

ESCUELA POLITECNICA NACIONAL
FACULTAD DE INGENIERIA ELECTRICA

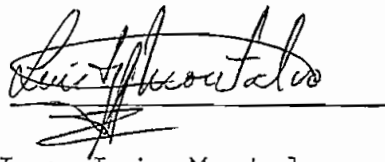
SISTEMA DE CONVERSION DE INTERFACES
SERIAL - PARALELO

ANA C. QUINTANA A.

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

ABRIL 1992

Certifico que la presente Tesis
ha sido elaborada en su totalidad
por la Srta. Ana C. Quintana A.

A handwritten signature in dark ink, appearing to read 'Luis Montalvo', is written over a horizontal line. Below the line, there is a small, stylized mark that looks like a cross or a star.

Ing. Luis Montalvo

DEDICATORIA

A MIS PADRES Y HERMANOS

AGRADECIMIENTO

Agradezco a todas las personas que de una u otra manera han ayudado a la realización de esta Tesis y en especial al Ing. Luis Montalvo por su acertada dirección en el desarrollo del presente trabajo de Tesis.

INDICE

PAG.

INTRODUCCION	i
--------------------	---

CAPITULO I

ASPECTOS GENERALES

1.1 Características de la Norma EIA-RS232C

1.1.1	Introducción	1
1.1.2	Comunicación Sincrónica y Asincrónica ...	3
1.1.3	Características Eléctricas	6
1.1.4	Características Mecánicas	8
1.1.5	Descripción de las Señales	9
1.1.6	Control de la Comunicación	16
1.1.7	Usos no Normalizados de las señales de Control	18
1.1.8	Formas de Conexión de la Interfaz	19
1.1.9	Pasos para la Comunicación Usando una Interfaz EIA-RS232C	26

1.2 Especificaciones de la norma IEEE-488

1.2.1	Introducción	27
1.2.2	Características del Bus	29
1.2.3	Estructura del Bus	30
1.2.4	Líneas y Señales de Bus	33
1.2.5	Especificaciones Mecánicas	37
1.2.6	Funciones de Interfaz	38
1.2.7	Características Eléctricas	47
1.2.8	Temporización para la Transición de Estados	50
1.2.9	Tipos de Mensajes	51
1.2.10	Transferencia y Codificación de Mensajes Remotos	53
1.2.11	Velocidad de Transferencia de Mensajes ..	56
1.2.12	Direccionamiento	58
1.2.13	Petición de Servicio del Bus	61
1.2.14	Diálogo	65

1.3	Definición de las Características del Sistema ..	70
-----	--	----

CAPITULO II

DESARROLLO DEL HARDWARE

2.1	Introducción	72
2.2	Diagrama de Bloques General del Sistema	75
2.3	Descripción y Diseño de los Bloque Individuales	
2.3.1	Bloque EIA-RS232C	78
2.3.2	Bloque IEEE-488	80
2.3.3	Bloque de Control	112
2.3.4	Funcionamiento del Sistema	115
2.3.5	Diagrama de Conexiones del Sistema de Conversión de Interfaces Serial-Paralelo	122

CAPITULO III

DESARROLLO DEL SOFTWARE

3.1	Descripción General del Software	124
3.2	Módulo para la Parte EIA-RS232C	124
3.2.1	Inicialización del Puerto Serial	126
3.2.2	Diseño de la Subrutina para Transmisión de Datos	134
3.2.3	Diseño de la Subrutina para Recepción de Datos	136
3.2.4	Diseño de la Subrutina de Detección Automática de Ritmo Binario	138
3.3	Módulo para el Bloque IEEE-488	144
3.3.1	Diseño de la Subrutina NO_CONT	145
3.3.2	Diseño de la Subrutina CONT	148
3.4	Software para las Pruebas del Sistema	
3.4.1	Diseño de Programas para la Parte EIA-RS232C	178
3.4.2	Diseño de Programas para la Parte IEEE-488	193

CAPITULO IV

RESULTADOS EXPERIMENTALES COMENTARIO Y CONCLUSIONES

4.1	Resultado Experimentales	198
-----	--------------------------------	-----

4.1.1	Equipos de Pruebas	198
4.1.2	Protocolo de Pruebas	200
4.2	Conclusiones y Recomendaciones	201

BIBLIOGRAFIA	205
--------------------	-----

ANEXO A	Manual de Uso
ANEXO B	Representación del Código ISO 7-Bits
ANEXO C	Información Técnica de los Circuito Integrados

INTRODUCCION

El propósito principal de la presente Tesis es realizar un sistema que permita la interconexión de equipos compatibles con la norma EIA-RS232C y equipos con salida IEEE-488.

Cada norma tiene diferentes especificaciones, por lo tanto el sistema debe convertir las señales y protocolos de una interfaz a los adecuados para la otra interfaz.

La normalización de una interfaz incluye cuatro elementos básicos: mecánico, eléctrico, funcional y operacional. El elemento mecánico se refiere a los medios físicos de la interconexión, esto es cables y conectores. El aspecto eléctrico comprende los límites de voltajes y corrientes, convenciones lógicas y sobretodo las especificaciones del medio de intercambio.

Los elementos funcionales describen las definiciones precisas de cada línea de señal, su propósito y método de uso, el protocolo y la temporización relativa requerida para el intercambio de mensajes, este elemento es el que tiene un significativo impacto en la compatibilidad entre productos diseñados independientemente. El cuarto elemento se relaciona con los parámetro dependientes del dispositivo (por ejemplo aplicaciones de software, códigos

de programas, rutinas de diagnóstico).

