sea ésta de programa (EPROM) o de datos (RAM), es necesario colocar permanentemente en nivel bajo a la señal \overline{EA} , por cuanto el microcontrolador, el 8031, no posee memoria ROM interna. Adicionalmente, el puerto 2 mantiene durante todo el ciclo de lectura o escritura la parte superior del bus de direcciones (A8 a A15).

2.2.1 LECTURA EN LA MEMORIA EXTERNA DE PROGRAMA

En la figura 2.3 se muestra el diagrama de tiempos para un ciclo de lectura de la memoria externa de programa (EPROM), se puede apreciar que: si el puerto 0 tiene la parte baja del bus de direcciones (A0 a A7), la señal ALE es

activada y en el flanco negativo permite retener el bus bajo de direcciones, luego se activa la señal PSEN para la lectura del código correspondiente.

La figura 2.4 muestra el circuito que permite tener acceso a la memoria externa de programa (EPROM).

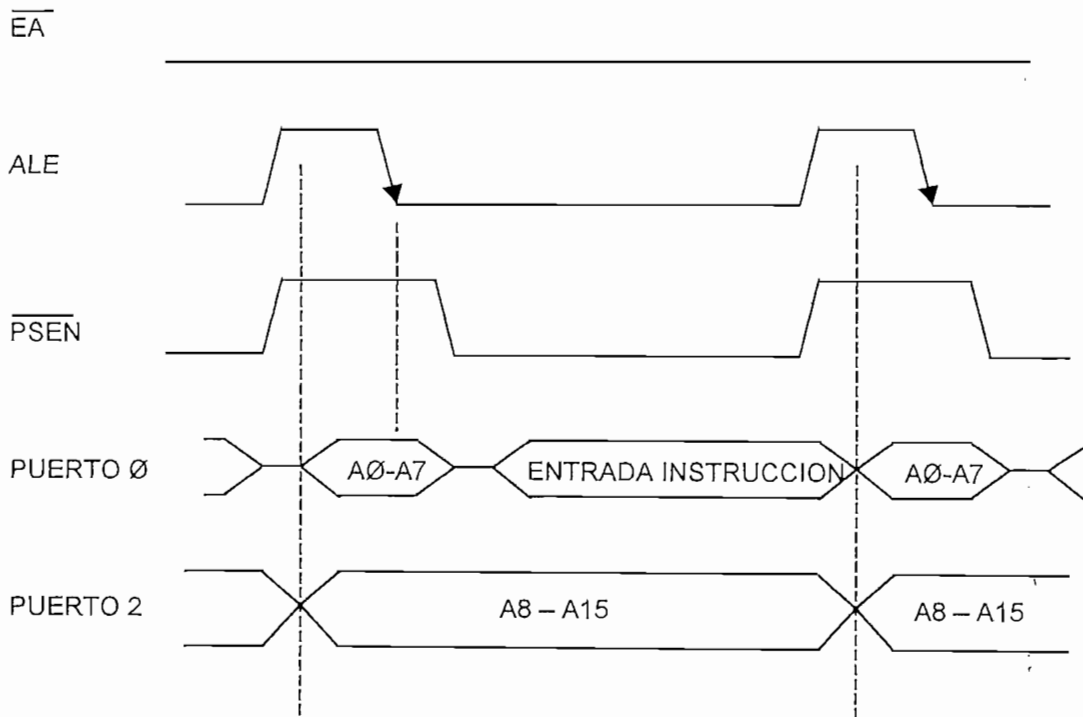


Figura 2.3 Ciclo de lectura de la memoria externa de programa

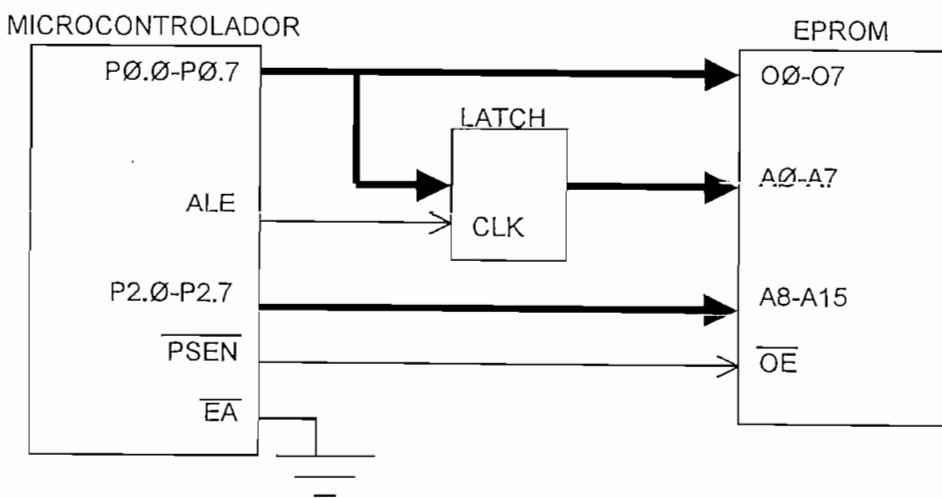


Figura 2.4 Circuito para lectura de la memoria de programa externa

2.2.2 LECTURA EN LA MEMORIA EXTERNA DE DATOS

La figura 2.5 muestra el diagrama de tiempos para un ciclo de lectura de la memoria externa de datos (RAM), se puede apreciar que:

Cuando el puerto \emptyset tiene la parte baja del bus de direcciones (A_0 a A_7), la señal ALE es activada y en el flanco negativo permite retener el bus bajo de direcciones, la señal $\overline{\text{PSEN}}$ permanece deshabilitada, esto es en nivel alto, por cuanto se la usa solo para acceder a la memoria externa de programa.

Para leer en la memoria externa de datos se utiliza la señal de control $\overline{\text{RD}}$, el byte de entrada es aceptada por el puerto \emptyset justo antes de que la señal de control $\overline{\text{RD}}$ que autoriza la lectura sea desactivada.

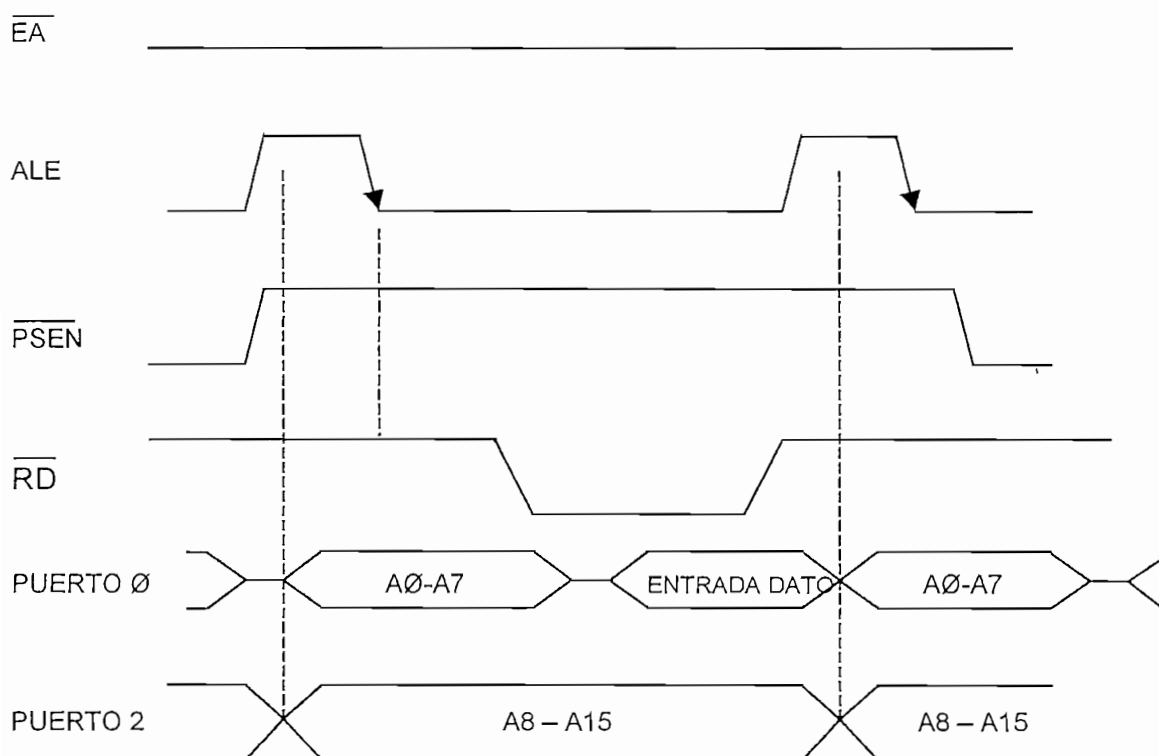


Figura 2.5 Ciclo de lectura de la memoria externa de datos

2.2.3 ESCRITURA EN LA MEMORIA EXTERNA DE DATOS

La figura 2.6 muestra el diagrama de tiempos para un ciclo de escritura de la memoria externa de datos (RAM), se puede apreciar que:

Cuando el puerto \emptyset tiene la parte baja del bus de direcciones (A_0 a A_7), la señal ALE es activada y en el flanco negativo permite retener el bus bajo de direcciones, la señal $\overline{\text{PSEN}}$ permanece deshabilitada, esto es en nivel alto, por cuanto se la usa solo para acceder a la memoria externa de programa.

Para escribir en la memoria externa de datos se utiliza la señal de control WR, el byte de entrada debe permanecer sobre el puerto \emptyset antes y después de que la señal de control de escritura WR sea desactivada.

En la figura 2.7 se muestra el circuito que permite leer o escribir en la memoria externa de datos (RAM).

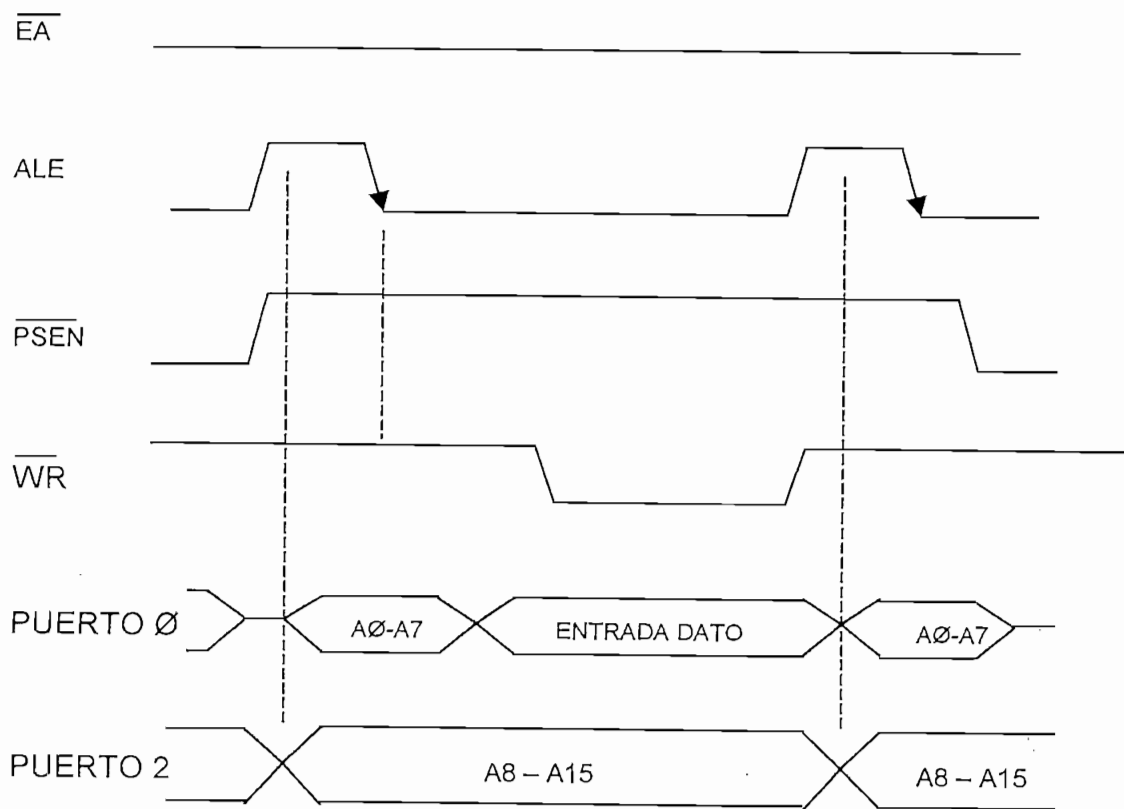


Figura 2.6 Ciclo de escritura de la memoria externa de datos

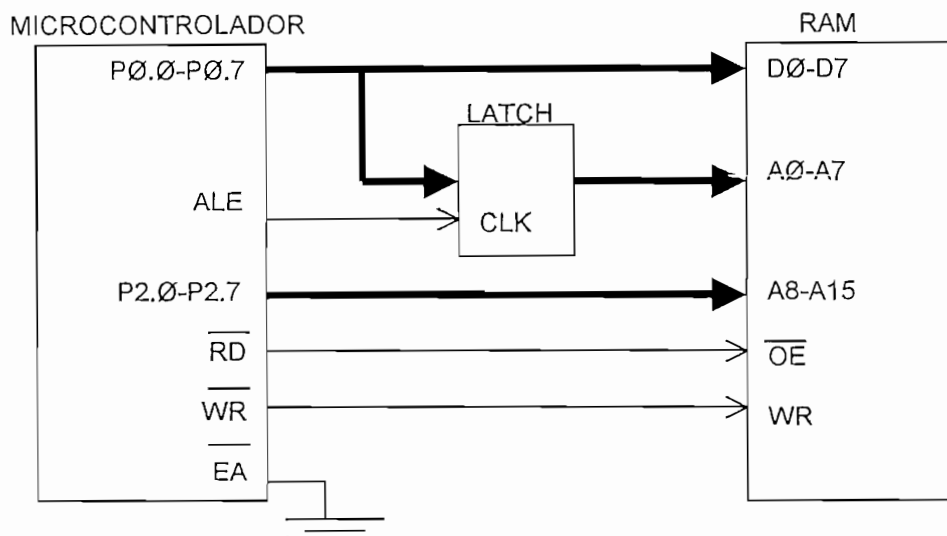


Figura 2.7 Circuito para lectura-escritura de la memoria de datos externa

La figura 2.8 muestra el circuito que permite al microcontrolador acceder a los dos tipos de memoria: EPROM y RAM.

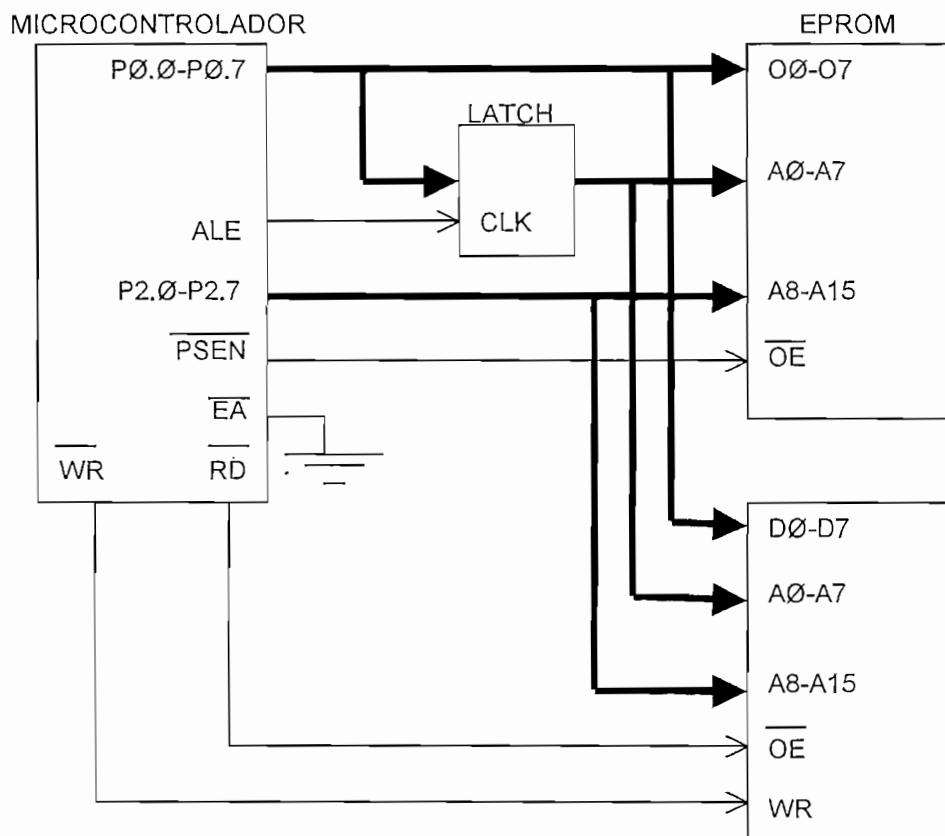


Figura 2.8 Circuito para acceder a la memoria externa de programa y de dato

2.3 HARDWARE PARA COMUNICACIONES

Todo elemento de la familia de microcontroladores MCS-51 tiene la posibilidad de mantener una comunicación con cualquier elemento externo siempre y cuando ambos utilicen el mismo formato de comunicación serial.

El microcontrolador dispone de un puerto serial el cual puede trabajar en el modo full-duplex, esto es, que puede transmitir y recibir simultáneamente.

A los registros de transmisión y recepción del puerto serial se puede acceder por un registro único denominado SBUF.

En el presente trabajo, la comunicación serial se realiza entre la computadora personal y el microcontrolador, el primero utiliza niveles de voltaje diferentes a los del segundo, por lo cual hay que utilizar un interfaz que permita acopiarlos.

Los niveles de voltaje en el microcontrolador de la familia MCS-51 para la comunicación serial utilizan la lógica TTL positiva, esto es:

- Un 0 lógico se representa por un voltaje que puede estar en el rango de [0; 0.8] voltios.
- Un 1 lógico se representa por un voltaje que puede estar en el rango de [2.4; 5] voltios.
- El rango de [0.8; 2.4] voltios es una zona de degeneración de los niveles lógicos.
- Si ningún carácter se transmite (estado permanente o de reposo), encontraremos un voltaje correspondiente a 1 lógico en el canal de transmisión.

Para la comunicación serial de información, la mayoría de computadoras disponen de un puerto serial en base a un conector de 25 pines, el cual está diseñado en base de la norma de comunicación serial EIA RS-232C :

- Los niveles de voltaje que representan al 0 y 1 lógicos transmitidos ya no son TTL.

- Un 0 lógico se representa por un voltaje que puede estar en el rango de [3; 25] voltios.
- Un 1 lógico se representa por un voltaje que puede estar en el rango de [-3; -25] voltios.
- El rango de [-3; 3] voltios es una zona de transito o ambigüedad.
- Si ningún carácter se transmite (estado permanente o de reposo), encontraremos un voltaje correspondiente a 1 lógico en el canal de transmisión.

Para la comunicación serial, la mayoría de las computadoras personales, operan dentro de los niveles +/- 9 voltios hasta +/- 12 voltios.

En la figura 2.9 se muestra el circuito interfaz entre el PC y el microcontrolador.

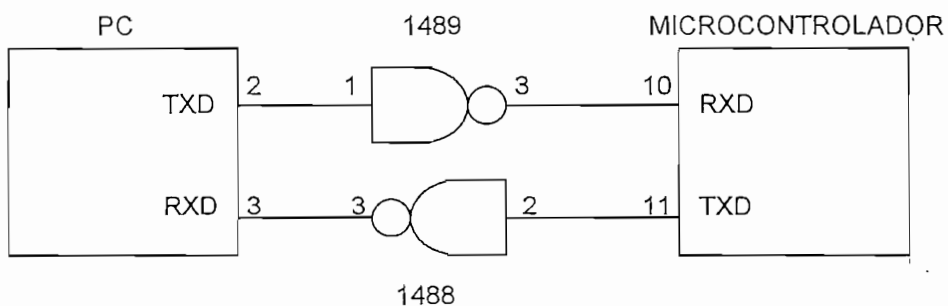


Figura 2.9 Circuito para comunicación serial

Para implementar este circuito, se ha utilizado dos chips de la Motorola:

- El MC 1489 que permite la conversión de niveles de la norma RS-232-C a TTL
- y el MC 1488 que hace la conversión de TTL a RS-232-C.

Toda computadora personal posee un conector para la comunicación serial, el cual puede ser de 25 pines (DB25) o de 9 pines (DB9) como se indica en la figura 2.10

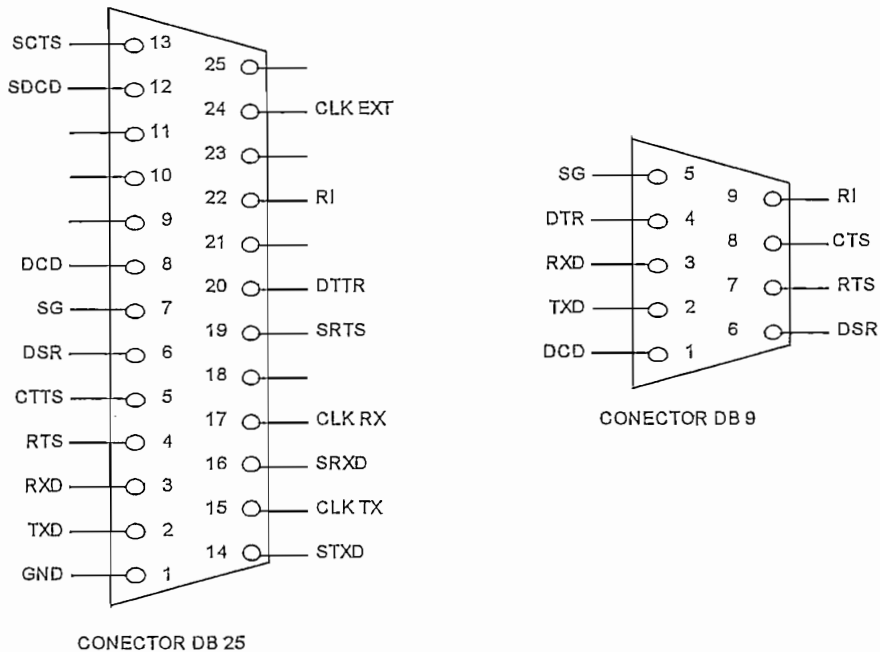


Figura 2.10 Conectores DB

Todos estos pines tienen una función específica, sin embargo en la mayoría de las aplicaciones, solo un número limitado de estos pines son utilizados.

La norma indica que el transmisor (computadora personal) debe poseer un conector macho y el receptor (Tarjeta # 1) un conector hembra.

Las señales de estos conectores se dividen en cuatro bloques: datos, control, sincronismo y tierras.

Para establecer la comunicación serial, el transmisor de datos debe conocer cuando el receptor está listo, estas indicaciones del estado de operación tanto del receptor como del transmisor son conocidas como Handshake, y que son de dos tipos: por hardware o por software.

En la comunicación serial por medio del protocolo RS232 las líneas DSR, DTR y RTS son dedicadas al Handshake por hardware, mientras que en el caso del Handshake por software es necesario transmitir caracteres especiales a través de las líneas de datos; los protocolos más comunes son:

XON/XOFF : El carácter ASCII 13H es enviado por el receptor al transmisor cuando se requiere parar la transmisión, y el carácter ASCII 11H cuando se quiera reiniciar la transmisión.

ETX/ACK : Es el método de Transmisión/Reconocimiento, en el cual los datos se envían en bloques de longitud fija, después de estos el dispositivo transmisor envía un carácter ASCII 3H (EXT End of transmission). El receptor por su parte recibe el bloque transmitido sin errores enviando el carácter ASCII 6H, en el caso de tener errores en el bloque recibido el receptor envía el carácter ASCII 21H para indicar al transmisor que se debe enviar nuevamente el bloque anterior.

De todas los pines que tiene este conector, se ha empleado las siguientes:

El pin 2 TXD y el pin 3 RXD, que se usan para transmitir y recibir , respectivamente. Ambas señales deben permanecer en 1L cuando no hay intercambio de información.

El pin 7 SG es un potencial de referencia es decir una línea de retorno para todas las señales excepto GND.

2.4 DESCRIPCION DEL SISTEMA

El microcontrolador escogido es el MCS-8031, el cual necesita de la presencia de una memoria externa de programa, la Eprom 2764 de la casa Intel puede almacenar hasta 8 kilobytes de información, en ella reside el programa que controla la recepción, grabación en la RAM y verificación del programa a emular.

En la Tarjeta # 2 se puede ver que la memoria RAM, los multiplexores y transceptores reciben energía de alimentación desde fuentes diferentes:

- Desde la Tarjeta # 1
- Desde un banco de baterías
- Desde el Sistema Microprocesado 1 o 2

La desventaja de trabajar con una RAM que va a emular a una EPROM, es que si se corta la energía de alimentación todos los datos que contiene la RAM se pierden. Para evitar este problema, la Tarjeta # 2 cuenta con un banco de baterías auxiliares como fuente de alimentación DC. De esta manera los datos permanecen en la RAM hasta que se restablezca la alimentación normal.

Cuando la Tarjeta # 2 esta trabajando con la alimentación de la red, el banco de baterías esta desconectada; pero si se pierde la energía de la red o si ésta decae a un valor crítico entra a funcionar las baterías alimentando principalmente a la memoria RAM.

La presencia de los diodos de silicio 1N4007 es permitir que una de las fuentes de alimentación sea la que provea de energía a los chips.

En la figura 2.11 se muestra el circuito que provee a la RAM de energía ininterrumpida.

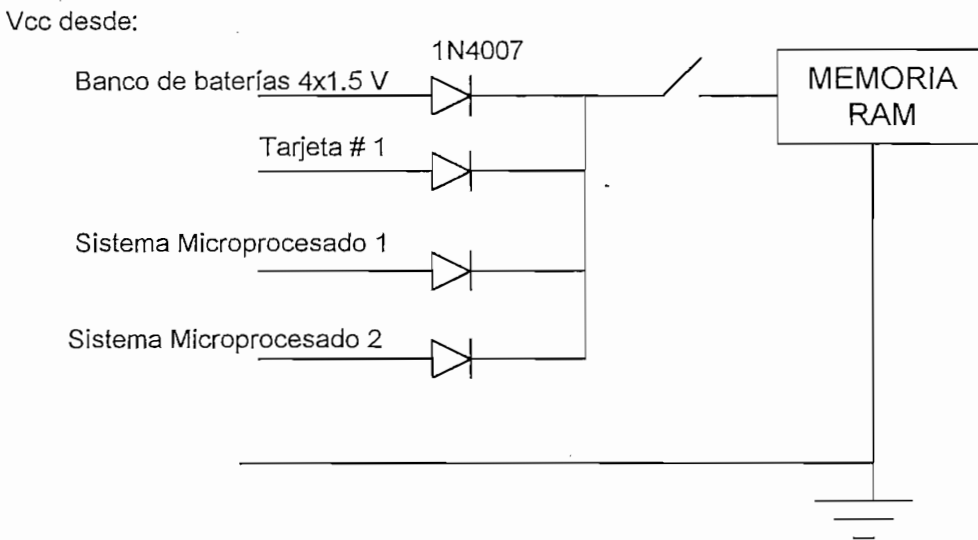


Figura 2.11 Fuente de alimentación ininterrumpida DC

Es importante anotar que la tierra es común para todos los elementos de las tarjetas involucradas.

La Tarjeta # 2 posee dos zócalos, uno de 24 pines y otro de 28 pines. El primero es para cuando se va a emular a una EPROM de 2 o de 4 kilobytes, mientras que el segundo para cuando se va a emular una EPROM de 8 kilobytes.

La figura 2.12 muestra el circuito que permite acceder a la memoria RAM. Si el microprocesador 8031 va a leer y grabar en la RAM, el pin P1.0 debe estar en nivel bajo, y así habilitar solo el transceptor 1. Cuando la señal OE está a nivel bajo se lee la RAM y en nivel alto se graba en ella. El sistema microprocesado

solo puede leer la RAM, para esto el pin P1.0 debe estar en nivel alto y el pin DIR del transceptor 2 en nivel bajo. Se ha escogido como transceptor al 74245. La señal de control a través del pin P1.0 también permite a los multiplexores 74157 dejar pasar el bus de direcciones y señales de control desde la Tarjeta # 1 o desde el Sistema Microprocesado.

Para la emulación, el Sistema Microprocesado lee los códigos de máquina desde la RAM.

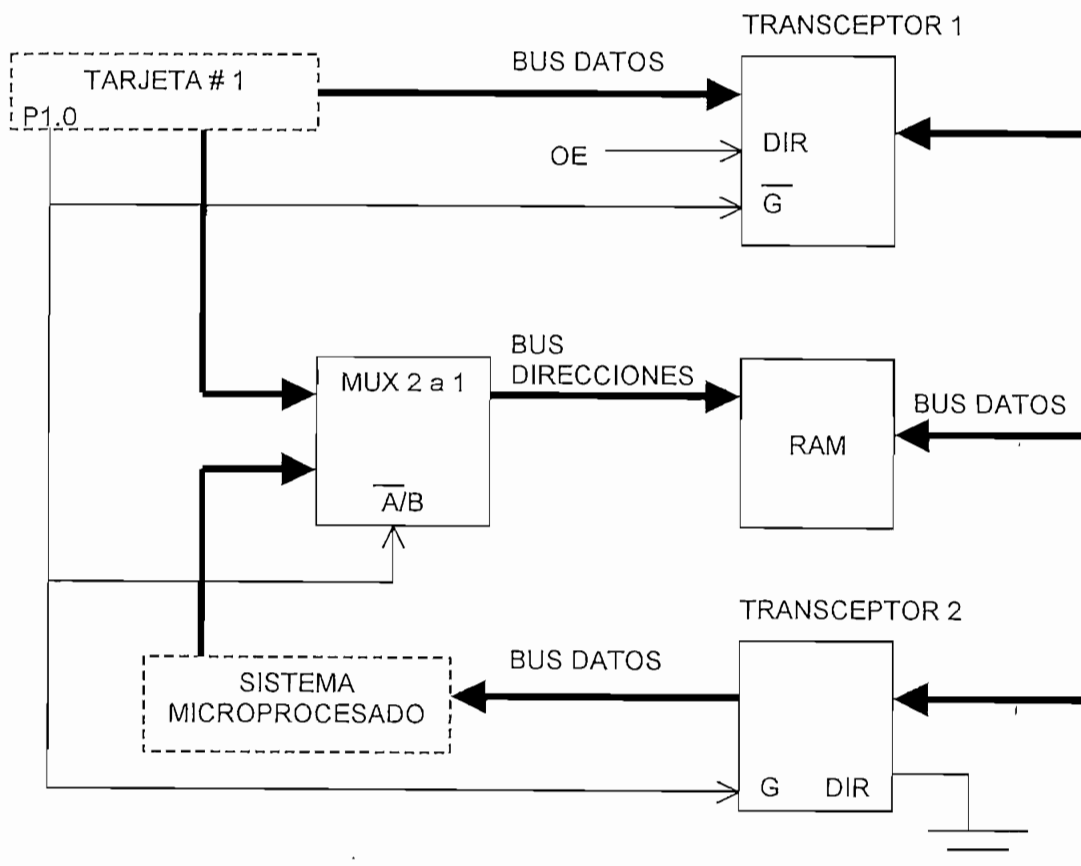


Figura 2.12 Acceso a la memoria RAM

2.5 DIAGRAMA DE LAS TARJETAS

En las figuras 2.13 y 2.14 se muestran los diagramas de las Tarjetas # 1 y # 2.

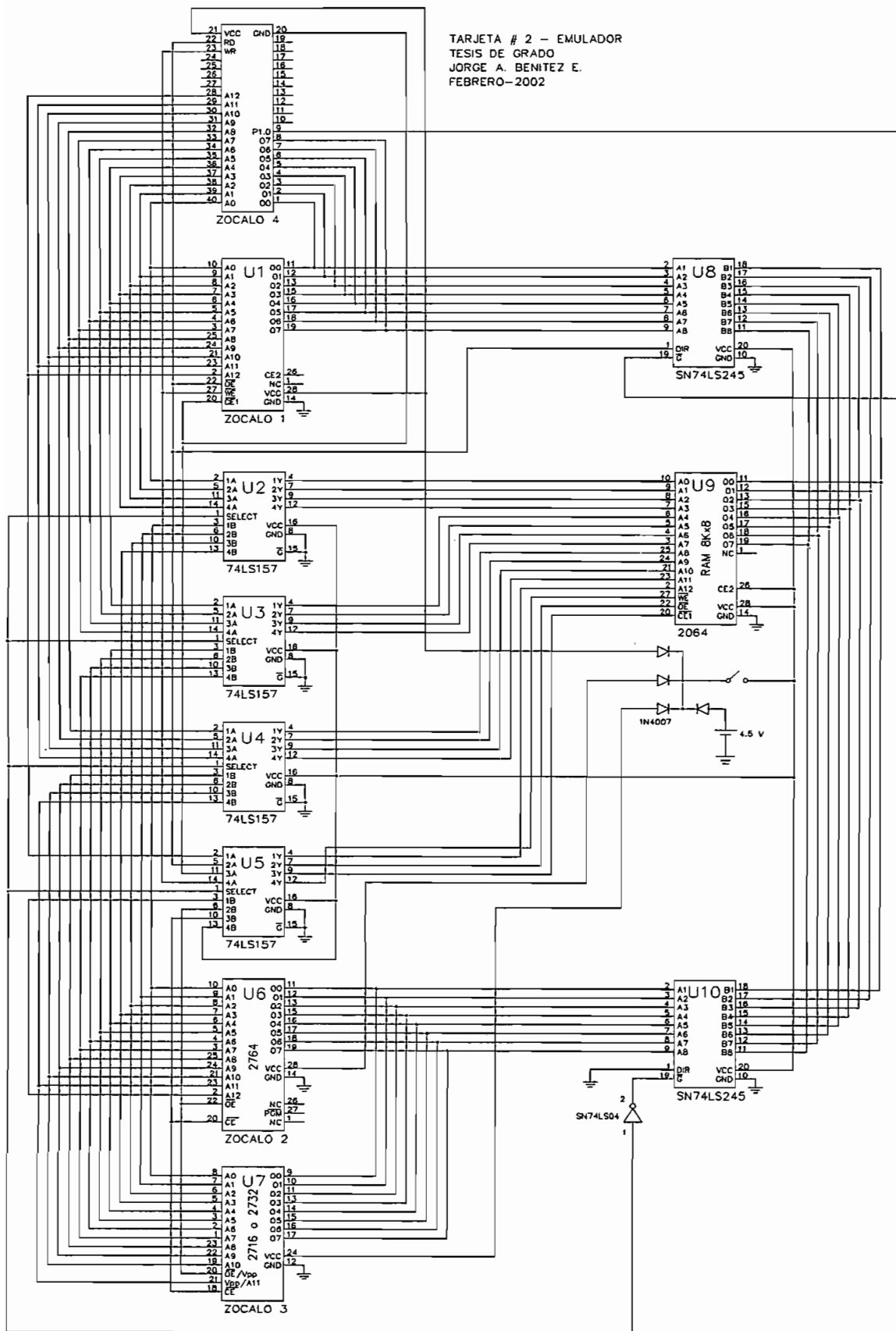


Figura 2.14 Diagrama general de la Tarjeta # 2

CAPITULO 3

SOFTWARE

3.1.- Interface serial de la familia MCS-51

3.2.- Protocolo de comunicaciones

3.3.- Programa de control en el MCS-8031

3.3.1 Diagramas de flujo del programa PROG

3.3.2 Listado del programa PROG

3.4.- Programa de control en la computadora

3.4.1 Diagramas de flujo del programa EMULA

3.4.2 Listado del programa EMULA

3.1 INTERFACE SERIAL DE LA FAMILIA MCS-51

Al puerto serial del microprocesador se lo puede programar para que trabaje de diferentes maneras y a diferente velocidad, para lo cual se tiene el registro SCON.

Registro SCON : es direccionable bit a bit y controla el modo de trabajo del puerto serial

b7	b6	b5	b4	b3	b2	b1	b0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Los bits b7 y b6 (SM0 y SM1) permiten escoger el modo de trabajo (Modo 0, 1, 2 y 3) y de selección de velocidad de comunicación.

Los bits b0 y b1 (RI y TI) son banderas de interrupción de la recepción y transmisión respectivamente, se activan por hardware y deben ser desactivadas por hardware.

En este trabajo, para la comunicación serial, se escogió como modo de trabajo el Modo 1, por estar más acorde con los estándares habituales.

En esta opción, 10 bits se transmiten a través del pin 11 (TXD) o se reciben del pin 10 (RXD) de la siguiente manera:

- 1 bit de inicio (0L) que señala el comienzo de la palabra de datos
- 8 bits de datos, primero el LSB, y
- 1 bit de parada (1L) que señala el fin de la palabra de datos.

Adicionalmente, la velocidad de transmisión, en baudios, puede ser ajustada por el usuario en un amplio rango.

En el caso del microcontrolador 8031, los baudios vienen dados por la razón de sobrepasamiento del Timer 1 y también dependen del valor del bit b7 (SMOD) del registro PCON .

Este registro PCON que es direccionable bit a bit, permite configurar los modos de trabajo de bajo consumo y de asignar parcialmente la velocidad de comunicación del puerto serial en los modos 1, 2 y 3

Si SMOD es igual a 0 o 1, los baudios son 1/64 o 1/32 de la frecuencia de oscilación del microcontrolador, respectivamente.

El timer 1 se ha configurado como temporizador en modo aut recarga (TMOD=20H). En este caso los baudios vienen dados por la fórmula:

$$\text{Baudios} = \frac{2^{\text{SMOD}}}{32} \cdot \frac{\text{frecuencia oscilador}}{12 \cdot (256 - (\text{TH1}))}$$

Por lo general, se conoce la velocidad en baudios y lo que se requiere es conocer el valor de recarga del timer TH1, el cual despejado de la fórmula anterior da:

$$\text{TH1} = 256 - \frac{2^{\text{SMOD}}}{32} \cdot \frac{\text{frecuencia oscilador}}{12 \cdot \text{baudios}}$$

Considerando SMOD=0, frecuencia del oscilador=12 Mhz y la velocidad de comunicación de 2400 baudios, se tiene:

$$\text{TH1} = 256 - 13 = 243$$

$$\text{TH1} = \text{F3H}$$

La transmisión comienza escribiendo el dato en el registro SBUF y termina cuando se activa (=1L) por hardware la bandera TI.

La recepción inicia cuando REN=1 y se detecta un flanco descendente en RXD y termina cuando se activa (=1L) por hardware la bandera RI.

En cualquier caso, antes de transmitir o recibir otro dato, es necesario desactivar (0L) por software las banderas TI y RI, respectivamente.

3.2 PROTOCOLO DE COMUNICACIONES

Para que dos dispositivos puedan comunicarse es necesario tener un protocolo

preestablecido por una norma determinada.

Para la comunicación serial existen varios tipos de protocolos con sus respectivas normas, se ha escogido las normas del formato INTEL de ocho bits para la transmisión y recepción de archivos.

La estructura en el formato Intel de 8 bits es la que se indica a continuación:

- El carácter de cabecera se representa por los dos puntos " : " (3AH) y es señal del comienzo de un mensaje.
- La longitud del mensaje es representada por dos dígitos hexadecimales que indican el número de bytes de datos del mensaje.
- La dirección de partida representada por cuatro dígitos hexadecimales que indican la localidad de memoria desde la cual debe grabarse el primer dato del mensaje.
- El tipo de mensaje esta representado por dígitos hexadecimales , si es 00 indica mensaje de datos y si es 01 fin de mensaje.
- El checksum representado por dos dígitos hexadecimales y que es el complemento de dos de la suma módulo 256 de todos los bytes precedentes (excepto " : ")

Por ejemplo, el mensaje en formato Intel en hexadecimal puede ser:

:1403700002AF1203401202A3E5F0540F1202AF120340223218

en donde:

:	es la cabecera
14	número de datos (hex) en el mensaje
0370	dirección donde debe almacenarse el primer dato
00	mensaje de datos
02... 32	datos
18	checksum

Se debe recalcar que esto es en hexadecimal, pero por el puerto serial de la computadora personal salen los datos en formato ASCII , siguiendo con el ejemplo anterior, el mensaje a través del puerto serial del PC será:

3A313430333730303030324146313230333430313230324133453546303534304631

3230324146313230333430323233323138

Estos códigos ASCII son los que el microprocesador está recibiendo, razón por la cual hay que hacer una decodificación para transformar los códigos ASCII a dígitos hexadecimales. De igual manera, los datos procesados y almacenados por el microprocesador están en hexadecimal, para enviarlos a la computadora personal es necesario transformarlos de hexadecimal a código ASCII.

3.3 PROGRAMA DE CONTROL EN EL MCS-8031

Después de resetear el microprocesador, el banco de registros por defecto es el banco 0. Para seleccionar uno de los otros tres bancos debe hacerse por software escribiendo en el registro de estado PSW.

Del banco 0 se ha utilizado los registros desde R0 hasta R7, alguno de ellos tienen doble función, según se indica en la descripción siguiente:

R0 : contiene la dirección de la RAM interna en la cual está, primero la parte alta de la capacidad de la emu-eprom y luego la parte alta del número de localidades usadas.

R1 : contiene la parte alta de la dirección desde la cual se almacenan los datos puros en la emu-eprom, según el tipo de memoria Eprom a emular,

R2 : contiene el checksum que envía el PC, luego es usado como contador.

R3 : contiene la suma de los datos recibidos de cada línea, excepto " : " y posteriormente el complemento de dos (checksum).

R4 : contiene el número, en hexadecimal, de datos puros.

R5 : contiene la parte alta de la dirección desde la cual se almacenan los datos puros de cada mensaje.

R6 : contiene la parte baja de la dirección desde la cual se almacenan los datos puros de cada mensaje.

R7 : primero contiene el tipo de mensaje y luego el dato puro en hexadecimal a guardar en la emu-eprom.

El programa PROG consta de un Programa Principal y de las siguientes subrutinas:

- ASCHEX : Transforma un dato de código ASCII a hexadecimal.
- ERRORTIPOEPROM : Envía el carácter 26H para indicar al PC un error en el tipo de eprom recibido y termina ejecución del programa.
- ESPERA11H : Entra en un lazo hasta recibir el carácter 11H.
- HEXASC : Transforma un dato de código hexadecimal a ASCII.
- MAL : Envía el carácter 25H para indicar al PC que una localidad de la emu-eprom esta en mal estado y termina ejecución del programa.
- RETX : Envía al PC el dato recibido, para confirmación.
- TXDATO : Transmite un dato al PC.
- TXRX : Ordena al PC transmitir un dato y espera por él.
- TXNUMRAM : Transmite al PC, primero la parte alta de la capacidad de la emu-eprom y posteriormente el número de localidades usadas por el archivo .HEX en la emu-eprom.
- TXBYTE : Transmite un byte al PC.