La interfaz IEEE-488 ofrece un medio normalizado de integrar instrumentos y un computador en un sistema de instrumentación automatizado. Su protocolo ofrece suficiente flexibilidad y compatibilidad para interconectar dispositivos de diferentes fabricantes. Con un computador personal puede ser usado para mediciones, adquisición de datos y control en el laboratorio y la industria.

La interfaz EIA-RS232C es la norma más común y de más amplio uso para la interconexión del equipo terminal de datos y el equipo de comunicación de datos. Conectando un terminal a través de un modem a la línea telefónica que lo conectará a un terminal lejano. Es utilizada también en otras aplicaciones como el conectar dos terminales directamente y diversos periféricos a un computador.

Para realizar el diseño del sistema de conversión de interfaces serial - paralelo la Tesis se ha distribuido en cuatro capítulos que contienen los siguientes puntos:

El Capítulo I contiene una revisión de las características más importantes de las norma EIA-RS23C e IEEE-488, definiendo las especificaciones que se tomarán

en cuenta de cada una de las normas en el diseño del sistema.

El Capítulo II presenta el diseño del hardware del sistema. Describiendo sus tres partes principales que son: la parte de la interfaz EIA-RS232C, la parte de la interfaz IEEE-488 y el bloque de control.

La descripción del software se trata en el Capítulo III, en donde se explican las subrutinas realizadas para el hardware diseñado, con los diagramas de flujo respectivos.

El Capítulo IV presenta los resultados de las pruebas realizadas con el equipo y las conclusiones y recomendaciones.

En los anexos se incluyen, el manual de uso, la información de los diversos circuitos integrados utilizados y los listados de los programas

CAPITULO I

ASPECTOS GENERALES

CAPITULO I

ASPECTOS GENERALES

1.1. CARACTERISTICAS DE LA NORMA EIA-RS232C

1.1.1 INTRODUCCION

La norma EIA-RS232C fue definida por la EIA (Electrical Industry Association), para la interconexión del equipo terminal de datos (DTE) con el equipo de comunicación de datos (DCE). Se la usa ampliamente en la comunicación serial entre computadores con periféricos como: impresoras, trazadores gráficos, modems, etc. Su norma CCITT correspondiente es la V.24. A través de la interfaz se realiza la transferencia de datos binarios, señales de control y sincronismo, a distancias de máximo 15 metros y velocidades de transferencia de máximo 20 Kbits/s.

La norma EIA-RS232C tiene aplicación en los sistemas electrónicos en los que los equipos tienen un retorno común simple (señal de tierra), que puede ser interconectado en un solo punto de la interfaz. No es aplicable cuando se requiere aislación eléctrica entre los

equipos conectados a los extremos de la interfaz ¹.

En una comunicación serial existen dos tipos de dispositivos ilustrados en la Figura 1.1:

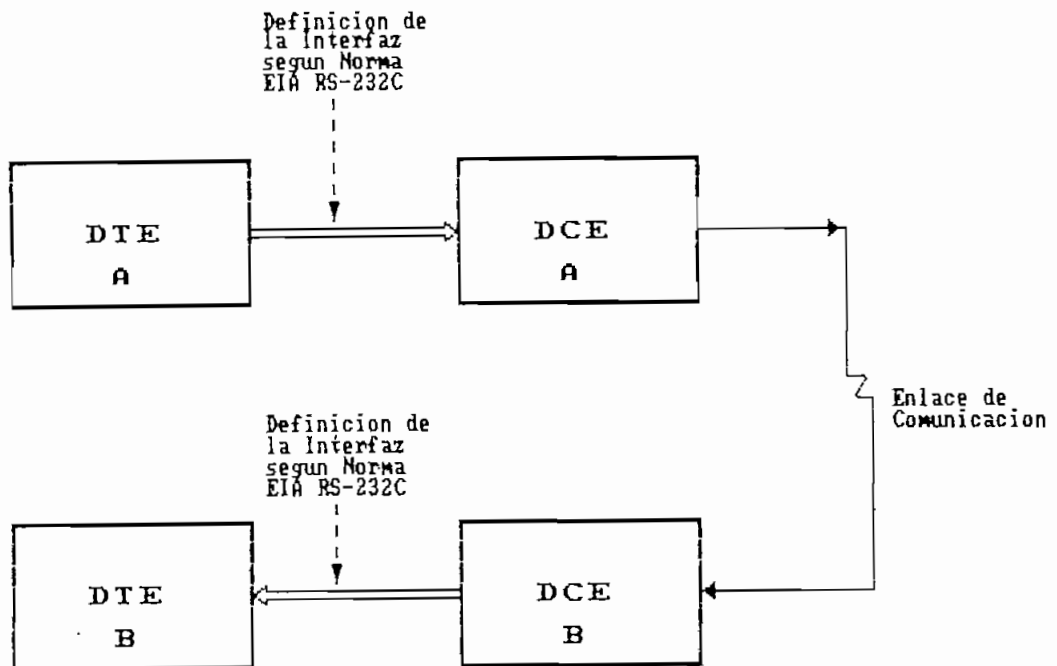


FIGURA 1.1 Sistema Básico de Transmisión Serial

- a) El DTE (Data terminal equipment) que cumple dos funciones básicas: ser fuente y destino final de los datos y controlar la comunicación.
- b) Y el DCE (Data communication equipment) que es un

¹ RS-232 MADE EASY, Seyer Martin, Anexo A, p. 79

equipo intermedio en la comunicación. Comprende convertidores de señales, generadores de sincronismo, regeneradores de impulsos y dispositivos de control, además de equipos con otras funciones como protección contra errores, llamadas y respuestas automáticas.