3.3.1 DIAGRAMAS DE FLUJO DEL PROGRAMA PROG

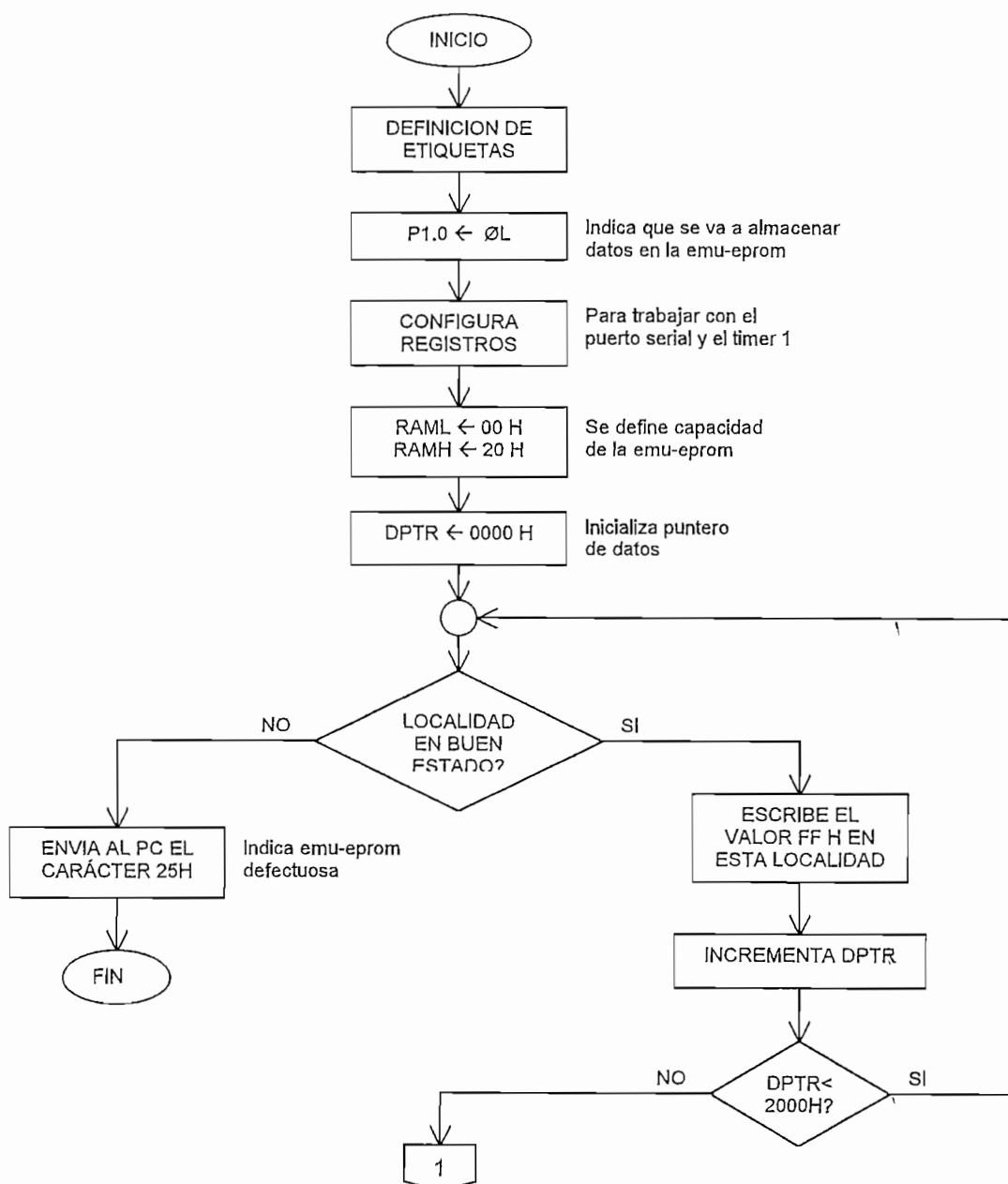
Para entender y visualizar de mejor manera el programa PROG y los saltos a las subrutinas, se presentan los diagramas de flujo con el comentario respectivo.

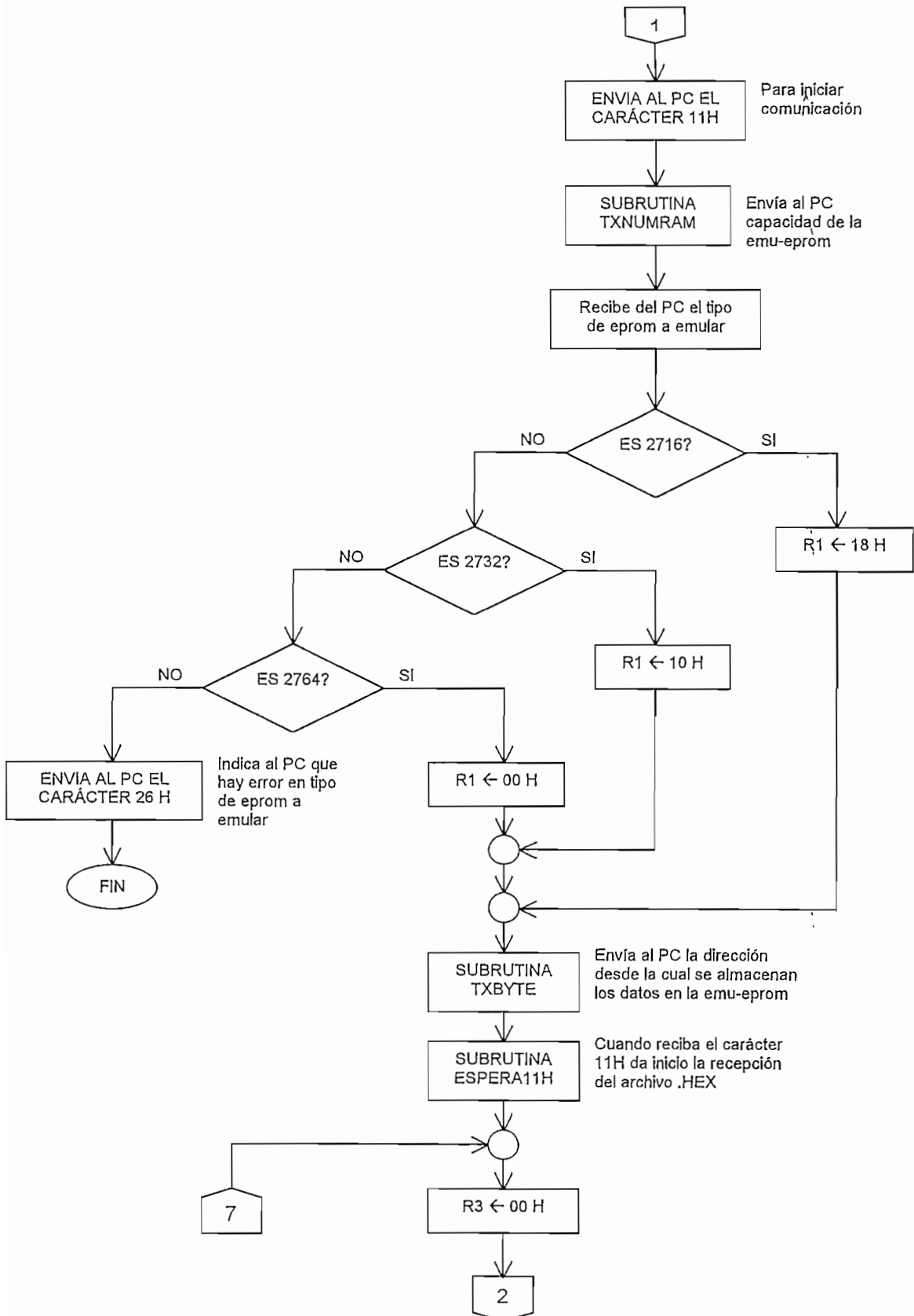
➤ Programa Principal

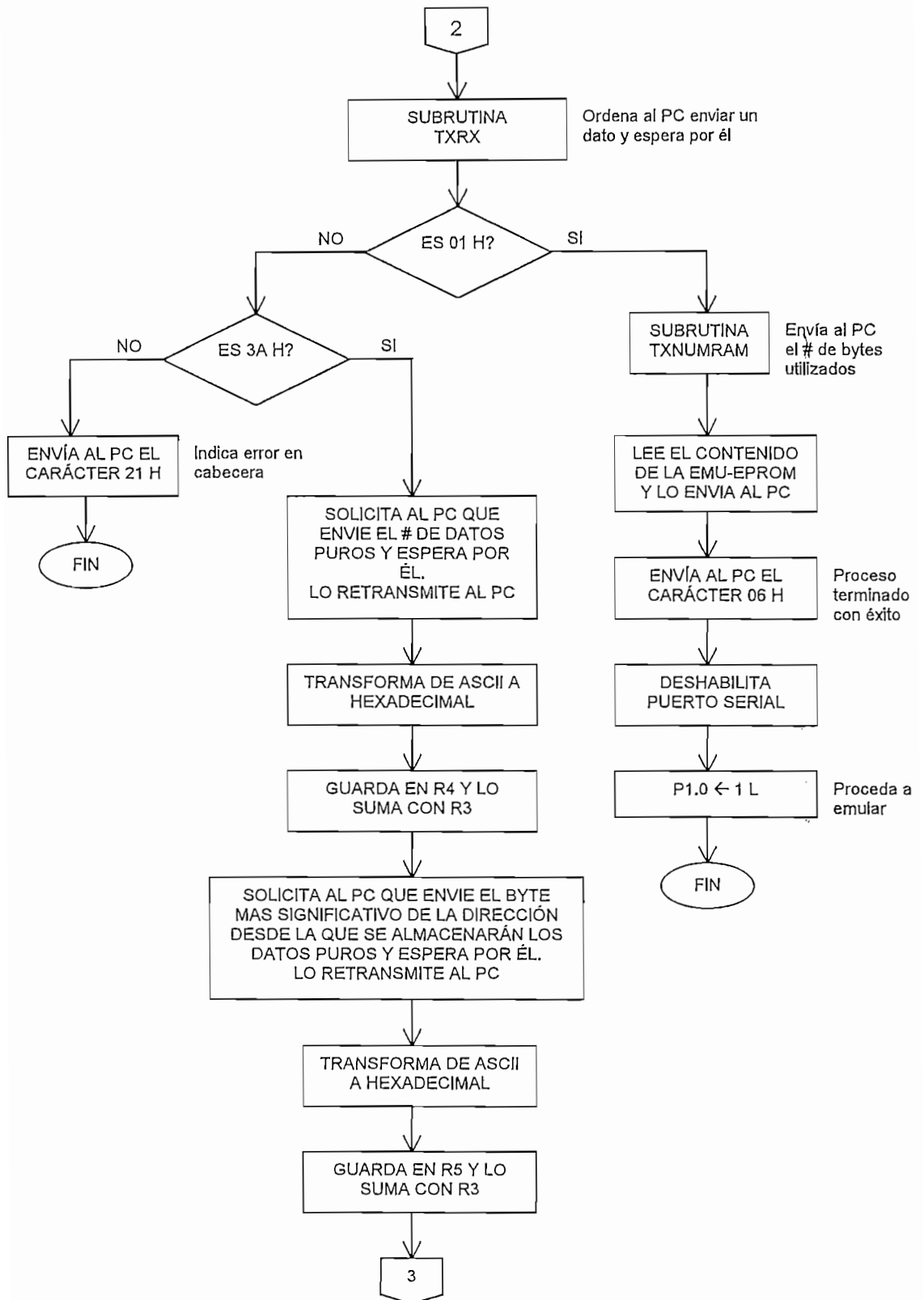
Inicia definiendo las etiquetas ha utilizarse, coloca en el pin P1.0 un nivel bajo para indicar que se grabarán datos en la emu-eprom, configura los registros para trabajar con el puerto serial y el timer 1, determina el estado de la emu-eprom, transmite a la computadora personal la capacidad de la misma, dependiendo del tipo de memoria Eprom a emular determina la dirección de inicio a partir de la cual se grabará el archivo .HEX . Luego recibe cada uno de los datos del archivo .HEX y los graba en la emu-eprom; si en algún momento se produce un error en la transmisión o que no se haya grabado el dato, lo indica a la computadora personal y termina ejecución del programa. Si no hay

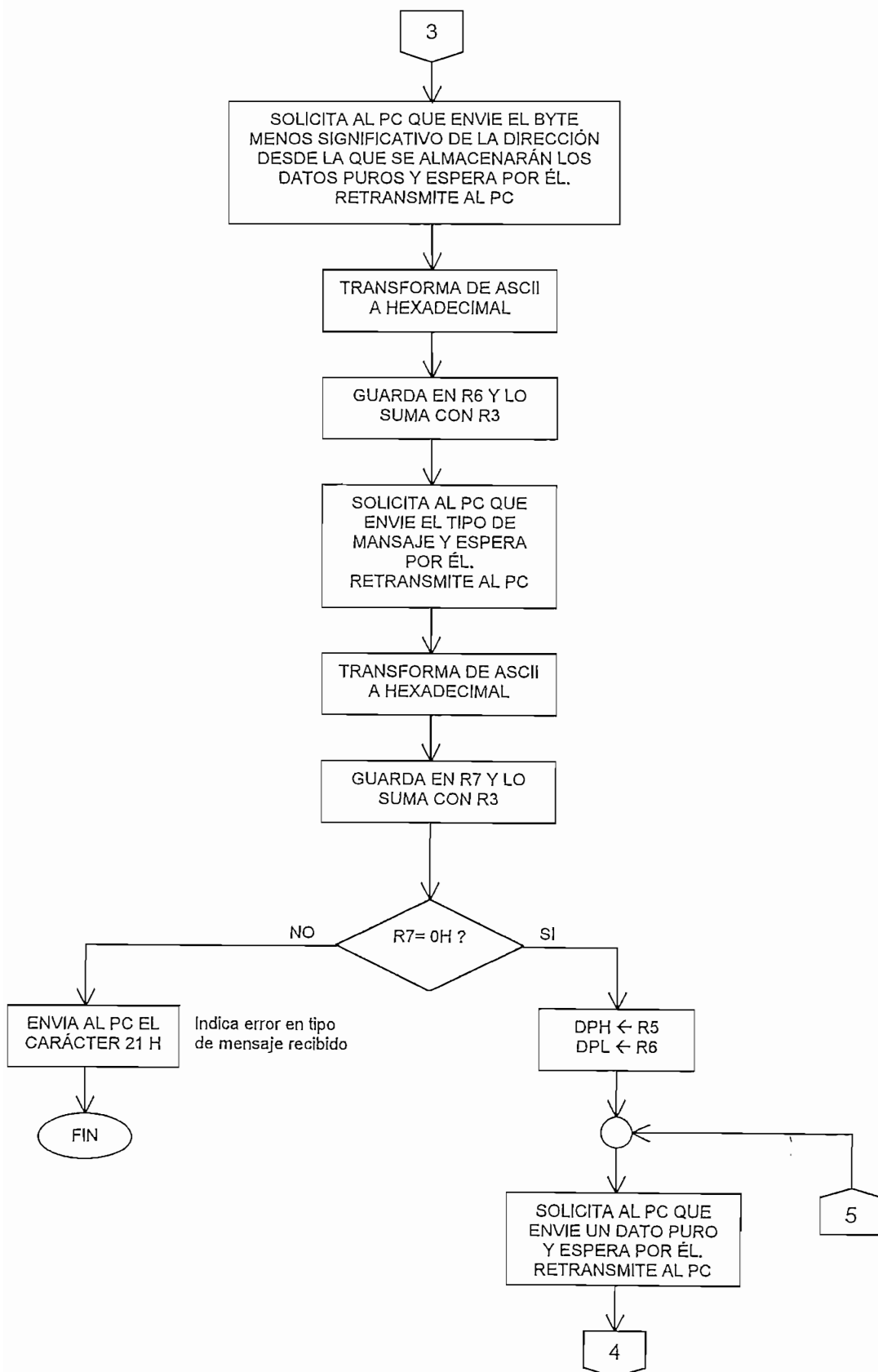
errores obtiene el número de localidades de la emu-eprom que contienen al archivo.HEX y lo envía a la computadora personal. Finalmente lee el contenido de la emu-eprom desde la dirección 0000 H y lo envía al PC, deshabilita el puerto serial y coloca en el pin P1.0 un nivel alto para indicar que se proceda a emular.

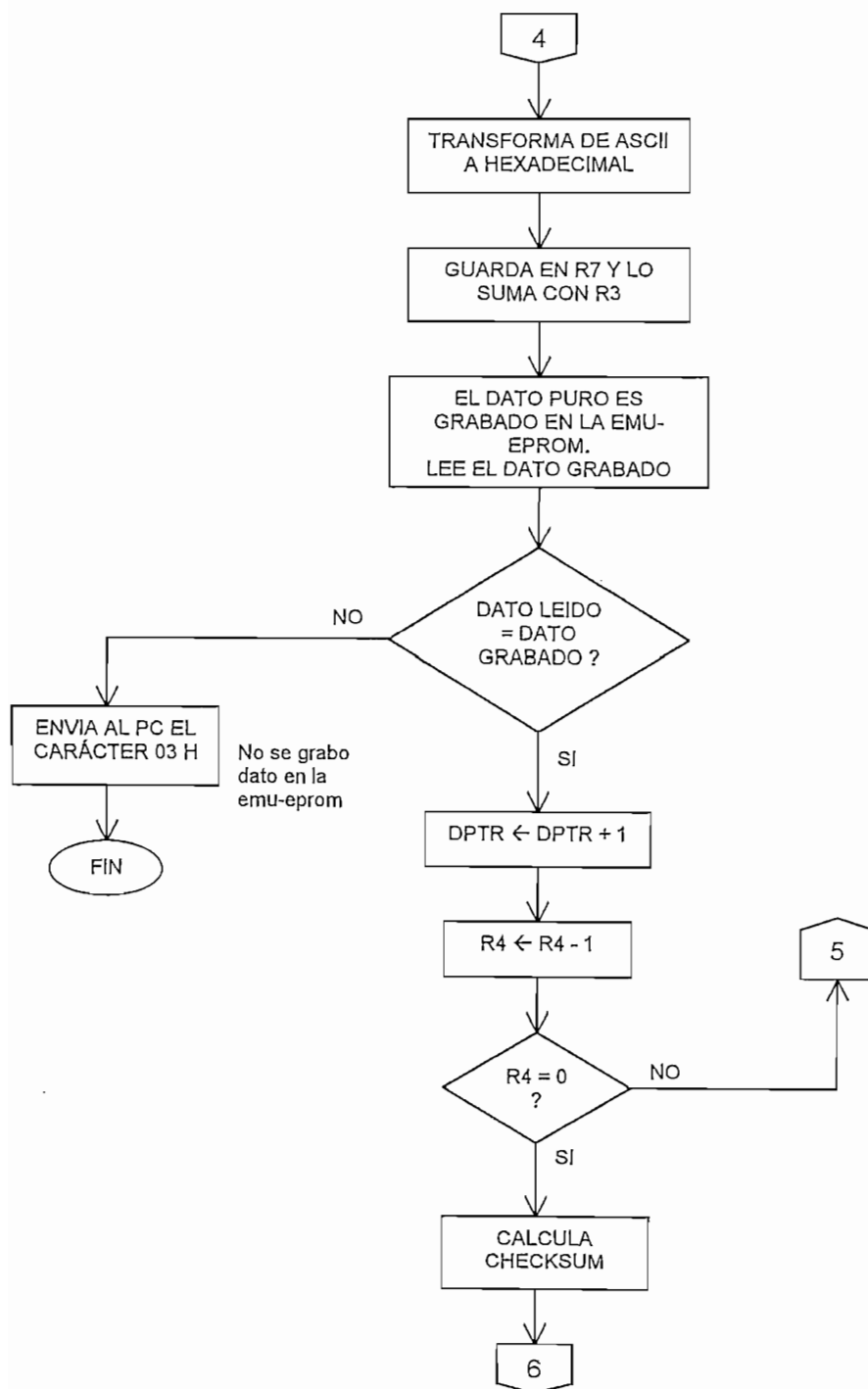
PROGRAMA PRINCIPAL

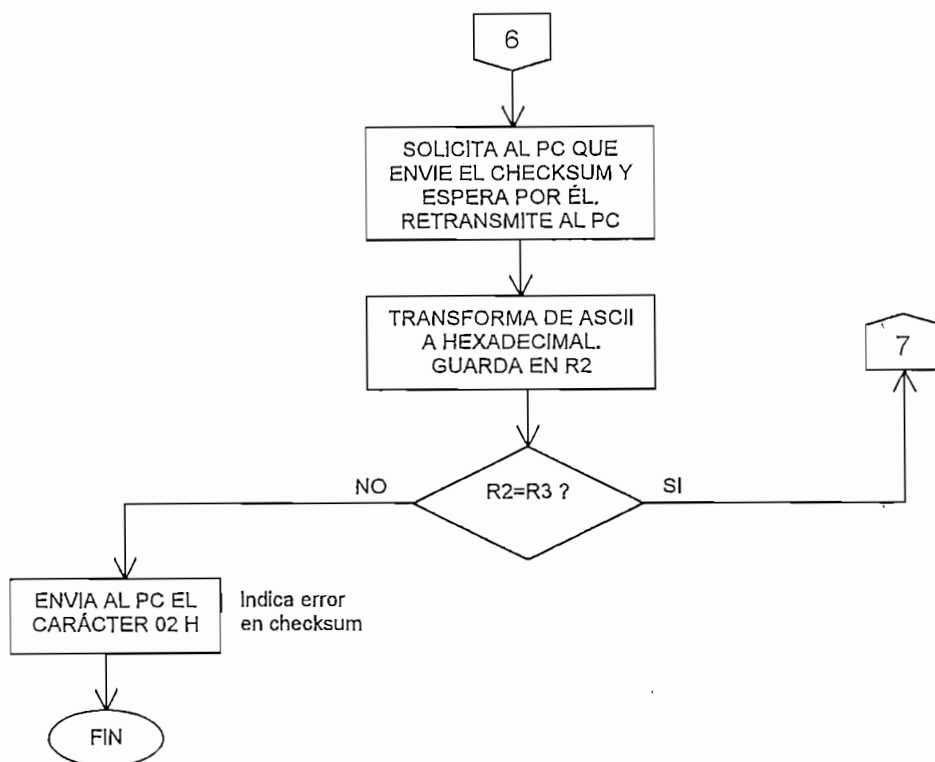












3.3.2 LISTADO DEL PROGRAMA PROG

```

; PROG.ASM
;
; ESCUELA POLITECNICA NACIONAL
; INGENIERIA EN ELECTRONICA Y TELECOMUNICACIONES
; TESIS DE GRADO
; TEMA: CONSTRUCCION DE UN EMULADOR DE MEMORIAS EPROM
; FECHA: MARZO-2002
; REALIZO: JORGE A. BENITEZ E.
; DIRECTOR: ING. RAMIRO MOREJON
;
; ETIQUETA OPCODE OPERANDOS COMENTARIO
; AREA PARA LA DEFINICION DE ETIQUETAS MEDIANTE LA DIRECTIVA "EQU"
;
BAUD EQU 0F3H ;Valor de recarga 2400 baudios
  
```

```

ENVIE      EQU    11H           ;Indica que transmite dato
ETIPOEPROM EQU    26H           ;Indica error en tipo de EMU-EPROM
OK         EQU    06H           ;Recepción sin errores
RAMALA     EQU    25H           ;Indica mal estado de la EMU-EPROM
RAML       EQU    40H           ;En estas localidades se guarda el
RAMH       EQU    RAML+1        ;número de bytes disponible en la
                                           ;EMU-EPROM
RAML1      EQU    42H           ;Contiene la última dirección donde
RAMH1      EQU    RAML1+1       ;se almacene un dato en la EMU-
                                           ;EPROM
;
;ETIQUETA  OPCODE      OPERANDOS      COMENTARIO
;
      DEFSEG  CERO, START=0
      SEG     CERO
      JMP     PRINCIPAL
      DEFSEG  SINT1, START=SINT, CLASS=CODE
      SEG     SINT1
      JMP     SERIAL
      DEFSEG  PROGRAMA, CLASS=CODE, START=30H
      SEG     PROGRAMA
;
PRINCIPAL: CLR     P1.0          ;Grabación en la EMU-EPROM
;
;Configura puerto serial

      MOV     IE, #90H          ;Habilita interrupción del puerto
                                           ;serial

      MOV     PCON, #00H
      MOV     TCON, #00H
      MOV     TMOD, #20H        ;TIMER 1 usado como temporizador en
      MOV     TH1, #BAUD        ;modo 2 (8 bit con autorecarga)
      MOV     TL1, #BAUD
      SETB    TR1               ;Habilita el TIMER 1
      MOV     SCON, #70H        ;8 bit UART variable, habilita RI y
                                           ;recepción serial
;
      MOV     RAML, #00H        ;Máxima capacidad de la EMU-EPROM
      MOV     RAMH, #20H        ;2000 H bytes
;
;Determina el estado de cada celda de la EMU-EPROM
LAZO1:  MOV     DPTR, #0000H     ;Inicia el puntero de datos
      MOV     A, #0AAH          ;Se prueba las localidades de
      MOVX    @DPTR, A          ;memoria de la EMU-EPROM, si esta
      CLR     A                 ;bien se guarda el valor FFH, caso
      MOVX    A, @DPTR          ;contrario va a SALTO1 para
      CJNE   A, #0AAH, SALTO1   ;reportar falla y terminar
      MOV     A, #55H           ;ejecución del programa
      MOVX    @DPTR, A
      CLR     A
      MOVX    A, @DPTR
      CJNE   A, #55H, SALTO1
      MOV     A, #0FFH
      MOVX    @DPTR, A
      CLR     A
      MOVX    A, @DPTR
      CJNE   A, #0FFH, SALTO1
      INC     DPTR
      CLR     C

```

```

        MOV    A,DPH
        SUBB   A,RAMH
        CJNE  A,#00H,LAZO1
        CLR   C
        MOV   A,DPL
        SUBB  A,RAML
        CJNE  A,#00H,LAZO1
        JMP   SALTO2
SALTO1:  JMP   MAL
;
;Indica al PC que esta listo para iniciar comunicaci3n
SALTO2:  MOV   A,#ENVIE
        CALL  TXDATO
;
;Envía al PC el número de bytes disponible
;en la EMU-EPROM, primero envía el MSB
;
        CALL  TXNUMRAM
;
;Se determina el tipo de memoria a emular: 2716 , 2732 o 2764, caso
;contrario reporta un error
;
        LCALL TXRX           ;Ordena al PC que transmita dato, espera
        LCALL ASCHEX        ;por él y transforma de código ASC a HEX
        SUBB  A,#02H
        JZ    SIGA1
        LCALL ERRORTIPOEPROM
SIGA1:   LCALL TXRX
        LCALL ASCHEX
        SUBB  A,#07H
        JZ    SIGA2
        LCALL ERRORTIPOEPROM
SIGA2:   LCALL TXRX
        LCALL ASCHEX
        MOV   B,A
        SUBB  A,#01H
        JZ    SIGA3
        MOV   A,B
        SUBB  A,#03H
        JZ    SIGA5
        MOV   A,B
        SUBB  A,#06H
        JZ    SIGA7
        LCALL ERRORTIPOEPROM
SIGA3:   LCALL TXRX
        LCALL ASCHEX
        SUBB  A,#06H
        JZ    SIGA4
        LCALL ERRORTIPOEPROM
SIGA5:   LCALL TXRX
        LCALL ASCHEX
        SUBB  A,#02H
        JZ    SIGA6
        LCALL ERRORTIPOEPROM
SIGA7:   LCALL TXRX
        LCALL ASCHEX
        SUBB  A,#04H
        JZ    SIGA8
        LCALL ERRORTIPOEPROM
SIGA4:   MOV   R1,#18H        ;R1 contiene el DPH desde la cual se

```

```

                JMP     SIGA9           ;almacenan los datos en la EMU-EPRQM
SIGA6:         MOV     R1,#10H
                JMP     SIGA9
SIGA8:         MOV     R1,#00H
SIGA9:         MOV     B,R1
                CALL   TXBYTE         ;Transmite al PC la dirección desde la
                                      ;cual se almacenan los datos en la EMU-
                                      ;EPROM
                CALL   ESPERA11H
;
;Inicia la recepción del archivo .HEX
LINEA:        MOV     R3,#00H         ;R3 contiene la suma de los datos
                MOV     A,R1
                MOV     R5,A
                CLR     A
                CALL   TXRX
                CJNE   A,#01H,SALTO    ;Si recibe 01H termina la recepción
                AJMP   FIN
SALTO:        CJNE   A,#3AH,ERR1
                SJMP   SALTO4
ERR1:         AJMP   ERROR
;
;Recibe el # de datos puros y lo guarda en R4
;
SALTO4:       LCALL   TXRX           ;Ordena al PC que transmita dato, espera
                LCALL   RETX         ;por él y lo retransmite
                LCALL   ASCHEX
                SWAP   A
                MOV     R4,A         ;R4 contiene el número de datos puros
                LCALL   TXRX         ;Ordena al PC que transmita dato, espera
                LCALL   RETX         ;por él y lo retransmite
                LCALL   ASCHEX
                ADD    A,R4
                MOV     R4,A
                ADD    A,R3
                MOV     R3,A
;
;Recibe la dirección desde la cual se almacenan los datos puros
;
                LCALL   TXRX         ;Ordena al PC que transmita dato,
                LCALL   RETX         ;espera por él y lo retransmite
                LCALL   ASCHEX
                SWAP   A
                ADD    A,R5         ;R5 contiene el DPH de la dirección de la
                MOV     R5,A         ;EMU-EPROM desde la cual se almacenan los
                                      ;datos puros
                LCALL   TXRX         ;Ordena al PC que transmita dato, espera
                LCALL   RETX         ;por él y lo retransmite
                LCALL   ASCHEX
                ADD    A,R5
                MOV     R5,A
                ADD    A,R3
                MOV     R3,A
;
                LCALL   TXRX
                LCALL   RETX
                LCALL   ASCHEX
                SWAP   A
                MOV     R6,A         ;R6 contiene el DPL de la dirección de la

```



```

        LCALL TXRX          ;EMU-EPROM desde la cual se almacenan los
                           ;datos puros

        LCALL RETX
        LCALL ASCHEX
        ADD  A,R6
        MOV  R6,A
        ADD  A,R3
        MOV  R3,A

;
;Recibe el tipo de mensaje y lo guarda en R7
;
        LCALL TXRX
        LCALL RETX
        LCALL ASCHEX
        SWAP A
        MOV  R7,A          ;R7 contiene el tipo de mensaje
        LCALL TXRX
        LCALL RETX
        LCALL ASCHEX
        ADD  A,R7
        MOV  R7,A
        ADD  A,R3
        MOV  R3,A
        CJNE R7,#00H,ERROR
        MOV  DPH,R5
        MOV  DPL,R6

;
;Recibe, uno a la vez, los datos puros a guardar en la EMU-EPROM
SALTO6:  LCALL TXRX
        LCALL RETX
        LCALL ASCHEX
        SWAP A
        MOV  R7,A          ;R7 contiene el dato hexadecimal puro a
                           ;guardar en la EMU-EPROM
        LCALL TXRX
        LCALL RETX
        LCALL ASCHEX
        ADD  A,R7
        MOV  R7,A
        ADD  A,R3
        MOV  R3,A
        MOV  A,R7
        MOV  B,A
        MOVX @DPTR,A
        CLR  A
        MOVX A,@DPTR      ;Si el dato hexadecimal no es igual al
        CJNE A,B,ERROR3   ;que se grabo, envía al PC un mensaje de
                           ;error

        INC  DPTR
        DJNZ R4,SALTO6

;
        CLR  C
        MOV  A,R3          ;Calcula el CHECKSUM y lo guarda en R3
        SUBB A,R1
        CPL  A
        ADD  A,#01H
        MOV  R3,A
        LCALL TXRX        ;Recibe el CHECKSUM y lo guarda en R2
        LCALL RETX
        LCALL ASCHEX
        SWAP A

```

```

        MOV    R2,A
        LCALL TXRX
        LCALL RETX
        LCALL ASCHEX
        ADD   A,R2
        MOV   R2,A
        CLR   C
        SUBB  A,R3
        CJNE A,#00H,ERROR2
        JMP   LINEA
ERROR2:  MOV   A,#02H
        CALL TXDATO
        SJMP FINAL
ERROR3:  MOV   A,#03H
        CALL TXDATO
        SJMP FINAL
ERROR:   MOV   A,#21H
        CALL TXDATO
        SJMP FINAL

FIN:
;Envía al PC el número de bytes utilizados
;en la EMU-EPROM, primero envía el MSB
        MOV   RAML1,DPL ;RAML1 y RAMH1 contienen la última
        MOV   RAMH1,DPH ;dirección donde se almaceno un dato
        MOV   RAML,DPL  ;del programa a emular
        MOV   A,DPH
        SUBB  A,R1
        MOV   RAMH,A
        MOV   R0,#RAMH
        MOV   R2,#02H
        CALL TXNUMRAM
;Programa para leer la EMU-EPROM
        MOV   A,#ENVIE
        CALL TXDATO
        CALL ESPERA11H
        MOV   DPTR,#0000H
LAZO3:  MOVX  A,@DPTR ;Lee el contenido hexadecimal de la
        MOV   B,A ;EMU-EPROM y lo envía al PC
        CALL TXBYTE
        CALL ESPERA11H
        INC  DPTR
        CLR  C
        MOV  A,DPH
        SUBB A,RAMH1
        CJNE A,#00H,SALTO7
        CLR  C
        MOV  A,DPL
        SUBB A,RAML1
        CJNE A,#00H,SALTO7
        SJMP FINAL
SALTO7: JMP   LAZO3
FIN2:   MOV   A,#06H
        CALL TXDATO
        CLR  A
        MOV  SCON,A ;Deshabilita el puerto serial
        SETB P1.0 ;Para emular
FINAL:  SJMP  $ ;FIN DEL PROGRAMA
;
;
;
```

```

;Area para subrutinas
;
;
ASCHEX:    CJNE  A, #30H, ASC1    ;Transforma un dato en formato
          MOV   A, #00H          ;ASCII a formato hexadecimal
          JMP   ASCFIN
ASC1:      CJNE  A, #31H, ASC2
          MOV   A, #01H
          JMP   ASCFIN
ASC2:      CJNE  A, #32H, ASC3
          MOV   A, #02H
          JMP   ASCFIN
ASC3:      CJNE  A, #33H, ASC4
          MOV   A, #03H
          JMP   ASCFIN
ASC4:      CJNE  A, #34H, ASC5
          MOV   A, #04H
          JMP   ASCFIN
ASC5:      CJNE  A, #35H, ASC6
          MOV   A, #05H
          JMP   ASCFIN
ASC6:      CJNE  A, #36H, ASC7
          MOV   A, #06H
          JMP   ASCFIN
ASC7:      CJNE  A, #37H, ASC8
          MOV   A, #07H
          JMP   ASCFIN
ASC8:      CJNE  A, #38H, ASC9
          MOV   A, #08H
          JMP   ASCFIN
ASC9:      CJNE  A, #39H, ASC10
          MOV   A, #09H
          JMP   ASCFIN
ASC10:     CJNE  A, #41H, ASC11
          MOV   A, #0AH
          JMP   ASCFIN
ASC11:     CJNE  A, #42H, ASC12
          MOV   A, #0BH
          JMP   ASCFIN
ASC12:     CJNE  A, #43H, ASC13
          MOV   A, #0CH
          JMP   ASCFIN
ASC13:     CJNE  A, #44H, ASC14
          MOV   A, #0DH
          JMP   ASCFIN
ASC14:     CJNE  A, #45H, ASC15
          MOV   A, #0EH
          JMP   ASCFIN
ASC15:     CJNE  A, #46H, ASCFIN
          MOV   A, #0FH
ASCFIN:    RET
;
ERRORTIPOEPROM:
          MOV   A, #ETIPOEPROM    ;Indica al PC que hay un error en
          LCALL TXDATO           ;el tipo de EPROM a emular, y
          AJMP  FINAL            ;termina ejecución del programa
          RET
;
ESPERA11H: JNB   RI, ESPERA11H    ;Lazo hasta recibir dato
          CLR   RI               ;Limpiar la bandera RX

```



```

CALL TXDATO
CALL ESPERA11H
MOV A, B
RET

;
TXDATO: MOV SBUF, A ;Transmite un dato al PC
ESPERA1: JNB TI, ESPERA1
CLR TI
RET

;
TXRX: MOV A, #ENVIE ;Envía orden para que el PC
MOV SBUF, A ;transmita
ESPERA6: JNB TI, ESPERA6
CLR TI
ESPERA7: JNB RI, ESPERA7 ;Espera por un dato
CLR RI
MOV A, SBUF
RET

;
TXNUMRAM: MOV R0, #RAMH
MOV R2, #02H
TX1: CALL ESPERA11H
MOV A, @R0
MOV B, A
CALL TXBYTE
DEC R0
DJNZ R2, TX1
RET

;
TXBYTE: MOV A, B ;Transmite al PC un byte
SWAP A
ANL A, #0FH
CALL HEXASC
CALL TXDATO
CALL ESPERA11H
MOV A, B
ANL A, #0FH
CALL HEXASC
CALL TXDATO
RET

;
SERIAL: RETI

;
END ;FIN DEL ENSAMBLADO

```

A continuación se muestra el archivo PROG.HEX correspondiente:

```

:03000000020030CB
:0300230002038352
:20003000C29075A890758700758800758920758DF3758BF3D28E75987075400075412090BA
:20005000000074AAFOE4E0B4AA247455F0E4E0B4551C74FFFOE4E0B4FF14A3C3E583954108
:20007000B400DFC3E5829540B400D702008102032D741112034012035912034812021D9434
:2000900002600312029B12034812021D9407600312029B12034812021DF5F09401600FE5A5
:2000B000F094036016E5F09406601D12029B12034812021D9406601D12029B1203481202D3
:2000D0001D9402601512029B12034812021D9404600D12029B79180200F179100200F1797E
:2000F0000089F012036A1202A37B00E9FDE4120348B4010221D3B43A02800221CC12034838
:2001100012033512021DC4FC12034812033512021D2CFC2BFB12034812033512021DC42DA5
:20013000FD12034812033512021D2DFD2BFB12034812033512021DC4FE1203481203351237

```

```

:20015000021D2EFE2BFB12034812033512021DC4FF12034812033512021D2FFF2BFBBF0098
:200170005B8D838E8212034812033512021DC4FF12034812033512021D2FFF2BFBEFF5F059
:20019000F0E4E0B5F02FA3DCDC3EB99F42401FB12034812033512021DC4FA120348120309
:2001B0003512021D2AFAC39BB400030200F9740212034080567403120340804F74211203AF
:2001D000408048858242858343858240E58399F54178417A0212035974111203401202A3C1
:2001F000900000E0F5F012036A1202A3A3C3E5839543B4000AC3E5829542B4000280028047
:20021000E27406120340E4F598D29080FEB43005740002029AB43105740102029AB43205E9
:20023000740202029AB43305740302029AB43405740402029AB43505740502029AB43605A2
:20025000740602029AB43705740702029AB43805740802029AB43905740902029AB441055B
:20027000740A02029AB44205740B02029AB44305740C02029AB44405740D02029AB4450506
:20029000740E02029AB44602740F227426120340411B223098FDC298E4E599B411F522B40F
:2002B0000005743002032CB40105743102032CB40205743202032CB40305743302032CB4EA
:2002D0000405743402032CB40505743502032CB40605743602032CB40705743702032CB4AA
:2002F0000805743802032CB40905743902032CB40A05744102032CB40B05744202032CB45C
:200310000C05744302032CB40D05744402032CB40E05744502032CB40F0274462274251228
:200330000340411B22F5F01203401202A3E5F022F5993099FDC299227411F5993099FDC298
:20035000993098FDC298E5992278417A021202A3E6F5F012036A18DAF422E5F0C4540F12E9
:1403700002AF1203401202A3E5F0540F1202AF120340223218
:00000001FF

```

3.4 PROGRAMA DE CONTROL EN LA COMPUTADORA PERSONAL

La computadora personal debe contener el paquete Qbasic 45, en base al cual se desarrollo el programa de control EMULA.BAS que sirve de interfaz entre el operador y el microprocesador. Este programa de control consta de un Programa Principal y de las siguientes subrutinas:

- capacidademueprom
- continuar
- diremueprom
- espera11h
- esperadato
- fuente
- hexdec
- leerarchivo
- recepción
- resetup
- tipoeprom
- txarchivo
- txtipoeprom

3.4.1 DIAGRAMAS DE FLUJO DEL PROGRAMA EMULA

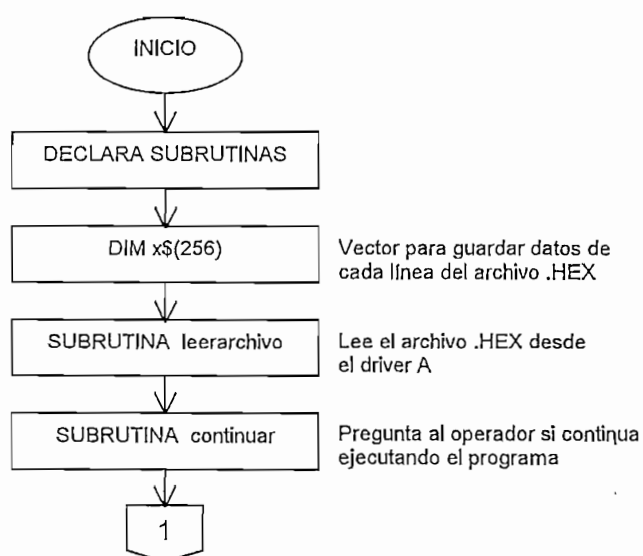
Con el fin de visualizar cada parte del programa EMULA y los saltos a las subrutinas, se presentan los diagramas de flujo con la explicación sobre su función.