La norma EIA-RS232C define cuatro aspectos fundamentales de la interfaz que son:

- 1) Las características eléctricas de las señales.
- 2) Las características mecánicas de la interfaz.
- 3) Las características funcionales de las señales.
- 4) La configuración de los sistemas de comunicación.

1.1.2 COMUNICACION SINCRONICA Y ASINCRONICA

Hay dos tipos básicos de transmisión serial: la transmisión sincrónica y la transmisión asincrónica. Ambos tipos de transmisión usan el concepto de tiempo de bit. Puesto que en una transmisión serial los bits son transmitidos sobre la línea de enlace uno a la vez, es necesario identificar donde finaliza un bit y comienza el siguiente. Para esto se define el período de tiempo en que un bit está presente en la línea. Este período se llama un tiempo de bit.

La transmisión asincrónica se llama así porque cada caracter se transmite sin tiempo fijo entre éste y el precedente y los caracteres sucesivos. Puesto que no hay información de sincronismo entre caracteres, cada caracter debe tener bits de sincronismo adicionales.

Un bit de inicio precede cada caracter transmitido en formato asincrónico. Esto indica al receptor que un caracter está llegando. El caracter es terminado por uno o más bits de parada que permiten al receptor un período de descanso antes de que el próximo caracter se transmita. No se envía ni reloj ni señal de sincronización con los datos. El transmisor y el receptor tienen relojes internos y el bit de inicio se usa para sincronizarlos.

Siguiendo al bit de inicio están los bits de datos en el caracter, el bit menos significativo primero. Cada bit se mantiene en la línea de datos por una longitud de tiempo precisamente controlada. Este tiempo, es el tiempo de bit.

El receptor se prepara a recibir el caracter entrante al detectar el bit de inicio. Y realiza un muestreo de cada bit tan cerca del centro del tiempo de bit como sea posible. Naturalmente, el transmisor y el receptor deben concordar en la longitud de tiempo que un bit se mantendrá

en la línea de datos o la transmisión será errónea porque los muestreos se hacen en tiempos incorrectos.

Este tipo de comunicación se basa en la existencia en el receptor de un reloj teóricamente igual al existente en el transmisor. El sincronismo de bit se consigue arrancando el reloj de recepción en el instante en que comienza o se detecta el bit de inicio, por tanto el sincronismo de carácter es simple ya que el receptor sabe que el primer bit de información es siempre el que sigue al bit de inicio.

La transmisión sincrónica en cambio se orienta a bloques. Mientras la información de sincronismo se incluye con cada carácter en la transmisión asincrónica añadiendo los bits de inicio y de parada, mensajes de caracteres múltiples se sincronizan en la transmisión sincrónica añadiendo uno o más caracteres de sincronismo al comienzo del mensaje. Estos caracteres de sincronismo sirven para sincronizar el receptor con el transmisor. Ellos indican el inicio del mensaje.

A diferencia de la transmisión asincrónica, los transmisores y receptores sincrónicos comparten un reloj común. Por tanto no es necesario el bit de inicio porque el transmisor y el receptor están siempre en

sincronización de bit. Únicamente debe indicarse el inicio de un bloque y el final del mismo ya que una vez recibido el primer carácter el receptor puede predecir exactamente cuándo llegará el siguiente.

Utilizando la norma EIA-RS232C se puede realizar los dos tipos de comunicaciones, pues existen pines para las señales de sincronismo en el conector que se encargan de controlar la relación en la cual los datos se transmiten y reciben en las líneas.

1.1.3 CARACTERÍSTICAS ELÉCTRICAS

Como norma general las señales de datos se consideran como marca (1L) cuando en ellas hay una tensión negativa, y como espacio (0L) cuando hay una tensión positiva. En cuanto a las señales de control y sincronismo se consideran en estado abierto (ON) cuando están a tensión positiva y en estado cerrado (OFF) cuando están a tensión negativa.

Las variaciones permitidas para uno y otro nivel son:

VOLTAJE POSITIVO: Corresponde a cero lógico (0L), con una variación permitida de +5V a +15V para la salida, y de +3V a +15V para

la entrada.

VOLTAJE NEGATIVO: Corresponde a uno lógico (1L), con una variación permitida de -5V a -15V para la salida y de -3V a -15V para la entrada.

Las capacidades parásitas presentes pueden afectar la calidad de la señal en la transición entre niveles, por lo que la norma recomienda 15 metros como la máxima distancia para cables normales y velocidades usuales. Para distancias grandes es necesario un MODEM en cada extremo de la línea.

Existen otras características eléctricas que también debe cumplir la interfaz y son las siguientes:

- Las entradas deben soportar sobrevoltajes de hasta 25V sin daños.
- Las salidas deben tolerar un cortocircuito sin daños.
- La impedancia de salida debe ser mayor o igual a 300Ω con o sin fuente de alimentación.
- La impedancia de entrada debe estar entre $3K\Omega$ y $7K\Omega$.

1.1.4 CARACTERISTICAS MECANICAS

La interfaz EIA-RS232C utiliza un conector de 25 pines generalmente un conector DB-25 indicado en la Figura 1.2. Los 25 pines de la interfaz tienen una función preasignada; sin embargo, sólo un número limitado de estos son usados en la mayoría de aplicaciones por esta razón con frecuencia no es necesario un conector de 25 pines y en su lugar se utiliza un conector de 9 pines DB9 que permite tener las líneas más utilizadas de la interfaz, este conector se indica en la Figura 1.3.

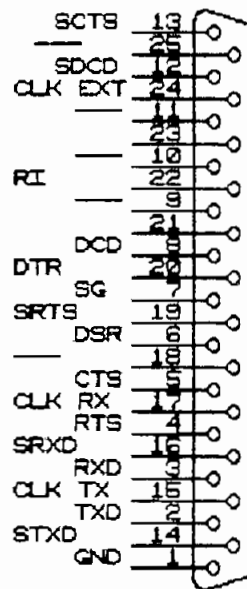


FIGURA 1.2 Conector DB25

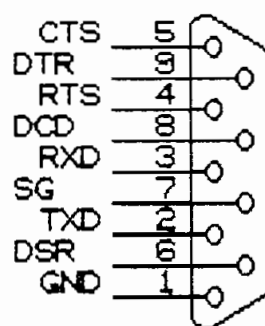


FIGURA 1.3 Conector DB9 1,2

La norma EIA-RS232C especifica que el equipo DCE debe utilizar un conector hembra y el DTE un conector macho.

Las señales son de cuatro clases: datos, control, sincronismo y las tierras. En la Tabla 1.1 se indican el pin de la señal en el conector DB25 y sus denominaciones: común, de acuerdo al EIA y de acuerdo al CCITT. En la Tabla 1.2 en cambio se indica el pin de la señal para cuando se utiliza un conector DB9 y el nombre común de la línea de señal.

1.1.5 DESCRIPCION DE LAS SEÑALES

a) SEÑALES DE TIERRA

Son los pines 1 y 7 de la interfaz. El pin 1 (GND) se

PIN	NOMBRE EIA	NOMBRE CCITT	NOMBRE COMUN	FUENTE	DESCRIPCION
1	AA	101	GND	---	Tierra de protección
2	BA	103	TXD	DTE	Transmisión de datos
3	BB	104	RXD	DCE	Recepción de datos
4	CA	105	RTS	DTE	Petición para Tx
5	CB	106	CTS	DCE	Preparado para Tx
6	CC	107	DSR	DCE	Equipo datos listo
7	AB	102	SG	---	Tierra de las señales
8	CF	109	DCD	DCE	Detección de portadora
9	---	---	---	---	Reservado para pruebas
10	---	---	---	---	Reservado para pruebas
11	---	---	---	---	Sin asignación
12	SCF	122	SDCD	DCE	Detección de portadora del canal secundario
13	SCB	121	---	DCE	Canal secundario listo para transmitir
14	SBA	118	STXD	DTE	Transmisión de datos por canal secundario
15	DB	114	---	DCE	Sincronismo de Tx con fuente en el DCE
16	SBB	119	SRXD	DCE	Recepción de datos por canal secundario
17	DD	115	---	DCE	Sincronismo de Rx con fuente en el DCE
18	---	---	---	---	Sin asignación
19	SCA	120	SRTS	DTE	Petición para Tx por el canal secundario
20	CD	108/2	DTR	DTE	Terminal de datos listo
21	CG	110	---	DTE	Detector de calidad de las señales de datos
22	CE	125	RI	DCE	Indicador de llamada
23	CH	111	---	Ambos	Selector de velocidad binaria origen DTE/DCE
24	CI	112	---		
24	DA	113	---	DTE	Sincronismo de Tx con fuente en el DTE
25	---	---	---	---	Sin asignación

TABLA 1.1 Señales de Intercambio de la Norma
EIA RS232C en un Conector DB25

PIN	NOMBRE EIA	NOMBRE CCITT	NOMBRE COMUN	FUENTE	DESCRIPCION
1	AA	101	GND	---	Tierra de protección
2	BA	103	TXD	DTE	Transmisión de datos
3	BB	104	RXD	DCE	Recepción de datos
4	CA	105	RTS	DTE	Petición para Tx
5	CB	106	CTS	DCE	Preparado para Tx
6	CC	107	DSR	DCE	Equipo datos listo
7	AB	102	SG	---	Tierra de las señales
8	CF	109	DCD	DCE	Detección de portadora
9	--	---	DTR	DTE	Terminal de datos listo

**TABLA 1.3 Señales de Intercambio de la Norma
EIA-RS232C en un Conector DB9**

llama tierra de protección, debe ser conectada a la carcasa correspondiente del equipo y sirve como protección contra descargas eléctricas y cortocircuitos. El pin 7 (SG) en cambio establece el retorno común de la señal para todas las señales excepto la tierra de protección, es decir provee el potencial de referencia. Esta línea debe terminar en un sólo punto en la interfaz; siempre debe estar incluido en el cable.

b) SEÑALES DE DATOS

TRANSMISION DE DATOS TXD : Es el pin 2 del conector. Usado para la transmisión de datos desde el DTE hacia el DCE. Debe mantenerse en condición de marca durante los

intervalos entre caracteres o palabras, y en todo momento en que no se transmitan datos. El DTE no transmitirá datos a menos que las señales DTR, DSR, RTS y CTS estén en estado abierto (condición ON).

RECEPCION DE DATOS RXD : Corresponde al pin 3 del conector y se usa para la recepción de datos por parte del DTE desde el DCE. Debe mantenerse en condición de marca mientras la señal Data Carrier Detect esté en estado cerrado (condición OFF). En un sistema half duplex debe mantenerse en la condición de marca cuando la señal Request to Send esté en estado abierto (ON) y por un breve intervalo después de la transición del estado abierto al cerrado de la señal RTS para permitir que se complete la transmisión.

c) SEÑALES DE CONTROL

TERMINAL DE DATOS PREPARADO DTR : Este pin (20) se usa para indicar en estado abierto que el DTE se encuentra funcionando correctamente. Prepara al DCE para conectarse al canal de comunicación manteniendo la conexión.