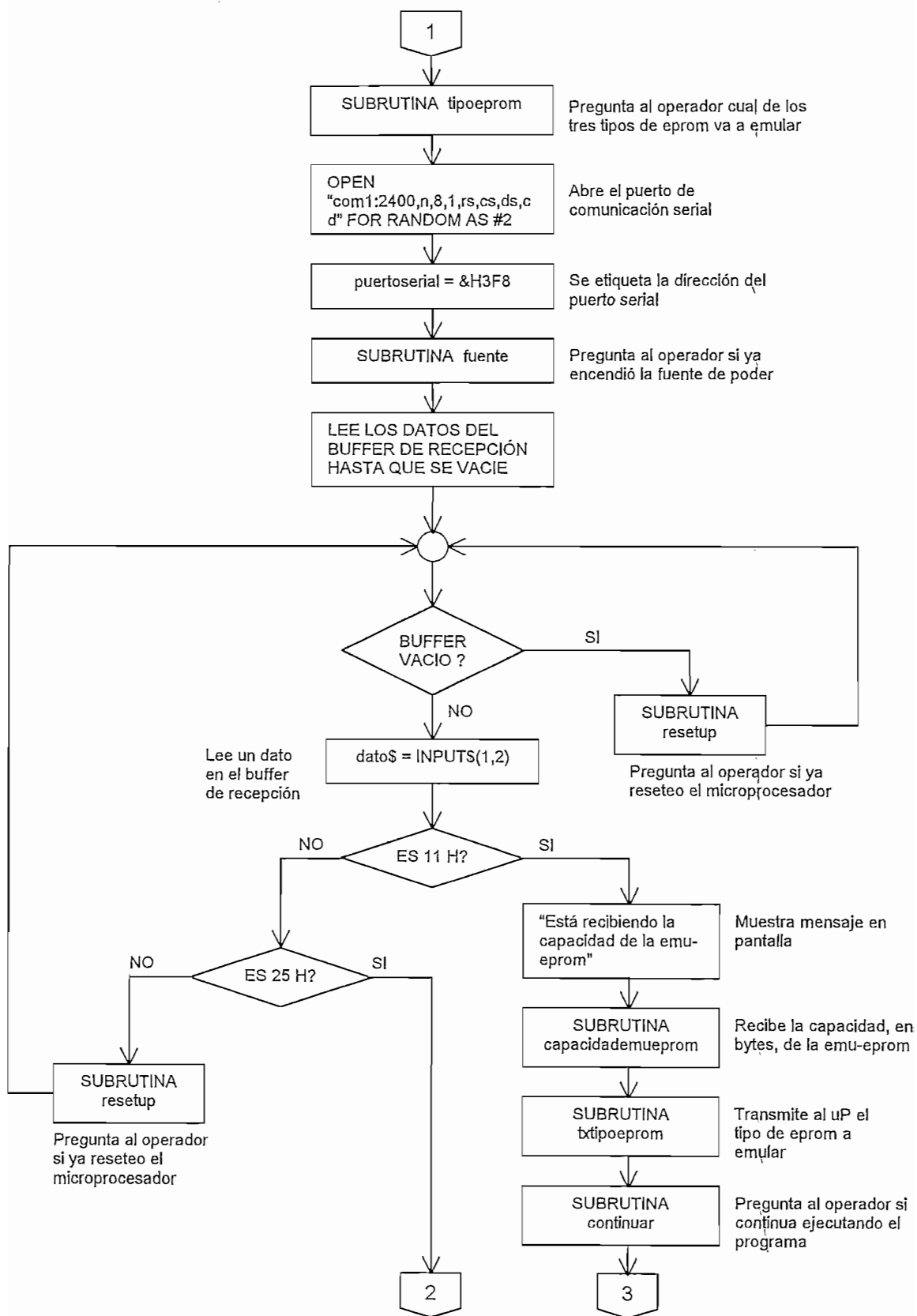
➤ Programa Principal

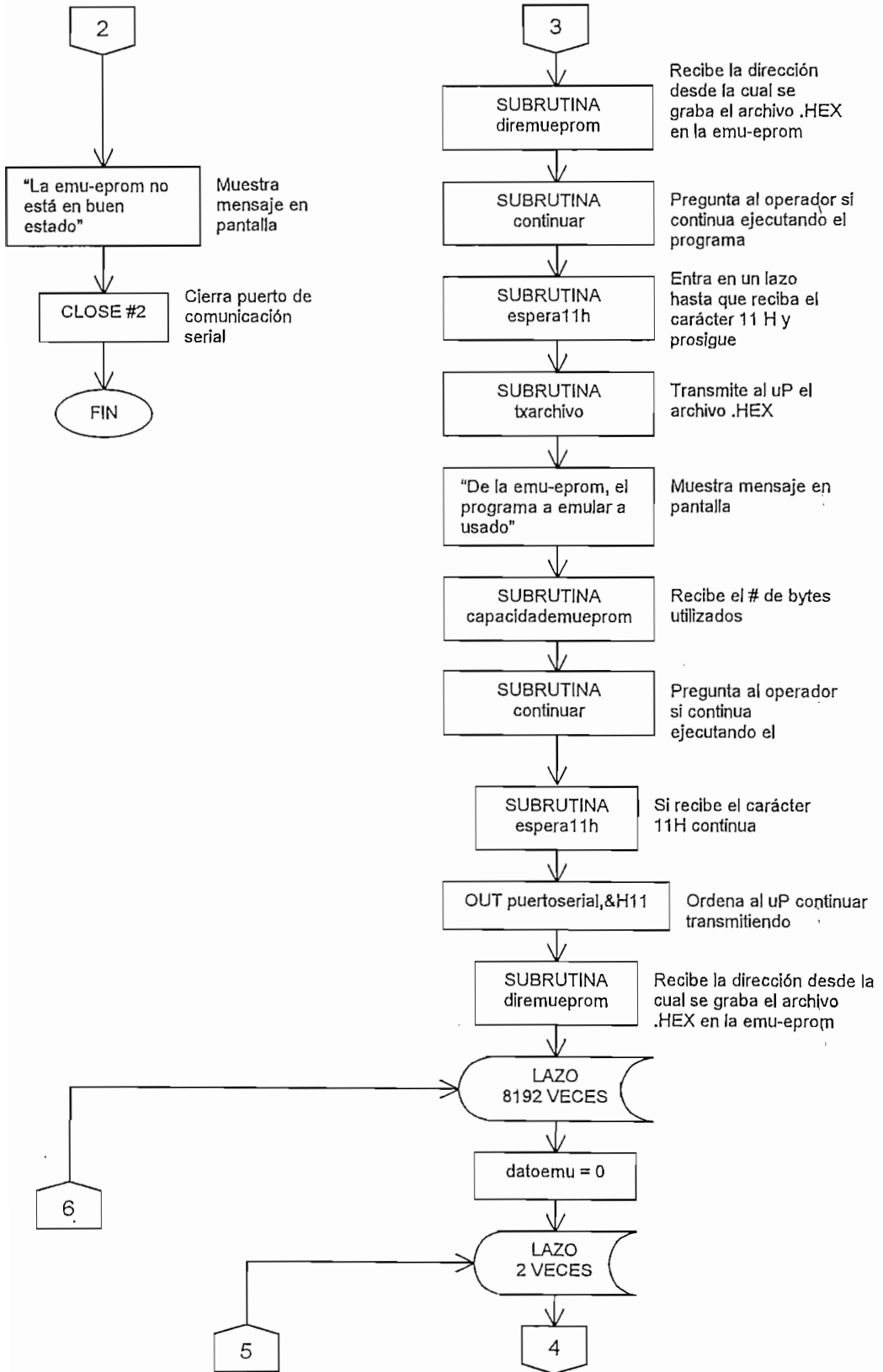
Inicia declarando las subrutinas a ejecutar, define un vector para almacenar datos, lee el archivo .HEX a emular, permite al operador ingresar el tipo de eprom a emular y lo transmite al microprocesador. Abre el puerto de comunicación serial, recibe la capacidad de la emu-eprom y la dirección de inicio desde la cual se grabará el archivo .HEX; luego transmite al microprocesador el archivo .HEX y finalmente recibe los datos guardados en la emu-eprom desde la dirección 0000 H y los presenta en pantalla.

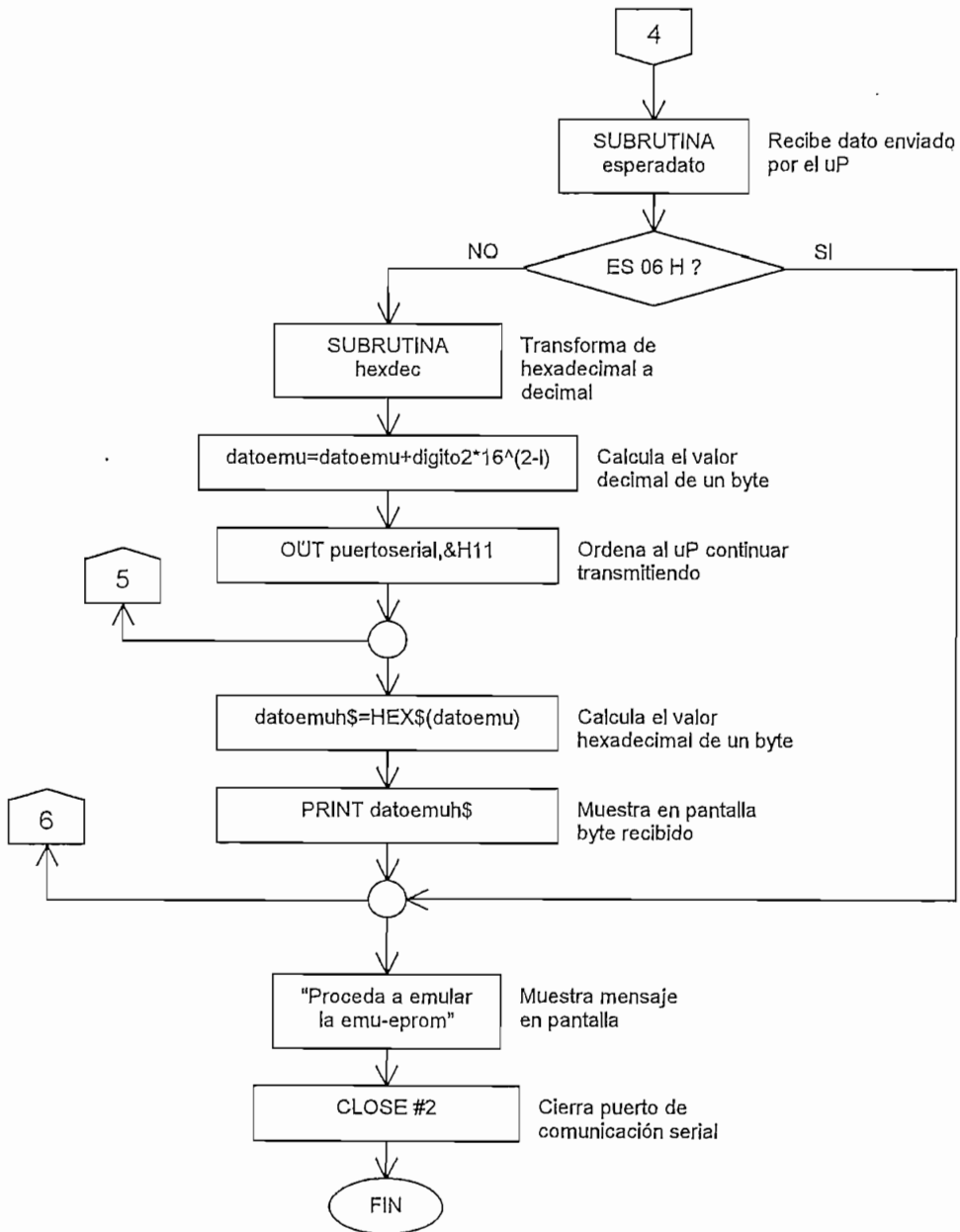
Durante la ejecución de todo el programa, la computadora interactúa con el operador permitiéndole ingresar datos y tomar decisiones, inclusive poder salir en cualquier momento del programa si fuese necesario.

PROGRAMA PRINCIPAL



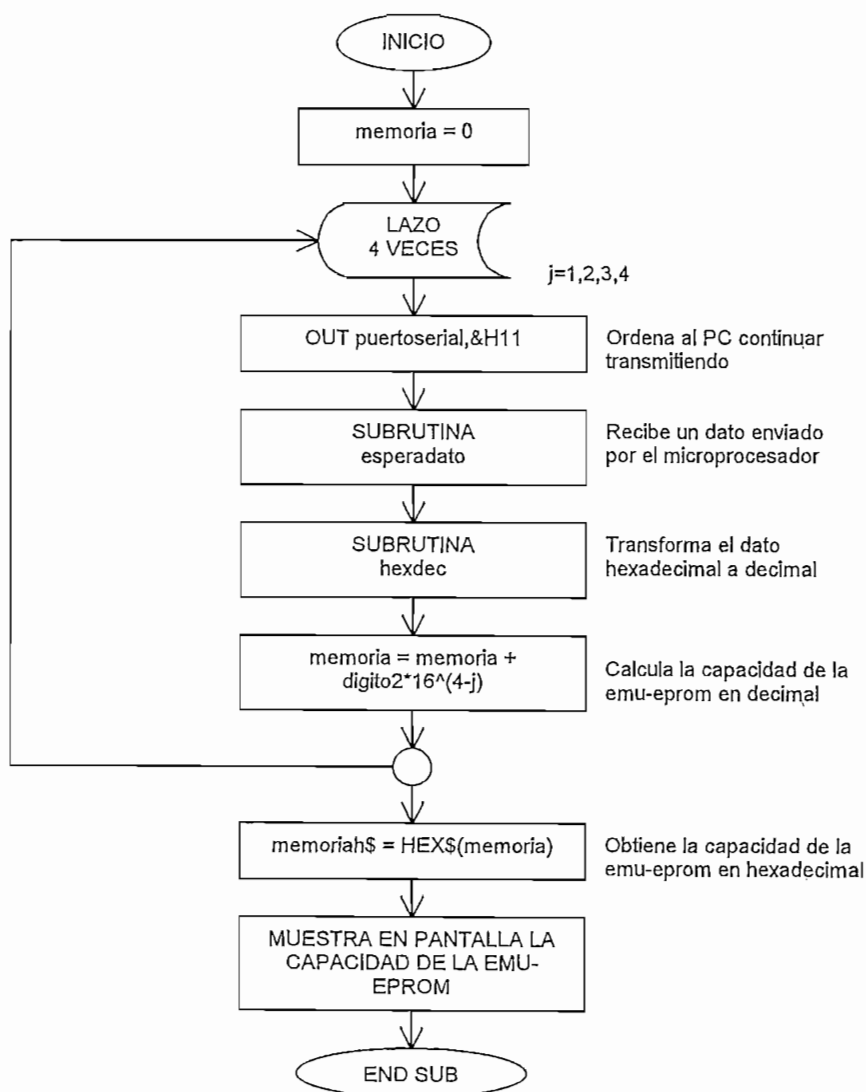






➤ SUBROUTINA capacidademueprom

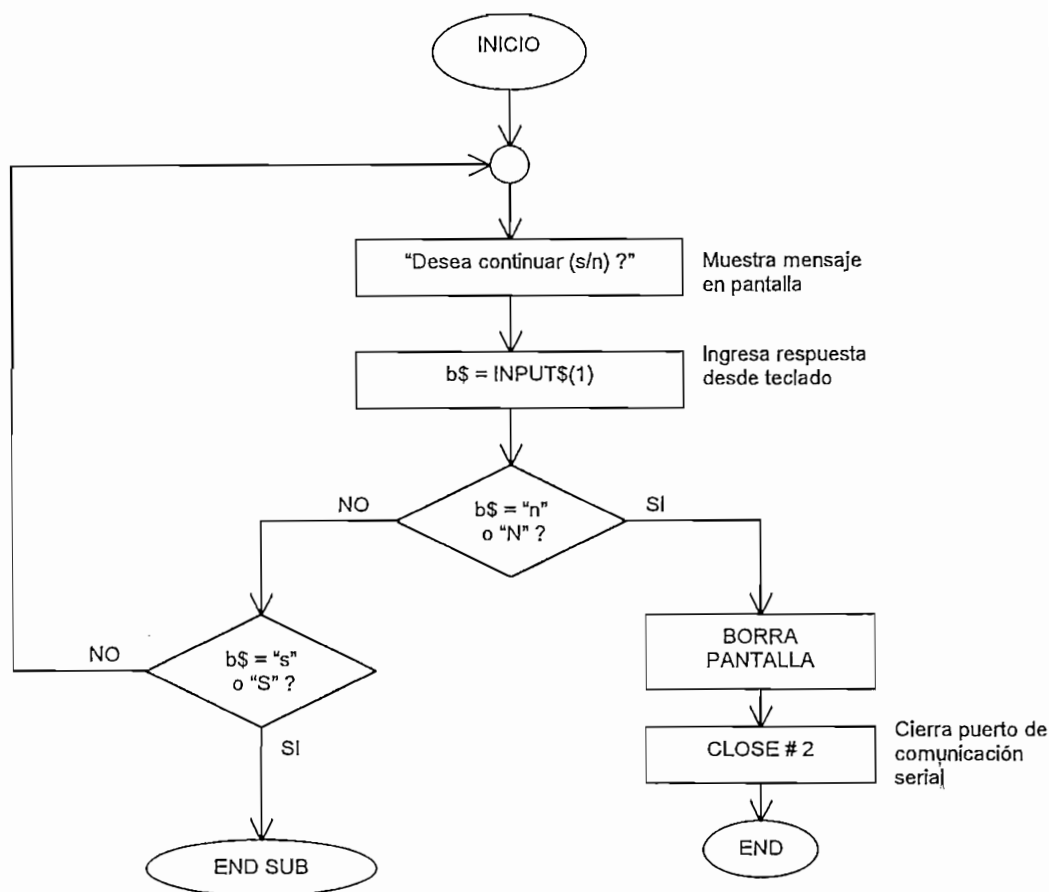
La computadora personal recibe primero el bit más significativo de la capacidad de la emu-eprom. Por cada bit se determina su valor decimal, al final se obtiene el valor en decimal y en hexadecimal de la capacidad de la emu-eprom.



➤ SUBROUTINA continua

Pregunta al operador si continua ejecutando el programa. Si la respuesta es positiva (s) termina subrutina, en caso negativo (n) borra la pantalla, cierra el puerto de comunicación serial y termina ejecución del programa.

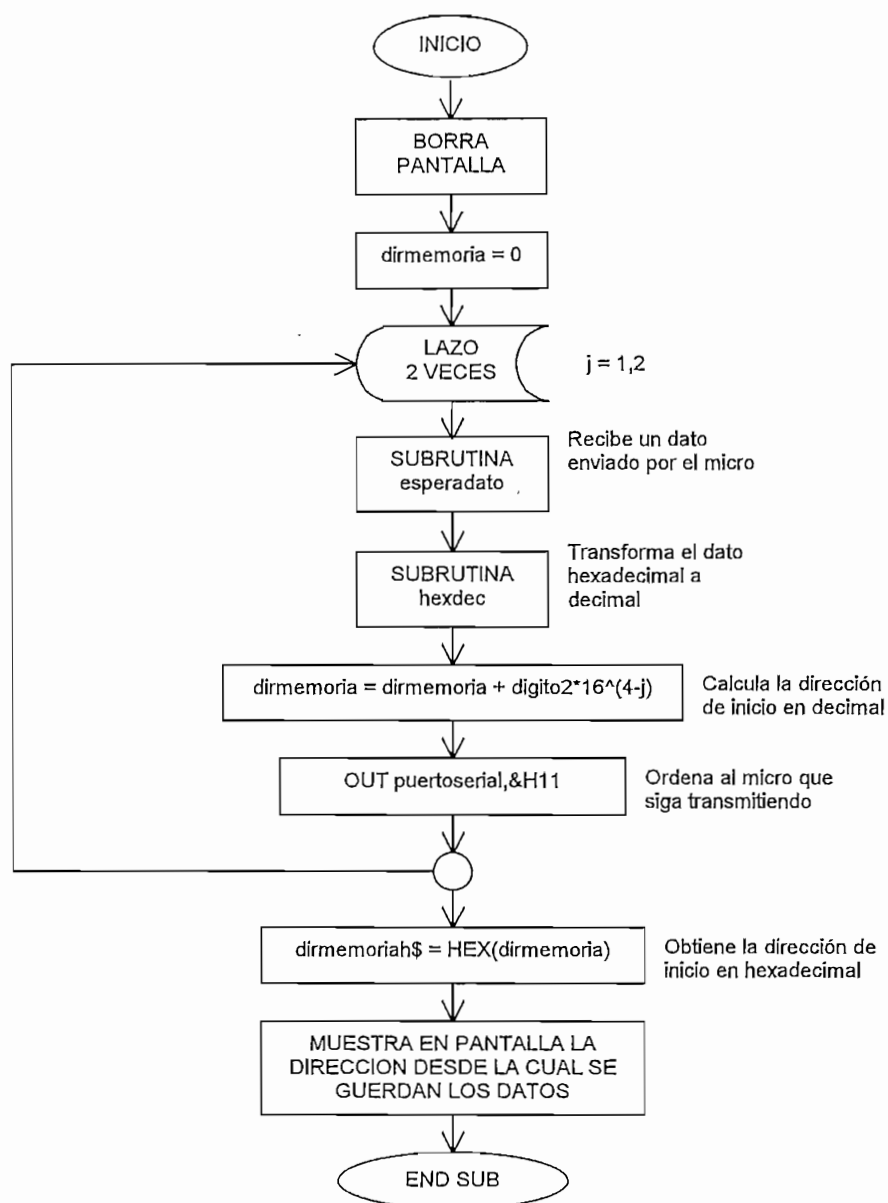
Si por error se presiona una tecla diferente a la letra s o n, vuelve a formular la misma pregunta. Es indiferente si el teclado está en modalidad mayúsculas o minúsculas.



➤ SUBROUTINA diremueprom

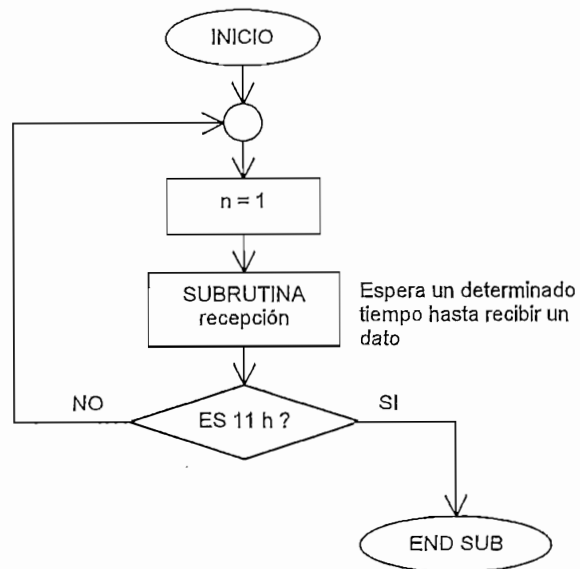
Existen tres posibles direcciones de inicio : 0000 H , 1000 H o 1800 , desde la cual se graba el programa .HEX en la emu-eprom. Se observa que el byte menos significativo siempre es 00 H ; por lo tanto, será suficiente que la computadora personal reciba solo el byte más significativo, esto es 00 H , 10 H o 18 H . Por ejemplo, si la dirección de inicio es 1800 H , la computadora recibirá primero el carácter "1" y luego el carácter "8", En cada caso habrá que

transformarlo a decimal, y finalmente obtener el valor en decimal y luego en hexadecimal de la dirección de inicio.



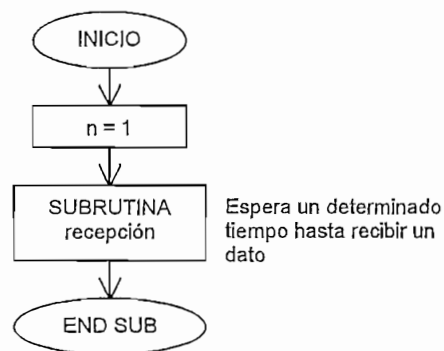
➤ SUBRUTINA espera11h

Si el dato recibido por la computadora personal es 11H termina subrutina, si es cualquier otro dato regresa al inicio de la subrutina.



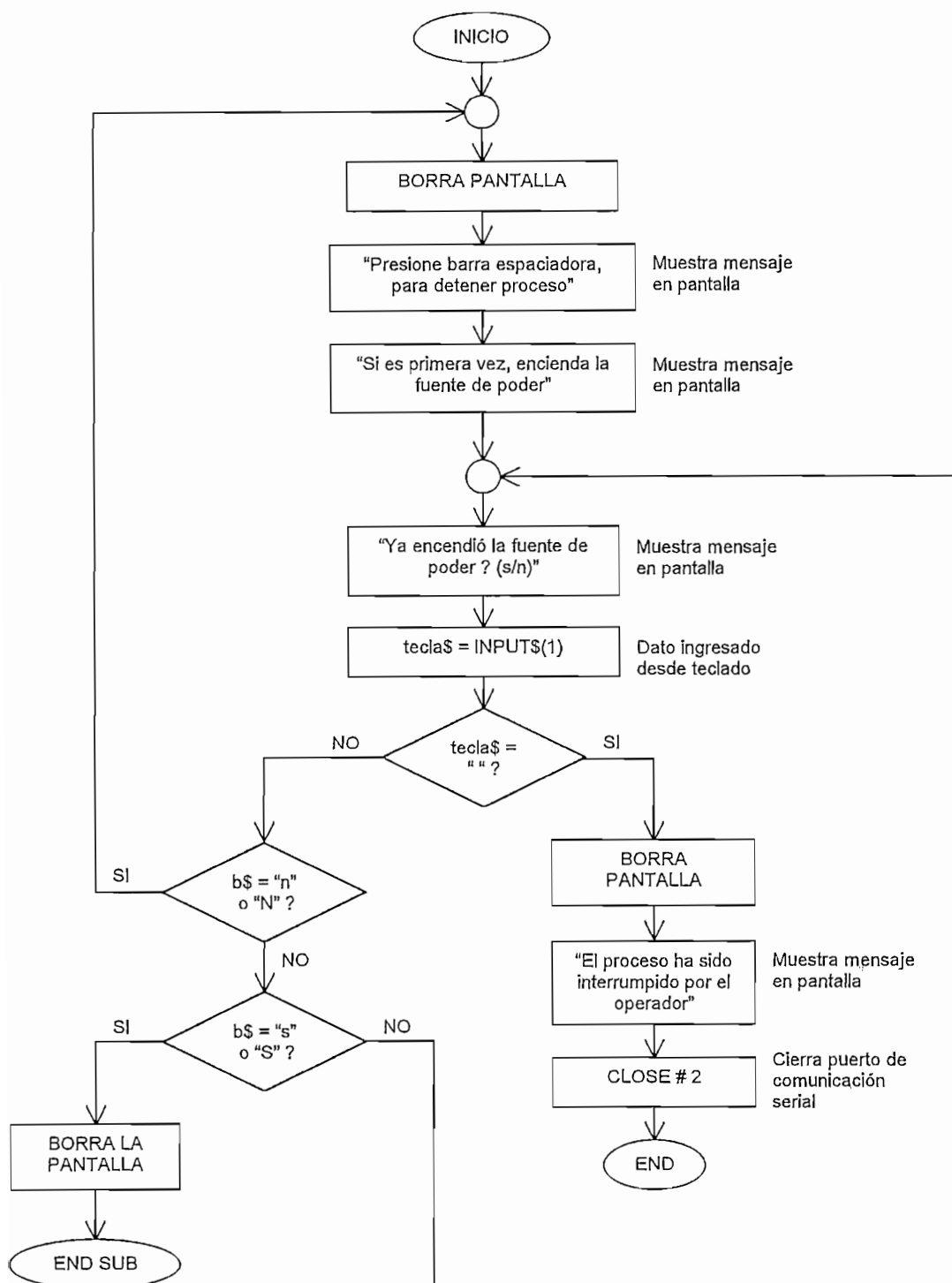
➤ SUBROUTINA esperadato

Recibe un dato que le envía el microprocesador.



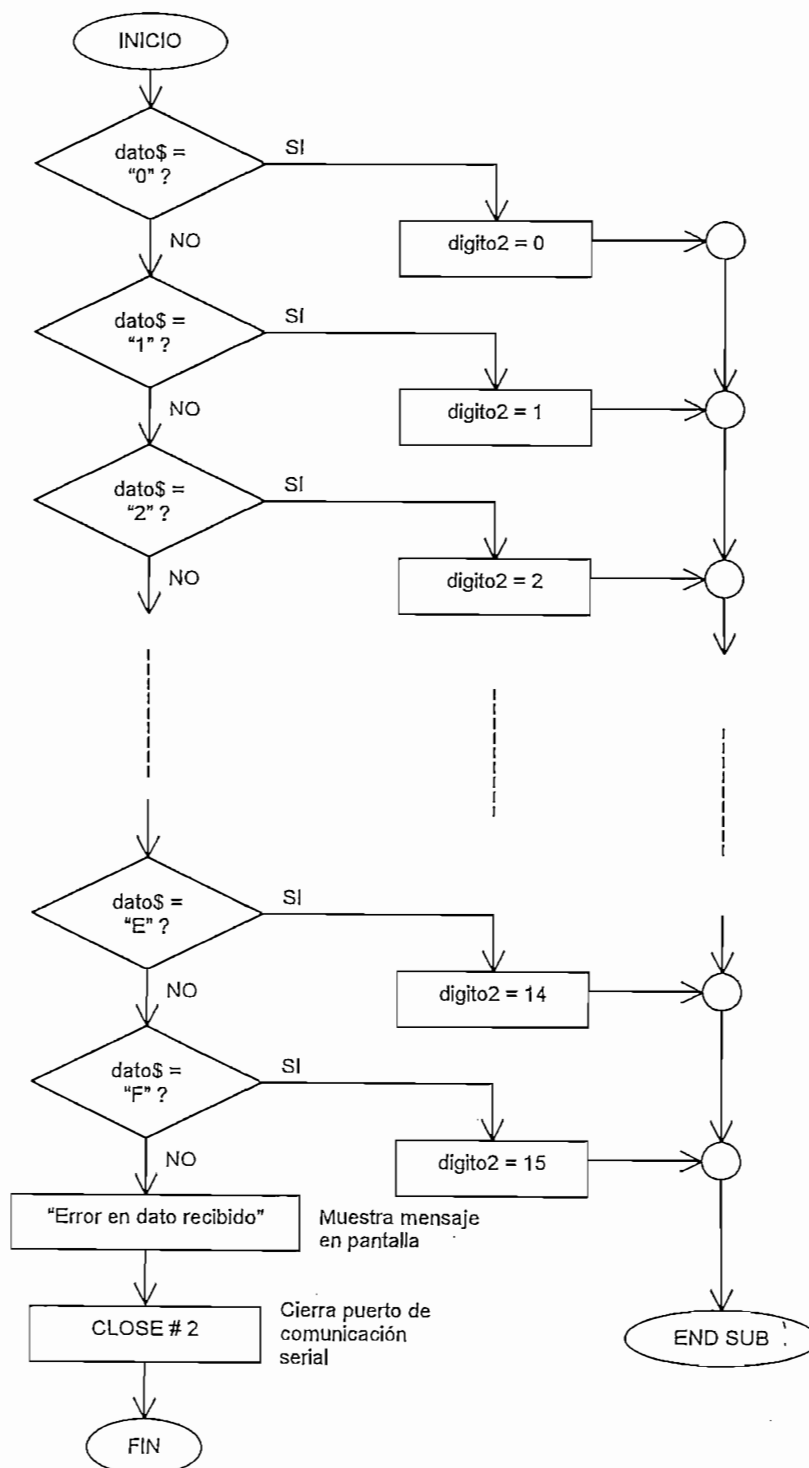
➤ SUBROUTINA fuente

Indica al operador que primero debe encender la fuente de poder. El teclado puede estar en modalidad de mayúsculas o minúsculas. Existe la opción de salir del programa presionando la barra espaciadora.



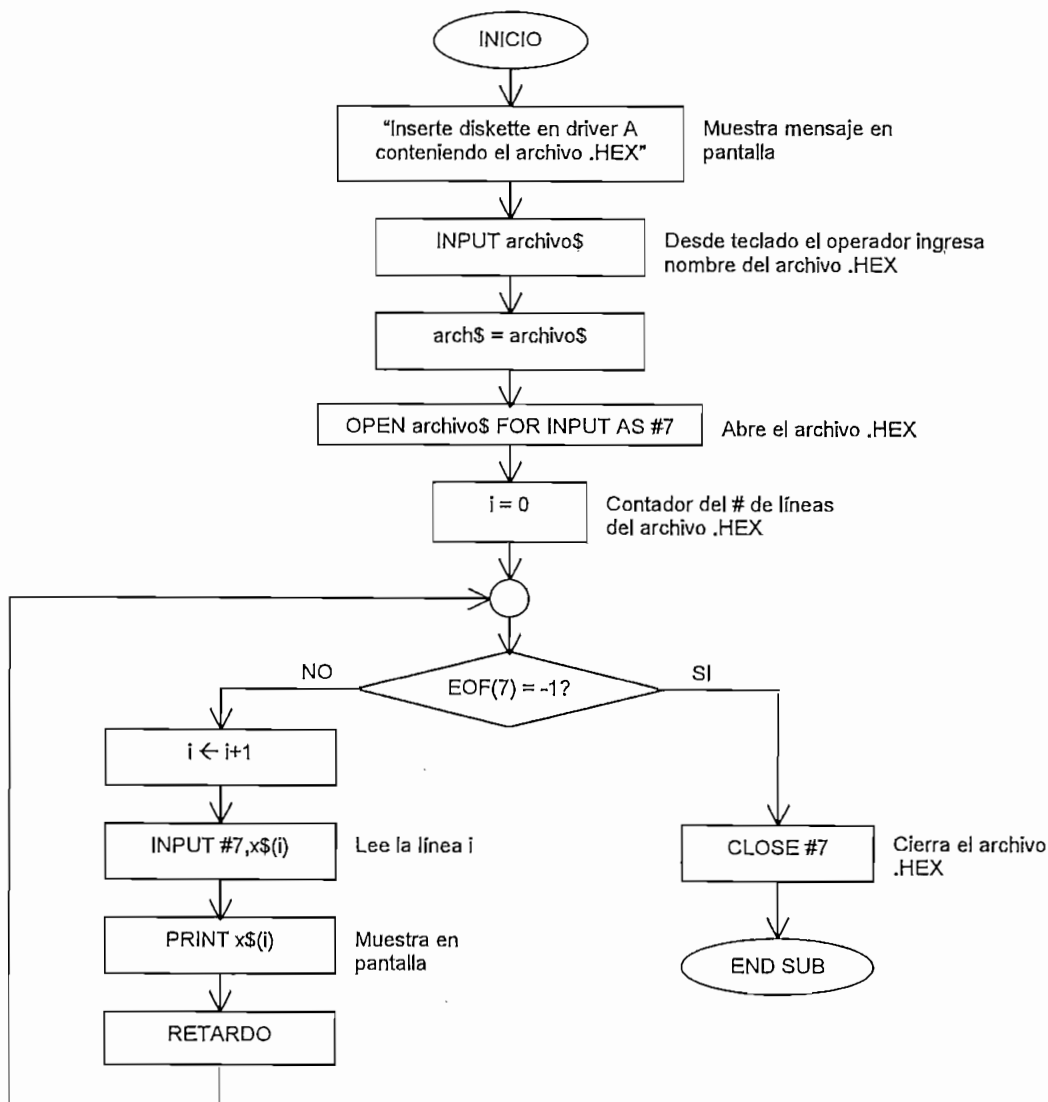
➤ SUBROUTINA hexdec

Esta subrutina transforma los caracteres "0" , "1" hasta "E" y "F" a número decimal 0 , 1 hasta 14 y 15 , respectivamente.



➤ SUBROUTINA leerarchivo

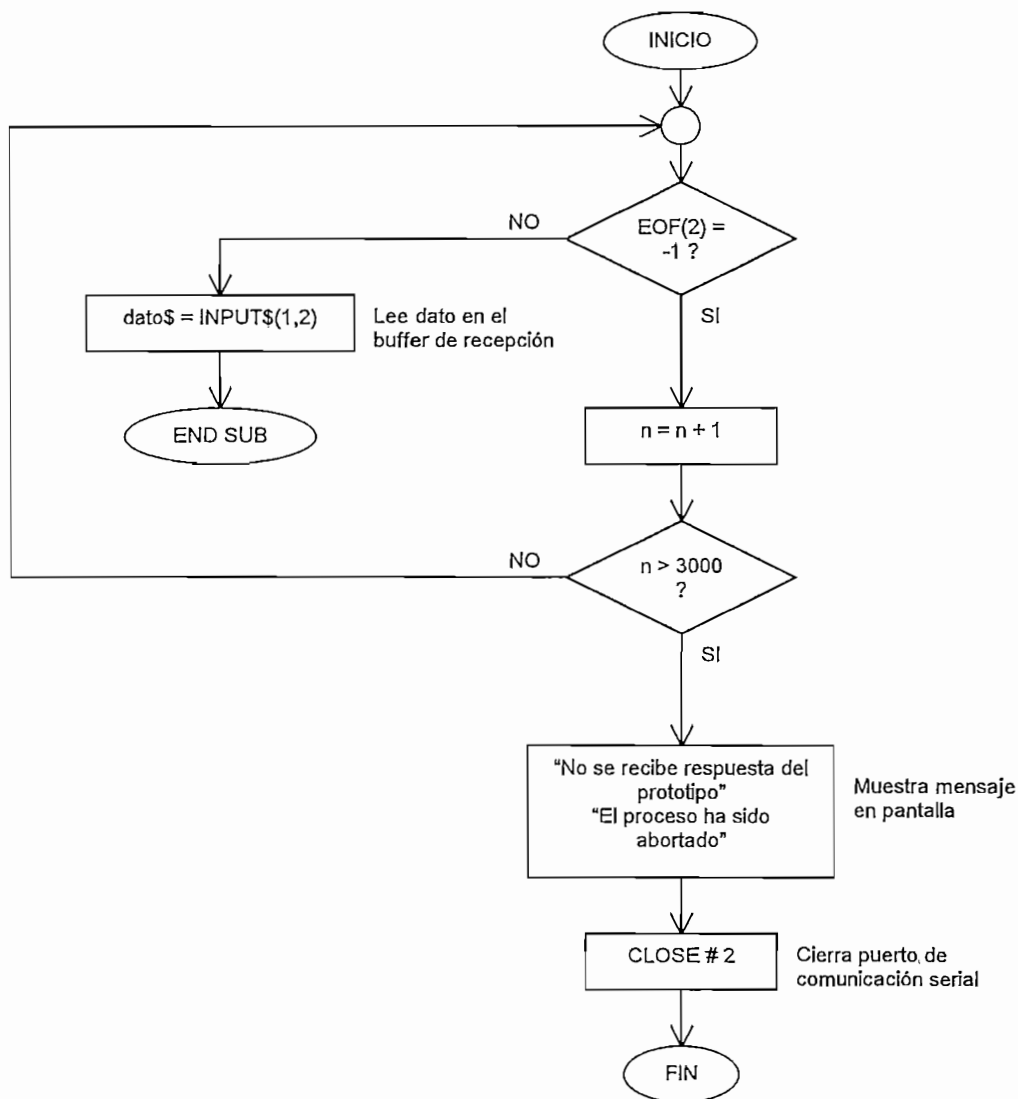
Permite leer el archivo .HEX desde la driver A , previamente pide al operador que ingrese el nombre del archivo .HEX . El archivo esta estructurado de varios mensajes (líneas) , esta subrutina los lee de uno en uno, guarda en el vector $x\$(i)$ y los presenta en pantalla.



➤ SUBROUTINA recepción

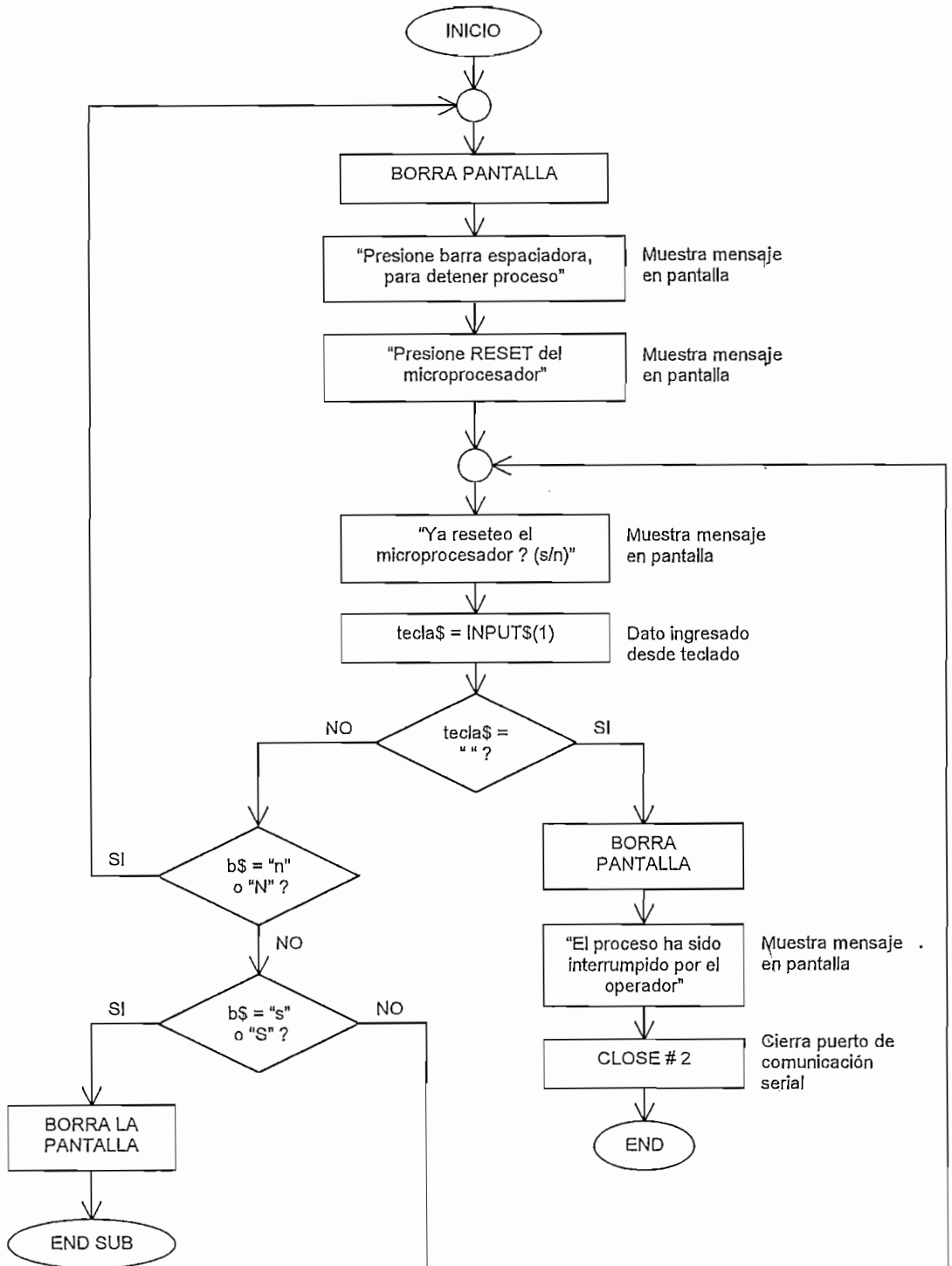
El programa pregunta a la computadora personal, hasta 3000 veces, si ha recibido algún dato por el puerto serial, en caso afirmativo lee el dato y termina

la subrutina; caso contrario, muestra un mensaje de que no recibe respuesta del microprocesador y finaliza el proceso.



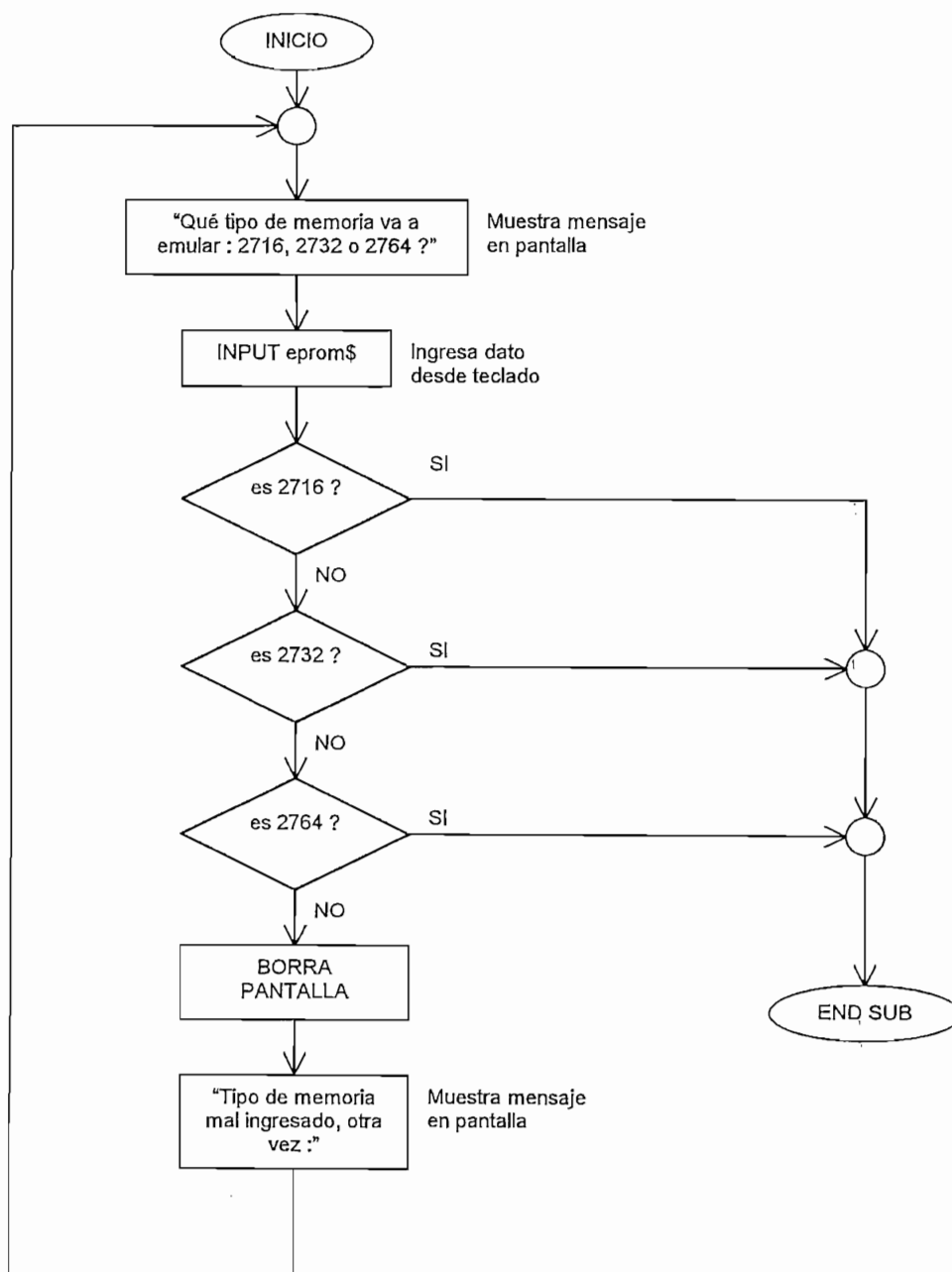
➤ SUBRUTINA resetup

Pregunta al operador si ya reseteo el microprocesador. El teclado puede estar en modalidad de mayúsculas o minúsculas. Existe la opción de salir del programa presionando la barra espaciadora.



➤ SUBROUTINA tipoeprom

Pregunta al operador y éste a través del teclado ingresa uno de los tres tipos de memoria Eprom a emular. Si ingresa mal, pregunta otra vez.



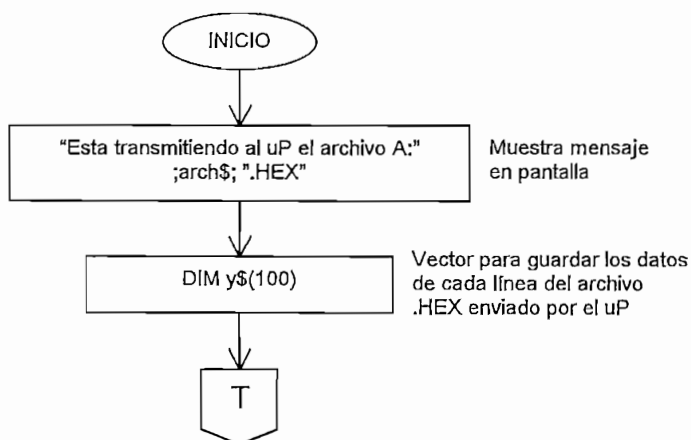
➤ SUBROUTINA txarchivo

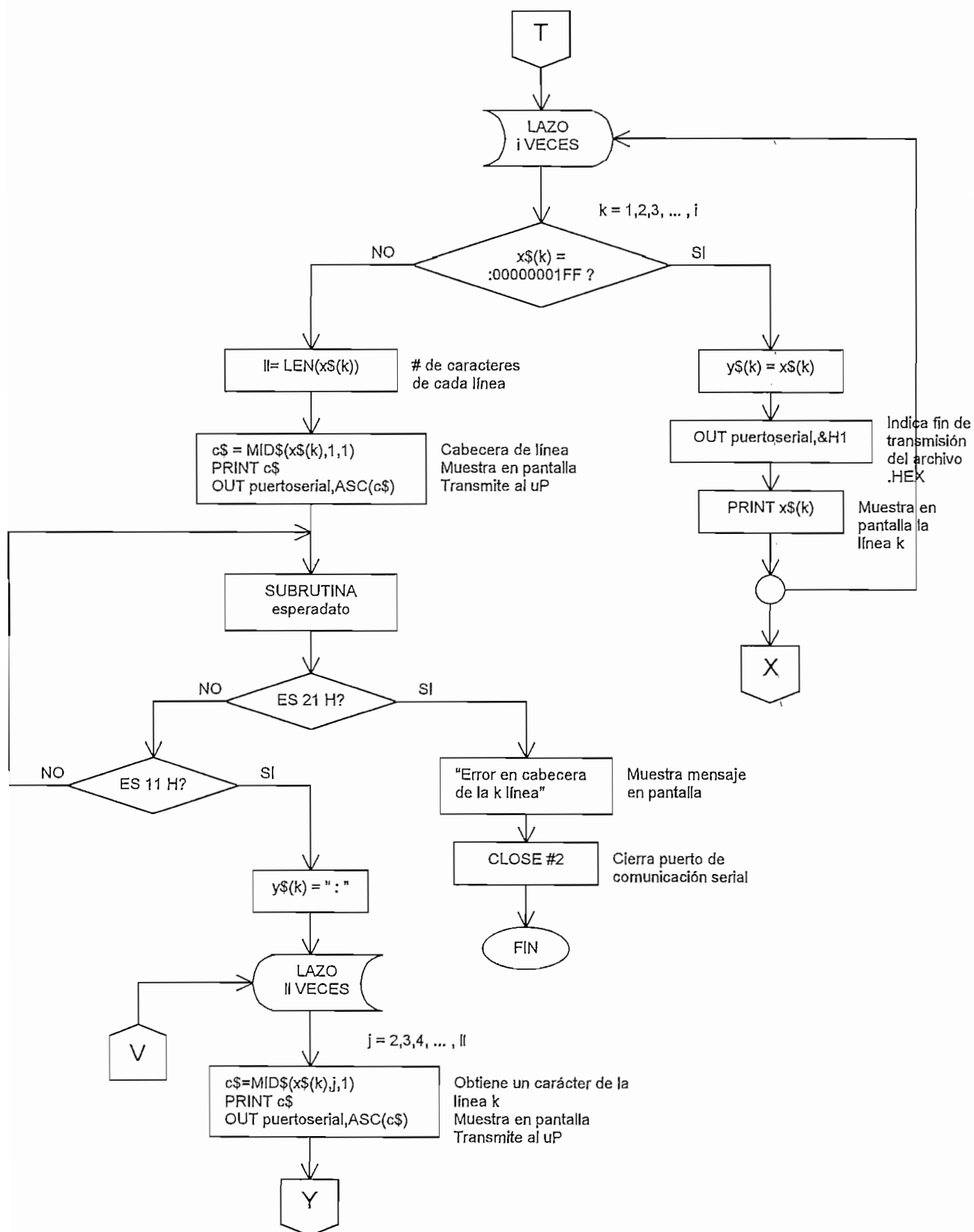
Ejecuta un lazo1 tantas veces como líneas de mensaje tenga el archivo .HEX.

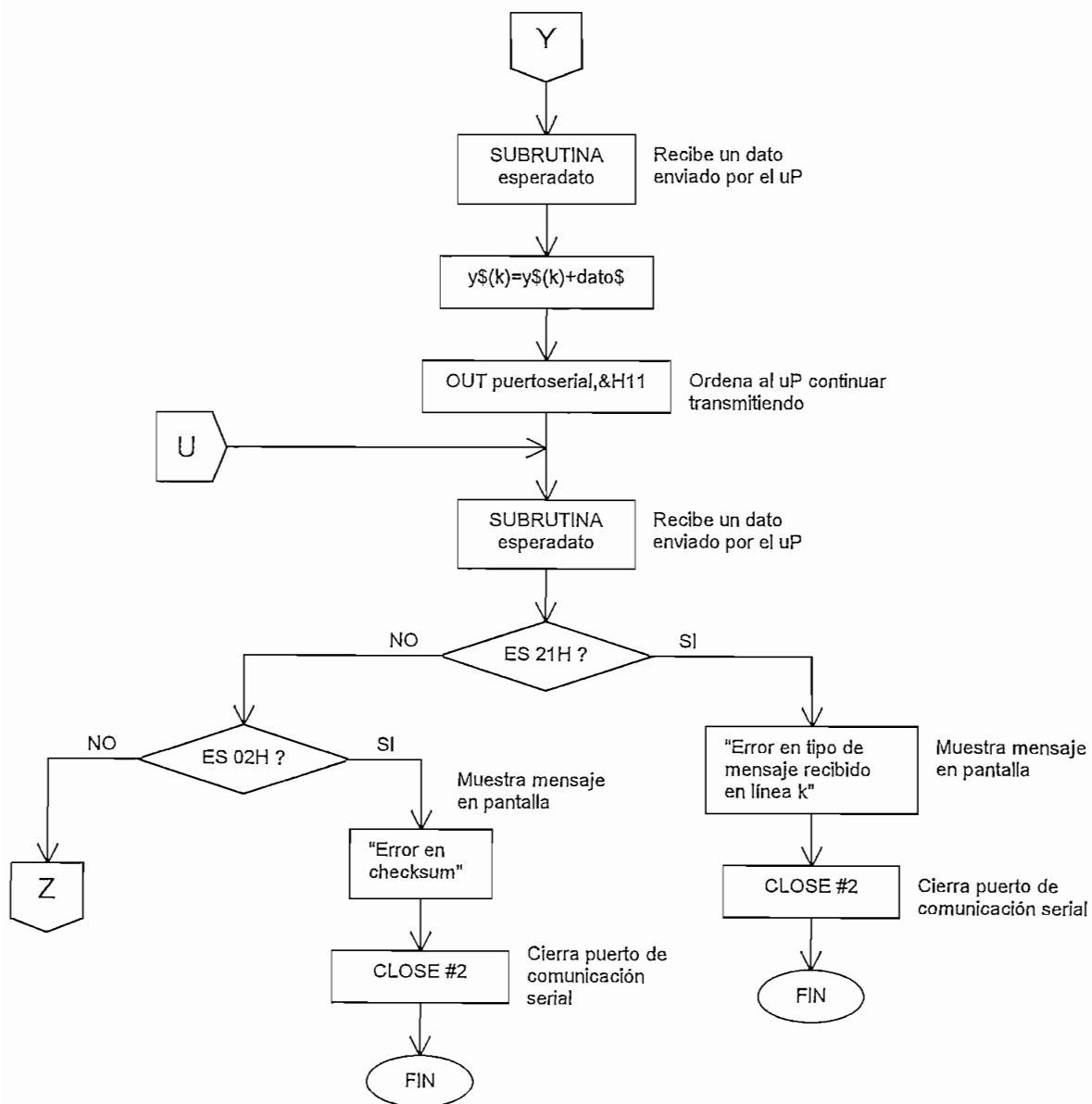
Si la línea leída corresponde a " :00000001FF " , indica al microprocesador fin de transmisión del archivo .HEX, muestra en pantalla los caracteres retransmitidos por el microprocesador y que se guardaron en el vector y\$(l). Borra la pantalla y termina la subrutina.

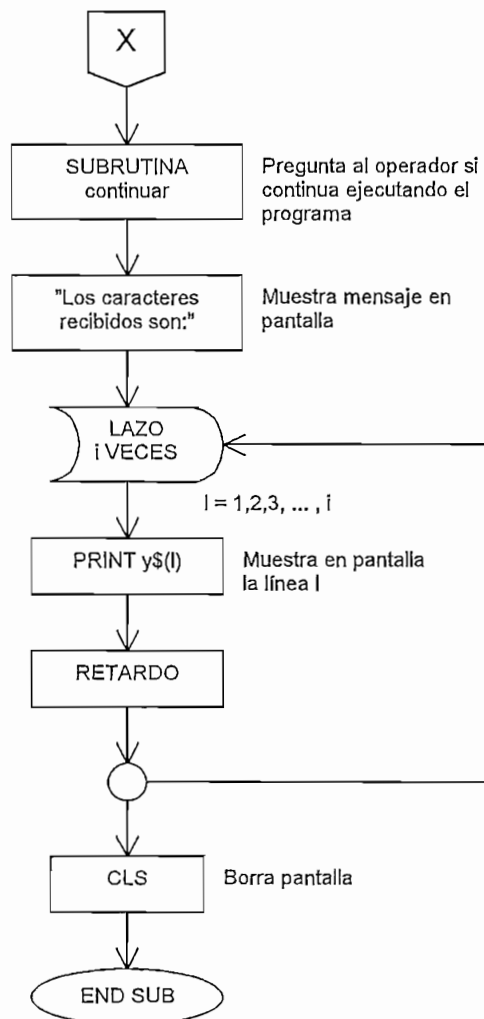
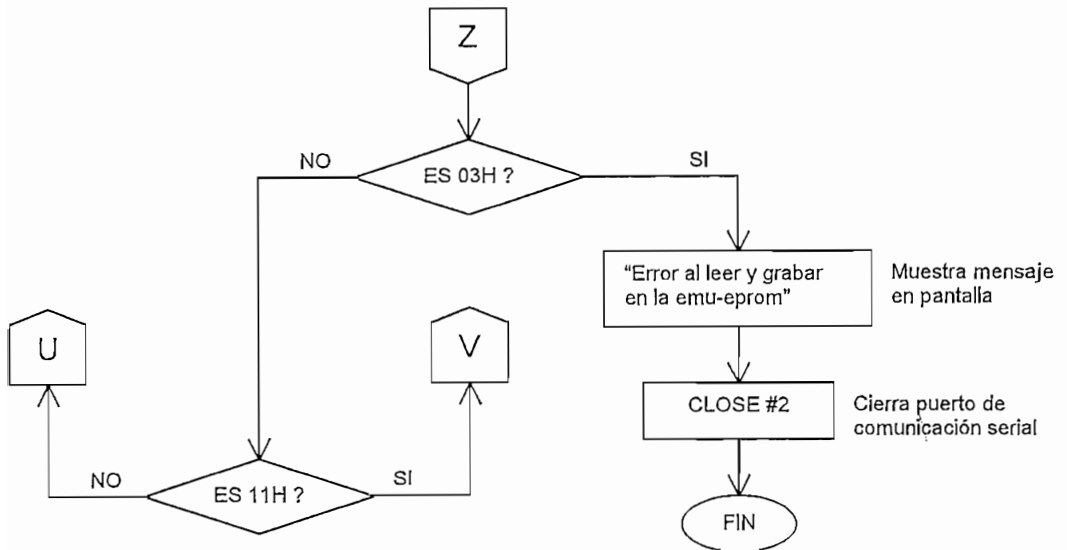
Si la línea leída es diferente a " :00000001FF " , determina el número de caracteres, toma le primer carácter que corresponde a " :", muestra en pantalla y lo transmite al microprocesador. Entra en un lazo hasta recibir respuesta del microprocesador y continua, o termina ejecución si ha ocurrido error en la transmisión. Si recibe el carácter 21H indica error en cabecera de mensaje recibido; si es 11H continua ejecutando el programa, caso contrario repite el lazo. Ejecuta un lazo2 tantas veces como caracteres tenga el mensaje, excepto el " : " .Obtiene el siguiente carácter de la línea y lo transmite al microprocesador, espera que él lo retransmita y lo guarda en el vector y\$(k).

Entra en un lazo hasta recibir respuesta del microprocesador y continua, o termina ejecución si ha ocurrido error en la transmisión. Si el carácter recibido es 21H indica error en tipo de mensaje recibido, si es 02H error en checksum, si es 03H indica que no se grabo dato en la emu-eprom, y si es 11H repite el lazo2, esto es obtiene el siguiente carácter del mensaje, caso contrario regresa al lazo de recibir respuesta del microprocesador.



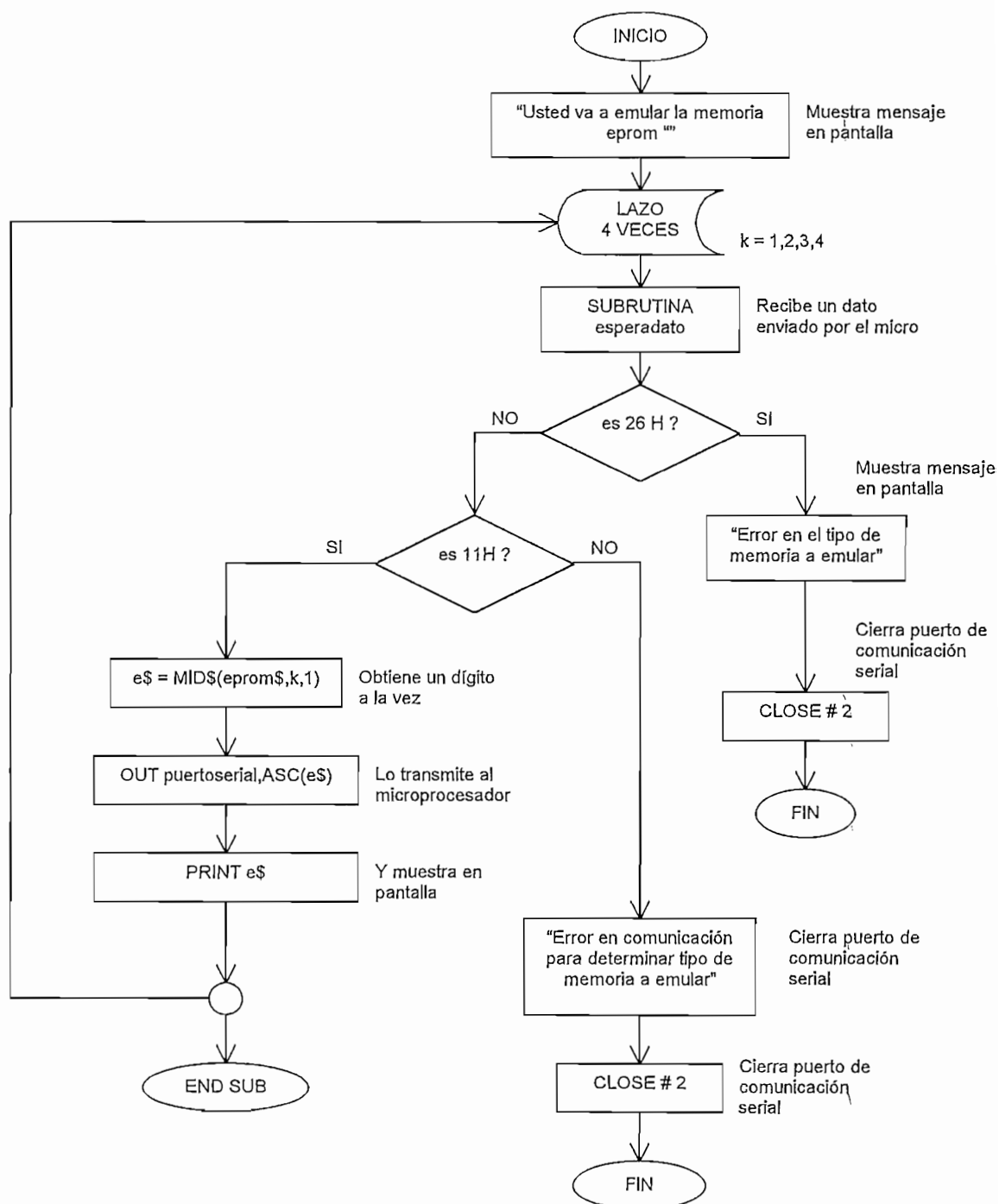






➤ SUBROUTINA txtipoeprom

Muestra en pantalla el tipo de memoria Eprom a emular, para luego transmitir al microprocesador, empezando por el dígito más significativo, Si hubiera algún error en la transmisión, reporta un mensaje de error.



3.4.2 LISTADO DEL PROGRAMA EMULA

```
'EMULA.BAS
'
'Programa de control en la computadora personal
'Este programa es el complemento del programa PROG
'
'          ESCUELA POLITECNICA NACIONAL
'          INGENIERIA EN ELECTRONICA Y TELECOMUNICACIONES
'          TESIS DE GRADO
'TEMA:     EMULADOR DE MEMORIAS EPROM
'FECHA:    MARZO-2002
'REALIZO:   JORGE A. BENITEZ E.
'DIRIGIO:  ING. RAMIRO MOREJON
'
DECLARE SUB capacidademueprom (puertoserial!, texto$, memoria)
DECLARE SUB continuar ()
DECLARE SUB diremueprom (puertoserial, dirmemoria)
DECLARE SUB esperalih ()
DECLARE SUB esperadato (dato$)
DECLARE SUB fuente ()
DECLARE SUB final (dato$, n)
DECLARE SUB hexdec (dato2$, digito2)
DECLARE SUB leerarchivo (i, x$(), arch$)
DECLARE SUB recepcion (dato$, n)
DECLARE SUB resetup ()
DECLARE SUB tipoeprom (eprom$)
DECLARE SUB txarchivo (i!, x$(), puertoserial!, arch$)
DECLARE SUB txtipoeprom (eprom$, puertoserial!)
'
DIM x$(256) 'Vector para almacenar de cada línea del archivo .HEX
CALL leerarchivo(i, x$(), arch$) 'Lee el archivo .HEX
CALL continuar
CALL tipoeprom(eprom$)
'Abre el puerto de comunicación serial
OPEN "com1:2400,n,8,1,rs,cs,ds,cd" FOR RANDOM AS #2
puertoserial = &H3F8
CALL fuente
WHILE EOF(2) = 0 'Lee los datos del buffer de recepción hasta
    dato$ = INPUT$(1, 2) 'que se vacie
WEND
inicio:
    IF EOF(2) = -1 THEN
        CALL resetup
        GOTO inicio
    ELSE
        dato$ = INPUT$(1, 2)
        IF ASC(dato$) = &H11 THEN
            ELSE
                IF ASC(dato$)=&H26 THEN
                    PRINT "La EMU-EPROM no esta en buen estado"
                    CLOSE #2
                END
            END IF
        CALL resetup
        GOTO inicio
    END IF
END IF
CLS
```

```

PRINT "Esta recibiendo la capacidad de la emu-eprom"
PRINT
'Recibe la capacidad de memoria de la emu-eprom
texto$ = "Emu-eprom tiene una capacidad de"
CALL capacidademueprom(puertoserial, texto$, memoria)
CALL txtipoeprom(eprom$, puertoserial)
CALL continuar
CALL diremueprom(puertoserial, dirmemoria)
CALL continuar
CALL esperallh
'Transmite al uP el archivo .HEX
CALL txarchivo(i, x$(), puertoserial, arch$)
CALL continuar
CLS
PRINT "Esta recibiendo la capacidad de memoria de la emu-eprom"
PRINT "usada por el programa a emular"
PRINT
texto$ = "De la emu-eprom, el programa a emular a usado"
CALL capacidademueprom(puertoserial, texto$, memoria)
CALL continuar
'
'
'Programa para recibir el contenido almacenado en la emu-eprom
CALL esperallh
OUT puertoserial, &H11
CLS
PRINT "Lectura de datos almacenados en la emu-eprom"
PRINT
memoria = 8192
FOR i = 1 TO memoria
  datoemu = 0
  FOR l = 1 TO 2
    CALL esperadato(dato$)
    IF ASC(dato$) = &H6 THEN
      GOTO fin2
    END IF
    CALL hexdec(dato$, digito2)
    datoemu = datoemu + digito2 * 16 ^ (2 - l)
    OUT puertoserial, &H11
  NEXT
  datoemuh$ = HEX$(datoemu)
  PRINT datoemuh$;
NEXT
fin2:PRINT
  PRINT "Proceda a emular la emu-eprom"
  CLOSE #2
  END
'
'
Area de subrutinas:
'
SUB capacidademueprom (puertoserial, texto$, memoria)
  'Inicializa parámetro para almacenar la capacidad de
  'memoria de la emu-eprom
  memoria = 0
  FOR j = 1 TO 4
    'Transmite orden para que el prototipo envíe datos
    OUT puertoserial, &H11
    'Recibe un dígito a la vez
    CALL esperadato(dato$)

```

```

'Obtiene la capacidad de memoria de la emu-eprom en
'formato decimal
CALL hexdec(dato$, digito2)
memoria = memoria + digito2 * 16 ^ (4 - j)
NEXT
memoriah$ = HEX$(memoria)
PRINT texto$; memoria; "o ";
PRINT memoriah$; "H bytes"
END SUB

SUB continuar
continua: PRINT
PRINT "Desea continuar (s/n)? "
b$ = INPUT$(1)
IF b$ = "n" OR b$ = "N" THEN
CLS
CLOSE #2
END
ELSE
IF b$ = "s" OR b$ = "S" THEN
ELSE
GOTO continua
END IF
END IF
END SUB

SUB diremueprom (puertoserial, dirmemoria)
'Inicializa parametro para almacenar la dirección desde
'la cual se almacenan los datos en la emu-eprom
CLS
dirmemoria = 0
FOR j = 1 TO 2
CALL esperadato(dato$)
CALL hexdec(dato$, digito2)
dirmemoria = dirmemoria + digito2 * 16 ^ (4 - j)
OUT puertoserial, &H11
NEXT
dirmemoriah$ = HEX$(dirmemoria)
PRINT "La dirección desde la cual se almacenan los datos"
PRINT "en la emu-eprom es "; dirmemoriah$; " H"
END SUB

SUB esperallh
'
'Si el dato recibido es igual a 11H continua con el proceso
'
n = 1
esperal: CALL recepcion(dato$, n)
IF ASC(dato$) = &H11 THEN
ELSE
GOTO esperal
END IF
END SUB

SUB esperadato (dato$)
n = 1
CALL recepcion(dato$, n)
END SUB

```

```

SUB fuente
'Indica que primero se debe encender la fuente de poder
inicio11:  CLS
           PRINT "Presione barra espaciadora para detener proceso"
           PRINT
           PRINT "Si es primera vez encienda la fuente de poder"
inicio21:  PRINT
           PRINT "Ya encendio la fuente de poder?(s/n)"
           tecla$ = INPUT$(1)
           IF tecla$ = " " THEN
               CLS
               PRINT "El proceso ha sido interrumpido por el operador"
               CLOSE #2
               END
           END IF
           IF tecla$ = "n" OR tecla$ = "N" THEN
               GOTO inicio11
           ELSE
               IF tecla$ = "s" OR tecla$ = "S" THEN
                   ELSE
                       GOTO inicio21
                   END IF
           END IF
           CLS
END SUB

```

```

SUB hexdec (dato$, digito2)
'Transforma del formato hexadecimal al formato decimal
IF dato$ = "0" THEN
    digito2 = 0
    GOTO fin
END IF
IF dato$ = "1" THEN
    digito2 = 1
    GOTO fin
END IF
IF dato$ = "2" THEN
    digito2 = 2
    GOTO fin
END IF
IF dato$ = "3" THEN
    digito2 = 3
    GOTO fin
END IF
IF dato$ = "4" THEN
    digito2 = 4
    GOTO fin
END IF
IF dato$ = "5" THEN
    digito2 = 5
    GOTO fin
END IF
IF dato$ = "6" THEN
    digito2 = 6
    GOTO fin
END IF
IF dato$ = "7" THEN
    digito2 = 7
    GOTO fin
END IF

```

```

IF dato$ = "8" THEN
    digito2 = 8
GOTO fin
END IF
IF dato$ = "9" THEN
    digito2 = 9
    GOTO fin
END IF
IF dato$ = "A" THEN
    digito2 = 10
    GOTO fin
END IF
IF dato$ = "B" THEN
    digito2 = 11
    GOTO fin
END IF
IF dato$ = "C" THEN
    digito2 = 12
    GOTO fin
END IF
IF dato$ = "D" THEN
    digito2 = 13
    GOTO fin
END IF
IF dato$ = "E" THEN
    digito2 = 14
    GOTO fin
END IF
IF dato$ = "F" THEN
    digito2 = 15
    GOTO fin
END IF
'Si el dato recibido no corresponde a ninguno de los digitos
'hexadecimales, presenta mensaje de error y finaliza
CLS
PRINT "Error 1 : Error en dato recibido de capacidad de emu-eprom"
CLOSE #2
END
fin:
END SUB

SUB leerarchivo (i, x$(), arch$)
CLS
PRINT "Inserte el diskette conteniendo el archivo .HEX"
PRINT
INPUT "Nombre del archivo .HEX a ser leído desde la unidad A: ",
archivo$
arch$ = archivo$
archivo$ = "A:" + archivo$ + ".HEX"
CLS
PRINT "Espere, esta leyendo el archivo A: "; arch$; ".HEX"
OPEN archivo$ FOR INPUT AS #7          'Abre el archivo .HEX
i = 0                                  'Contador de # de líneas de archivo .HEX
DO UNTIL EOF(7)
    i = i + 1
    INPUT #7, x$(i)
    PRINT x$(i)
    FOR m = 1 TO 15000
        NEXT
LOOP

```

```

CLOSE #7
END SUB

SUB recepcion (dato$, n)
'
'El programa pregunta al PC, hasta 3000 veces, si ha recibido algún
dato 'por el puerto serial, en caso afirmativo lee el dato y continua
con el 'proceso; caso contrario, muestra un mensaje indicando que no
se recibe 'respuesta del prototipo, y finaliza el proceso
'
espera3: IF EOF(2) = -1 THEN
        n = n + 1
        IF n > 3000 THEN
            PRINT "No se recibe respuesta del prototipo"
            PRINT "El proceso ha sido abortado"
            CLOSE #2
            END
        ELSE
            GOTO espera3
        END IF
    ELSE
        dato$ = INPUT$(1, 2)
    END IF
END SUB

SUB resetup
'Indica que primero se debe resetear el uP
inicio1: CLS
        PRINT "Presione barra espaciadora para detener proceso"
        PRINT
        PRINT "Presione RESET del microprocesador"
inicio2: PRINT
        PRINT "Ya reseteo el microprocesador?(s/n)"
        tecla$ = INPUT$(1)
        IF tecla$ = " " THEN
            CLS
            PRINT "El proceso ha sido interrumpido por el operador"
            CLOSE #2
            END
        END IF
        IF tecla$ = "n" OR tecla$ = "N" THEN
            GOTO inicio1
        ELSE
            IF tecla$ = "s" OR tecla$ = "S" THEN
                ELSE
                    GOTO inicio2
            END IF
        END IF
        CLS
END SUB

SUB tipoeprom (eprom$)
otravez1:
PRINT
INPUT "Qué tipo de memoria va a emular: 2716, 2732 o 2764 ?", eprom$
IF eprom$ = "2716" THEN
    GOTO siga
ELSE
    IF eprom$ = "2732" THEN
        GOTO siga

```

```

ELSE
  IF eprom$ = "2764" THEN
    GOTO siga
  ELSE
    CLS
    PRINT "Tipo de memoria mal ingresado, otra vez: "
  END IF
END IF
END IF
GOTO otravez1
siga:
END SUB

SUB txarchivo (i, x$(), puertoserial, arch$)
  CLS
  PRINT "Espere"
  PRINT "Esta transmitiendo al uP el archivo A: "; arch$; ".HEX"
  PRINT
  DIM y$(256)
  FOR k = 1 TO i      'i es el número de líneas del archivo . HEX
    IF x$(k) = ":00000001FF" THEN
      y$(k) = x$(k)
      OUT puertoserial, &H1
      PRINT x$(k)
      GOTO fintx
    END IF
    'Obtiene el # de caracteres de una línea
    ll = LEN(x$(k))
    'Transmite la cabecera de la línea (: )
    c$ = MID$(x$(k), 1, 1)
    PRINT c$;
    OUT puertoserial, ASC(c$)
    'Espera confirmación del uP de que la cabecera ha sido
    'recibida correctamente
repital: CALL esperadato(dato$)
    IF ASC(dato$) = &H21 THEN
      PRINT "Error en cabecera de la "; k; " línea"
      CLOSE #2
      END
    END IF
    IF ASC(dato$) = &H11 THEN
      ELSE
        GOTO repital
      END IF
    y$(k) = ":"
    'Transmite el resto de caracteres de la línea
    FOR j = 2 TO ll
      'Obtiene un carácter a la vez de esa línea
      c$ = MID$(x$(k), j, 1)
      PRINT c$;
      'Transmite dicho carácter
      OUT puertoserial, ASC(c$)
      'Espera a que el uP le envíe el último dato
      CALL esperadato(dato$)
      y$(k) = y$(k) + dato$
      'Ordena al uP que continúe
      OUT puertoserial, &H11
      'Espera orden del uP para continuar o abortar
repita2: CALL esperadato(dato$)
      IF ASC(dato$) = &H21 THEN

```



```

        PRINT
        PRINT "Error en tipo de mensaje recibido en línea "; k
        CLOSE # 2
        END
    END IF
    IF ASC(dato$) = &H2 THEN
        PRINT
        PRINT "Error en CHECKSUM"
        CLOSE #2
        END
    END IF
    IF ASC(dato$) = &H3 THEN
        PRINT
        PRINT "Error al grabar y leer en la emu-eprom"
        CLOSE #2
        END
    END IF
    IF ASC(dato$) = &H11 THEN
    ELSE
        GOTO repita2
    END IF
NEXT
PRINT
fintx: NEXT
CALL continuar
CLS
PRINT "Los caracteres recibidos son :"
FOR l = 1 TO i
    PRINT y$(l)
NEXT
END SUB

SUB txtipoeeprom (eprom$, puertoserial)
    PRINT "Usted va a emular la memoria eprom ";
    FOR k = 1 TO 4
        CALL esperadato(dato$)
        IF ASC(dato$) = &H26 THEN
            PRINT "Error en el tipo de memoria a emular"
            CLOSE #2
            END
        ELSE
        END IF
        IF ASC(dato$) = &H11 THEN
            e$ = MID$(eprom$, k, 1)
            OUT puertoserial, ASC(e$)
            PRINT e$;
        ELSE
            PRINT "Error en comunicación para determinar tipo";
            PRINT " de memoria a emular"
            CLOSE # 2
            END
        END IF
    END IF
NEXT
PRINT
END SUB

```

CAPITULO 4

CONSTRUCCION Y PRUEBAS

- 4.1.- Construcción del prototipo
- 4.2.- Construcción de un sistema microprocesado
- 4.3.- Pruebas

4.1 CONSTRUCCION DEL PROTOTIPO

No obstante que el sistema en su totalidad presenta un hardware relativamente reducido, su implantación se realiza en dos módulos. El diseño modular permite un desarrollo más efectivo, de esta manera, al realizar las pruebas con el segundo módulo y si se tiene alguna falla en los resultados esperados, ésta se centra en el segundo módulo.

Una vez terminado el diseño del hardware, se procede a la implantación.

Con el fin de facilitar la corrección de cualquier error de diseño en el hardware, el primer paso fue armar, en un protoboard, el diseño que se presenta en la figura 2.13 de la página 25 y que corresponde a la Tarjeta # 1. Solo cuando se estuvo seguro de su correcto funcionamiento, se construyó el circuito impreso.

Sabiendo que el software funciona con el hardware de la Tarjeta # 1, se armó el diseño que se presenta en la figura 2.14 de la página 26 y que corresponde a la Tarjeta # 2, igual que la anterior se hizo primero en un protoboard y finalmente en un circuito impreso.

Siempre, deben estar conectadas las dos tarjetas a través de un cable plano de 28 pines y un cable de 1 hilo que conecte los pines 9 de los headers de 40 postes de las Tarjetas # 1 y # 2. Al ejecutar nuevamente el programa, se presentaron fallas, pero dado que ya se realizaron las pruebas en la Tarjeta # 1, el problema era en la segunda tarjeta, una vez realizados los cambios necesarios en el hardware todo funcionó como se esperaba.

Las dos tarjetas están dentro de una caja, con llave triangular, de dimensiones 20x30x15 centímetros; de ésta sale un cable de 3 hilos, terminado en conector DB9, para conectarse al puerto serial de la computadora personal; y, para conectarse al sistema microprocesado existen dos cables planos, uno de 28 pines para cuando se emule una memoria Eprom 2764 y otro cable de 24 pines para cuando se emule a una memoria Eprom 2732 o 2716. Solo uno de estos cables planos debe conectar a la vez.

La fuente de alimentación utilizada es externa al sistema y tiene los siguientes voltajes: +5 V , -5 V, +12 V y -12 V .

4.2 CONSTRUCCION DE UN SISTEMA MICROPROCESADO

Hasta aquí, el prototipo construido ejecutaba bien todos los algoritmos explicados anteriormente, la prueba final y definitiva consistía en conectarlo a un sistema microprocesado y que éste lea los códigos de máquina desde la memoria RAM (emu-eprom).

Dado que no era factible conseguir un sistema microprocesado para cada tipo de memoria Eprom, se procedió a diseñar y construir un circuito impreso que contenga un sistema microprocesado que permita usar, uno a la vez, los tres tipos de memoria Eprom a emular.

Este circuito no es más que una variación del circuito que se presenta en la figura 2.13 de la página 25.

El sistema microprocesado construido se muestra en la figura 4.1, se observa la presencia de un jumper el cual en una posición permite emular la memoria Eprom 2764 o 2732, y en la otra posición a una memoria Eprom 2716. Esto es necesario por cuanto en la memoria 2716, el pin 21 siempre debe estar en nivel alto.

4.3 PRUEBAS

En lo referente al software, se elaboró en forma paralela los programas PROG y EMULA en el microprocesador y computadora personal, respectivamente. Primero se realizó el programa que garantice siempre una comunicación serial entre el microprocesador y la computadora personal. La velocidad de comunicación serial escogida fue de 2400 baudios.

Luego se realizó, en el microprocesador, el algoritmo para determinar el estado de cada una de las celdas en la memoria RAM (emu-eprom) . Para esto, primero se define la máxima capacidad de la emu-eprom (2 kilobytes) , el programa primero escribe en cada celda el valor AAH, lo lee y compara, luego escribe el valor 55H, lo lee y compara; en cualquier caso, si el valor escrito y el leído no es el mismo, el programa se detiene e indica a la computadora personal que la memoria esta en mal estado y termina ejecución del programa;

caso contrario se almacena el valor FFH, y transmite a la computadora personal el valor de la máxima capacidad de la emu-eprom.

Una vez verificado el estado de la emu-eprom, se desarrolló un programa que permita transmitir, desde la computadora personal al microprocesador, el tipo de memoria Eprom a emular; esto es importante por la siguiente razón:

la memoria RAM (emu-eprom) utilizada es de 8Kx8 bits, es decir que con ésta se puede emular a una memoria Eprom 2716 (2Kx8 bits) , 2732 (4Kx8 bits) o 2764 (8Kx8 bits), pero no en todos los tres casos, los datos se deben almacenar desde la localidad 0000H , la explicación es la siguiente:

la memoria Eprom 2716 tiene líneas de direcciones A0 a A10, entonces las direcciones A11 y A12 no están definidas y por lo tanto son consideradas como uno lógico (1L); y cuando se proceda a emular, la primera localidad que leerá el microprocesador sería la 1800H y no la 0000H. Ver cuadro 4.1.

Por tal razón es necesario considerar en el programa, que los datos en la emu-eprom se almacenen desde la dirección 1800H.

A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	0	0	0	0	0	0
1	8			0				0				

Cuadro 4.1

En el caso de la memoria EPROM 2732, la dirección que no está definida es A12. Al emular, la primera localidad que lee el microprocesador es la 1000H y no la 0000H, por tal motivo los datos deben ser guardados desde la localidad 1000H. Ver cuadro 4.2

A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	0	0	0	0	0	0	0	0	0	0	0	0
1	0			0				0				

Cuadro 4.2

Y, solo en el caso de emular a una memoria EPROM 2764, la lectura se realiza desde la localidad 0000H. Ver cuadro 4.3

A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0			0			0					

Cuadro 4.3

Resumiendo, a través de la computadora personal se envía al microprocesador el tipo de memoria EPROM a emular, el microprocesador determina desde que localidad de memoria se guardan los datos en la emu-eprom, y lo transmite a la computadora personal para verificación del operador.

A continuación, se elaboró el algoritmo para transmitir, desde la computadora personal al microprocesador, el archivo (.HEX) en hexadecimal que se será almacenado en la emu-eprom. Complementariamente, determina el número de localidades de memoria que han sido necesarias para guardar el archivo .HEX y lo envía a la computadora personal para verificación del operador.

Finalmente se consideró, un algoritmo que permita leer, desde la dirección 0000H, el contenido de las localidades de la emu-eprom, y enviarlo a la computadora personal para verificación del operador.

Al ejecutar los programas PROG y EMULA hubo fallas, fue necesario hacer cambios, aumentos, hasta que el programa se ejecute adecuadamente. La depuración, del programa que ejecuta el microprocesador 8031, se realizó en base a los utilitarios AVMAC51 y el AVSIM51; mientras que el programa que ejecuta la computadora personal se realizó y depuro con ayuda del QBASIC 45.

La prueba definitiva consistió en emular a una memoria Eprom 2764, para lo cual se conecto un cable plano de 28 pines entre el zócalo U6 de la Tarjeta # 2 del prototipo construido y el otro extremo en el sitio de Eprom del sistema microprocesado.

El programa para probar el funcionamiento del sistema se denomina "ONDA", el cual consiste en obtener en el pin P1.0 del microcontrolador 8031, del

sistema microprocesado, una onda cuadrada de período 1 segundo, como se indica en la figura 4.2

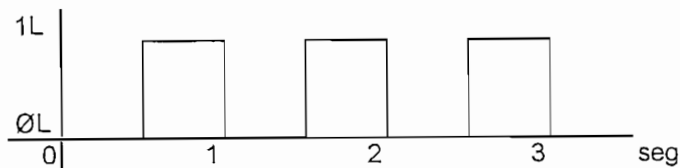


Figura 4.2 Onda periódica de 1 segundo

El listado del programa se indica a continuación:

```
;ONDA.ASM
;
;          ESCUELA POLITECNICA NACIONAL
;          INGENIERIA EN ELECTRONICA Y TELECOMUNICACIONES
;          TESIS DE GRADO
;          TEMA:          EMULADOR DE MEMORIAS EPROM
;          FECHA:        MARZO-2002
;          REALIZO:      JORGE A. BENITEZ E.
;          DIRECTOR:     ING. RAMIRO MOREJON
;
;Programa de prueba para emular
;Genera onda cuadrada de período 1 segundo, cuando la frecuencia del
;oscilador en el microcontrolador del sistema microprocesado es 50 Khz
;La salida se muestra en el pin 0 del puerto 1 (P1.0)
;
;ETIQUETA  OPCODE      OPERANDOS      COMENTARIO
;AREA PARA LA DEFINICION DE ETIQUETAS MEDIANTE LA DIRECTIVA "EQU"
;
PERIODO    EQU    01H    ;Onda de período 1 segundo
;
;ETIQUETA  OPCODE      OPERANDOS      COMENTARIO
;
          DEFSEG  CERO, START=0
          SEG     CERO
          JMP     CICLO
          DEFSEG  SINT1, START=SINT, CLASS=CODE
          SEG     SINT1
          JMP     SERIAL
          DEFSEG  PROGRAMA, CLASS=CODE, START=30H
          SEG     PROGRAMA
;
CICLO:    MOV     R1, #PERIODO ;Nivel bajo
          CLR     P1.0
SIGA1:    LCALL  TIEMPO
          DJNZ   R1, SIGA1
          MOV    R1, #PERIODO ;Nivel alto
```



```

                SETB  P1.0
SIGA2          LCALL  TIEMPO
                DJNZ  R1,SIGA2
                SJMP  CICLO
                SJMP  $           ;FIN DEL PROGRAMA
;
;
;Area para subrutinas
TIEMPO:        MOV    TMOD,#10H           ;Esta subrutina realiza un lapso de
                MOV    TCON,#00H         ;tiempo de 0.50 segundos
                MOV    TH1,#0F7H
                MOV    TL1,#0DDH
                SETB   TR1
ESPERA1:       JNB    TF1,ESPERA1        ;cuenta 2.083 ciclos de máquina
                CLR    TF1
                CLR    TR1
                RET
SERIAL:
                RETI
                END           ;FIN DEL ENSAMBLADO

```

A continuación se muestra el archivo ONDA.HEX

```

:03000000020030CB
:0300230002005C7C
:2000300007901C290120046D9FB7901D290120046D9FB80EC80FE758910758800758DF77544D
:0D0050008BDDD28E308FFDC28FC28E22322A
:00000001FF

```

Siguiendo las instrucciones del Manual del Usuario, ver Anexo 1, se obtuvo como resultado lo esperado, esto es, emular a una memoria Eprom 2764; con ayuda de un osciloscopio y la punta de prueba concetada en el pin P1.0 del microprocesador del sistema microprocesado se vio en su pantalla la forma de onda que se indica en la figura 4.2 . El siguiente paso fue variar el programa ONDA, por ejemplo que la forma de onda sea de período 2 segundos, para lo cual bastaba con modificar el valor de la etiqueta PERIODO de 01H a 02H, grabar el nuevo programa y con ayuda de los utilitarios AVMAC51 y AYSIM51 obtener el nuevo archivo .HEX.