CONJUNTO DE DATOS LISTO DSR : Esta señal (pin 6) cuando se encuentra en estado abierto indica que el DCE está listo o preparado para funcionar.

PETICION PARA TRANSMITIR RTS : (pin 4) Enviado por el DTE hacia el DCE para indicar en estado abierto que quiere realizar una transmisión. Cuando se realiza el cambio de cerrado a abierto, el DCE responde cambiando la señal CTS a estado abierto. Los datos a transmitir pueden ser enviados únicamente después de que el DTE detecte este cambio en la señal CTS.

PREPARADO PARA TRANSMITIR CTS : (pin 5) Señal enviada desde el DCE para indicar al DTE en estado cerrado que está preparado para transmitir datos. Se activa en respuesta a una señal RTS.

DETECTADA PORTADORA DE DATOS DCD : Es el pin 8 de la interfaz y la envía el DCE hacia el DTE. Detecta la señal de portadora desde el equipo DCE distante. El estado de esta señal indica si las señales de línea recibidas por el canal de datos están o no dentro de los límites especificados para el DCE.

INDICADOR DE LLAMADA RI : (pin 22) En estado abierto indica la recepción de una llamada en el canal de comunicación y la necesidad de que se establezca una conexión. La señal se desactiva en el intervalo entre llamadas.

d) SEÑALES DE SINCRONISMO

SINCRONIZACION PARA BITS DE DATOS EN LA TRANSMISION
(Proveniente del DCE): (pin 15) Va desde el DCE hacia el DTE. Esta señal se usa para proveer al DTE información de sincronismo de bit en transmisión. El DTE deberá cambiar el estado de la línea TXD cuando se produzca una transición de estado cerrado a abierto en esta línea. Tal como se muestra en la Figura 1.4a.

SINCRONIZACION PARA BITS DE DATOS EN LA TRANSMISION:
(Proveniente del DTE): (pin 24) Va desde el DTE hacia el DCE. El cambio del estado abierto a cerrado indica al DCE el centro de cada bit a transmitir. Se utiliza cuando la sincronización proviene del DTE. La señal se indica en la Figura 1.4b.

SINCRONIZACION PARA BITS DE DATOS EN LA RECEPCION
(Proveniente del DCE): (Pin 17) Esta señal va desde el DCE hacia el DTE. Se usa para proveer al DTE la información de sincronismo de bit en la señal recibida. La transición de estado abierto a cerrado en esta línea indica al DTE el centro de cada bit en la línea RXD. Esta señal se indica en la Figura 1.5.

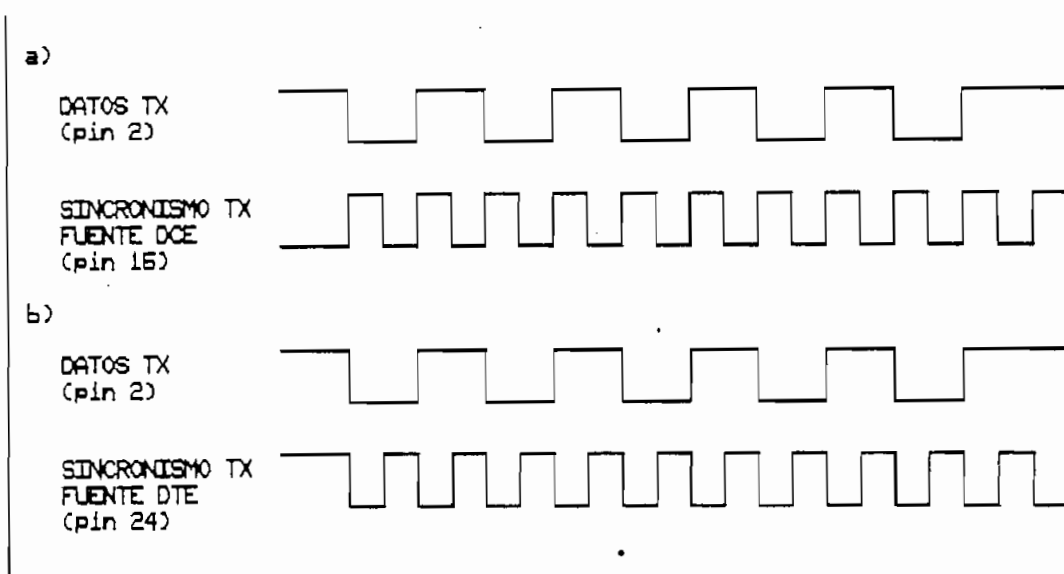


FIGURA 1.4 Sincronismo de Bit en Transmisión

a) Con fuente en DCE

b) Con fuente en DTE

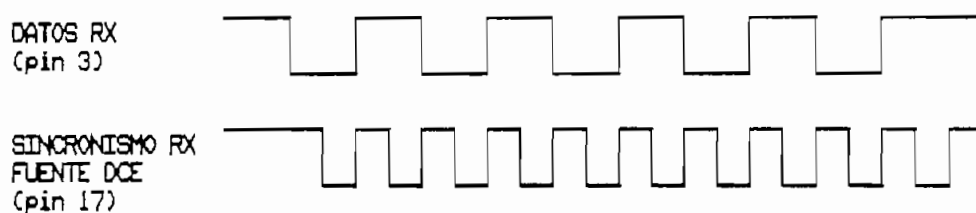


FIGURA 1.5 Sincronismo de Bit para Recepción

e) SEÑALES SECUNDARIAS

Las señales secundarias funcionan de la misma manera que las señales primarias correspondientes y controlan los canales secundarios de la facilidad de comunicación. Hay señales secundarias RTS, CTS, DCD además de los canales de

transmisión y recepción secundarios.