Repitiendo los pasos anteriores se obtuvo en la pantalla del osciloscopio la forma de onda que se muestra en la figura 4.2.

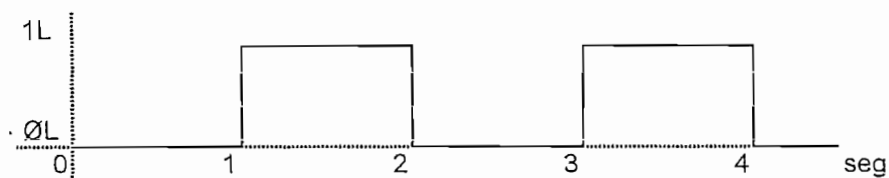


Figura 4.2 Onda periódica de 2 segundos

Es importante anotar que estos resultados no se obtuvieron en la primera prueba, es más, aparentemente el sistema microprocesado no leía a la emu-eprom; después de revisar cuidadosamente el hardware y software y ver que sus diferentes etapas no presentaban ningún error, se optó por variar la frecuencia a la que trabaja el sistema microprocesado, para sorpresa, éste lee a la emu-eprom solo en un rango de frecuencias, y no en todas como se pensaba en un inicio, el rango de frecuencias a las que funciona el prototipo es entre 30 KHz y 270 KHz.

A pesar de este limitante, todo el hardware y software funciona como se concibió desde un inicio.

Al realizar las siguientes pruebas, esto es, emulando a una memoria Eprom 2732 o 2716 los resultados fueron los mismos.

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1.- Conclusiones y recomendaciones.

5.1 CONCLUSIONES Y RECOMENDACIONES

Del diseño y construcción de este prototipo denominado EMULADOR DE MEMORIAS EPROM se concluye lo siguiente:

- El sistema desarrollado está sujeto a mejorarlo con el fin de optimizar su funcionamiento tanto en lo que a hardware y software se refiere.
- No funciona para todas las frecuencias, debido a que la memoria RAM utilizada trabaja a frecuencias menores que las del microprocesador. Si se quiere aumentar el rango de frecuencias a las que funcione el prototipo será necesario conseguir una memoria RAM (emu-eprom) de mayor velocidad.
- La computadora personal a usarse no necesariamente funciona a la misma frecuencia del microprocesador, es necesario variar un retardo, en la subrutina "recepcion" del programa EMULA, hasta que el dato enviado desde el microprocesador llegue a la computadora personal.
- Las conexiones de los diferentes componentes del prototipo se realizaron en circuitos impresos de doble lado, para realizar mejores soldaduras de lado y lado sin sobrecalentar a los integrados; se utilizaron zócalos de características adecuadas lo cual permite el reemplazo cómodo de cualquier integrado defectuoso.
- Si se requiere construir otro prototipo de similares características, esto tomaría poco tiempo debido a la facilidad de construir los circuitos impresos con la técnica fotomecánica, la cual permite a partir de los negativos construir los circuitos impresos que se requieran.
- El desarrollo del programa de control PROG, para el microcontrolador 8031, requirió de hacer cambios, y por cada cambio fue necesario grabar el programa con un grabador de memorias Eprom, para el siguiente trabajo se grabó unas 25 veces. En el mercado existen personas que tienen los grabadores de memorias Eprom, el costo por cada grabación es de 3 dólares, y en algunos casos le entregan al otro día. Entonces, para este trabajo se necesitaría hacer un egreso de 75 dólares. Se ve claramente que el costo para desarrollar un solo programa es alto, sin tomar en cuenta el tiempo que se tarda en mandar a grabarlo.

- Los tiempos en desarrollar y depurar un programa se reduce, ya que los cambios y aumentos en el programa se hacen en ese mismo momento, se graba en la emu-eprom y se prueba inmediatamente.
- El costo del prototipo realizado es alrededor de 280 dólares, incluido la fuente de poder; se asume que se dispone de una computadora personal, por lo que se concluye que solo al desarrollar los primeros programas se recupera la inversión.
- Para este trabajo se utilizó una fuente de voltaje externa, es posible que una mejora a este prototipo consista en tener como fuente de voltaje a la propia de la computadora personal.
- El archivo .HEX se lee desde el driver A, una mejora sería que se lo pueda hacer también desde el disco duro.
- Los programas de control PROG y EMULA se los presenta de manera escrita y con el diagrama de flujo respectivo, permitiendo entender rápida y globalmente su funcionamiento.
- El programa EMULA permite al operador salir en cualquier etapa de ejecución del programa.
- El prototipo construido es de gran utilidad como herramienta de depuración de programas, ahorrando tiempo y esfuerzo.
- Las simulaciones poco objetivas que entorpecen la depuración de programas ya no son un problema, porque con este prototipo los programas se ejecutan en tiempo real.
- Los elementos utilizados en la construcción están disponibles en el mercado.

BIBLIOGRAFIA

- 1.- GONZALEZ, JOSÉ , Introducción a los microcontroladores
Hardware, software y aplicaciones
McGraw-Hill 1992
- 2.- AVOCET SYSTEMS INC. Avmac 8051 User's Manual
1986
- 3.- AVOCET SYSTEMS INC. Avsim51 8051 Family User's Manual
1986
- 4.- TEXAS INSTRUMENTS TTL Data Book
1985
- 5.- NATIONAL SEMICONDUCTOR Interface Databook
- 6.- WASHINGTON, SIDEL Modulo de desarrollo para sistemas
basados en los microcontroladores de
la familia MCS-51/52 .
Tesis de Grado 1993

A N E X O S

ANEXO 1 : Manual del usuario del emulador de memorias

- Descripción general
- Operación

ANEXO 2 : Descripción de los elementos utilizados

ANEXO 3 : Características de los elementos utilizados

ANEXO 4 : Manual de la familia MCS-51/52

ANEXO 5 : Distribución de los componentes en las Tarjetas 1 y 2

ANEXO 6 : CD con los archivos PROG y EMULA

ANEXO 1 : MANUAL DEL USUARIO DEL EMULADOR DE MEMORIAS

El Emulador de Memorias Eprom esta formado por :

- La Tarjeta # 1
- La Tarjeta # 2
- Cable para comunicación serial, terminado en conector DB 9 macho en un extremo y en DB 9/ DB 25 hembra en el otro extremo.
- Dos cables plano terminado en conectores de 28 pines.
- Un cable plano terminado en conector de 24 pines.
- Un cable plano terminado en conector de 40 pines.
- Baterías de 1.5 V x 4 (opcional)
- Fuente externa de alimentación para voltajes de +/- 5 V y +/- 12 V.

DESCRIPCION GENERAL

En la Tarjeta # 1 se tiene un pulsador de reset para iniciar la ejecución del programa PROG.

Un extremo del primer cable plano terminado en conector de 28 pines se conecta en el sitio de la RAM (zócalo U4) de la Tarjeta # 1 y el otro extremo en el zócalo U1 de la Tarjeta # 2 .

Para emular a una memoria Eprom 2764, se conecta un extremo, del segundo cable plano terminado en conector de 28 pines, en el zócalo de la Tarjeta # 2 identificado como U6 y el otro extremo en el sitio de la Eprom 2764 del Sistema Microprocesado.

Al emular una memoria Eprom 2732 o 2716, se conecta un extremo, del cable plano terminado en conector de 24 pines, en el zócalo de la Tarjeta # 2 identificado como U7 y el otro extremo en el sitio de la Eprom 2732 o 2716 del Sistema Microprocesado.

En cualquier caso, siempre debe estar conectado, un cable desde el pin 9 del header de 40 postes de la Tarjeta # 1 con el pin 9 del header de 40 postes de la Tarjeta # 2. Si este pin esta en un nivel bajo indica que se está procediendo

a recibir el archivo .HEX a emular y se graba en la emu-eprom; si está a nivel alto indica que se puede proceder a emular la memoria eprom.

También hay la alternativa de no utilizar el primer cable plano para 28 pines, en su lugar se puede emplear el cable plano para 40 pines, el cual conecta los headers de 40 pines de las Tarjetas #1 y #2.

En la Tarjeta # 2 se dispone de un switch que permite energizar a los componentes.

OPERACIÓN

La operación es bastante simple: primero se debe asegurar que el programa PROG.HEX esté grabado en la memoria Eprom 2764 de la Tarjeta # 1 y el programa EMULA.BAS en la computadora personal.

Seguidamente se debe verificar que la Tarjeta # 1 y el Sistema Microprocesado estén conectados a la fuente de alimentación, y el switch de la Tarjeta # 2 en la posición ON.

Verificar que los pines 9 de los headers de 40 postes estén conectados.

Al ejecutar, en la computadora personal, el programa EMULA.BAS , éste va interactuando paso a paso con el usuario. Al inicio le indica que debe energizar a las tarjetas y seguidamente que pulse el reset de la Tarjeta # 1, solicita que ingrese el nombre del archivo .HEX a emular, etc.

Conforme avanza la ejecución del programa, después de cada etapa, pregunta al usuario si desea continuar o terminar.

Finalmente, le indica al usuario que proceda a emular la memoria Eprom, para lo cual solo basta con presionar el reset del Sistema Microprocesado.

ANEXO 2 : DESCRIPCION DE LOS ELEMENTOS UTILIZADOS

CANTIDAD	DENOMINACION	DESCRIPCION
1	MCS 8031	Microcontrolador Intel de 40 pines
1	2764	Memoria Eprom de 8 K x 8 bits
1	MCM 6064	Memoria Ram de 8K x 8 bits
1	74LS373	Latch
1	MC 1488	Interfaz de TTL-RS232
1	MC 1489	Interfaz de RS232-TTL
4	74LS157	Multiplexores 2 a 1
2	74LS245	Transceptor
2	74LS04	Inversor
1	Oscilador	12 Mhz
1	Resistencia	680 ohmios
4	Condensadores electroliticos	1 uF - 50 V
2	Condensadores	27 uF
4	1N4007	Diodo de silicio

**ANEXO 3 : CARACTERÍSTICAS DE LOS ELEMENTOS
UTILIZADOS**



2764 (5133)
64K EPROM
27128 (5143)
128K EPROM

PRODUCT DESCRIPTION

December 1983

Features

- 200 ns Access Times at 0 to 70°C
- Programmed Using Intelligent Algorithm
 - Typically 5 ms/byte Programming Time
 - 2 Minutes for 27128 (5143)
 - 1 Minute for 2764 (5133)
- JEDEC Approved Byte-wide Pin Configuration
 - 2764 8K x 8 Organization
 - 27128 16K x 8 Organization
- Low Power Dissipation
 - 100 mA Active Current
 - 30 mA Standby Current
- Extended Temperature Range Available
- Silicon Signature™

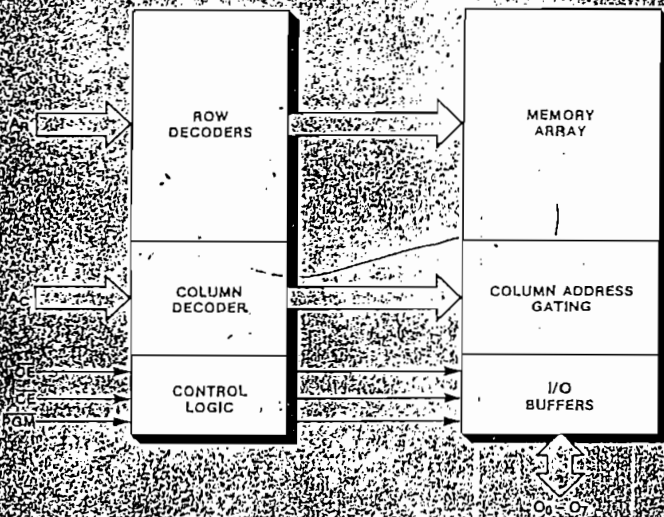
Description

SEEQ's 2764 (5133) and 27128 (5143) are ultraviolet light erasable EPROMs which are organized 8K x 8 and 16K x 8 respectively. They are pin for pin compatible to JEDEC approved 64K and 128K EPROMs in all operational/programming modes. Both devices have access times as fast as 200 ns over the 0 to 70°C temperature and V_{CC} tolerance range. The access time is achieved without sacrificing power, since the maximum active and standby currents are 100 mA and 30 mA respectively. The 200 ns allows higher system efficiency by eliminating the need for wait states in today's 8- or 16-bit microcomputers.

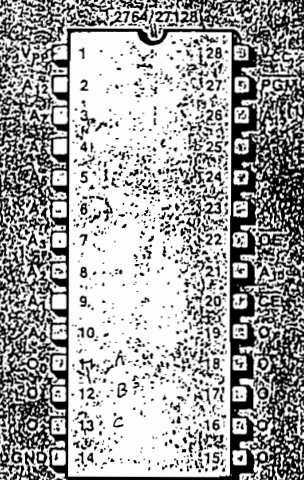
Initially, and after erasure, all bits are in the "1" state. Data is programmed by applying 21 V to V_{PP} and a TTL "0" to pin 27 (program pin). The 2764 (5133) and 27128 (5143) may be programmed with an intelligent

(continued on page 2)

Block Diagram



Pin Configuration



Mode Selection

MODE	PINS	CE (20)	OE (22)	PGM (27)	V _{pp} (1)	V _{CC} (28)	Outputs (11-13, 15-19)
Read		V _{IL}	V _{IL}	V _{IH}	V _{CC}	V _{CC}	DOUT
Standby		V _{IH}	X	X	V _{CC}	V _{CC}	High Z
Program		V _{IL}	V _{IH}	V _{IL}	V _{PP}	V _{CC}	DIN
Program Verify		V _{IL}	V _{IL}	V _{IH}	V _{PP}	V _{CC}	DOUT
Program Inhibit		V _{IH}	X	X	V _{PP}	V _{CC}	High Z
Silicon Signature™		V _{IL}	V _{IL}	V _{IH}	V _{CC}	V _{CC}	Encoded Data

Pin Names

A _C	ADDRESSES — COLUMN (LSB)
A _R	ADDRESSES — ROW
CE	CHIP ENABLE
OE	OUTPUT ENABLE
O ₀ - O ₇	OUTPUTS
PGM	PROGRAM

can be either V_{IL} or V_{IH}.
 For Silicon Signature™: A₀-A₃ are toggled, A₄=V_{IL}, A₅=12V, all other addresses are at any TTL level.

2764 (5133) 27128 (5143)

PRODUCT DESCRIPTION

algorithm that is now available on commercial programmers. The programming time is typically 5 ms/byte or 2 minutes for all 16K bytes of the 27128. The 2764 requires only half of this time, about a minute for 8K bytes. This faster time improves manufacturing throughput time by hours over conventional 50 ms algorithms. Commercial programmers (e.g. Data I/O, Pro-log, Digelec, Kontron, and Stag) have implemented this fast algorithm for SEEQ's EPROMs. If desired, both EPROMs may be

programmed using the conventional 50 ms programming specification of older generation EPROMs.

Incorporated on SEEQ's EPROMs is Silicon Signature™. Silicon Signature contains encoded data which identifies SEEQ as the EPROM manufacturer, the product's fab location, and programming information. This data is encoded in ROM to prevent erasure by ultraviolet light.

Absolute Maximum Stress Ratings

Temperature

Storage -65° C to +150° C

Under Bias -10° C to +80° C

All Inputs or Outputs with

Respect to Ground +7V to -0.6V

V_{PP} During Programming with

Respect to Ground +22V to -0.6V

Voltage on A₉ with*

Respect to Ground +15.5V to -0.6V

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

Recommended Operating Conditions (27XX = 2764 and 27128)††

	27XX-200, 27XX-250 27XX-300, 27XX-450	27XX-2, 27XX-3, 27XX-4
V _{CC} Supply Voltage ²	5 V ± 10%	5 V ± 5%
Temperature Range (Read Mode)	0 to 70° C	0 to 70° C
V _{PP} During Programming	21 ± 0.5 V	21 ± 0.5 V

DC Operating Characteristics During Read or Programming

Symbol	Parameter	Limits		Unit	Test Conditions
		Min.	Max.		
I _{IN}	Input Leakage Current		10	μA	V _{IN} = V _{CC} Max.
I _O	Output Leakage Current		10	μA	V _{OUT} = V _{CC} Max.
I _{PP1} ²	V _{PP} Current	Read Mode	5	mA	V _{PP} = V _{CC} Max.
		Prog. Mode	30	mA	V _{PP} = 21.5V
I _{CC1} ²	V _{CC} Standby Current		30	mA	$\overline{CE} = V_{IH}$
I _{CC2} ²	V _{CC} Active Current		100	mA	$\overline{CE} = \overline{OE} = V_{IL}$
V _{IL}	Input Low Voltage	-0.1	0.8	V	
V _{IH}	Input High Voltage	2	V _{CC} + 1	V	
V _{OL}	Output Low Voltage		0.45	V	I _{OL} = 2.1 mA
V _{OH}	Output High Voltage	2.4		V	I _{OH} = -400 μA

NOTES:

1. The 5133 and 5143 have the same dash numbers and operate with the same operating conditions as the 2764 and 27128 respectively. The specifications are exactly the same.

2. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.

2764 (5133)
27128 (5143)
PRODUCT DESCRIPTION

AC Operating Characteristics During Read

Symbol	Parameter	Limits (nsec)								Test Conditions
		27XX-2 27XX-200		27XX-250		27XX-3 27XX-300		27XX-4 27XX-450		
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
t_{ACC}	Address to Data Valid		200		250		300		450	$\overline{CE} = \overline{OE} = V_{IL}$
t_{CE}	Chip Enable to Data Valid		200		250		300		450	$\overline{OE} = V_{IL}$
t_{OE}	Output Enable to Data Valid		75		100		120		150	$\overline{CE} = V_{IL}$
t_{DF}	Output Enable to Output Float	0	60	0	60	0	105	0	130	$\overline{CE} = V_{IL}$
t_{OH}	Output Hold from Chip Enable, Addresses, or Output Enable whichever occurred first	0		0		0		0		$\overline{CE} = \overline{OE} = V_{IL}$

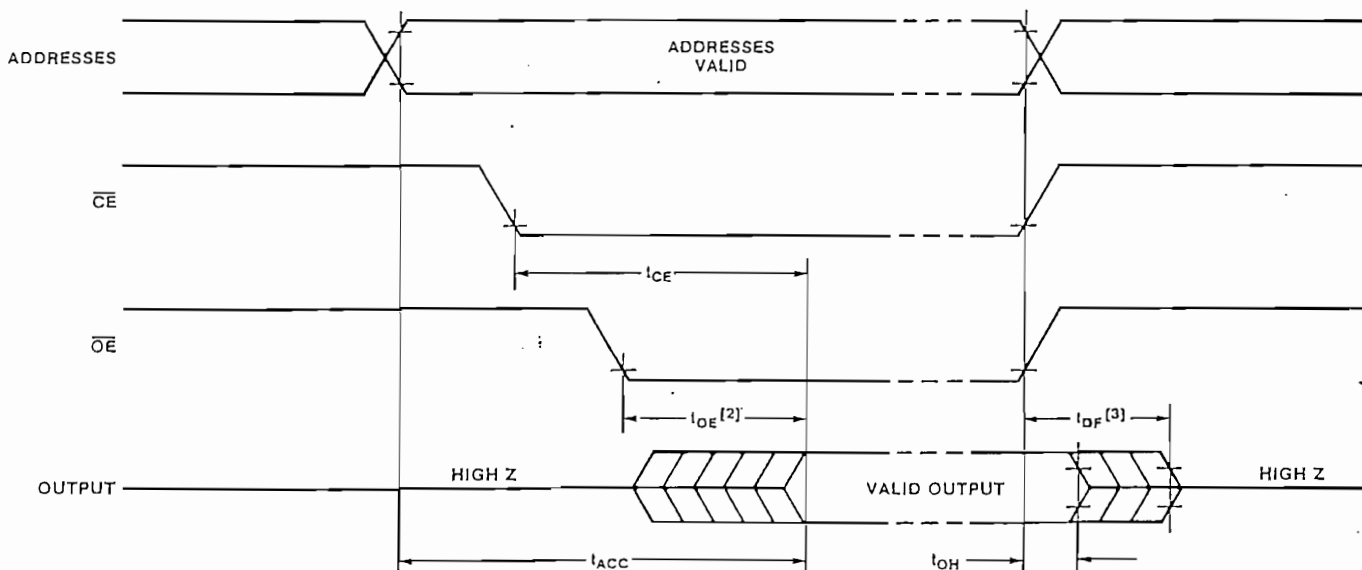
Capacitance¹⁾

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT} = 0V$

A.C. Test Conditions

Output Load: 1 TTL gate and $C_L = 100$ pF
 Input Rise and Fall Times: ≤ 20 ns
 Input Pulse Levels: 0.45V to 2.4V
 Timing Measurement Reference Level:
 Inputs 1V and 2V
 Outputs 0.8V and 2V

A.C. Waveforms



- NOTES:
 1. THIS PARAMETER IS SAMPLED AND IS NOT 100% TESTED.
 2. \overline{OE} MAY BE DELAYED UP TO $t_{ACC} - t_{OE}$ AFTER THE FALLING EDGE OF \overline{CE} WITHOUT IMPACT ON t_{ACC} .
 3. t_{DF} IS SPECIFIED FROM \overline{OE} OR \overline{CE} , WHICHEVER OCCURS FIRST.

MEMORIA RAM DE 8Kx8

PIN CONFIGURATION
(Top View)

NC	1	28	VCC
A12	2	27	WE
A7	3	26	CE2
A6	4	25	A8
A5	5	24	A9
A4	6	23	A11
A3	7	22	OE
A2	8	21	A10
A1	9	20	CE1
A0	10	19	I/O 7
I/O 0	11	18	I/O 6
I/O 1	12	17	I/O 5
I/O 2	13	16	I/O 4
GND	14	15	I/O 3

PIN DESCRIPTION

Symbol	Description
A0 to A12	Address input
I/O 0 to I/O 7	Data input output
CE1, CE2	Chip enable 1, 2 input
WE	Write enable input
OE	Output enable input
VCC	+ 5V Power supply
GND	Ground
NC	Non connection

TRUTH TABLE

CE1	CE2	OE	WE	MODE	I/O 0 to I/O 7
H	X	X	X	Not selected	High Z
X	L	X	X	Not selected	High Z
L	H	H	H	Output disable	High Z
L	H	L	H	Read	Data out
L	H	X	L	Write	Data in

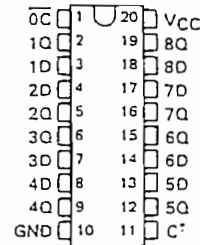
X: "H" or "L"

TYPES SN54LS373, SN54LS374, SN54S373, SN54S374, SN74LS373, SN74LS374, SN74S373, SN74S374[†] OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

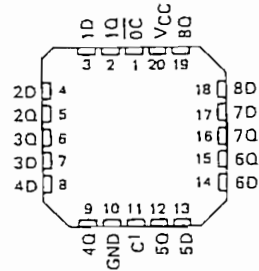
OCTOBER 1975—REVISED APRIL 1985

- Choice of 8 Latches or 8 D-Type Flip-Flops in a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel-Access for Loading
- Buffered Control Inputs
- Clock/Enable Input Has Hysteresis to Improve Noise Rejection ('S373 and 'S374)
- P-N-P Inputs Reduce D-C Loading on Data Lines ('S373 and 'S374)

SN54LS373, SN54LS374, SN54S373,
SN54S374 ... J PACKAGE
SN74LS373, SN74LS374, SN74S373,
SN74S374 ... DW, J OR N PACKAGE
(TOP VIEW)



SN54LS373, SN54LS374, SN54S373,
SN54S374 ... FK PACKAGE
SN74LS373, SN74LS374, SN74S373,
SN74S374 ... FN PACKAGE
(TOP VIEW)



'LS373, 'S373
FUNCTION TABLE

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	[†] Q ₀
H	X	X	Z

'LS374, 'S374
FUNCTION TABLE

OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q ₀
H	X	X	Z

description

These 8-bit registers feature three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-type latches meaning that while the enable (C) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

[†]C for 'LS373 and 'S373; CLK for 'LS374 and 'S374

TYPES SN54LS373, SN54LS374, SN54S373, SN54S374,
 SN74LS373, SN74LS374, SN74S373, SN74S374
 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

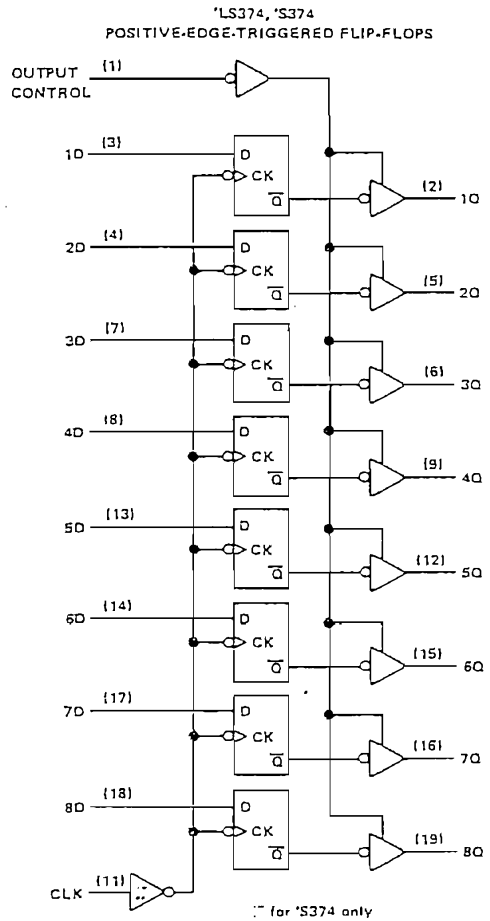
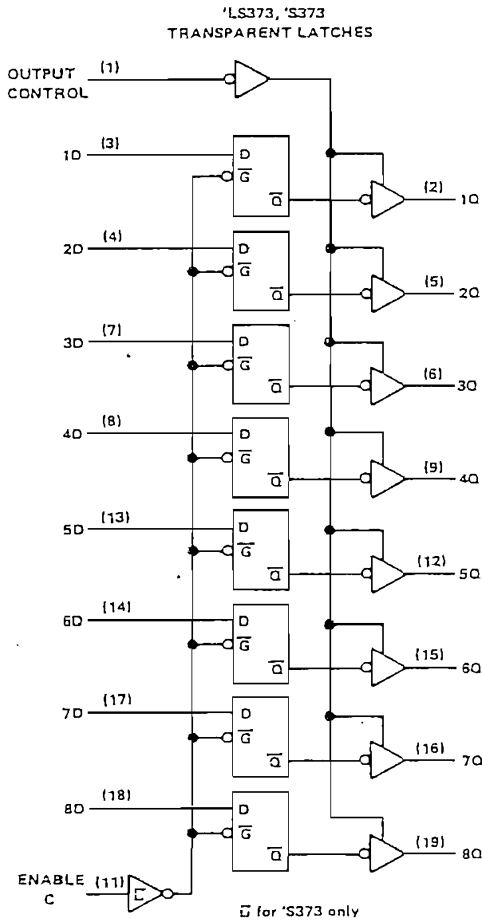
description (continued)

The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were setup at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines of the 'S373 and 'S374 devices, simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

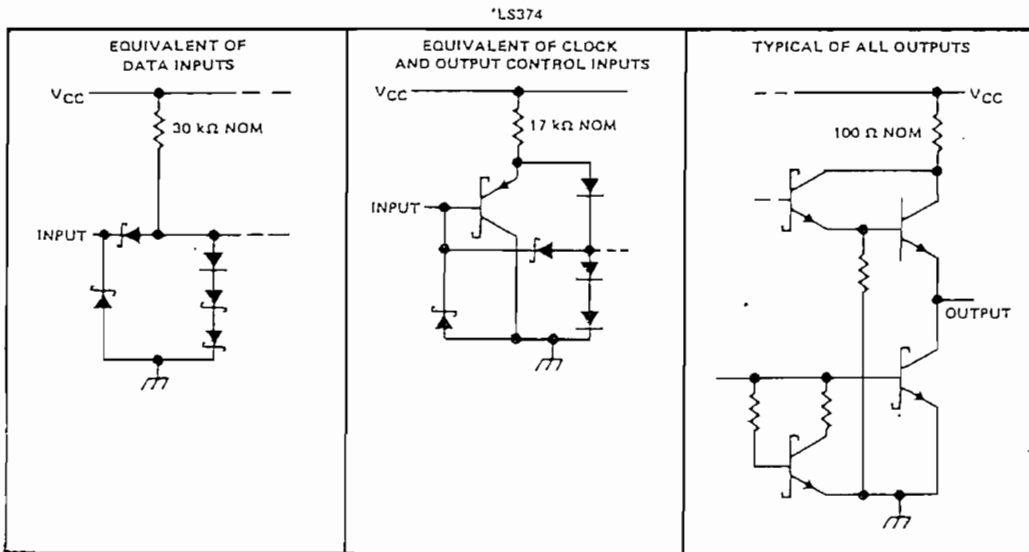
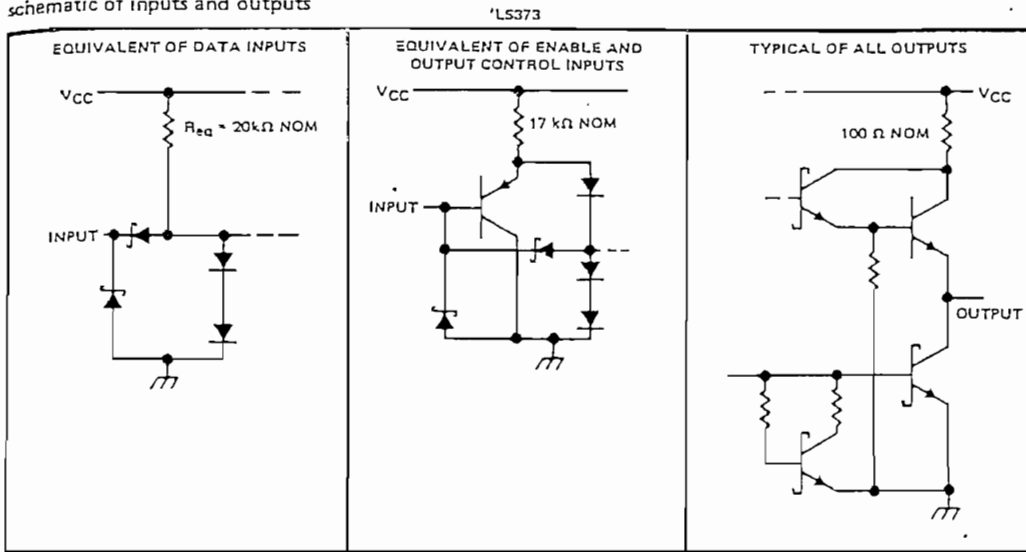
logic diagrams



Pin numbers shown on logic notation are for DW J or P packages

TYPES SN54LS373, SN54LS374, SN74LS373, SN74LS374
 OCTAL D-TYPE TRANSPARENT LATCHES AND
 EDGE-TRIGGERED FLIP-FLOPS

schematic of inputs and outputs



TYPES SN54LS373, SN54LS374, SN74LS373, SN74LS374
OCTAL D-TYPE TRANSPARENT LATCHES AND
EDGE-TRIGGERED FLIP-FLOPS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	7 V
Off-state output voltage	5.5 V
Operating free-air temperature range: SN54LS*	-55°C to 125°C
SN74LS*	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

		SN54LS*			SN74LS*			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{OH}	High-level output voltage			5.5			5.5	V
I_{OH}	High-level output current			-1			-2.6	mA
I_{OL}	Low-level output current			12			24	mA
t_w	Pulse duration	CLK high		15			15	ns
		CLK low		15			15	ns
t_{SU}	Data setup time	*LS373		5			5	ns
		*LS374		20†			20†	ns
t_H	Data hold time	*LS373		20			20	ns
		*LS374‡		0†			0†	ns
T_A	Operating free-air temperature	-55		125	0		70	°C

† The t_H specification applies only for data frequency below 10 MHz. Designs above 10 MHz should use a minimum of 5 ns.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54LS*			SN74LS*			UNIT	
		MIN	TYP‡	MAX	MIN	TYP‡	MAX		
V_{IH}	High-level input voltage	2			2			V	
V_{IL}	Low-level input voltage			0.7			0.8	V	
V_{IK}	Input clamp voltage	$V_{CC} = \text{MIN}$, $I_I = -18 \text{ mA}$		-1.5			-1.5	V	
V_{OH}	High-level output voltage	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = V_{IL \text{ max}}$, $I_{OH} = \text{MAX}$	2.4	3.4		2.4	3.1	V	
V_{OL}	Low-level output voltage	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = V_{IL \text{ max}}$		0.25	0.4		0.25	0.4	V
		$I_{OL} = 12 \text{ mA}$ $I_{OL} = 24 \text{ mA}$					0.25	0.5	
I_{OZH}	Off-state output current, high-level voltage applied	$V_{CC} = \text{MAX}$, $V_{IH} = 2 \text{ V}$, $V_O = 2.7 \text{ V}$		20			20	μA	
I_{OZL}	Off-state output current, low-level voltage applied	$V_{CC} = \text{MAX}$, $V_{IH} = 2 \text{ V}$, $V_O = 0.4 \text{ V}$		-20			-20	μA	
I_I	Input current at maximum input voltage	$V_{CC} = \text{MAX}$, $V_I = 7 \text{ V}$		0.1			0.1	mA	
I_{IH}	High-level input current	$V_{CC} = \text{MAX}$, $V_I = 2.7 \text{ V}$		20			20	μA	
I_{IL}	Low-level input current	$V_{CC} = \text{MAX}$, $V_I = 0.4 \text{ V}$		-0.4			-0.4	mA	
I_{OS}	Short-circuit output current‡	$V_{CC} = \text{MAX}$	-30	-130	-30		-130	mA	
I_{CC}	Supply current	$V_{CC} = \text{MAX}$, Output control at 4.5 V		24	40		24	40	mA
		*LS373 *LS374		27	40		27	40	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.

TYPES SN54LS373, SN54LS374, SN74LS373, SN74LS374
OCTAL D-TYPE TRANSPARENT LATCHES AND
EDGE-TRIGGERED FLIP-FLOPS

switching characteristics, $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$

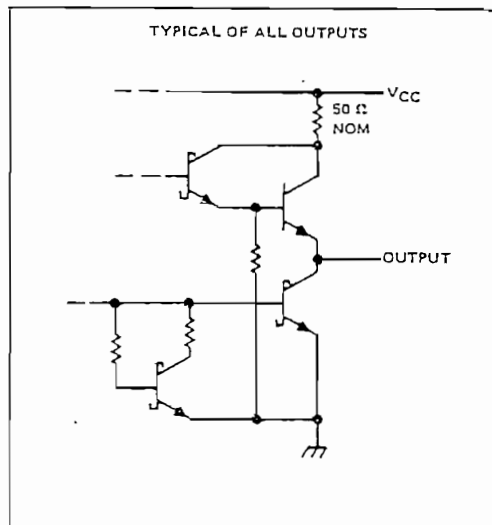
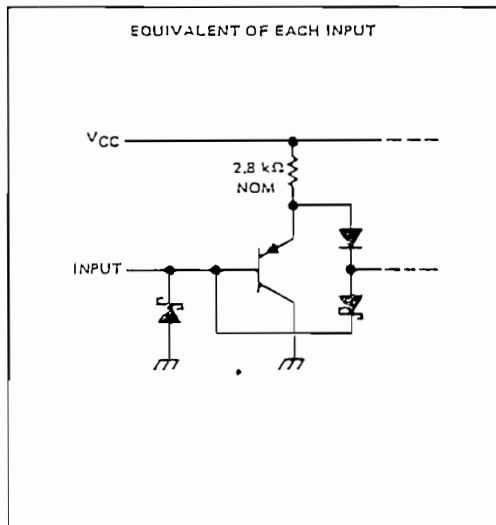
PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'LS373			'LS374			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}			$C_L = 45\text{ pF}$, $R_L = 667\ \Omega$ See Notes 2 and 3				35	50		MHz
t_{PLH}	Data	Any Q		12	18					ns
t_{PHL}				12	18					ns
t_{PLH}	Clock or enable	Any Q		20	30		15	28		ns
t_{PHL}				18	30		19	28		ns
t_{PZH}	Output Control	Any Q		15	28		20	26		ns
t_{PZL}			25	36		21	28		ns	
t_{PHZ}	Output Control	Any Q	$C_L = 5\text{ pF}$, $R_L = 667\ \Omega$ See Note 3	SN54	28	32		28	32	ns
				SN74	15	25		15	28	ns
t_{PLZ}	Output Control	Any Q _s		12	20		12	20		ns

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.
3. See General Information Section for load circuits and voltage waveforms.

f_{max} = maximum clock frequency
 t_{PLH} = propagation delay time, low-to-high-level output
 t_{PHL} = propagation delay time, high-to-low-level output
 t_{PZH} = output enable time to high level
 t_{PZL} = output enable time to low level
 t_{PHZ} = output disable time from high level
 t_{PLZ} = output disable time from low level

TYPES SN54S373, SN54S374, SN74S373, SN74S374
OCTAL D-TYPE TRANSPARENT LATCHES AND
EDGE-TRIGGERED FLIP-FLOPS

schematic of inputs and outputs



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	5.5 V
Off-state output voltage	5.5 V
Operating free-air temperature range: SN54S*	-55°C to 125°C
SN74S*	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1. Voltage values are with respect to network ground terminal.

recommended operating conditions

	SN54S*			SN74S*			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output voltage, V_{OH}	5.5			5.5			V
High-level output current, I_{OH}	-2			-5.5			mA
Width of clock/enable pulse, t_w	High	6		6		ns	
	Low	7.3		7.3			
Data setup time, t_{su}	'S373	0↓		0↓		ns	
	'S374	5↑		5↑			
Data hold time, t_h	'S373	10↓		10↓		ns	
	'S374	2↑		2↑			
Operating free-air temperature, T_A	-55		125	0		70	C

TYPES SN54S373, SN54S374, SN74S373, SN74S374
 OCTAL D-TYPE TRANSPARENT LATCHES AND
 EDGE-TRIGGERED FLIP-FLOPS

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†				MIN	TYP‡	MAX	UNIT
V _{IH}					2			V
V _{IL}							0.8	V
V _{IK}	V _{CC} = MIN, I _I = -18 mA						-1.2	V
V _{OH}	SN54S'	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OH} = MAX		2.4	3.4		V	
	SN74S'			2.4	3.1			
V _{OL}	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = 0.8 V, I _{OL} = 20 mA						0.5	V
I _{OZH}	V _{CC} = MAX, V _{IH} = 2 V, V _O = 2.4 V						50	μA
I _{OZL}	V _{CC} = MAX, V _{IH} = 2 V, V _O = 0.5 V						-50	μA
I _I	V _{CC} = MAX, V _I = 5.5 V						1	mA
I _{IH}	V _{CC} = MAX, V _I = 2.7 V						50	μA
I _{IL}	V _{CC} = MAX, V _I = 0.5 V						-250	μA
I _{OS} §	V _{CC} = MAX				-40		-100	mA
I _{CC}	V _{CC} = MAX		'S373	outputs high			160	mA
				outputs low			160	
				outputs disabled			190	
			'S374	outputs high			110	
				outputs low			140	
				outputs disabled			160	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at V_{CC} = 5 V, T_A = 25°C.

§ Not more than one output should be shorted at a time and duration of the short-circuit should not exceed one second.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'S373			'S374			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
f _{max}			C _L = 15 pF, R _L = 280 Ω, See Notes 2 and 4				75	100		MHz
t _{PLH}	Data	Any Q		7	12					ns
t _{PHL}				7	12					
t _{PLH}	Clock or enable	Any Q		7	14		8	15		ns
t _{PHL}				12	18		11	17		ns
t _{PZH}	Output Control	Any Q		8	15		8	15		ns
t _{PZL}			11	18		11	18		ns	
t _{PHZ}	Output Control	Any Q	C _L = 5 pF, R _L = 280 Ω, See Note 3		6	9	5	9		ns
t _{PLZ}			8	12		7	12		ns	

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.

4. See General Information Section for load circuits and voltage waveforms.

f_{max} = maximum clock frequency

t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

t_{PZH} = output enable time to high level

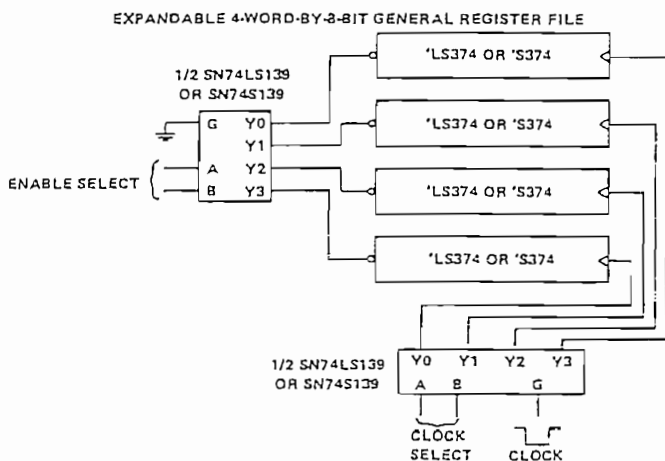
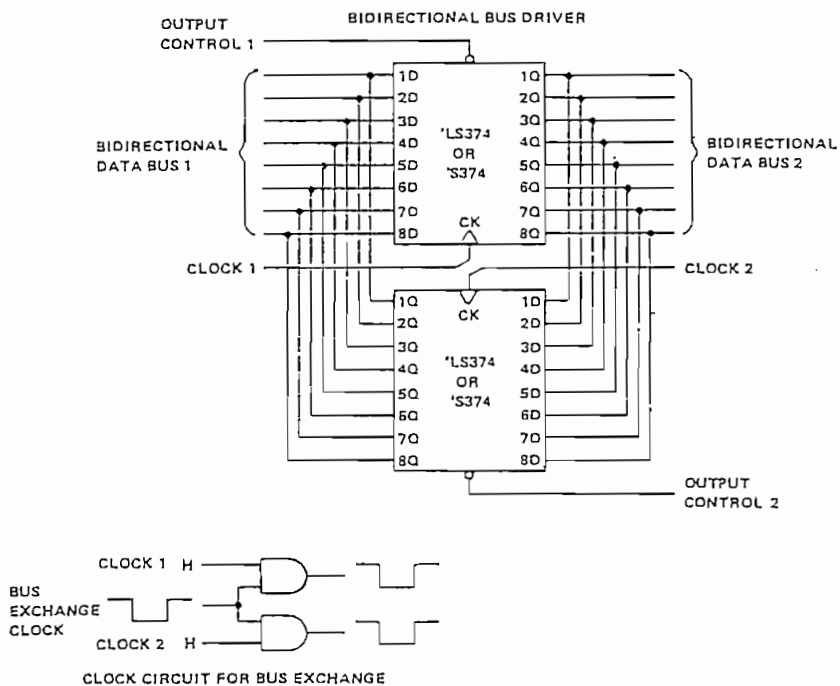
t_{PZL} = output enable time to low level

t_{PHZ} = output disable time from high level

t_{PLZ} = output disable time from low level

TYPES SN54LS373, SN54LS374, SN54S373, SN54S374,
 SN74LS373, SN74LS374, SN74S373, SN74S374
 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

TYPICAL APPLICATION DATA





Transmission Line Drivers/Receivers

DS1488 quad line driver

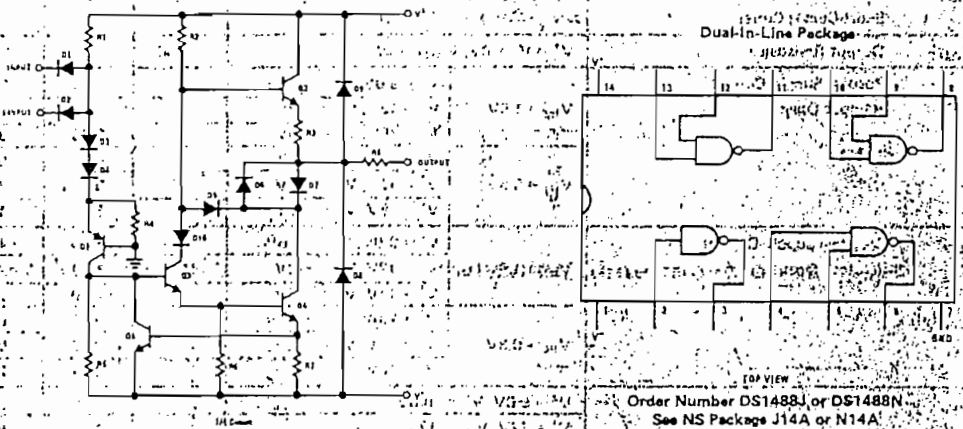
general description

The DS1488 is a quad line driver which converts standard DTL/TTL input logic levels through one stage of inversion to output levels which meet EIA Standard No. RS-232C and CCITT Recommendation V.24.

features

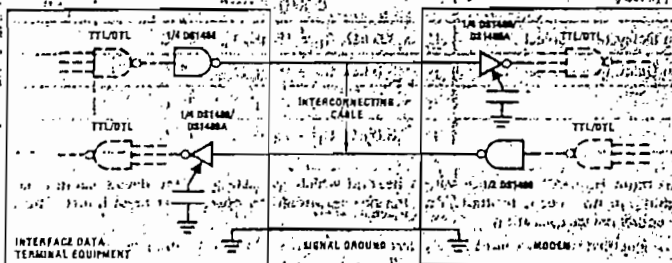
- Current limited output ± 10 mA typ
- Power-off source impedance 300Ω min
- Simple slew rate control with external capacitor
- Flexible operating supply range
- Inputs are DTL/TTL compatible

schematic and connection diagrams



typical applications

RS232C Data Transmission



DS1488

absolute maximum ratings (Note 1)

Supply Voltage	
V^+	+15V
V^-	-15V
Input Voltage (V_{IN})	$-15V \leq V_{IN} \leq 7.0V$
Output Voltage	$\pm 15V$
Operating Temperature Range	$0^\circ C$ to $+75^\circ C$
Storage Temperature Range	$-65^\circ C$ to $+150^\circ C$
Lead Temperature (Soldering, 10 sec)	$300^\circ C$

electrical characteristics (Notes 2, 3 and 4)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
I_{IL}	Logical "0" Input Current $V_{IN} = 0V$		-1.0	-1.3	mA
I_{IH}	Logical "1" Input Current $V_{IN} = +5.0V$		0.005	10.0	μA
V_{OH}	High Level Output Voltage $R_L = 3.0 k\Omega$, $V_{IN} = 0.8V$	$V^+ = 9.0V, V^- = -9.0V$ 6.0	7.0		V
		$V^+ = 13.2V, V^- = -13.2V$ 9.0	10.5		V
V_{OL}	Low Level Output Voltage $R_L = 3.0 k\Omega$, $V_{IN} = 1.9V$	$V^+ = 9.0V, V^- = -9.0V$ -6.8	-6.0	-6.0	V
		$V^+ = 13.2V, V^- = -13.2V$ -10.5	-9.0	-9.0	V
I_{OS}^+	High Level Output Short-Circuit Current $V_{OUT} = 0V, V_{IN} = 0.8V$	-6.0	-10.0	-12.0	mA
I_{OS}^-	Low Level Output Short-Circuit Current $V_{OUT} = 0V, V_{IN} = 1.9V$	6.0	10.0	12.0	mA
R_{OUT}	Output Resistance $V^+ = V^- = 0V, V_{OUT} = \pm 2V$	300			Ω
I_{CC}^+	Positive Supply Current (Output Open)				
	$V_{IN} = 1.9V$	$V^+ = 9.0V, V^- = -9.0V$	15.0	20.0	mA
		$V^+ = 12V, V^- = -12V$	19.0	25.0	mA
		$V^+ = 15V, V^- = -15V$	25.0	34.0	mA
	$V_{IN} = 0.8V$	$V^+ = 9.0V, V^- = -9.0V$	4.5	6.0	mA
		$V^+ = 12V, V^- = -12V$	5.5	7.0	mA
		$V^+ = 15V, V^- = -15V$	8.0	12.0	mA
I_{CC}^-	Negative Supply Current (Output Open)				
	$V_{IN} = 1.9V$	$V^+ = 9.0V, V^- = -9.0V$	-13.0	-17.0	mA
		$V^+ = 12V, V^- = -12V$	-18.0	-23.0	mA
		$V^+ = 15V, V^- = -15V$	-25.0	-34.0	mA
	$V_{IN} = 0.8V$	$V^+ = 9.0V, V^- = -9.0V$	-0.001	-0.015	mA
		$V^+ = 12V, V^- = -12V$	-0.001	-0.015	mA
		$V^+ = 15V, V^- = -15V$	-0.01	-2.5	mA
P_D	Power Dissipation				
		$V^+ = 9.0V, V^- = -9.0V$	252	333	mW
		$V^+ = 12V, V^- = -12V$	444	576	mW

switching characteristics ($V_{CC} = 9V, V_{EE} = -9V, T_A = 25^\circ C$)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
t_{pd1}	Propagation Delay to a Logical "1"		230	350	ns
t_{pd0}	Propagation Delay to a Logical "0"		70	175	ns
t_r	Rise Time		75	100	ns
t_f	Fall Time		40	75	ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range", they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified min/max limits apply across the $0^\circ C$ to $+75^\circ C$ temperature range for the DS1488.

Note 3: All currents into device pins shown as positive, out of device pins as negative, all voltages referenced to ground unless otherwise noted. All values shown as max or min on absolute value basis.

DS1488

applications

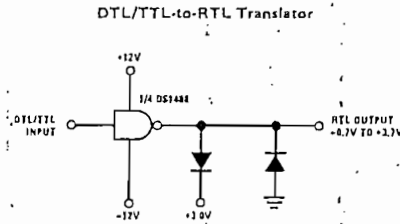
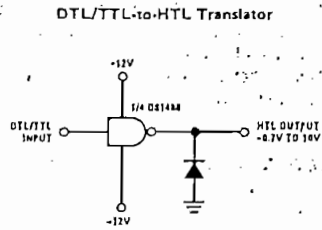
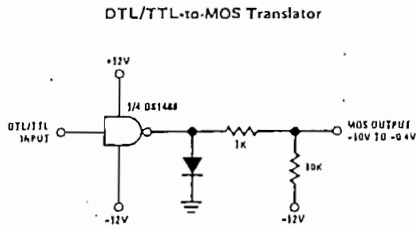
By connecting a capacitor to each driver output the slew rate can be controlled utilizing the output current limiting characteristics of the DS1488. For a set slew rate the appropriate capacitor value may be calculated using the following relationship

$$C = I_{SC} (\Delta T / \Delta V)$$

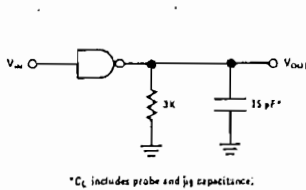
where C is the required capacitor, I_{SC} is the short circuit current value, and $\Delta V / \Delta T$ is the slew rate.

RS232C specifies that the output slew rate must not exceed 30V per microsecond. Using the worst case output short circuit current of 12 mA in the above equation, calculations result in a required capacitor of 400 pF connected to each output.

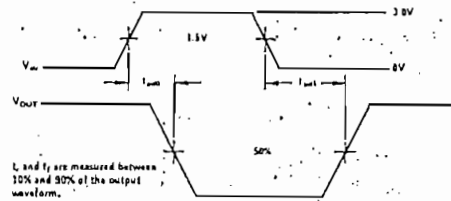
typical applications (con't)



ac load circuit

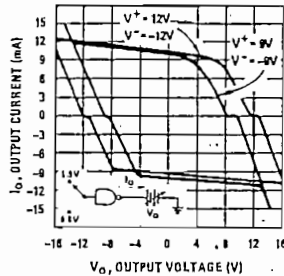


switching time waveforms



typical performance characteristics

Output Voltage and Current-Limiting Characteristics





Transmission Line Drivers/Receivers

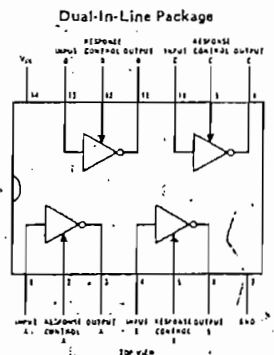
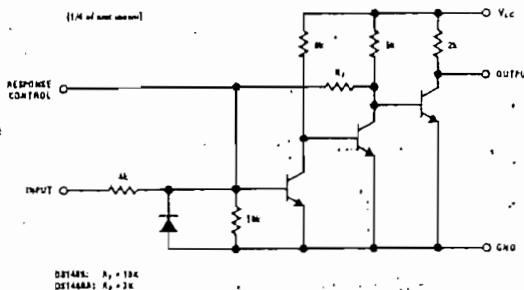
DS1489/DS1489A quad line receiver general description

The DS1489/DS1489A are quad line receivers designed to interface data terminal equipment with data communications equipment. They are constructed on a single monolithic silicon chip. These devices satisfy the specifications of EIA standard No. RS232C. The DS1489/DS1489A meet and exceed the specifications of MC1489/MC1489A and are pin-for-pin replacements. The DS1489/DS1489A are available in 14-lead ceramic dual-in-line package.

features

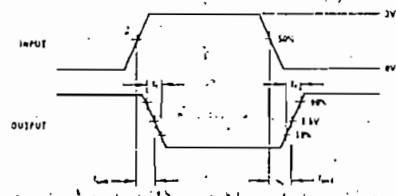
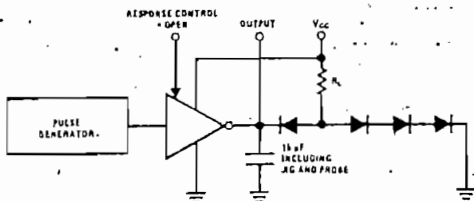
- Four totally separate receivers per package
- Programmable threshold
- Built-in input threshold hysteresis
- "Fail safe" operating mode
- Inputs withstand $\pm 30V$

schematic and connection diagrams

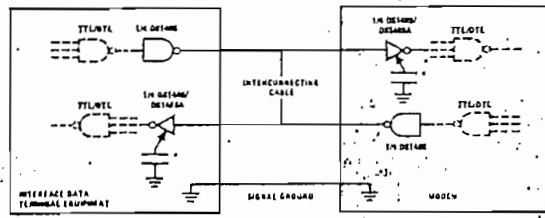


Order Number DS1489J, DS1489AJ, DS1489N or DS1489AN
See NS Package J14A or N14A

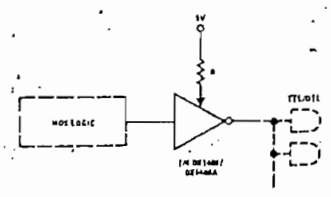
ac test circuit and voltage waveforms



typical applications



RS232C Data Transmission



MOS to TTL/DTL Translator

DS1489

absolute maximum ratings (Note 1)

The following apply for $T_A = 25^\circ\text{C}$ unless otherwise specified.

Power Supply Voltage	10V
Input Voltage Range	$\pm 30\text{V}$
Output Load Current	20 mA
Power Dissipation (Note 2)	1W
Operating Temperature Range	0°C to $+75^\circ\text{C}$
Storage Temperature Range	-65°C to $+150^\circ\text{C}$

electrical characteristics (Notes 2, 3 and 4)

DS1489/DS1489A: The following apply for $V_{CC} = 5.0\text{V} \pm 1\%$, $0^\circ\text{C} \leq T_A \leq +75^\circ\text{C}$ unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
V_{TH} Input High Threshold Voltage	$T_A = 25^\circ\text{C}$, $V_{OUT} \leq 0.45\text{V}$, $I_{OUT} = 10\text{ mA}$	DS1489	1.0	1.5	V
		DS1489A	1.75	2.25	V
V_{TL} Input Low Threshold Voltage	$T_A = 25^\circ\text{C}$, $V_{OUT} \geq 2.5\text{V}$, $I_{OUT} = -0.5\text{ mA}$	0.75		1.25	V
I_{IN} Input Current	$V_{IN} = +25\text{V}$	+3.6	+5.6	+8.3	mA
	$V_{IN} = -25\text{V}$	-3.6	-5.6	-8.3	mA
	$V_{IN} = +3\text{V}$	+0.43	+0.53		mA
	$V_{IN} = -3\text{V}$	-0.43	-0.53		mA
V_{OH} Output High Voltage	$I_{OUT} = -0.5\text{ mA}$, $V_{IN} = 0.75\text{V}$	2.6	3.8	5.0	V
	$I_{OUT} = -0.5\text{ mA}$, Input = Open	2.6	3.8	5.0	V
V_{OL} Output Low Voltage	$V_{IN} = 3.0\text{V}$, $I_{OUT} = 10\text{ mA}$		0.33	0.45	V
I_{SC} Output Short Circuit Current	$V_{IN} = 0.75\text{V}$		3.0		mA
I_{CC} Supply Current	$V_{IN} = 5.0\text{V}$		14	26	mA
P_d Power Dissipation	$V_{IN} = 5.0\text{V}$		70	130	mW

switching characteristics ($V_{CC} = 5\text{V}$, $T_A = 25^\circ\text{C}$)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
t_{pd1} Input to Output "High" Propagation Delay	$R_L = 3.9\text{k}$, (Figure 1) (ac Test Circuit)		28	85	ns
t_{pd0} Input to Output "Low" Propagation Delay	$R_L = 390\Omega$, (Figure 1) (ac Test Circuit)		20	50	ns
t_r Output Rise Time	$R_L = 3.9\text{k}$, (Figure 1) (ac Test Circuit)		110	175	ns
t_f Output Fall Time	$R_L = 390\Omega$, (Figure 1) (ac Test Circuit)		9	20	ns

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified min/max limits apply across the 0°C to $+75^\circ\text{C}$ temperature range for the DS1489 and DS1489A.

Note 3: All currents into device pins shown as positive, out of device pins as negative, all voltages referenced to ground unless otherwise noted. All values shown as max or min on absolute value basis.

Note 4: These specifications apply for response control pin = open.

TYPES SN54157, SN54L157, SN54LS157, SN54LS158, SN54S157, SN54S158, SN74157, SN74LS157, SN74LS158, SN74S157, SN74S158
 QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

MARCH 1974—REVISED DECEMBER 1983

- Buffered Inputs and Outputs
- Three Speed/Power Ranges Available

TYPES	TYPICAL AVERAGE PROPAGATION TIME	TYPICAL POWER DISSIPATION
'157	9 ns	150 mW
'L157	18 ns	75 mW
'LS157	9 ns	49 mW
'S157	5 ns	250 mW
'LS158	7 ns	24 mW
'S158	4 ns	195 mW

applications

- Expand Any Data Input Point
- Multiplex Dual Data Buses
- Generate Four Functions of Two Variables (One Variable Is Common)
- Source Programmable Counters

description

These monolithic data selectors/multiplexers contain inverters and drivers to supply full on-chip data selection to the four output gates. A separate strobe input is provided. A 4-bit word is selected from one of two sources and is routed to the four outputs. The '157, 'L157, 'LS157, and 'S157 present true data whereas the 'LS158 and 'S158 present inverted data to minimize propagation delay time.

FUNCTION TABLE

INPUTS				OUTPUT Y	
STROBE G	SELECT A/B	A	B	'157, 'L157, 'LS157, 'S157	'LS158 'S158
H	X	X	X	L	H
L	L	L	X	L	H
L	L	H	X	H	L
L	H	X	L	L	H
L	H	X	H	H	L

H = high level, L = low level, X = irrelevant

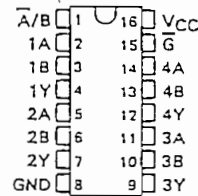
absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V _{CC} (see Note 1)	7 V
Input voltage: '157, 'L157, 'S158	5.5 V
'LS157, 'LS158	7 V
Operating free-air temperature range: SN54'	-55°C to 125°C
SN74'	0°C to 70°C
Storage temperature range	-55°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

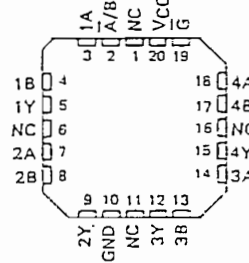
SN54157, SN54LS157, SN54S157, SN54LS158, SN54S158 ... J OR W PACKAGE
 SN54L157 ... J PACKAGE
 SN74157 ... J OR N PACKAGE
 SN74LS157, SN74S157, SN74LS158, SN74S158 ... D, J OR N PACKAGE

(TOP VIEW)



SN54LS157, SN54S157, SN54LS158, SN54S158 ... FK PACKAGE
 SN74LS157, SN74S157, SN74LS158, SN74S158 ... FN PACKAGE

(TOP VIEW)

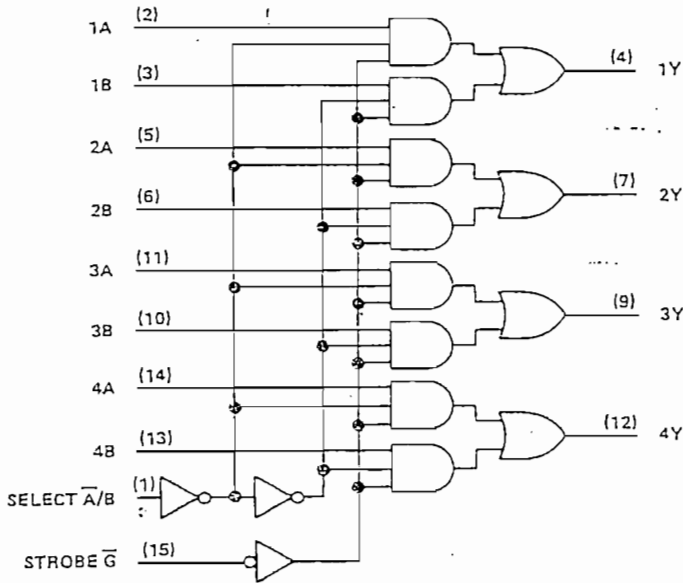


NC - No internal connection

TYPES SN54157, SN54L157, SN74157
 QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

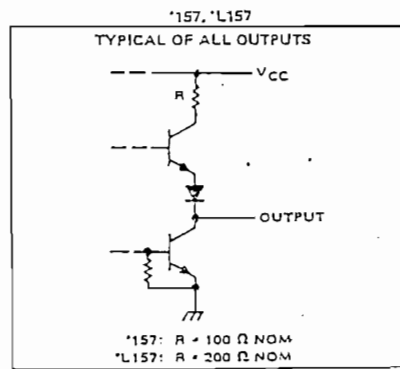
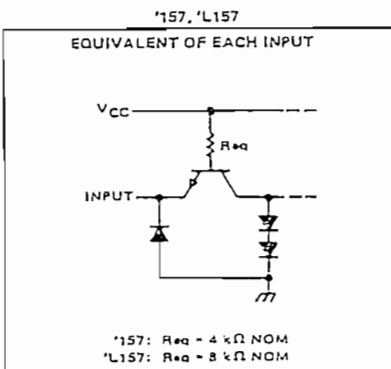
logic diagram

'157, 'L157



Pin numbers shown on logic notation are for D, J or N packages

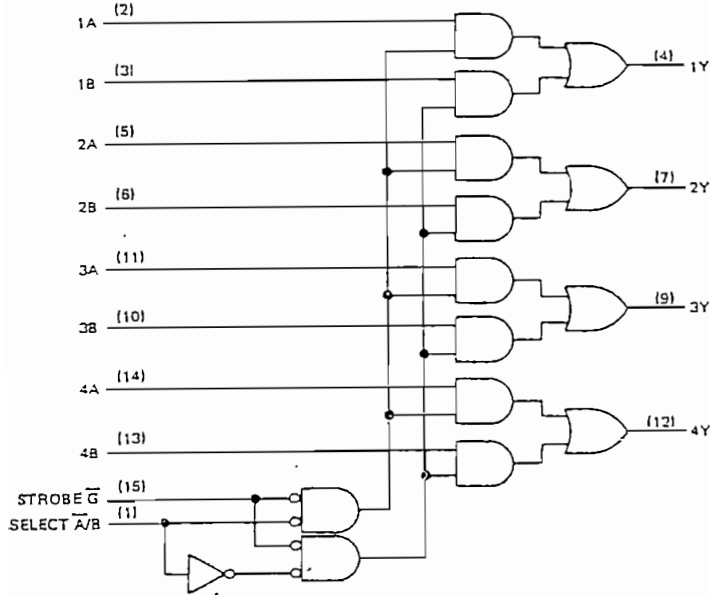
schematics of inputs and outputs



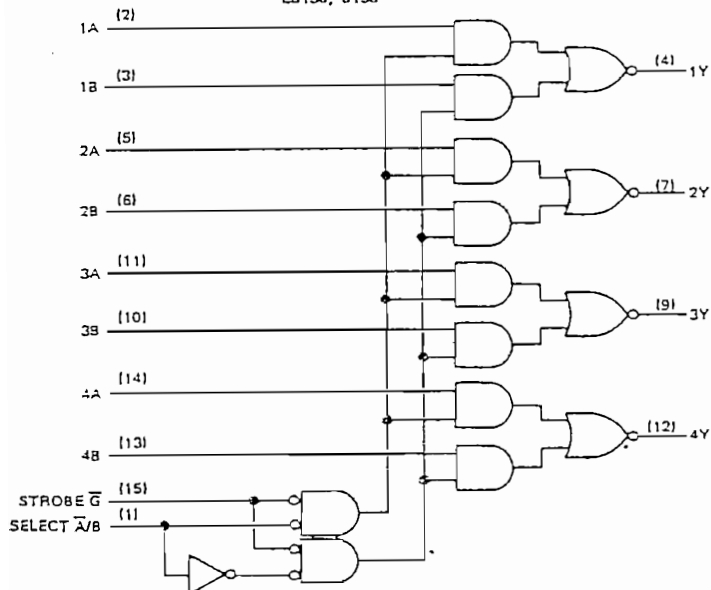
TYPES SN54LS157, SN54LS158, SN54S157, SN54S158,
SN74LS157, SN74LS158, SN74S157, SN74S158
QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

logic diagrams

'LS157, 'S157



'LS158, 'S158

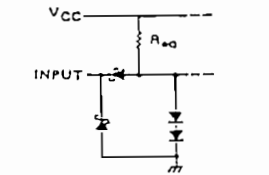


Pin numbers shown on logic notation are for D, J or N packages

schematics of inputs and outputs

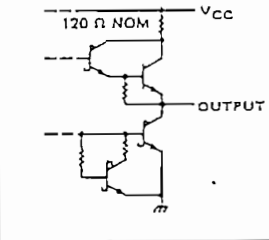
'LS157, 'LS158

EQUIVALENT OF EACH INPUT



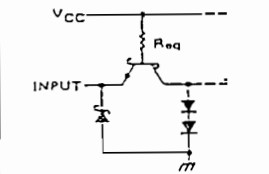
S or G inputs: $R_{eq} = 8.5 \text{ k}\Omega \text{ NOM}$
A or B inputs: $R_{eq} = 17 \text{ k}\Omega \text{ NOM}$

TYPICAL OF ALL OUTPUTS



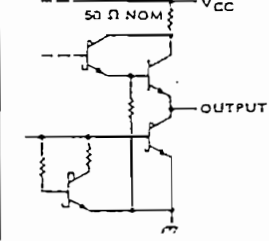
'S157, 'S158

EQUIVALENT OF EACH INPUT



S or G inputs: $R_{eq} = 1.4 \text{ k}\Omega \text{ NOM}$
A or B inputs: $R_{eq} = 2.8 \text{ k}\Omega \text{ NOM}$

TYPICAL OF ALL OUTPUTS



TYPES SN54157, SN74157
 QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

recommended operating conditions

	SN54157			SN74157			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-800			-300	μ A
Low-level output current, I_{OL}			16			16	mA
Operating free-air temperature, T_A	-55		125	0		70	$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS [†]	SN54157			SN74157			UNIT
		MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V_{IH} High-level input voltage		2			2			V
V_{IL} Low-level input voltage				0.8			0.8	V
V_{IK} Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -12 \text{ mA}$			-1.5			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -800 \mu\text{A}$	2.4	3.4		2.4	3.4		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = 0.8 \text{ V}$, $I_{OL} = 16 \text{ mA}$		0.2	0.4		0.2	0.4	V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = 5.5 \text{ V}$			1			1	mA
I_{IH} High-level input current	$V_{CC} = \text{MAX.}$, $V_I = 2.4 \text{ V}$			40			40	μ A
I_{IL} Low-level input current	$V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$			-1.6			-1.6	mA
I_{OS} Short-circuit output current [§]	$V_{CC} = \text{MAX.}$	-20		-55	-18		-55	mA
I_{CC} Supply current	$V_{CC} = \text{MAX.}$, See Note 2		30	48		30	48	mA

[†]For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡]All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$.

[§]Not more than one output should be shorted at a time and duration of short-circuit should not exceed one second.

NOTE 2: I_{CC} is measured with 4.5 V applied to all inputs and all outputs open.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$

PARAMETER [‡]	FROM (INPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Data	$C_L = 15 \text{ pF}$, $R_L = 400 \Omega$, See Note 3		9	14	ns
t_{PHL}				9	14	
t_{PLH}	Strobe \bar{G}			13	20	ns
t_{PHL}				14	21	
t_{PLH}	Select \bar{A}/\bar{B}			15	23	ns
t_{PHL}				18	27	

[‡] t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

NOTE 3: See General Information Section for load circuits and voltage waveforms

TYPE SN54L157
QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

recommended operating conditions

	MIN	NOM	MAX	UNIT
V_{CC} Supply voltage	4.5	5	5.5	V
I_{OH} High-level output current			-400	μ A
I_{OL} Low-level output current			8	mA
T_A Operating free-air temperature	-55		125	$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS [†]	MIN	TYP [‡]	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage				0.8	V
V_{IK} Input clamp voltage	$V_{CC} = \text{MIN}, I_I = -12 \text{ mA}$			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V},$ $V_{IL} = 0.8 \text{ V}, I_{OH} = -400 \mu\text{A}$	2.4	3.4		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V},$ $V_{IL} = 0.8 \text{ V}, I_{OL} = 8 \text{ mA}$		0.2	0.4	V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 5.5 \text{ V}$			1	mA
I_{IH} High-level input current	$V_{CC} = \text{MAX}, V_I = 2.4 \text{ V}$			20	μ A
I_{IL} Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.4 \text{ V}$			-0.8	mA
I_{OS} Short-circuit output current [§]	$V_{CC} = \text{MAX}$	-9		-28	mA
I_{CC} Supply current	$V_{CC} = \text{MAX},$ See Note 2		15	24	mA

[†] For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡] All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$.

[§] Not more than one output should be shorted at a time.

NOTE 2: I_{CC} is measured with 4.5 V applied to all inputs and all outputs open.

switching characteristics, $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$

PARAMETER [§]	FROM (INPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Data	$C_L = 15 \text{ pF},$ $R_L = 800 \Omega,$ See Note 3	18	28		ns
t_{PHL}			18	28		ns
t_{PLH}	Strobe \bar{C}		26	40		ns
t_{PHL}			28	42		ns
t_{PLH}	Select \bar{A}/\bar{B}		30	46		ns
t_{PHL}			36	54		ns

[§] t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

NOTE 3: See General Information Section for load circuits and voltage waveforms

TYPES SN54LS157, SN54LS158, SN74LS157, SN74LS158
 QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

recommended operating conditions

	SN54LS ¹			SN74LS ¹			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
I _{OH} High-level output current	-400			-400			μA
I _{OL} Low-level output current	4			8			mA
T _A Operating free-air temperature	-55			0			70 °C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS [†]	SN54LS ¹			SN74LS ¹			UNIT
		MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V _{IH} High-level input voltage		2			2			V
V _{IL} Low-level input voltage		0.7			0.8			V
V _{IK} Input clamp voltage	V _{CC} = MIN, I _I = -18 mA	-1.5			-1.5			V
V _{OH} High-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = MAX, I _{OH} = -400 μA	2.5	3.4		2.7	3.4		V
V _{OL} Low-level output voltage	V _{CC} = MIN, V _{IH} = 2 V, V _{IL} = MAX, I _{OL} = 4 mA	0.25	0.4		0.25	0.4		V
					0.35	0.5		
I _I Input current at maximum input voltage	\bar{A}/B or \bar{G}	0.2			0.2			mA
	A or B	0.1			0.1			
I _{IH} High-level input current	\bar{A}/B or \bar{G}	40			40			μA
	A or B	20			20			
I _{IL} Low-level input current	\bar{A}/B or \bar{G}	-0.8			-0.8			mA
	A or B	-0.4			-0.4			
I _{OS} Short-circuit output current [‡]	V _{CC} = MAX	-20	-100		-20	-100		mA
I _{CC} Supply current	V _{CC} = MAX, See Note 2	'LS157	9.7	16	9.7	16		mA
		'LS158	4.8	8	4.8	8		
		'LS158	6.5	11	6.5	11		

[†] For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡] All typical values are at V_{CC} = 5 V, T_A = 25°C.

[§] Not more than one output should be shorted at a time and duration of short-circuit should not exceed one second.

NOTE 2: I_{CC} is measured with 4.5 V applied to all inputs and all outputs open.

switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER [†]	FROM (INPUT)	TEST CONDITIONS	'LS157			'LS158			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
t _{PLH}	Data	C _L = 15 pF, R _L = 2 kΩ, See Note 3	9	14		7	12		ns
t _{PHL}			9	14		10	15		
t _{PLH}	Strobe \bar{G}		13	20		11	17		ns
t _{PHL}			14	21		18	24		
t _{PLH}	Select \bar{A}/B		15	23		13	20		ns
t _{PHL}			18	27		16	24		

[†] t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

NOTE 3. See General Information Section for load circuits and voltage waveforms

TYPES SN54S157, SN54S158, SN74S157, SN74S158
 QUADRUPLE 2-LINE TO 1-LINE DATA SELECTORS/MULTIPLEXERS

recommended operating conditions

	SN54S157 SN54S158			SN74S157 SN74S158			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-1			-1	mA
Low-level output current, I_{OL}			20			20	mA
Operating free-air temperature, T_A	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54S157 SN74S157		SN54S158 SN74S158		UNIT	
		MIN	TYP‡	MAX	MIN		TYP‡
V_{IH} High-level input voltage		2		2		V	
V_{IL} Low-level input voltage			0.8		0.8	V	
V_{IK} Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -18 \text{ mA}$		-1.2		-1.2	V	
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -1 \text{ mA}$	Series 54S 2.5	3.4	Series 74S 2.5	3.4	V	
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = 0.8 \text{ V}$, $I_{OL} = 20 \text{ mA}$		0.5		0.5	V	
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = 5.5 \text{ V}$		1		1	mA	
I_{IH} High-level input current	$\overline{A/B}$ or \overline{G}		100		100	μA	
	A or B		50		50		
I_{IL} Low-level input current	$\overline{A/B}$ or \overline{G}		-4		-4	mA	
	A or B		-2		-2		
I_{OS} Short-circuit output current§	$V_{CC} = \text{MAX.}$	-40	-100	-40	-100	mA	
I_{CC} Supply current	$V_{CC} = \text{MAX.}$, All inputs at 4.5 V, See Note 2		50	76	39	61	mA
	$V_{CC} = \text{MAX.}$, A inputs at 4.5 V, B,G,S, inputs at 0 V, See Note 2					81	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

Note 2: I_{CC} is measured with all outputs open.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$

PARAMETER‡	FROM (INPUT)	TEST CONDITIONS	SN54S157 SN74S157		SN54S158 SN74S158		UNIT
			MIN	TYP	MAX	MIN	
t_{PLH}	Data	$C_L = 15 \text{ pF}$, $R_L = 280 \Omega$, See Note 3	5	7.5	4	6	ns
t_{PHL}			4.5	6.5	4	6	
t_{PLH}	Strobe \overline{G}		8.5	12.5	6.5	11.5	ns
t_{PHL}			7.5	12	7	12	
t_{PLH}	Select $\overline{A/B}$		9.5	15	8	12	ns
t_{PHL}			9.5	15	8	12	

‡ t_{PLH} = propagation delay time, low-to-high-level output

‡ t_{PHL} = propagation delay time, high-to-low-level output

NOTE 3: See General Information Section for load circuits and voltage waveforms

TYPES SN54LS245, SN74LS245 OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

OCTOBER 1976—REVISED APRIL 1985

- Bi-directional Bus Transceiver in a High-Density 20-Pin Package
- 3-State Outputs Drive Bus Lines Directly
- PNP Inputs Reduce D-C Loading on Bus Lines
- Hysteresis at Bus Inputs Improve Noise Margins
- Typical Propagation Delay Times, Port-to-Port . . . 8 ns

TYPE	I _{OL} (SINK CURRENT)	I _{OH} (SOURCE CURRENT)
SN54LS245	12 mA	-12 mA
SN74LS245	24 mA	-15 mA

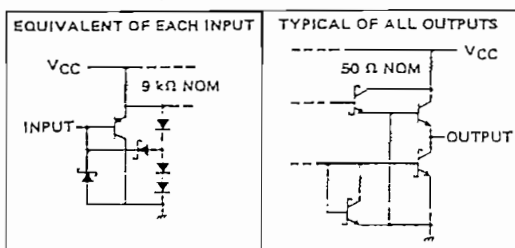
description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

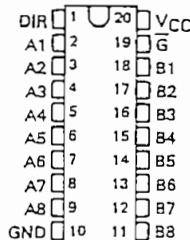
The devices allow data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input (\bar{G}) can be used to disable the device so that the buses are effectively isolated.

The SN54LS245 is characterized for operation over the full military temperature range of -55°C to 125°C. The SN74LS245 is characterized for operation from 0°C to 70°C.

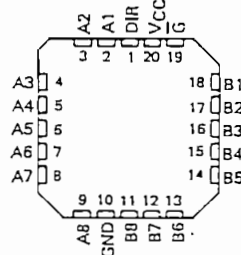
schematics of inputs and outputs



SN54LS245 . . . J PACKAGE
SN74LS245 . . . DW, J OR N PACKAGE
(TOP VIEW)



SN54LS245 . . . FK PACKAGE
SN74LS245 . . . FN PACKAGE
(TOP VIEW)



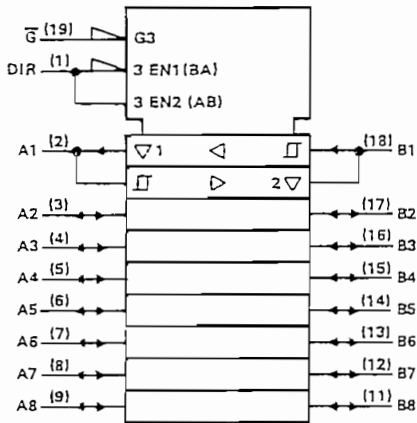
FUNCTION TABLE

ENABLE \bar{G}	DIRECTION CONTROL DIR	OPERATION
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

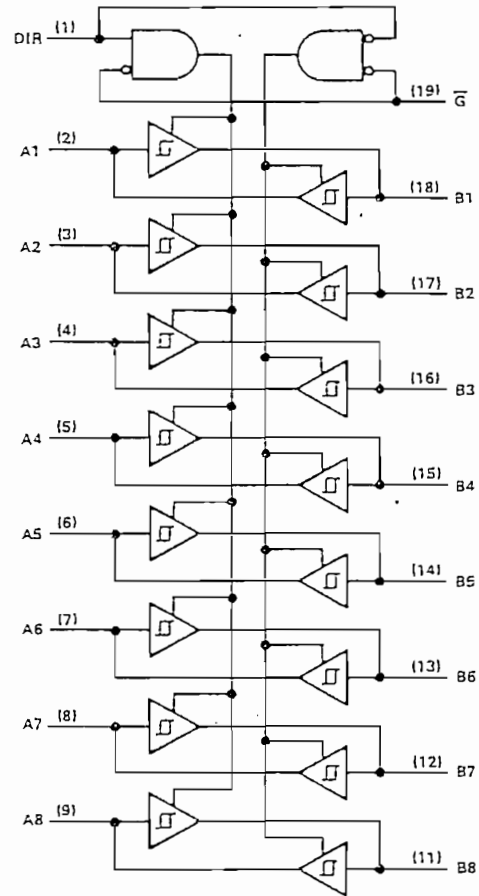
H = high level, L = low level, X = irrelevant

TYPES SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

logic symbol



logic diagram (positive logic)



Pin numbers shown on logic notation are for DW, J or N packages

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	7 V
Off-state output voltage	5.5 V
Operating free-air temperature range, SN54LS*	-55°C to 125°C
SN74LS*	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

TYPES SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

recommended operating conditions

PARAMETER	SN54LS245			SN74LS245			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-12			-15	mA
Low-level output current, I_{OL}			12			24	mA
Operating free-air temperature, T_A	-55		125	0		70	$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS [†]	SN54LS245			SN74LS245			UNIT
		MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V_{IH} High-level input voltage		2			2		V	
V_{IL} Low-level input voltage				0.7			0.8	
V_{IK} Input clamp voltage	$V_{CC} = \text{MIN.}$ $I_I = -12 \text{ mA}$			-1.5			-1.5	
Hysteresis ($V_{T+} - V_{T-}$) A or B input	$V_{CC} = \text{MIN.}$	0.2	0.4		0.2	0.4	V	
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$ $V_{IH} = 2 \text{ V.}$ $V_{IL} = V_{IL \text{ max}}$	$I_{OH} = -3 \text{ mA}$	2.4	3.4	2.4	3.4	V	
		$I_{OH} = \text{MAX}$	2		2			
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$ $V_{IH} = 2 \text{ V.}$ $V_{IL} = V_{IL \text{ max}}$	$I_{OL} = 12 \text{ mA}$		0.4		0.4	V	
		$I_{OL} = 24 \text{ mA}$				0.5		
I_{OZH} Off-state output current, high-level voltage applied	$V_{CC} = \text{MAX.}$ \bar{G} at 2 V			20		20	μ A	
I_{OZL} Off-state output current, low-level voltage applied				-200		-200		
I_I Input current at maximum input voltage	A or B DIR or \bar{G}	$V_{CC} = \text{MAX.}$	$V_I = 5.5 \text{ V}$		0.1	0.1	mA	
			$V_I = 7 \text{ V}$		0.1	0.1		
I_{IH} High-level input current	$V_{CC} = \text{MAX.}$	$V_{IH} = 2.7 \text{ V}$		20		20	μ A	
I_{IL} Low-level input current	$V_{CC} = \text{MAX.}$	$V_{IL} = 0.4 \text{ V}$		-0.2		-0.2	mA	
I_{OS} Short-circuit output current [‡]	$V_{CC} = \text{MAX.}$			-40	-225	-40	-225	
I_{CC} Supply current	Total, outputs high	$V_{CC} = \text{MAX.}$	Outputs open		48	70	48	70
	Total, outputs low				62	90	62	90
	Outputs at Hi-Z				64	95	64	95

[†] For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡] All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$.

[§] Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT	
t_{PLH} Propagation delay time, low-to-high-level output	$C_L = 45 \text{ pF}$, $R_L = 667 \Omega$, See Note 2		8	12	ns	
t_{PHL} Propagation delay time, high-to-low-level output			8	12	ns	
t_{PZL} Output enable time to low level				27	40	ns
t_{PZH} Output enable time to high level				25	40	ns
t_{PLZ} Output disable time from low level		$C_L = 5 \text{ pF}$, $R_L = 667 \Omega$, See Note 2		15	25	ns
t_{PHZ} Output disable time from high level			15	28	ns	

NOTE 2: See General Information Section for load circuits and voltage waveforms

TYPES SN5404, SN54H04, SN54L04, SN54LS04, SN54S04,
SN7404, SN74H04, SN74LS04, SN74S04
HEX INVERTERS

REVISED DECEMBER 1983

- Package Options Include Both Plastic and Ceramic Chip Carriers in Addition to Plastic and Ceramic DIPs
- Dependable Texas Instruments Quality and Reliability

description

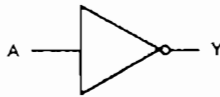
These devices contain six independent inverters.

The SN5404, SN54H04, SN54L04, SN54LS04 and SN54S04 are characterized for operation over the full military temperature range of -55°C to 125°C. The SN7404, SN74H04, SN74LS04 and SN74S04 are characterized for operation from 0°C to 70°C.

FUNCTION TABLE (each inverter)

INPUTS A	OUTPUT Y
H	L
L	H

logic diagram (each inverter)

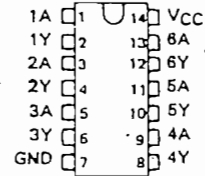


positive logic

$$Y = \bar{A}$$

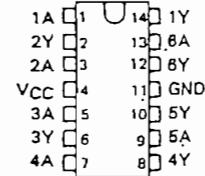
SN5404, SN54H04, SN54L04 ... J PACKAGE
SN54LS04, SN54S04 ... J OR W PACKAGE
SN7404, SN74H04 ... J OR N PACKAGE
SN74LS04, SN74S04 ... D, J OR N PACKAGE

(TOP VIEW)



SN5404, SN54H04 ... W PACKAGE

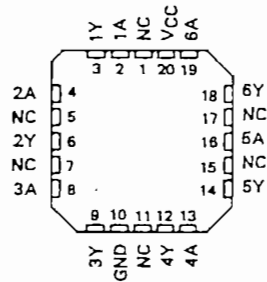
(TOP VIEW)



SN54LS04, SN54S04 ... FK PACKAGE

SN74LS04, SN74S04 ... FN PACKAGE

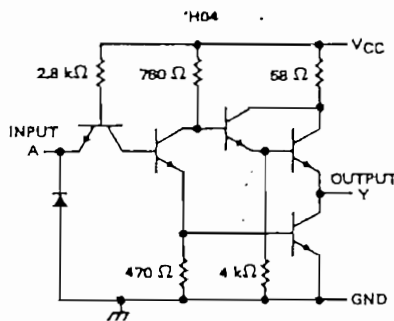
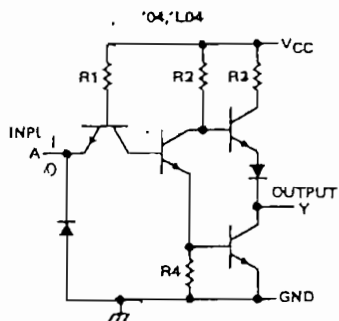
(TOP VIEW)



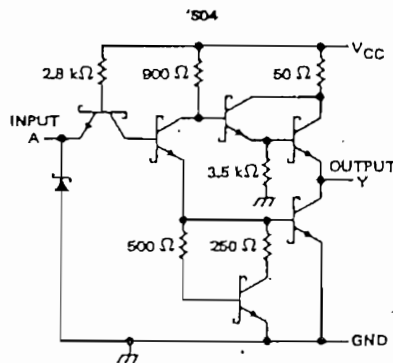
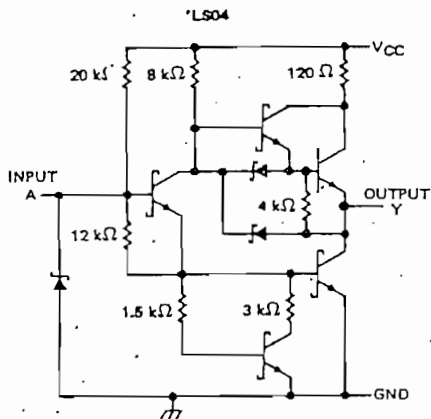
NC - No internal connection

TYPES SN5404, SN54H04, SN54L04, SN54LS04, SN54S04,
SN7404, SN74H04, SN74LS04, SN74S04
HEX INVERTERS

schematics (each gate)



CIRCUIT	R1	R2	R3	R4
'04	4 kΩ	1.8 kΩ	130 Ω	1 kΩ
'L04	40 kΩ	20 kΩ	500 Ω	12 kΩ



Resistor values shown are nominal.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1): '04, 'H04, 'LS04, 'S04	7 V
'L04	8 V
Input voltage: '04, 'H04, 'L04, 'S04	5.5 V
'LS04	7 V
Operating free-air temperature range: SN54'	-55° C to 125° C
SN74'	0° C to 70° C
Storage temperature range	-65° C to 150° C

NOTE 1: Voltage values are with respect to network ground terminal.