Los Equipos de Comunicación de Datos inteligentes capaces de transmitir información de diagnóstico, usan esos canales secundarios de datos para pruebas y reporte de errores. Sin embargo, generalmente los canales de datos secundarios rara vez son utilizados.

1.1.6. CONTROL DE LA COMUNICACION

Para que se establezca adecuadamente una comunicación, es necesario que el transmisor de datos conozca cuándo el receptor está listo para recibir información. Las indicaciones del estado de operación del transmisor y del receptor se conocen como handshake e involucran señales que viajan desde el transmisor hasta el receptor y viceversa.

Hay dos métodos por los cuales, el receptor puede controlar el flujo de datos: el handshake por hardware y el handshake por software.

En el primer método, el receptor envía un voltaje positivo a través de las líneas de interfaz destinadas para este propósito cuando está listo para recibir, en cambio cuando no está listo para recibir envía un voltaje

negativo, para que éste deje de enviar datos. En el EIA-RS232C las líneas DSR, CTS, DTR y RTS están dedicadas al handshake por hardware.

En el segundo método las señales consisten de caracteres especiales transmitidos a través de las líneas de datos. Se usa generalmente cuando dos computadores se comunican directamente o vía modem y cuando la comunicación en los dos sentidos es posible. Los protocolos más comunes para el handshake por software son:

XON/XOFF: El dispositivo receptor envía el carácter ASCII (13H), al equipo que transmite cuando quiere parar la transmisión y envía el carácter ASCII (11H) cuando quiere reiniciar la transmisión.

ETX/ACK: Es el método conocido como Fin 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 (ETX End of Transmission). El receptor acusa recibo del bloque transmitido sin errores enviando el carácter ACK (ASCII 6H), si se detectan errores en el bloque transmitido, el receptor envía el carácter NACK (21H) para indicar al transmisor que debe enviar nuevamente el bloque anterior.

1.1.7 USOS NO NORMALIZADOS DE LAS SEÑALES DE CONTROL

La mayoría de enlaces de comunicación no utilizan todos los pines de la interfaz. Los más importantes son los pines 2 y 3 para transmitir y recibir datos respectivamente, el resto de las señales con excepción de las tierras son señales de control para establecer y mantener un enlace de comunicación. Desafortunadamente no siempre se usan las señales de control de la manera normalizada lo que causa problemas cuando se quiere interconectar equipos.

De acuerdo a la definición estricta del EIA-RS232C, el DTE activa el pin 5 (RTS) cuando tiene un dato para transmitir y espera que el DCE active el pin 4 (CTS) antes de transmitir. Por tanto el pin 4 no puede ser usado apropiadamente por el DCE como un indicador de flujo, ya que al DCE no se le permite bajar CTS hasta que el DTE baje RTS, puesto que CTS y RTS son señales que permiten al DTE tomar el control del enlace de comunicación desde el DCE. El DTE asume que éste mantendrá el enlace tanto como sea necesario y por lo tanto el DCE no puede bajar CTS arbitrariamente.

Algunos fabricantes ignoran la definición estricta de los pines 4 y 5 y los usan para handshake de datos, al

igual que los pines 6 (DSR) y 20 (DTR). El uso de cualquier pin del EIA-RS232C para handshake no garantiza el reconocimiento del handshake por el dispositivo en el otro extremo del cable. Ya que la norma no considera la idea de que la interfaz se use como un puerto de I/O serial general.

Porque esta posibilidad no se considera en la norma, el resultado de implementar un mecanismo de handshake en una interfaz no puede determinarse sin estudiar los manuales de interfaz de los dos dispositivos.

1.1.8 FORMAS DE CONEXION DE LA INTERFAZ

Los pines más usados y que tienen mayor importancia en la interfaz son: el pin 2 (TXD), el pin 3 (RXD) y el pin 7 (GND), pudiendo prescindir del resto de los pines. Estos tres pines son suficientes para una comunicación bidireccional entre el DCE y el DTE como se indica en la Figura 1.6.

Sin embargo, esta forma de conexión presenta la desventaja de que ninguno de los equipos sabe si el otro está listo para comunicarse, por lo que adicionalmente se usan las señales de control DTR, DSR, RTS y CTS con lo que las conexiones quedarían como se indica en la Figura 1.7.