TYPES SN5404, SN7404
HEX INVERTERS

recommended operating conditions

	SN5404			SN7404			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH} High-level input voltage ¹	2			2			V
V_{IL} Low-level input voltage	0.8			0.8			V
I_{OH} High-level output current	-0.4			-0.4			mA
I_{OL} Low-level output current	16			16			mA
T_A Operating free-air temperature	-55	125		0	70		°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS [†]	SN5404		SN7404		UNIT
		MIN	TYP [‡]	MAX	MIN	
V_{IK}	$V_{CC} = \text{MIN.}$, $I_I = -12 \text{ mA}$			-1.5		V
V_{OH}	$V_{CC} = \text{MIN.}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -0.4 \text{ mA}$	2.4	3.4	2.4	3.4	V
V_{OL}	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 16 \text{ mA}$	0.2	0.4	0.2	0.4	V
I_I	$V_{CC} = \text{MAX.}$, $V_I = 5.5 \text{ V}$	1		1		mA
I_{IH}	$V_{CC} = \text{MAX.}$, $V_I = 2.4 \text{ V}$	40		40		μA
I_{IL}	$V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$	-1.6		-1.6		mA
I_{OS} [§]	$V_{CC} = \text{MAX.}$	-20	-55	-18	-56	mA
I_{CCH}	$V_{CC} = \text{MAX.}$, $V_I = 0 \text{ V}$	6	12	6	12	mA
I_{CCL}	$V_{CC} = \text{MAX.}$, $V_I = 4.5 \text{ V}$	18	33	18	33	mA

[†] For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡] All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

[§] Not more than one output should be shorted at a time.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$ (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS		MIN	TYP	MAX	UNIT	
t_{PLH}	A	Y	$R_L = 400 \Omega$,	$C_L = 15 \text{ pF}$			12	22	ns
t_{PHL}					8	15	ns		

NOTE 2: See General Information Section for load circuits and voltage waveforms.

TYPES SN54H04, SN74H04
HEX INVERTERS

recommended operating conditions

	SN54H04			SN74H04			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH} High-level input voltage	2			2			V
V_{IL} Low-level input voltage			0.8			0.8	V
I_{OH} High-level output current			-0.5			-0.5	mA
I_{OL} Low-level output current			20			20	mA
T_A Operating free-air temperature ¹	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS [†]	MIN	TYP [‡]	MAX	UNIT
V_{IK}	$V_{CC} = \text{MIN}$, $I_I = -8 \text{ mA}$			-1.5	V
V_{OH}	$V_{CC} = \text{MIN}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -0.5 \text{ mA}$	2.4	3.5		V
V_{OL}	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 20 \text{ mA}$		0.2	0.4	V
I_I	$V_{CC} = \text{MAX}$, $V_I = 5.5 \text{ V}$			1	mA
I_{IH}	$V_{CC} = \text{MAX}$, $V_I = 2.4 \text{ V}$			50	μA
I_{IL}	$V_{CC} = \text{MAX}$, $V_I = 0.4 \text{ V}$			-2	mA
I_{OS}^{\S}	$V_{CC} = \text{MAX}$	-40		-100	mA
I_{CCH}	$V_{CC} = \text{MAX}$, $V_I = 0 \text{ V}$		16	26	mA
I_{CCL}	$V_{CC} = \text{MAX}$, $V_I = 4.5 \text{ V}$		40	58	mA

[†] For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

[‡] All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

[§] Not more than one output should be shorted at a time, and the duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$ (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	$R_L = 280 \Omega$, $C_L = 25 \text{ pF}$		6	10	ns
t_{PHL}					6.5	10	ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

TYPE SN54L04
HEX INVERTERS

recommended operating conditions

	SN54L04			UNIT
	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	V
V_{IH} High-level input voltage	2			V
V_{IL} Low-level input voltage			0.7	V
I_{OH} High-level output current			-0.1	mA
I_{OL} Low-level output current			2	mA
T_A Operating free-air temperature	-55		125	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54L04			UNIT
		MIN	TYP‡	MAX	
V_{OH}	$V_{CC} = \text{MIN}$, $V_{IL} = 0.7 \text{ V}$, $I_{OH} = -0.1 \text{ mA}$	2.4	3.3		V
V_{OL}	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 2 \text{ mA}$	0.15	0.3		V
I_I	$V_{CC} = \text{MAX}$, $V_I = 5.5 \text{ V}$			0.1	mA
I_{IH}	$V_{CC} = \text{MAX}$, $V_I = 2.4 \text{ V}$			10	μA
I_{IL}	$V_{CC} = \text{MAX}$, $V_I = 0.3 \text{ V}$			-0.18	mA
$I_{OS}§$	$V_{CC} = \text{MAX}$	-3		-15	mA
I_{CCH}	$V_{CC} = \text{MAX}$, $V_I = 0 \text{ V}$	0.66		1.2	mA
I_{CCL}	$V_{CC} = \text{MAX}$, $V_I = 4.5 \text{ V}$	1.74		3.06	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$ (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	$R_L = 4 \text{ k}\Omega$, $C_L = 50 \text{ pF}$		35	60	ns
t_{PHL}					31	60	ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

Types SN54LS04, SN74LS04
HEX INVERTERS

recommended operating conditions

	SN54LS04			SN74LS04			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4,5	5	5,5	4,75	5	5,25	V
V_{IH} High-level input voltage	2			2			V
V_{IL} Low-level input voltage			0,7			0,8	V
I_{OH} High-level output current			-0,4			-0,4	mA
I_{OL} Low-level output current			4			8	mA
T_A Operating free-air temperature	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS †	SN54LS04			SN74LS04			UNIT
		MIN	TYP ‡	MAX	MIN	TYP ‡	MAX	
V_{IK}	$V_{CC} = \text{MIN}$, $I_I = -18 \text{ mA}$			-1,5			-1,5	V
V_{OH}	$V_{CC} = \text{MIN}$, $V_{IL} = \text{MAX}$, $I_{OH} = -0,4 \text{ mA}$	2,5	3,4		2,7	3,4		V
V_{OL}	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 4 \text{ mA}$		0,25	0,4			0,4	V
	$V_{CC} = \text{MIN}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 8 \text{ mA}$					0,25	0,5	
I_I	$V_{CC} = \text{MAX}$, $V_I = 7 \text{ V}$			0,1			0,1	mA
I_{IH}	$V_{CC} = \text{MAX}$, $V_I = 2,7 \text{ V}$			20			20	µA
I_{IL}	$V_{CC} = \text{MAX}$, $V_I = 0,4 \text{ V}$			-0,4			-0,4	mA
$I_{OS} §$	$V_{CC} = \text{MAX}$	-20		-100	-20		-100	mA
I_{CCH}	$V_{CC} = \text{MAX}$, $V_I = 0 \text{ V}$		1,2	2,4		1,2	2,4	mA
I_{CCL}	$V_{CC} = \text{MAX}$, $V_I = 4,5 \text{ V}$		3,6	6,6		3,6	6,6	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and the duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$ (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	$R_L = 2 \text{ k}\Omega$, $C_L = 15 \text{ pF}$		9	15	ns
t_{PHL}					10	15	ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

TYPES SN54S04, SN74S04
HEX INVERTERS

recommended operating conditions

	SN54S04			SN74S04			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC} Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH} High-level input voltage	2			2			V
V_{IL} Low-level input voltage	0.8			0.8			V
I_{OH} High-level output current	-1			-1			mA
I_{OL} Low-level output current	20			20			mA
T_A Operating free-air temperature	-55			0			$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS †	SN54S04			SN74S04			UNIT
		MIN	TYP ‡	MAX	MIN	TYP ‡	MAX	
V_{IK}	$V_{CC} - \text{MIN}$, $I_I = -18 \text{ mA}$			-1.2			-1.2	V
V_{OH}	$V_{CC} - \text{MIN}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = -1 \text{ mA}$	2.5	3.4		2.7	3.4		V
V_{OL}	$V_{CC} - \text{MIN}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 20 \text{ mA}$			0.5			0.5	V
I_I	$V_{CC} - \text{MAX}$, $V_I = 5.5 \text{ V}$			1			1	mA
I_{IH}	$V_{CC} - \text{MAX}$, $V_I = 2.7 \text{ V}$			50			50	μ A
I_{IL}	$V_{CC} - \text{MAX}$, $V_I = 0.5 \text{ V}$			-2			-2	mA
$I_{OS} §$	$V_{CC} - \text{MAX}$	-40		-100	-40		-100	mA
I_{CCH}	$V_{CC} - \text{MAX}$, $V_I = 0 \text{ V}$		15	24		15	24	mA
I_{CCL}	$V_{CC} - \text{MAX}$, $V_I = 4.5 \text{ V}$		30	54		30	54	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$.

§ Not more than one output should be shorted at a time, and the duration of the short-circuit should not exceed one second.

switching characteristics, $V_{CC} = 5 \text{ V}$, $T_A = 25^{\circ}\text{C}$ (see note 2)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{PLH}	A	Y	$R_L = 280 \Omega$,	$C_L = 15 \text{ pF}$		3	4.5	ns
t_{PHL}						3	5	ns
t_{PLH}			$R_L = 280 \Omega$,	$C_L = 50 \text{ pF}$		4.5		ns
t_{PHL}						5		ns

NOTE 2: See General Information Section for load circuits and voltage waveforms.

Del 1N4001 al 1N4007

Hoja de datos Designers

RECTIFICADORES "SURMETIC"

... rectificadores axiales con montura de alambre, tamaño subminiatura para aplicaciones de baja potencia de propósito general.

Datos de los diseñadores para las condiciones de "el peor de los casos"

Las hojas de datos Designers permiten diseñar la mayor parte de los circuitos totalmente a partir de la información presentada. Las curvas límite, que representan las características extremas del dispositivo, se dan para facilitar el diseño para las peores condiciones de operación.

RECTIFICADORES DE SILICIO
CON MONTURA DE ALAMBRE

UNIÓN DIFUNDIRA
50-1000 VOLTS

*ESPECIFICACIONES MÁXIMAS

Especificación	Símbolo	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	Unid.
Voltaje inverso repetitivo pico Voltaje inverso pico de operación Voltaje de bloqueo en cd	V_{RRM} V_{RWM} V_R	50	100	200	400	600	800	1000	Volts
Voltaje inverso pico no repetitivo (media onda, una sola fase, 60 Hz)	V_{RSM}	60	120	240	480	720	1000	1200	Volts
Voltaje inverso rms	$V_{R(RMS)}$	35	70	140	280	420	560	700	Volts
Corriente prom. rectif. en polariz. dir. (una fase, carga resist., 60 Hz, véase la figura 8, $T_c = 75^\circ\text{C}$)	I_O	1.0							Amp
Oleada de cor. pico no repetitiva (la oleada se aplica en las cond. especificadas de carga, v. figura 2)	I_{FSM}	30 (para un ciclo)							Amp
Intervalo de temp. de operación y de almacenam. de la unión	T_J, T_{stg}	-65 to +175							$^\circ\text{C}$

*CARACTERÍSTICAS ELÉCTRICAS

Característica y condiciones	Símbolo	Tipo	máx.	Unid.
Máxima caída de voltaje instantánea en polarización directa	V_F	0.93	1.1	Volts
Máxima caída de voltaje promedio de ciclo completo en polarización directa	$V_F(AV)$	-	0.8	Volts
Corriente inversa máx. (voltaje especific. en cd) $T_J = 25^\circ\text{C}$ $T_J = 100^\circ\text{C}$	I_R	0.05 1.0	10 50	μA
Máxima corriente inversa prom. de ciclo completo ($I_O = 1.0 \text{ A}$, $T_c = 75^\circ\text{C}$, conductores de 1 pulg.)	$I_{R(AV)}$	-	30	μA

*Indica datos registrados JEDEC

CARACTERÍSTICAS MECÁNICAS

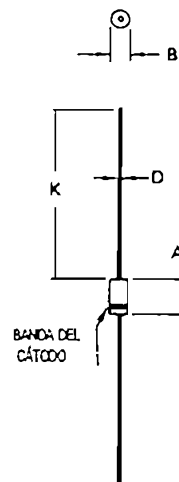
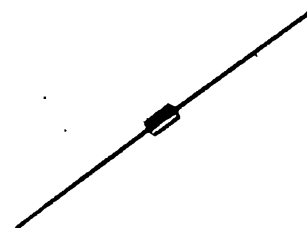
ENVASE: sin vacío, Transfer Molded

TEMPERATURA MÁXIMA DE LOS CONDUCTORES PARA PROPÓSITOS DE SOLDADURA: 350°C , a $3/8"$ del envase durante 10 segundos con una tensión de 5 lbs.

ACABADO: todas las superficies externas son resistentes a la corrosión y los conductores se pueden soldar con facilidad.

POLARIDAD: el cátodo se indica mediante la banda de color

PESO: 0.40 gramos (aproximadamente)



DIM	MILÍMETROS		PULGADAS	
	MIN	MAX	MIN	MAX
A	5.97	6.60	0.235	0.260
B	2.79	3.05	0.110	0.120
D	0.76	0.86	0.030	0.034
K	27.94	-	1.100	-

ENVASE 59-04

No se ajusta a la descripción DO-41

ANEXO 4 : MANUAL DE LA FAMILIA MCS-51/52



PRELIMINARY

8031/8051/8751 SINGLE COMPONENT 8-BIT MICROCOMPUTER

- 8031 - Control Oriented CPU With RAM and I/O
 - 8051 - An 8031 With Factory Mask-Programmable ROM
 - 8751 - An 8031 With User Programmable/Erasable EPROM
-
- 4K x 8 ROM/EPROM
 - 128 x 8 RAM
 - Four 8-Bit Ports, 32 I/O Lines
 - Two 16-Bit Timer/Event Counters
 - High-Performance Full-Duplex Serial Channel
 - External Memory Expandable to 128K
 - Compatible with MCS-80®/MCS-85® Peripherals
-
- Boolean Processor
 - MCS-48® Architecture Enhanced with:
 - Non-Paged Jumps
 - Direct Addressing
 - Four 8-Register Banks
 - Stack Depth Up to 128-Bytes
 - Multiply, Divide, Subtract, Compare
 - Most Instructions Execute in 1µs
 - 4µs Multiply and Divide

The Intel® 8031/8051/8751 is a stand-alone, high-performance single-chip computer fabricated with Intel's highly-reliable +5 Volt, depletion-load, N-Channel, silicon-gate HMOS technology and packaged in a 40-pin DIP. It provides the hardware features, architectural enhancements and new instructions that are necessary to make it a powerful and cost effective controller for applications requiring up to 64K bytes of program memory and/or up to 64K bytes of data storage.

The 8051/8751 contains a non-volatile 4K x 8 read only program memory; a volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/counters; a five-source, two-priority-level, nested interrupt structure; a serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. The 8031 is identical, except that it lacks the program memory. For systems that require extra capability, the 8051 can be expanded using standard TTL compatible memories and the byte oriented MCS-80 and MCS-85 peripherals.

The 8051 microcomputer, like its 8048 predecessor, is efficient both as a controller and as an arithmetic processor. The 8051 has extensive facilities for binary and BCD arithmetic and excels in bit-handling capabilities. Efficient use of program memory results from an instruction set consisting of 44% of one-byte, 41% two-byte, and 15% three-byte instructions. With a 12 MHz crystal, 58% of the instructions execute in 1µs, 40% in 2µs and multiply and divide require only 4µs. Among the many instructions added to the standard 8048 instruction set are multiply, divide, subtract and compare.

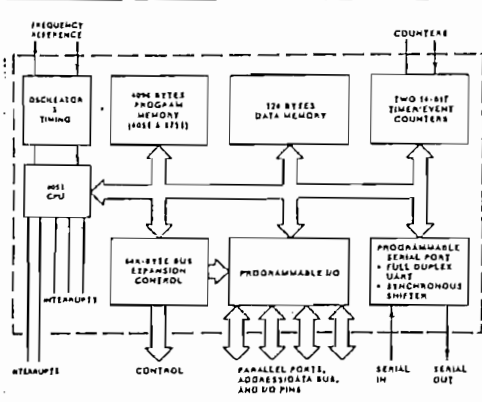


Figure 1.
Block Diagram

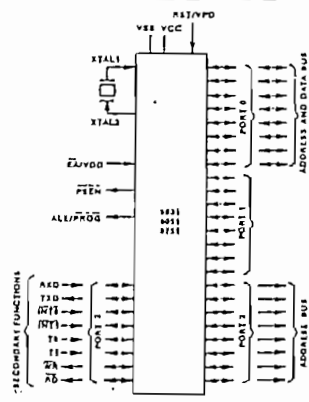


Figure 2.
Logic Symbol

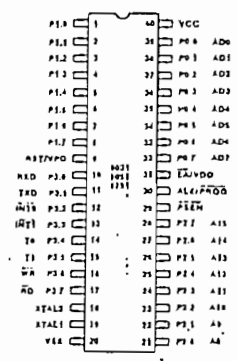


Figure 3. Pin
Configuration

intel

8031/8051/8751

PRELIMINARY

INTRODUCTION

This data sheet provides an introduction to the 8051 family. A detailed description of the hardware required to expand the 8051 with more program memory, data memory, I/O, specialized peripherals and into multiprocessor configurations is described in the 8051 Family User's Manual.

THE 8051 Family

The 8051 is a stand-alone high-performance single-chip computer intended for use in sophisticated real-time applications such as instrumentation, industrial control and intelligent computer peripherals. It provides the hardware features, architectural enhancements and new instructions that make it a powerful and cost effective controller for applications requiring up to 64K-bytes of program memory and/or up to 64K-bytes of data storage. A Block Diagram is shown in Figure 3.

The 8031 is a control-oriented CPU without on-chip program memory. It can address 64K-bytes of external Program Memory in addition to 64K-bytes of External Data Memory. For systems requiring extra capability, each member of the 8051 family can be expanded using standard memories and the byte oriented MCS-80 and MCS-85 peripherals. The 8051 is an 8031 with the lower 4K-bytes of Program Memory filled with on-chip mask programmable ROM while the 8751 has 4K-bytes of UV-light-erasable/electrically-programmable ROM.

The three pin-compatible versions of this component reduce development problems to a minimum and provide maximum flexibility. The 8751 is well suited for development, prototyping, low-volume production and applications requiring field updates; the 8051 for low-cost, high volume production; and the 8031 for applications desiring the flexibility of external Program Memory which can be easily

modified and updated in the field.

MACRO-VIEW OF THE 8051 ARCHITECTURE

On a single die the 8051 microcomputer combines CPU; non-volatile 4K x 8 read-only program memory; volatile 128 x 8 read/write data memory; 32 I/O lines; two 16-bit timer/event counters; a five-source, two-priority-level, nested interrupt structure; serial I/O port for either multi-processor communications, I/O expansion, or full duplex UART; and on-chip oscillator and clock circuits. This section will provide an overview of the 8051 by providing a high-level description of its major elements: the CPU architecture and the on-chip functions peripheral to the CPU. The generic term "8051" is used to refer collectively to the 8031, 8051, and 8751.

8051 CPU Architecture

The 8051 CPU manipulates operands in four memory spaces. These are the 64K-byte Program Memory, 64K-byte External Data Memory, 384-byte Internal Data Memory and 16-bit Program Counter spaces. The Internal Data Memory address space is further divided into the 256-byte Internal Data RAM and 128-byte Special Function Register (SFR) address spaces shown in Figure 4. Four Register Banks (each with eight registers), 128 addressable bits, and the stack reside in the Internal Data RAM. The stack depth is limited only by the available Internal Data RAM and its location is determined by the 8-bit stack pointer. All registers except the Program Counter and the four 8-Register Banks reside in the Special Function Register address space. These memory mapped registers include arithmetic registers, pointers, I/O ports, interrupt system registers, timers and serial port. 128 bit locations in the SFR address space are addressable as bits. The 8051 contains 128 bytes of Internal Data RAM and 20 SFRs.

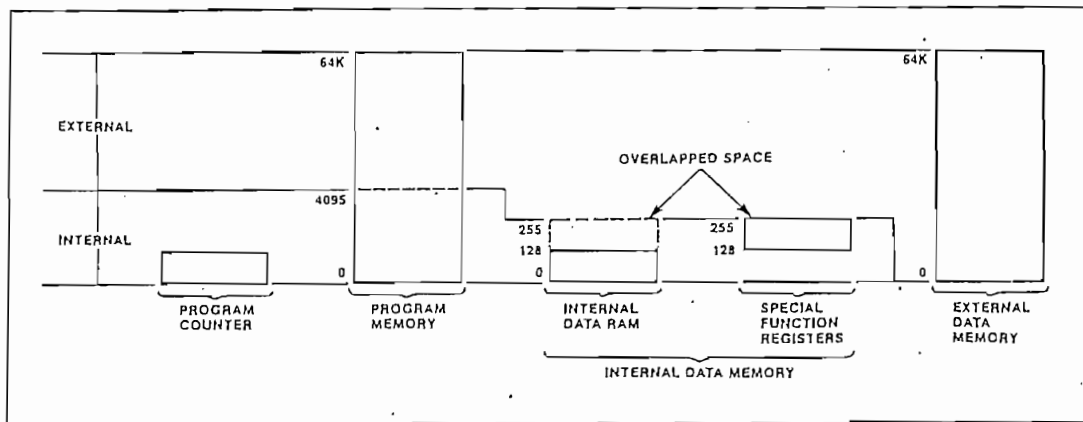


Figure 4. 8051 Family Memory Organization



The 8051 provides a non-paged Program Memory address space to accommodate relocatable code. Conditional branches are performed relative to the Program Counter. The register-indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. Sixteen-bit jumps and calls permit branching to any location in the contiguous 64K Program Memory address space.

The 8051 has five methods for addressing source operands: Register, Direct, Register-Indirect, Immediate and Base-Register- plus Index-Register-Indirect Addressing. The first three methods can be used for addressing destination operands. Most instructions have a "destination, source" field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand.

Registers in the four 8-Register Banks can be accessed through Register, Direct, or Register-Indirect Addressing; the 128 bytes of Internal Data RAM through Direct or Register-Indirect Addressing; and the Special Function Registers through Direct Addressing. External Data Memory is accessed through Register-Indirect Addressing. Look-Up-Tables resident in Program Memory can be accessed through Base-Register- plus Index-Register- Indirect Addressing.

The 8051 is classified as an 8-bit machine since the internal ROM, RAM, Special Function Registers, Arithmetic/Logic Unit and external data bus are each 8-bits wide. The 8051 performs operations on bit, nibble, byte and double-byte data types.

The 8051 has extensive facilities for byte transfer, logic, and integer arithmetic operations. It excels at bit handling since data transfer, logic and conditional branch operations can be performed directly on Boolean variables.

The 8051's instruction set is an enhancement of the instruction set familiar to MCS-48 users. It is enhanced to allow expansion of on-chip CPU peripherals and to optimize byte efficiency and execution speed. Op codes were reassigned to add new high-power operations and to permit new addressing modes which make the old operations more orthogonal. Efficient use of program memory results from an instruction set consisting of 49 single-byte, 45 two-byte and 17 three-byte instructions. When using a 12 MHz oscillator, 64 instructions execute in 1 μ s and 45 instructions execute in 2 μ s. The remaining instructions (multiply and divide) execute in only 4 μ s. The number of bytes in each instruction and the number of oscillator periods required for execution are listed in the appended 8051 Instruction Set Summary.

On-Chip Peripheral Functions

Thus far only the CPU and memory spaces of the 8051 have been described. In addition to the CPU and memories, an interrupt system, extensive I/O facilities, and several peripheral functions are integrated on-chip to relieve the CPU of repetitious, complicated or time-critical tasks and to permit stringent real-time control of external system interfaces. The extensive I/O facilities include the I/O pins, parallel I/O ports, bidirectional address/data bus and the serial port for I/O expansion. The CPU peripheral functions integrated on-chip are the two 16-bit counters and the serial port. All of these work together to greatly boost system performance.

INTERRUPT SYSTEM

External events and the real-time-driven on-chip peripherals require service by the CPU asynchronous to the execution of any particular section of code. To tie the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, two-priority-level, nested interrupt system is provided. Interrupt response latency ranges from 3 μ s to 7 μ s when using a 12 MHz crystal.

The 8051 acknowledges interrupt requests from five sources: Two from external sources via the INT0 and INT1 pins, one from each of the two internal counters and one from the serial I/O port. Each interrupt vectors to a separate location in Program Memory for its service program. Each of the five sources can be assigned to either of two priority levels and can be independently enabled and disabled. Additionally all enabled sources can be globally disabled or enabled. Each external interrupt is programmable as either level- or transition-activated and is active-low to allow the "wire or-ing" of several interrupt sources to the input pin. The interrupt system is shown diagrammatically in Figure 5.

I/O FACILITIES

The 8051 has instructions that treat its 32 I/O lines as 32 individually addressable bits and as four parallel 8-bit ports addressable as Ports 0, 1, 2 and 3. Ports 0, 2 and 3 can also assume other functions. Port 0 provides the multiplexed low-order address and data bus used for expanding the 8051 with standard memories and peripherals. Port 2 provides the high-order address bus when expanding the 8051 with external Program Memory or more than 256 bytes of External Data Memory. The pins of Port 3 can be configured individually to provide external interrupt request inputs, counter inputs, the serial port's receiver input and transmitter output, and to generate the control signals used for reading and writing External Data Memory. The generation or use of an alternate function on a Port 3 pin is done automatically by the 8051 as long as the pin

intel

8031/8051/8751

PRELIMINARY

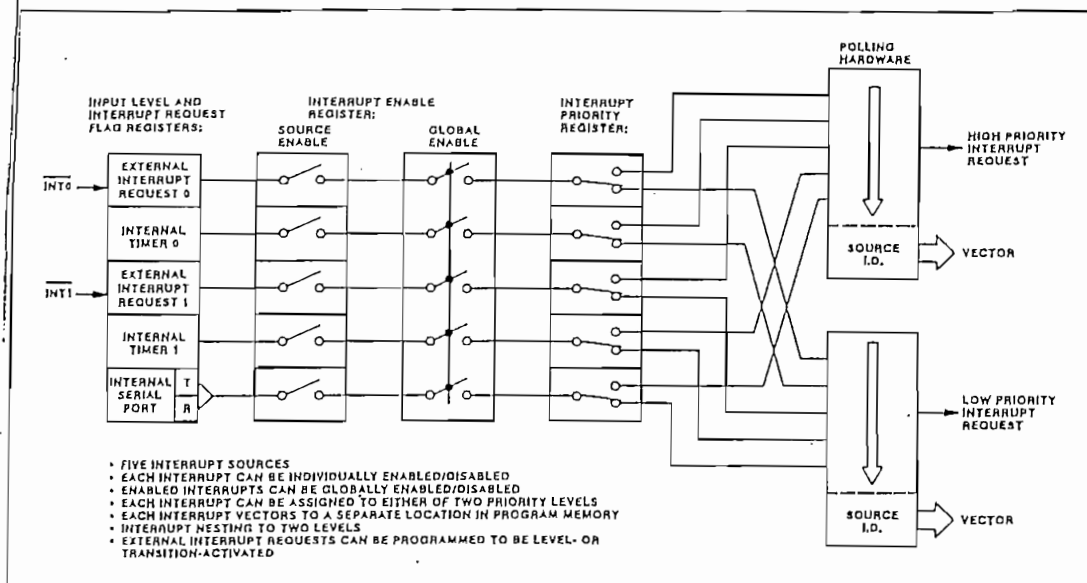


Figure 5. 8051 Interrupt System

is configured as an input. The configuration of the ports is shown on the 8051 Family Logic Symbol of Figure 2.

Open Drain I/O Pins

Each pin of Port 0 can be configured as an open drain output or as a high impedance input. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Re-writing the pin to a one (1) will place its output driver in a high-impedance state and configure the pin as an input. Each I/O pin of Port 0 can sink two TTL loads.

Quasi-Bidirectional I/O Pins

Ports 1, 2 and 3 are quasi-bidirectional buffers. Resetting the microcomputer programs each pin as an input by writing a one (1) to the pin. If a zero (0) is later written to the pin it becomes configured as an output and will continuously sink current. Any pin that is configured as an output will be reconfigured as an input when a one (1) is written to the pin. Simultaneous to this reconfiguration the output driver of the quasi-bidirectional port will source current for two oscillator periods. Since current is sourced only when a bit previously written to a zero (0) is updated to a one (1), a pin programmed as an input will not source current into the TTL gate that is driving it if the pin is later written with another one (1). Since the quasi-bidirectional output driver sources current for only two oscillator periods, an internal

pullup resistor of approximately 20K- to 40K-ohms is provided to hold the external driver's loading at a TTL high level. Ports 1, 2 and 3 can sink/source one TTL load.

Microprocessor Bus

A microprocessor bus is provided to permit the 8051 to solve a wide range of problems and to allow the upward growth of user products. This multiplexed address and data bus provides an interface compatible with standard memories, MCS-80 peripherals and the MCS-85 memories that include on-chip programmable I/O ports and timing functions. These are summarized in the 8051 Microcomputer Expansion Components chart of Figure 6.

When accessing external memory the high-order address is emitted on Port 2 and the low-order address on Port 0. The ALE signal is provided for strobing the address into an external latch. The program store enable (\overline{PSEN}) signal is provided for enabling an external memory device to Port 0 during a read from the Program Memory address space. When the MOVX instruction is executed Port 3 automatically generates the read (\overline{RD}) signal for enabling an External Data Memory device to Port 0 or generates the write (\overline{WR}) signal for strobing the external memory device with the data emitted by Port 0. Port 0 emits the address and data to the external memory through a push/pull driver that can sink/source two TTL loads. At the end of the read/write bus cycle Port 0 is automatically reprogrammed to its high

intel

8031/8051/8751

PRELIMINARY

Category	I.D.	Description	Comments	Program Or Data Memory	Crystal Frequency MHz (Max)
I/O Expander		8 Line I/O Expander (Shift Register)	Low Cost I/O Expander		12
Standard EPROMs	2758	1K x 8 450 ns Light Erasable	User programmable and erasable.	P	9
	2716-1	2K x 8 350 ns Light Erasable		P	11
	2732	4K x 8 450 ns Light Erasable		P	9
	2732A	4K x 8 250 ns Light Erasable		P	12
Standard RAMs	2114A	1K x 4 100 ns RAM	Data memory can be easily expanded using standard NMOS RAMs.	D	12
	2148	1K x 4 70 ns RAM		D	12
	2142-2	1K x 4 200 ns RAM		D	12
Multiplexed Address/Data RAMs	8185A	1K x 8 300 ns RAM		D	12
Standard I/O	8212	8-Bit I/O Port	Serves as Address Latch or I/O port.	D	12
	8282	8-Bit I/O Port		D	12
	8283	8-Bit I/O Port	Three 8-bit programmable I/O ports. Serial Communications Receiver/Transmitter.	D	12
	8255A	Programmable Peripheral Interface		D	12
	8251A	Programmable Communications Interface		D	12
Standard Peripherals	8205	1 of 8 Binary Decoder	MCS-80 and MCS-85 peripheral devices are compatible with the 8051 allowing easy addition of specialized interfaces. Future MCS-80/85 devices will also be compatible.	D	12
	8286	Bi-directional Bus Driver		D	12
	8287	Bi-directional Bus Driver (Inverting)		D	12
	8253A	Programmable Interval Timer		D	12
	8279	Programmable Keyboard/Display Interface (128 Keys)		D	12
	8291 8292	GPIB Talker/Listener GPIB Controller		D D	12 11.7
Universal Peripheral Interfaces	8041A	ROM Program Memory	User programmable to perform custom I/O and control functions.	D/P	12/11.7
	8741A	EPROM Program Memory		D/P	12/11.7
Memories with on-chip I/O and Peripheral Functions.	8155-2	256 x 8 330 ns RAM		D	12
	8355-2	2K x 8 330 ns ROM		P	11.6
	8755-2	2K x 8 330 ns EPROM		P	11.6

Compatible MCS-80/85 Components

Figure 6. 8051 Microcomputer Expansion Components

Impedance state and Port 2 is returned to the state it had prior to the bus cycle. The 8051 generates the address, data and control signals needed by memory and I/O devices in a manner that minimizes the requirements placed on external program and data memories. At 12 MHz, the Program Memory cycle time is 500ns and the access times required from stable address and \overline{PSEN} are approximately 320ns and 150ns respectively. The External Data Memory cycle time is 1 μ s and the access times required from stable address and from read (\overline{RD}) or write (\overline{WR}) command are approximately 600ns and 250ns respectively.

TIMER/EVENT COUNTERS

The 8051 contains two 16-bit counters for measuring time intervals, measuring pulse widths, counting events and generating precise, periodic interrupt requests. Each can be programmed independently to operate similar to an 80488-bit timer with divide by 32 prescaler or as an 8-bit counter with divide by 32 prescaler (Mode 0), as a 16-bit time-interval or event counter (Mode 1), or as an 8-bit time-interval or event counter with automatic reload upon overflow (Mode 2).

Additionally, counter 0 can be programmed to a mode that divides it into one 8-bit time-interval or



event counter and one 8-bit time-interval counter (Mode 3). When counter 0 is in Mode 3, counter 1 can be programmed to any of the three aforementioned modes, although it cannot set an interrupt request flag or generate an interrupt. This mode is useful because counter 1's overflow can be used to pulse the serial port's transmission-rate generator. Along with their multiple operating modes and 16-bit precision, the counters can also handle very high input frequencies. These range from 0.1 MHz to 1.0 MHz (for 1.2 MHz to 12 MHz crystal) when programmed for an input that is a division by 12 of the oscillator frequency and from 0 Hz to an upper limit of 50 KHz to 0.5 MHz (for 1.2 MHz to 12 MHz crystal) when programmed for external inputs. Both internal and external inputs can be gated to the counter by a second external source for directly measuring pulse widths.

The counters are started and stopped under software control. Each counter sets its interrupt request flag when it overflows from all ones to all zeros (or auto-reload value). The operating modes and input sources are summarized in Figures 7 and 8. The effects of the configuration flags and the status flags are shown in Figures 9 and 10.

Serial Communications

The 8051 has a serial I/O port that is useful for serially linking peripheral devices as well as multiple 8051s through standard asynchronous protocols with full-duplex operation. The serial port also has a synchronous mode for expansion of I/O lines using CMOS and TTL shift registers. This hardware serial communications interface saves ROM code and permits a much higher transmission rate than could be achieved through software. In response to a serial port interrupt request the CPU has only to read/write the serial port's buffer to service the serial link. A block diagram of the serial port is shown in Figures 11 and 12. Methods for linking UART (universal asynchronous receiver/transmitter) devices are

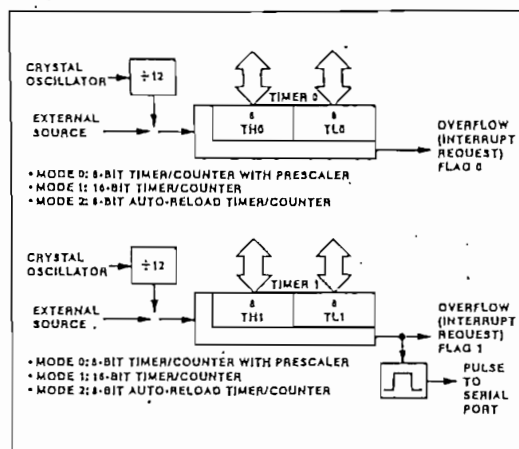


Figure 7. Timer/Event Counter Modes 0, 1 and 2

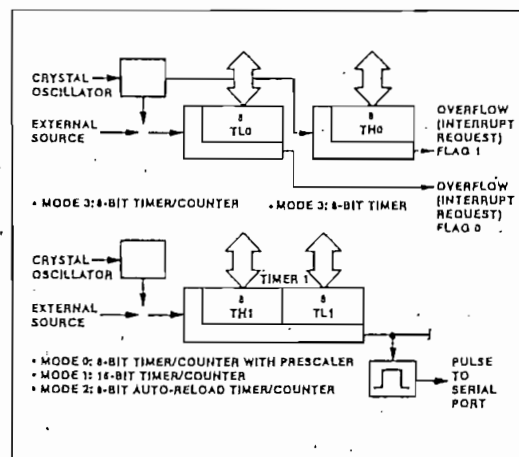


Figure 8. Timer/Event Counter 0 In Mode 3

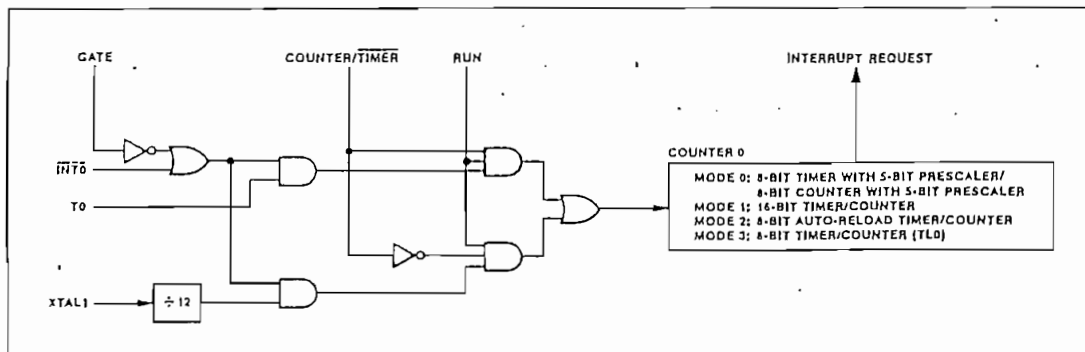


Figure 9. Timer/Counter 0 Control and Status Flag Circuitry

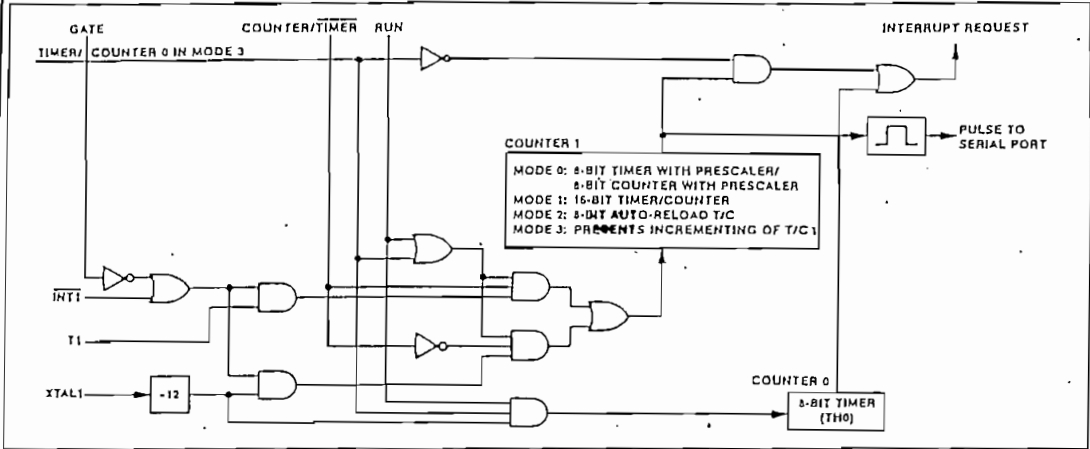


Figure 10. Timer/Counter 1 Control and Status Flag Circuitry

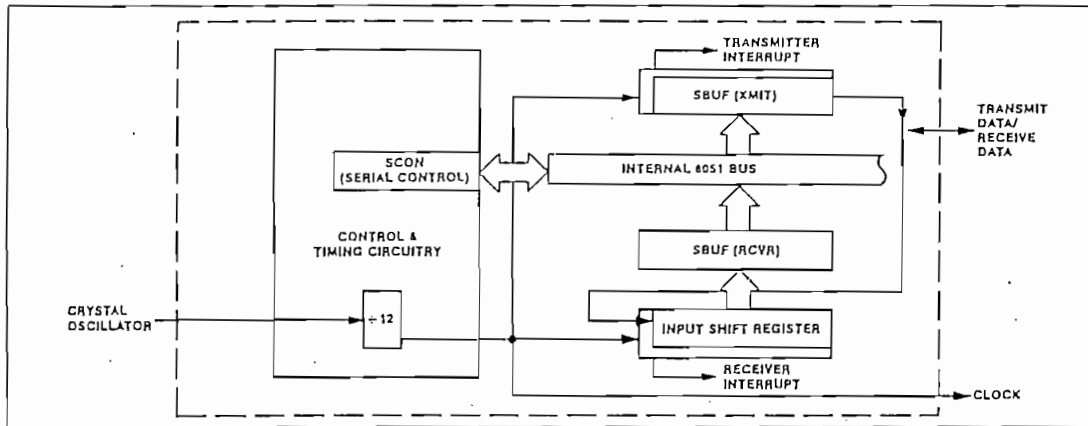


Figure 11. Serial Port—Synchronous Mode 0

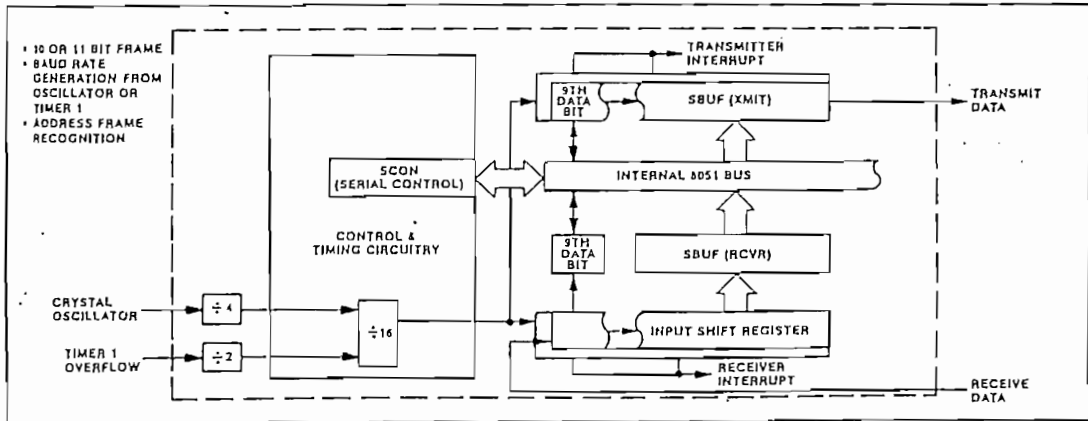


Figure 12. Serial Port—UART Modes 1, 2, and 3

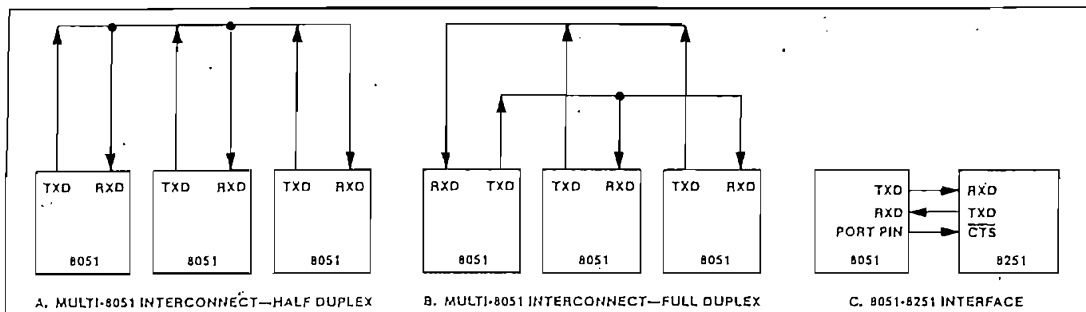


Figure 13. UART Interfacing Schemes

shown in Figure 13 and a method for I/O expansion is shown in Figure 14.