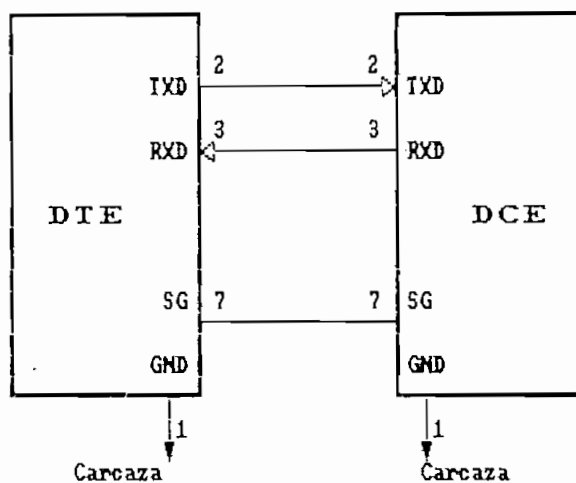


FIGURA 1.6 Mínimas Conexiones para Operación Full Duplex

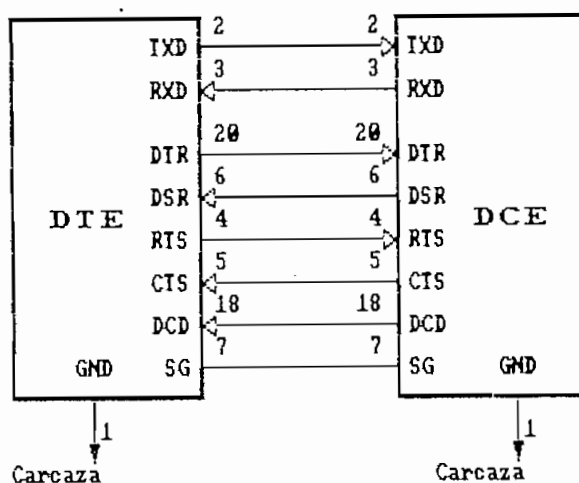


FIGURA 1.7 Conexión Full Duplex con las Principales Líneas de Control

Para una comunicación sincrónica hace falta además de las anteriores las señales de sincronismo. La

configuración de la Figura 1.8 se sincroniza con la señal de sincronismo de bit en transmisión con fuente DCE del pin 15 de la interfaz y la señal de sincronismo de bit en recepción del pin 17.

Otra posible fuente de sincronismo de bit en transmisión tiene como fuente el DTE; en este caso se suministrará la señal de sincronismo por el pin 24 y la señal de sincronismo se recepción por el pin 17. Esta configuración se indica en la Figura 1.9.

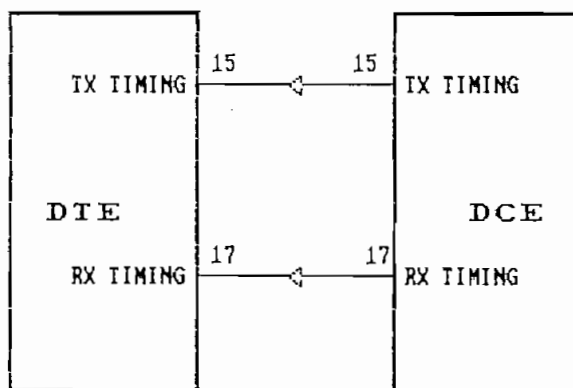


FIGURA 1.8 Conexión de las señales de sincronismo

Cuando el sincronismo derivado en el pin 17 se conecta al pin 15, la sincronización se mantiene fácilmente, porque viene de una fuente simple. Por tanto se minimiza el número de fuentes de sincronismo. La Figura

1.10 muestra esta configuración.

Si se quieren conectar dos DTE directamente no es necesaria la presencia de equipos DCE por lo que se deben realizar conexiones cruzadas o null modem, ya que en este caso los dos equipos generan unas señales y esperan otras para funcionar adecuadamente; por lo tanto hay ciertas señales que deben generarse de las señales existentes en los DTE teniendo en cuenta que siempre una señal de salida debe estar conectada a una entrada.

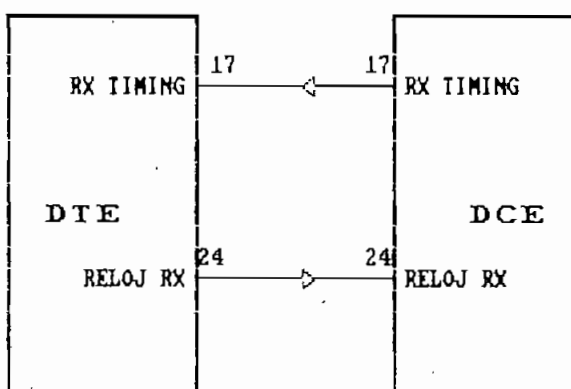


FIGURA 1.9 Conexión de las señales de sincronismo

Así en la configuración de la Figura 1.11 el pin 2 (TXD) del DTE1 debe conectarse con el pin 3 (RXD) del DTE2 y de la misma manera en el sentido contrario. En cuanto a las señales de control los DTE tienen como salidas las

señales DTR y RTS y necesitan como entradas las señales DSR, CTS y DCD por tanto en la conexión de la Figura 1.11 se conecta la señal DTR de uno de los DTE a la señal DSR del otro DTE en ambos extremos, la señal RTS de uno de los DTE a la señal CTS del otro DTE en ambos extremos el mismo y a la señal DCD del otro extremo. Con esto se tienen las conexiones necesarias para una comunicación full duplex.

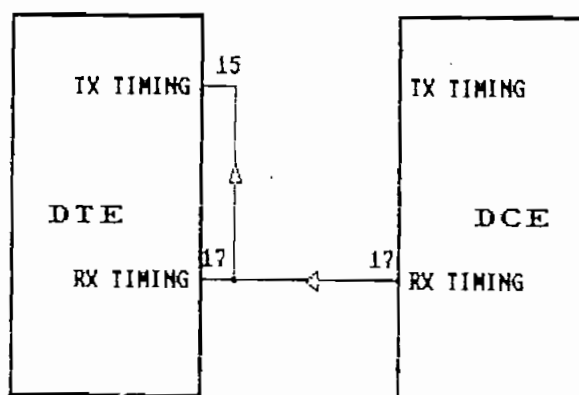


FIGURA 1.10 Conexión de las señales de sincronismo

El cable de conexión de la configuración Null Modem de la Figura 1.12 tiene solamente tres cables TXD, RXD y SG, al contrario de la configuración anterior en la que las señales de control se conectaban de un extremo a otro, en este caso las señales de control se conectan las salidas con las entradas en el mismo extremo sin pasar por

el cable. Así la señal DTR se conecta a la señal DSR del mismo DTE y las señales RTS, CTS y DCD se interconectan juntas en el mismo extremo, se conectan los pines de datos y tierra igual que en la Figura 1.11. Esta configuración también permite la comunicación full duplex entre los dos DTE.