The full-duplex serial I/O port provides asynchronous modes to facilitate communications with standard UART devices, such as printers and CRT terminals, or communications with other 8051s in multi-processor systems. The receiver is double buffered to eliminate the overrun that would occur if the CPU failed to respond to the receiver's interrupt before the beginning of the next frame. Double buffering of the transmitter is not needed since the 8051 can generally maintain the serial link at its maximum rate without it. A minor degradation in transmission rate can occur in rare events such as when the servicing of the transmitter has to wait for a lengthy interrupt service program to complete. In asynchronous modes, false start-bit rejection is provided on received frames. For noise rejection a best two-out-of-three vote is taken on three samples near the center of each received bit.

When interfacing with standard UART devices the serial channel can be programmed to a mode (Mode 1) that transmits/receives a ten-bit frame or programmed to a mode (Mode 2 or 3) that transmits/receives an eleven-bit frame as shown in Figure 15. The frame consists of a start bit, eight or nine data bits and a stop bit. In Modes 1 and 3, the transmission-rate timing circuitry receives a pulse from counter 1 each time the counter overflows. The input to counter 1 can be an external source or a division by 12 of the oscillator frequency. The auto-reload mode of the counter provides communication rates of 122 to 31,250 bits per second (including start and stop bits) for a 12 MHz crystal. In Mode 2 the communication rate is a division by 64 of the oscillator frequency yielding a transmission rate of 187,500 bits per second (including start and stop bits) for a 12 MHz crystal.

Distributed processing offers a faster, more powerful system than can be provided by a single CPU processor. This results from a hierarchy of interconnected processors, each with its own memories and

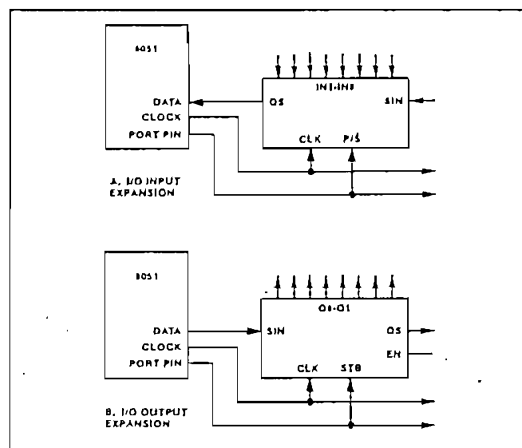


Figure 14. I/O Expansion Technique

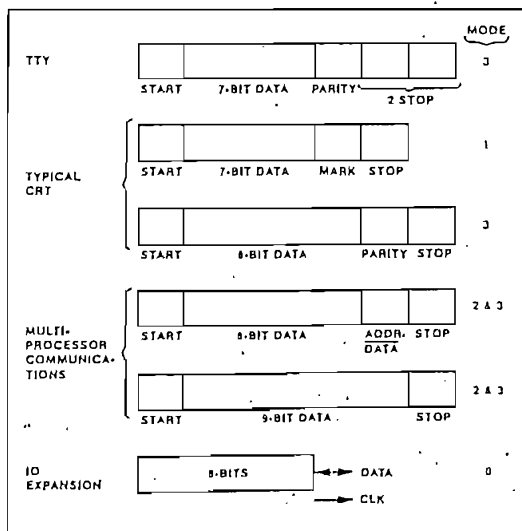


Figure 15. Typical Frame Formats



1. Slaves—Configure serial port to interrupt CPU if the received ninth data bit is a one (1).
2. Master—Transmit frame containing address in first 8 data bits and set ninth data bit (i.e. ninth data bit designates address frame).
3. Slaves—Serial port interrupts CPU when address frame is received. Interrupt service program compares received address to its address. The slave which has been addressed reconfigures its serial port to interrupt the CPU on all subsequent transmissions.
4. Master—Transmit control frames and data frames (these will be accepted only by the previously addressed slave).

Figure 16. Protocol for Multi-Processor Communications

I/O In multiprocessing, a host 8051 microcomputer controls a multiplicity of 8051s configured to operate simultaneously on separate portions of the program, each controlling a portion of the overall process. The interconnected 8051s reduce the load on the host processor and result in a low-cost system of data transmission. This form of distributed processing is especially effective in systems where controls in a complex process are required at physically separated locations.

In Modes 2 and 3 the automatic wake-up of slave processors through interrupt driven address-frame recognition is provided to facilitate interprocessor communications. The protocol for interprocessor communications is shown in Figure 16. In synchronous mode (Mode 0) the high speed serial port provides an efficient, low-cost method of expanding I/O lines using standard TTL and CMOS shift registers. The serial channel provides a clock output for synchronizing the shifting of bits to/from an external register. The data rate is a division by 12 of the oscillator frequency and is 1M bits per second at 12 MHz.

8051 Family Pin Description

V_{SS}

Circuit ground potential

V_{CC}

+5V power supply during operation, programming and verification

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port

It is also the multiplexed low-order address and data bus when using external memory. It is used for data input and output during programming and verification. Port 0 can sink/source two TTL loads.

PORT 1

Port 1 is an 8-bit quasi-bidirectional I/O port. It is used for the low-order address byte during programming and verification. Port 1 can sink/source one TTL load.

PORT 2

Port 2 is an 8-bit quasi-bidirectional I/O port. It also emits the high-order 8 bits address when accessing external memory. It is used for the high-order address and the control signals during programming and verification. Port 2 can sink/source one TTL load.

PORT 3

Port 3 is an 8-bit quasi-bidirectional I/O port. It also contains the interrupt, timer, serial port and RD and WR pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. Port 3 can sink/source one TTL load. The secondary functions are assigned to the pins of Port 3, as follows:

- RXD/data (P3.0). Serial port's receiver data input (asynchronous) or data input/output (synchronous).
- TXD/clock (P3.1). Serial port's transmitter data output (asynchronous) or clock output (synchronous).
- $\overline{INT0}$ (P3.2). Interrupt 0 input or gate control input for counter 0.
- $\overline{INT1}$ (P3.3). Interrupt 1 input or gate control input for counter 1.
- T0 (P3.4). Input to counter 0.
- T1 (P3.5). Input to counter 1.
- \overline{WR} (P3.6). The write control signal latches the data byte from Port 0 into the External Data Memory.
- \overline{RD} (P3.7). The read control signal enables External Data Memory to Port 0.

RST/ V_{PD}

A low to high transition on this pin (at approximately 3V) resets the 8051. If V_{PD} is held within its spec (approximately +5V), while V_{CC} drops below spec, V_{PD} will provide standby power to the RAM. When V_{PD} is low, the RAM's current is drawn from V_{CC} . A small internal resistor permits power-on reset using only a capacitor connected to V_{CC} .

ALE/ $\overline{P\overline{R}\overline{O}\overline{G}}$

Provides Address Latch Enable output used for latching the address into external memory during normal operation. Receives the program pulse

intel

8031/8051/8751

PRELIMINARY

input during EPROM programming.

PSEN

The Program Store Enable output is a control signal that enables the external Program Memory to the bus during normal fetch operations.

 \overline{EA}/VDD

When held at a TTL high level, the 8051 executes instructions from the internal ROM/EPROM when the PC is less than 4096. When held at a TTL low level, the 8051 fetches all instructions from external Program Memory. The pin also receives the 21V EPROM programming supply voltage.

XTAL1

Input to the oscillator's high gain amplifier. A crystal or external source can be used.

XTAL2

Output from the oscillator's amplifier. Required when a crystal is used.

8051 FAMILY DEVELOPMENT SYSTEM AND SOFTWARE SUPPORT

The 8051 is supported by a total range of Intel development tools. This broad range of support shortens the product development cycle and thus brings the product to market sooner.

- ASM51 Absolute macro assembler for the 8051.
- CONV51 8048 assembly language source code to 8051 assembly source code conversion program.
- EM-51 8051/8751 emulator board that uses a modified 8051 and an EPROM.
- ICE-51™ Real-time in-circuit emulator.
- SDK-51 System Design Kit for Developing User Prototype around the 8051.
- UPP-551 8751 personality card for UPP-103 Universal PROM Programmer.
- 8051 Workshop.

8051 Software Development Package (ASM51 and CONV51)

The 8051 software development package provides development system support for the powerful 8051 family of single chip microcomputers. The package contains a symbolic macro assembler and a 8048 to 8051 source code converter. This diskette-based software package runs under ISIS-II on any Intel® Microcomputer Development System with 64K bytes of memory.

8051 Macro Assembler (ASM51)

The 8051 macro assembler translates symbolic 8051

assembly language instructions into machine executable object code. These assembly language mnemonics are easier to program and are more readable than binary or hexadecimal machine instructions. Also, by allowing the programmer to give symbolic names to memory locations rather than absolute addresses, software design and debug are performed more quickly and reliably.

ASM51 provides symbolic access for the many useful addressing methods in the 8051 architecture which reference bit, nibble and byte locations.

The assembler supports macro definitions and calls. This provides a convenient means of programming a frequently used code sequence only once. The assembler also provides conditional assembly capabilities. Cross referencing is provided in the symbol table listing, which shows the user the lines in which each symbol was defined and referenced.

If an 8051 program contains errors, the assembler provides a comprehensive set of error diagnostics, which are included in the assembly listing.

The object code generated may be used to program the 8751 EPROM version of the chip or sent to Intel for fabricating the 8051 ROM version. The assembler output can also be debugged using the ICE-51 in-circuit emulator.

8048 to 8051 Assembly Language Converter Utility Program (CONV51)

The 8048 to 8051 assembly language converter is a utility to help users of the MCS-48 family of microcomputers upgrade their designs to the high performance 8051 architecture. By converting 8048 source code to 8051 source code, the investment in software developed for the 8048 is maintained when the system is upgraded.

8051 Emulation Board (EM-51)

The EM-51 8051 emulation board is a small (2.85" x 5.25") board which emulates an 8031/8051/8751 microcomputer using standard PROMs or EPROMs in place of the 8051's on-chip program memory. The board includes a modified 8051 microcomputer, supporting circuits, and two sockets for program memory. The user may select two 2716 EPROMs, a 2732 EPROM, or two 3636 bipolar PROMs depending on crystal frequency and power requirements.

8051 In-Circuit Emulator (ICE-51™)

The 8051 In-Circuit Emulator resides in the Intel development system. The development system interfaces with the user's 8051 system through an in-cable buffer box with the cable terminating in an 8051 pin-compatible plug. Together these replace the 8051 device in the system. With the emulator plug in place, the designer can exercise the system in real-time while collecting up to 255 instruction



8031/8051/8751

PRELIMINARY

cycles of real-time data. In addition, he can single step the system program.

Static RAM memory is available in the ICE-51 buffer box to emulate the 8051's internal and external program memories and external data memory. The designer can display and alter the contents of the replacement memory in the ICE-51 buffer box, internal 8051 registers, internal data RAM, and Special Function Registers. Symbolic reference capability allows the designer to use meaningful symbols provided by ASM51 rather than absolute values when examining and modifying these memory, register, flag, and I/O locations in his system.

Universal PROM Programmer

Personality Card (UPP-551)

The UPP-551 is a personality card for the UPP-103 Universal PROM Programmer. The Universal PROM

Programmer is an Intellec system peripheral capable of programming and verifying the 8751 when the UPP-551 is inserted. Programming and verification operations are initiated from the Intellec development system console and are controlled by the Universal PROM Mapper (UPM) program.

8051 Workshop

The workshop provides the design engineer or system designer hands-on experience with the 8051 microcomputers. The course includes explanation of the Intel 8051 architecture, system timing and input/output design. Lab sessions will allow the attendee to gain detailed familiarity with the 8051 family and support tools.

INSITE™ Library

The INSITE Library contains 8051 utilities and applications programs.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	65°C to +150°C
Voltage on Any Pin With Respect to Ground (V _{SS})	-0.5V to 17V
Power Dissipation	2 Watts

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5V ± 5%, V_{SS} = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Units	Test Conditions
V _{IL}	Input Low Voltage (All except XTAL1)	-0.5		0.8	V	
V _{IL1}	Input Low Voltage (XTAL1)	-0.5		TBD	V	
V _{IH}	Input High Voltage (All Except XTAL1, RST, V _{P(J)})	2.0		V _{CC} -0.5	V	
V _{IH1}	Input High Voltage (XTAL1)	TBD		V _{CC} +0.5	V	
V _{IH2}	Input High Voltage (RST)	3.0		V _{CC} +0.5	V	
V _{IH3}	Input High Voltage (V _{P(J)})	4.5		5.5	V	Power Down Only (V _{CC} = 0)
V _{OL}	Output Low Voltage (All Outputs Except Port 0)			0.45	V	I _{OL} = 2 mA
V _{OL1}	Output Low Voltage (Port 0)			0.45	V	I _{OL} = 4 mA
V _{OH}	Output High Voltage (All Outputs Except Port 0, ALE and PSEN)	2.4			V	I _{OH} = -100 μA
V _{OH1}	Output High Voltage (ALE and PSEN, Port 0 in External Bus Mode)	2.4			V	I _{OH} = -400 μA
I _{LO}	Pullup Resistor Current (P1, P2, P3)			500	μA	.45V ≤ V _{IN} ≤ V _{CC}
I _{LO1}	Output Leakage Current (P0)			±10	μA	.45V ≤ V _{IN} ≤ V _{CC}
I _{CC}	Power Supply Current (All Outputs Disconnected)			150	mA	T _A = 25°C
I _{PD}	Power Down Supply Current			20	mA	T _A = 25°C, V _{PD} = 5V, V _{CC} = 0V
C _{IO}	Capacitance of I/O Buffer			10	pF	f _c = 1 kHz

intel

8031/8051/8751

PRELIMINARY

A.C. CHARACTERISTICS (TA = 0°C to 70°C; V_{CC} = 5V ± 5%; CL for Port 0, ALE and PSEN Outputs = 150 PF; CL for All Other Outputs = 80 PF)

Program Memory Characteristics

Symbol	Parameter	12MHz Clock			Variable Clock 1/TCLCL=1.2 MHz to 12 MHz		
		Min.	Max.	Units	Min.	Max.	Units
TCLCL	Oscillator Period	83		ns			ns
TCY	Min Instruction Cycle Time	1.0		μs	12TCLCL	12TCLCL	ns
TLHLL	ALE Pulse Width	140		ns	2TCLCL-30		ns
TAVLL	Address Set Up To ALE	60		ns	TCLCL-25		ns
TLLAX	Address Hold After ALE	50		ns	TCLCL-35		ns
TPLPH	PSEN Width	230		ns	3TCLCL-20		ns
TLHLH	PSEN, ALE Cycle Time	500		ns	6TCLCL		ns
TPLIV	PSEN To Valid Instr In		150	ns		3TCLCL-100	ns
TPHDX	Input Data Hold After PSEN	0		ns	0		ns
TPHDZ	Input Data Float After PSEN		75	ns		TCLCL-10	ns
TAVIV	Address To Valid Instr In		320	ns		5TCLCL-100	ns
TAZPL	Address Float To PSEN	0		ns	0		ns

External Data Memory Characteristics

Symbol	Parameter	12MHz Clock			Variable Clock		
		Min.	Max.	Units	Min.	Max.	Units
TRLRH	RD Pulse Width	400		ns	6TCLCL-100		ns
TWLWH	WR Pulse Width	400		ns	6TCLCL-100		ns
TRLDV	RD To Valid Data In		250	ns		5TCLCL-170	ns
TRHDX	Data Hold After RD	0		ns	0		ns
TRHDZ	Data Float After RD		100	ns		2TCLCL-70	ns
TAVDV	Address To Valid Data In		600	ns		9TCLCL-150	ns
TAVWL	Address To WR or RD	200		ns	4TCLCL-130		ns
TOVWH	Data Setup Before WR	400		ns	7TCLCL-180		ns
TWHOX	Data Held After WR	80		ns	2TCLCL-90		ns

NOTE:

There are 2 to 8 ALE cycles per instruction. Clocks and state timing are shown on the timing diagram for reference purposes only. They are not accessible outside the package. TCY is the minimum instruction cycle time which consists of 12 oscillator clocks or two ALE cycles. Address setup and hold time from ALE are the same for data and program memory.

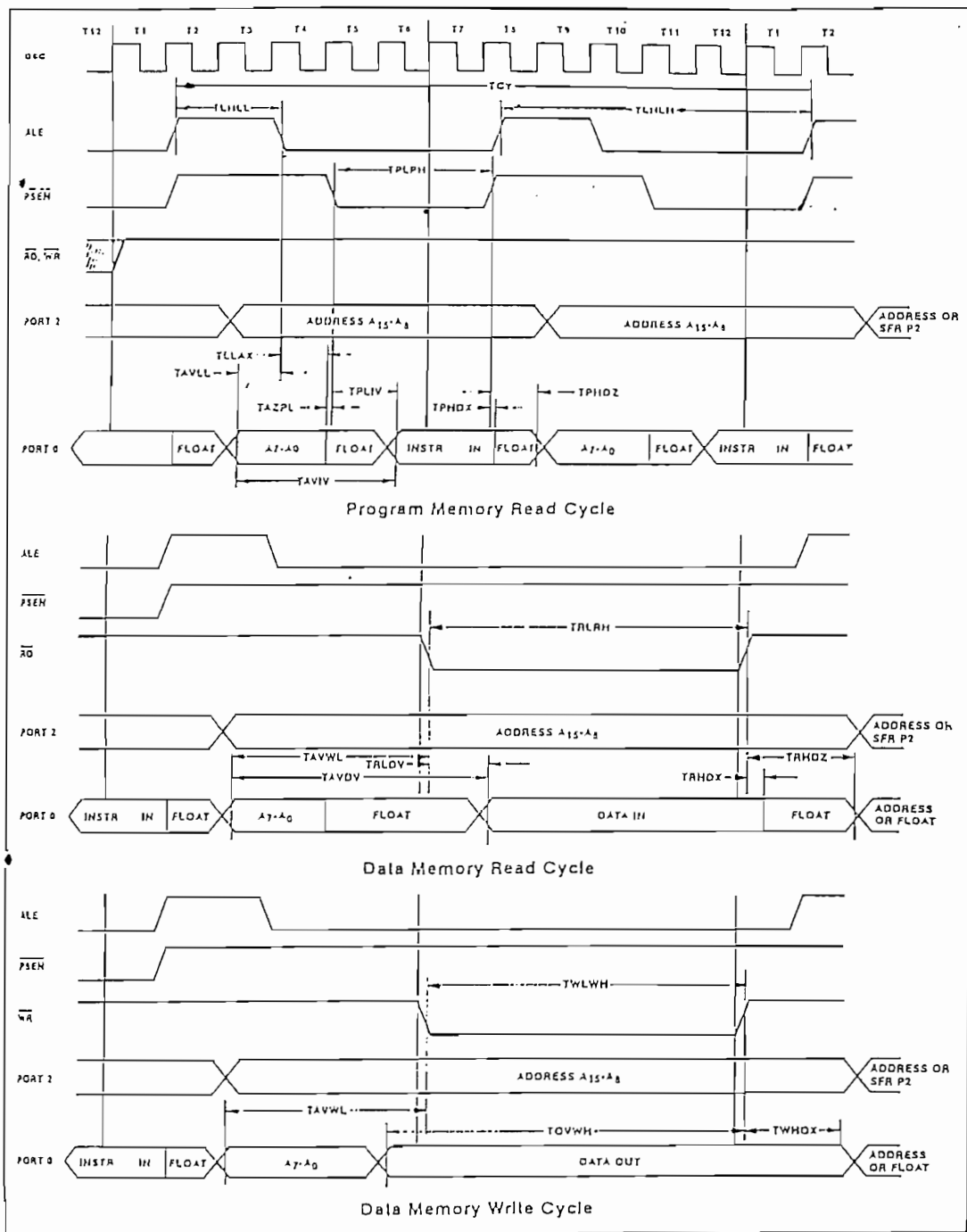




Table 1. 8051 Instruction Set Summary

Notes on instruction set and addressing modes:
Rn — Register R7-R0 of the currently selected Register Bank.
data — 8-bit internal data location's address. This could be an internal data RAM location (0-127) or a SFR (i.e. I/O port, control register, status register, etc. (128-255)).
@Ri — 8-bit internal data RAM location (0-255) addressed indirectly through register R1 or R0.
#data — 8-bit constant included in instruction.
#data16 — 16-bit constant included in instruction.
addr16 — 16-bit destination address. Used by LCALL & LJMP. A branch can be anywhere within the 64K-byte program memory address space.
addr11 — 11-bit destination address. Used by ACALL & AJMP. The branch will be within the same 2K-byte page of program memory as the first byte of the following instruction.
rel — Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit — Direct Addressed bit in internal data RAM or Special Function Register.
 — New operation not provided by 8048/8049.

Interrupt Response Time: To finish execution of current instruction, respond to the interrupt request, push the PC and to vector to the first instruction of the interrupt service program requires 38 to 81 oscillator periods (3 to 7 μ s @ 12MHz).

INSTRUCTIONS THAT AFFECT FLAG SETTINGS¹

INSTRUCTION	FLAG			INSTRUCTION	FLAG		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	O	X		ANL C, bit	X		
DIV	O	X		ORL C, bit	X		
DA	X			ORL C, bit	X		
RRC	X			MOV C, bit	X		
RLC	X			CJNE	X		
SETB C	1						

¹Note that operations on SFR byte address 208 or bit addresses 209-215 (i.e. the PSW or bits in the PSW) will also affect flag settings.

Data Transfer

Mnemonic	Description	Bytes	Oscillator Periods
MOV A,Rn	Move register to A	1	12
*MOV A,data	Move direct byte to A	2	12
MOV A,@Ri	Move indirect RAM to A	1	12
MOV A,#data	Move immediate data to A	2	12
MOV Rn,A	Move A to register	1	12
*MOV Rn,data	Move direct byte to register	2	24
MOV Rn,#data	Move immediate data to register	2	12
*MOV data,A	Move A to direct byte	2	12
*MOV data,Rn	Move register to direct byte	2	24
*MOV data,data	Move direct byte to direct byte	3	24
*MOV data,@Ri	Move indirect RAM to direct byte	2	24
*MOV data,#data	Move immediate data to direct byte	3	24
MOV @Ri,A	Move A to indirect RAM	1	12
*MOV @Ri,data	Move direct byte to indirect RAM	2	24
MOV @Ri,#data	Move immediate data to indirect RAM	2	12
*MOV DPTR,#data16	Move 16-bit constant to Data Pointer	3	24
*MOV C,bit	Move direct bit to carry	2	12
*MOV bit,C	Move carry to direct bit	2	24
*MOV A,@A+DPTR	Move Program Memory byte addressed by A+DPTR to A	1	24
*MOV C,A,@A+PC	Move Program Memory byte addressed by A+PC to A	1	24
MOVX A,@Ri	Move External Data (8-bit address) to A	1	24
*MOVX A,@DPTR	Move External Data (16-bit address) to A	1	24
MOVX @Ri,A	Move A to External Data (8-bit address)	1	24
*MOVX @DPTR,A	Move A to External Data (16-bit address)	1	24
*PUSH data	Move direct byte to stack and inc. SP	2	24
*POP data	Move direct byte from stack and dec. SP	2	24
XCH A,Rn	Exchange register with A	1	12
*XCH A,data	Exchange direct byte with A	2	12
XCH A,@Ri	Exchange indirect RAM with A	1	12
XCHD A,@Ri	Exchange indirect RAM's least sig nibble with A's LSN	1	12

Logic

Mnemonic	Description	Bytes	Oscillator Periods
ANL A,Rn	AND register to A	1	12
*ANL A,data	AND direct byte to A	2	12
ANL A,@Ri	AND indirect RAM to A	1	12
ANL A,#data	AND immediate data to A	2	12
*ANL data,A	AND A to direct byte	2	12
*ANL data,#data	AND immediate data to direct byte	3	24
*ANL C,bit	AND direct bit to carry	2	24
*ANL C,/bit	AND complement of direct bit to carry	2	24
ORL A,Rn	OR register to A	1	12
*ORL A,data	OR direct byte to A	2	12
ORL A,@Ri	OR indirect RAM to A	1	12
ORL A,#data	OR immediate data to A	2	12
*ORL data,A	OR A to direct byte	2	12
*ORL data,#data	OR immediate data to direct byte	3	24
*ORL C,bit	OR direct bit to carry	2	24
*ORL C,/bit	OR complement of direct bit to carry	2	24
XRL A,Rn	Exclusive-OR register to A	1	12
*XRL A,data	Exclusive-OR direct byte to A	2	12
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	12
XRL A,#data	Exclusive-OR immediate data to A	2	12
*XRL data,A	Exclusive-OR A to direct byte	2	12
*XRL data,#data	Exclusive-OR immediate data to direct byte	3	24
*SETB C	Set carry	1	12
*SETB bit	Set direct bit	2	12
CLR A	Clear A	1	12
CLR C	Clear carry	1	12
*CLR bit	Clear direct bit	2	12
CPL A	Complement A	1	12
CPL C	Complement carry	1	12
*CPL bit	Complement direct bit	2	12
RL A	Rotate A Left	1	12
RLC A	Rotate A Left through carry	1	12
RR A	Rotate A Right	1	12
RRC A	Rotate A Right through carry	1	12
SWAP A	Rotate A left four (exchange nibbles within A)	1	12

intel

8031/8051/8751

PRELIMINARY

Arithmetic			
Mnemonic	Description	Bytes	Oscillator Periods
ADD A,Rn	Add register to A	1	12
ADD A,data	Add direct byte to A	2	12
ADD A,@RI	Add indirect RAM to A	1	12
ADD A,#data	Add immediate data to A	2	12
ADDC A,Rn	Add register and carry flag to A	1	12
ADDC A,data	Add direct byte and carry flag to A	2	12
ADDC A,@RI	Add indirect RAM and carry flag to A	1	12
ADDC A,#data	Add immediate data and carry flag to A	2	12
SUBB A,Rn	Subtract register and carry flag from A	1	12
SUBB A,data	Subtract direct byte and carry flag from A	2	12
SUBB A,@RI	Subtract indirect RAM and carry flag from A	1	12
SUBB A,#data	Subtract immediate data and carry flag from A	2	12
INC A	Increment A	1	12
INC Rn	Increment register	1	12
INC data	Increment direct byte	2	12
INC @RI	Increment indirect RAM	1	12
DEC A	Decrement A	1	12
DEC Rn	Decrement register	1	12
DEC data	Decrement direct byte	2	12
DEC @RI	Decrement indirect RAM	1	12
INC DPTR	Increment Data Pointer	1	24
MUL AB	Multiply A times B	1	48
DIV AB	Divide A by B	1	48
DA A	Decimal add Adjust of A	1	12

Other			
Mnemonic	Description	Bytes	Oscillator Periods
HOP	No Operation	1	12

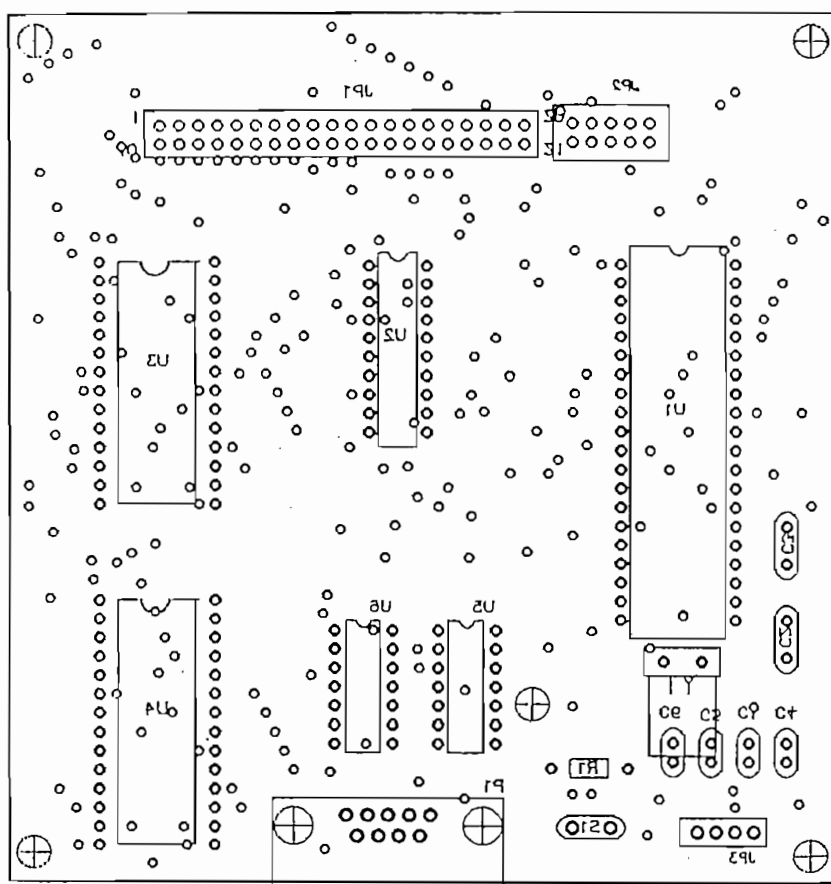
Control Transfer (Branch)			
Mnemonic	Description	Bytes	Oscillator Periods
AJMP addr11	Absolute Jump	2	24
LJMP addr16	Long Jump	3	24
SJMP rel	Short Jump	2	24
JMP @A+DPTR	Jump indirect relative to the DPTR	1	24
JZ rel	Jump if A is zero	2	24
JNZ rel	Jump if A is not zero	2	24
JC rel	Jump if carry is set	2	24
JNC rel	Jump if carry is not set	2	24
JB bit,rel	Jump relative if direct bit is set	3	24
JNB bit,rel	Jump relative if direct bit is not set	3	24
JBC bit,rel	Jump relative if direct bit is set, then clear bit	3	24
CJNE A,data,rel	Compare direct byte to A & Jump if not Eq. See Note a.	3	24
CJNE A,#data,rel	Compare immed. to A & Jump if not Eq. See Note a.	3	24
CJNE Rn,#data,rel	Compare immed. to reg & Jump if not Eq. See Note a.	3	24
CNJE @RI,#data,rel	Compare immed. to indirect RAM & Jump if not Eq. See Note a.	3	24
DJNZ Rn,rel	Decrement register & Jump if not zero	2	24
DJNZ data,rel	Decrement direct byte & Jump if not zero	3	24

Note a) Set C if the first operand is less than the second operand; else clear

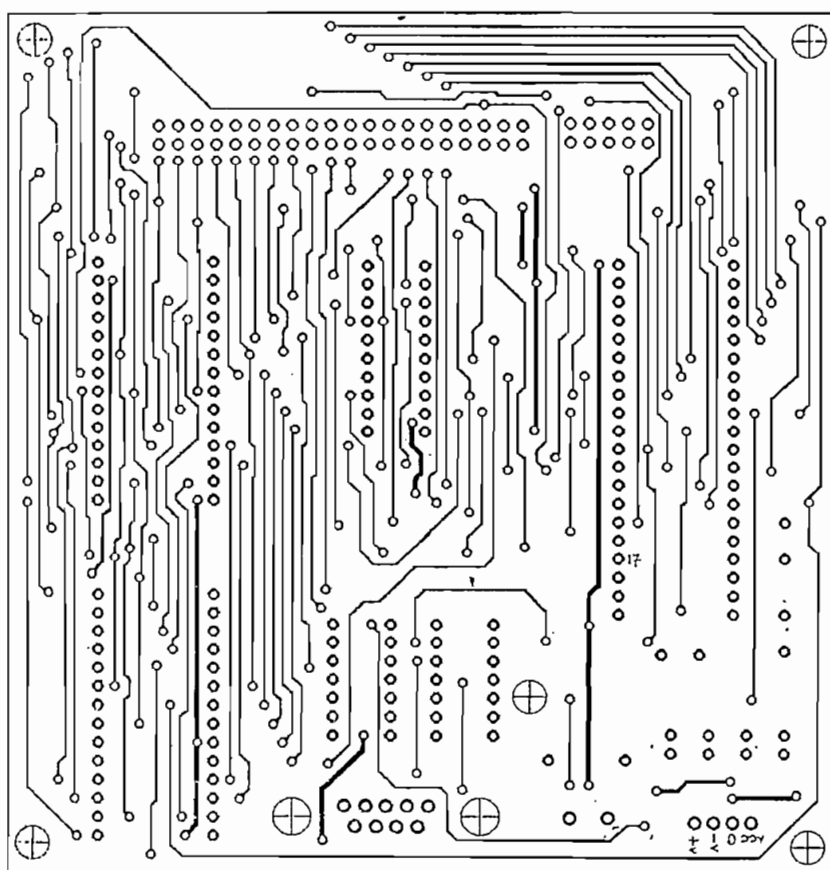
Control Transfer (Subroutine)			
Mnemonic	Description	Bytes	Oscillator Periods
ACALL addr11	Absolute Subroutine Call	2	24
LCALL addr16	Long Subroutine Call	3	24
RET	Return from Subroutine Call	1	24
RETI	Return from Interrupt Call	1	24

**ANEXO 5 : DISTRIBUCION DE LOS COMPONENTES EN LAS
TARJETAS 1 y 2**

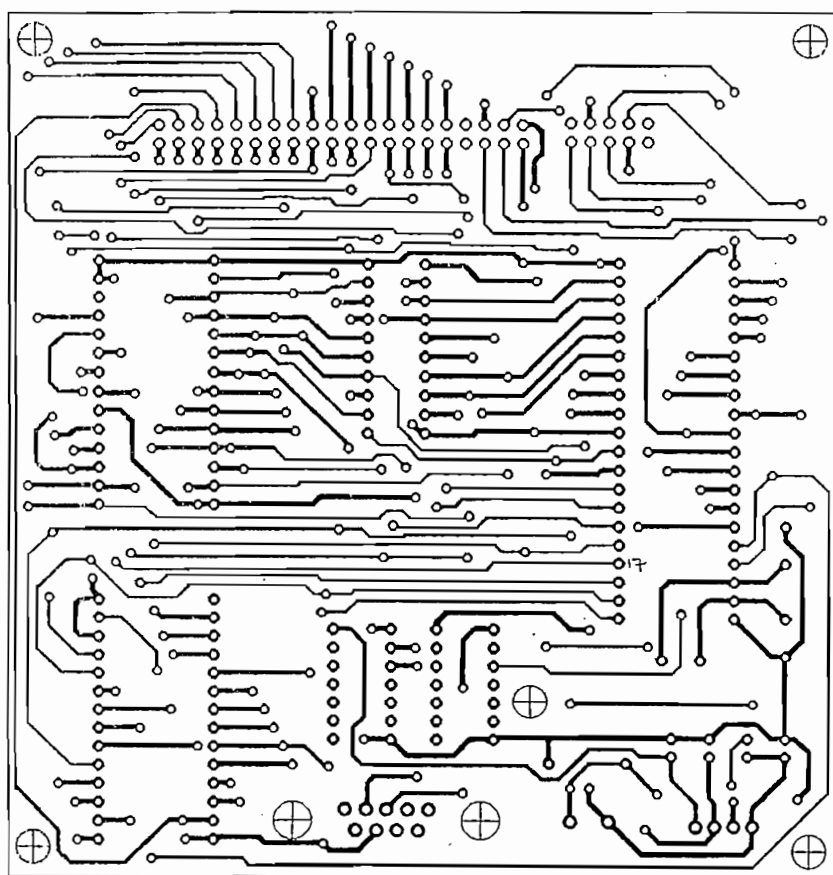
TARJETA # 1 : DISTRIBUCIÓN DE LOS COMPONENTES



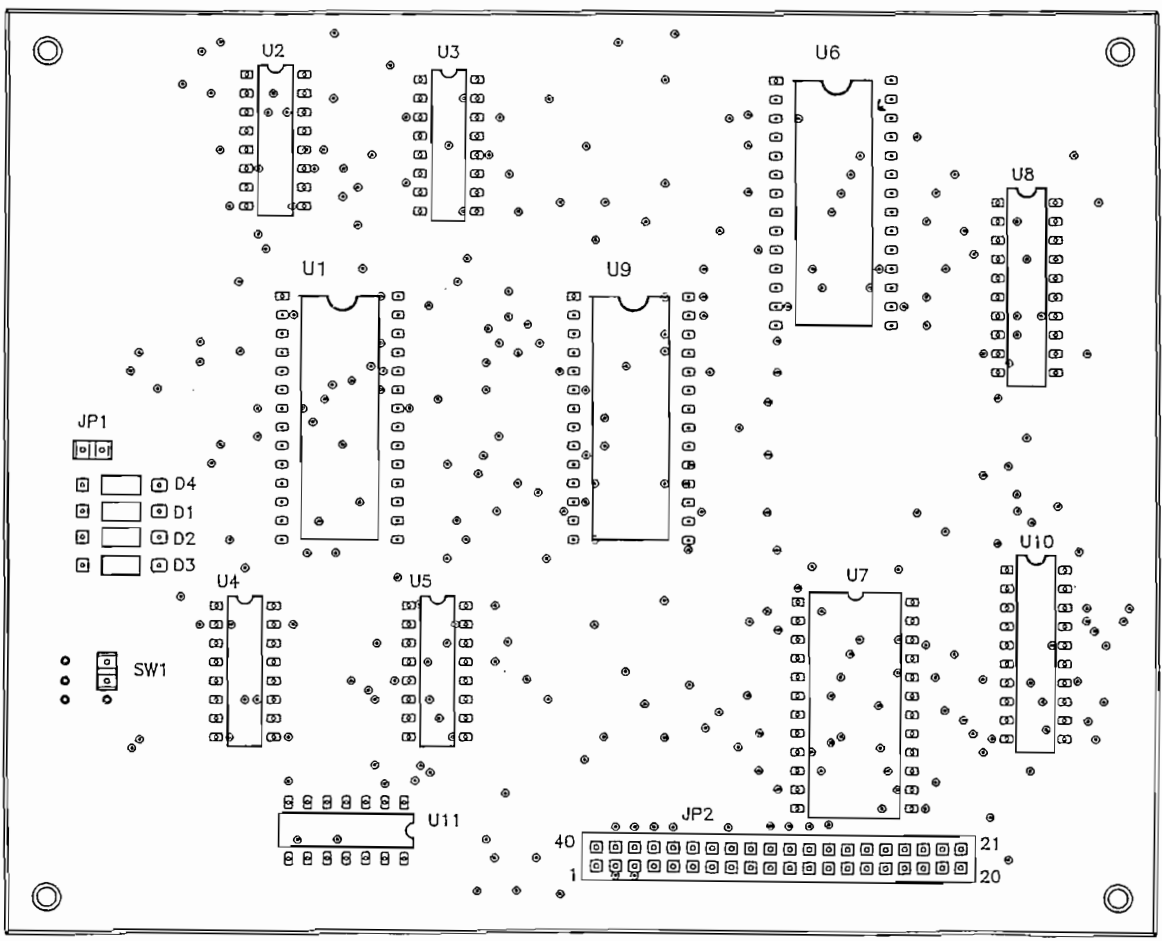
TARJETA # 1 : LADO DE LOS COMPONENTES



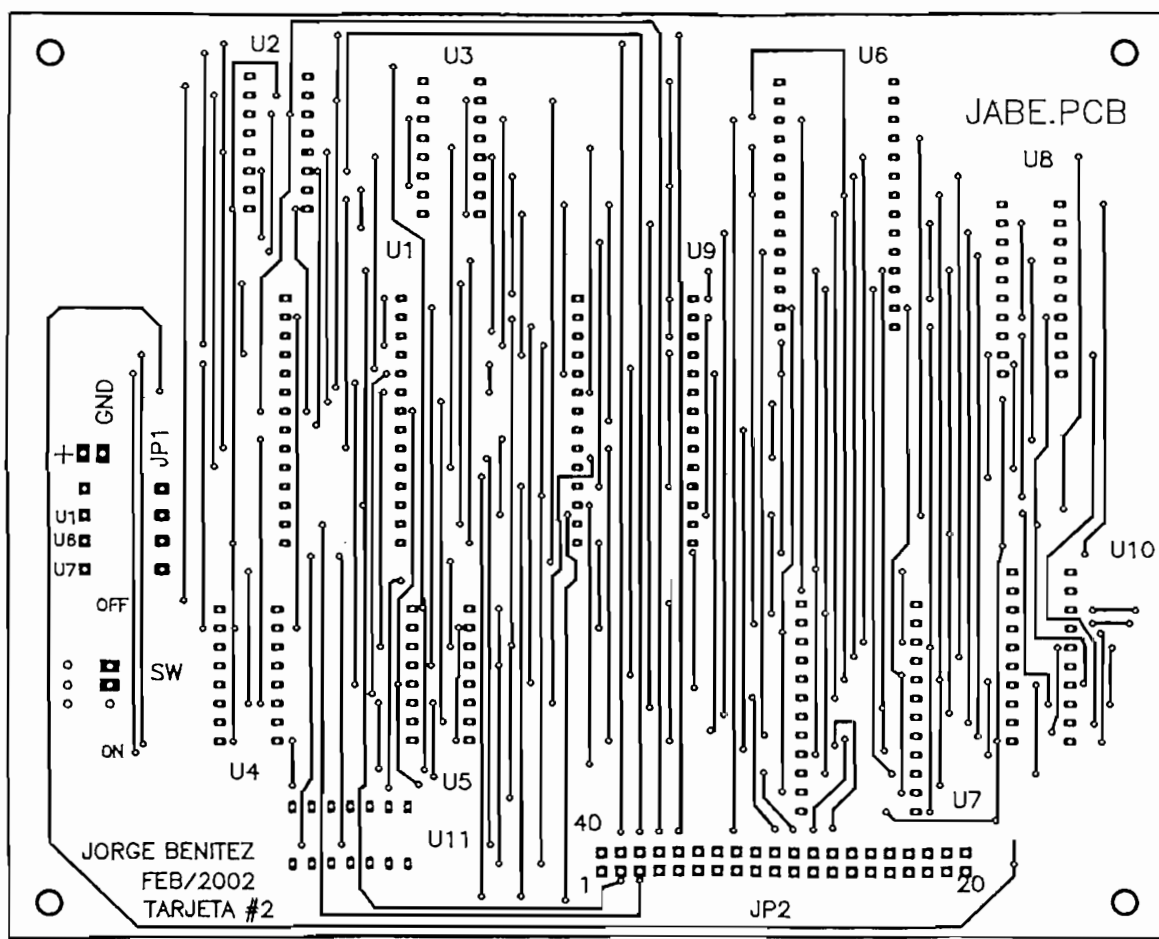
TARJETA # 1 : LADO DE LAS SUELDAS



TARJETA # 2 : DISTRIBUCIÓN DE LOS COMPONENTES



TARJETA # 2 : LADO DE LOS COMPONENTES



TARJETA # 2 : LADO DE LAS SUELDAS