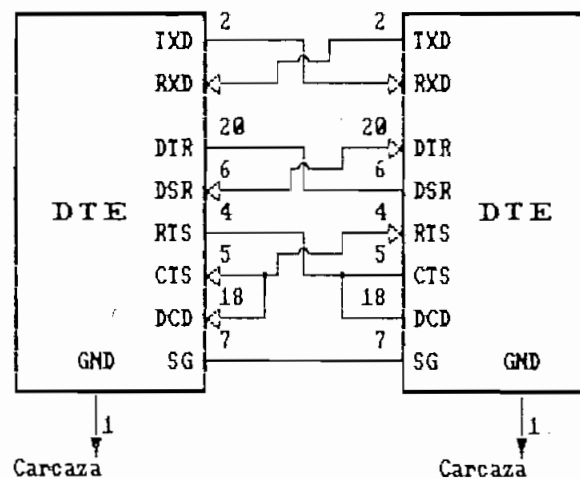


FIGURA 1.11 Conexiones Cruzadas

Finalmente si la comunicación entre los DTE es sincrónica existe una posibilidad adicional de conexión de las señales de sincronismo tal como se muestra en la Figura 1.13, en donde la señal de sincronismo de bit en transmisión con fuente DTE proveniente del DTE1 es utilizada como fuente única de sincronismo.

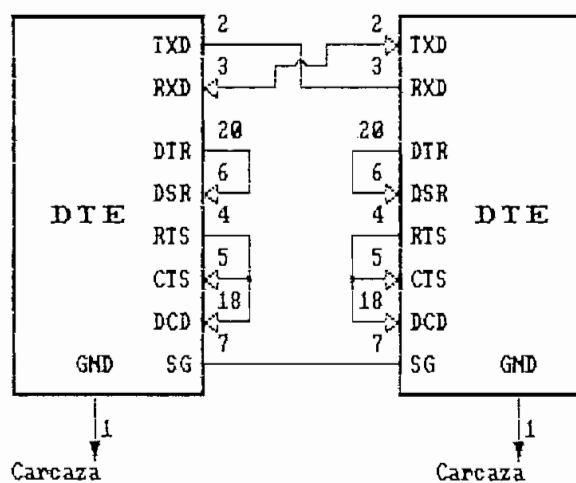


FIGURA 1.12 Conexión Null Modem

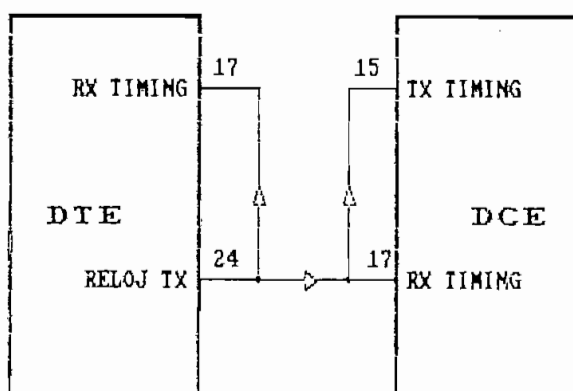


FIGURA 1.13 Conexión Null Modem Sincrónica

1.1.9 PASOS PARA LA COMUNICACION USANDO UNA INTERFAZ EIA-RS232C

La secuencia de eventos que ocurre en la interfaz EIA-RS232C para que el DTE A se comunique con el DTE B de la Fig. 1.1 son los siguientes² :

- 1.- Los pines de las señales y la tierra deben estar conectados.
- 2.- Los DTE activan sus respectivos pines 20 (DTR) para indicar que están encendidos, y los DCE activan sus pines 6 (DSR) en respuesta.
- 3.- El DTE A solicita una transmisión activando el pin 4 (RTS). EL DCE A envía la señal portadora al DCE B y activa el pin 5 (CTS) para el DTE A. El DCE B detecta la señal portadora y activa el pin 8 (DCD) para el DTE B.
- 4.- El DTE A comienza la transmisión de los datos sobre el pin 2 (TXD) al DCE A. El DTE A genera la señal de sincronismo de transmisión para el DCE A. El dato es modulado y transmitido por el DCE A hacia el DCE B, que lo demodula y transmite los datos al DTE B usando el pin 3, el DCE B envía la señal de sincronismo apropiadas al DTE B en el pin 17.

² JOURNAL OF DATA & COMPUTER COMMUNICATIONS, pg. 28

5.- El procedimiento de fin de transmisión (EOT) causa que el DTE A desactive el pin 4 (RTS), indicando al DCE A que debe desactivar su señal portadora y la señal CTS para el DTE A. EL DCE B detecta la ausencia de portadora y desactiva a su vez la señal DCD del pin 8 al DTE B.

1.2.ESPECIFICACIONES DE LA NORMA IEEE-488

1.2.1 INTRODUCCION

La norma IEEE-488 se utiliza en la interconexión digital de instrumentos electrónicos programables, debido a que facilita el diseño, ensamblado y uso de los sistemas de instrumentación y elimina la incompatibilidad entre equipos de diferentes fabricantes.

Por su versatilidad y alta velocidad de transferencia de datos, puede ser usada también para otras aplicaciones, como por ejemplo en la comunicación entre computadores, control de periféricos, etc.

Fue definida por la Hewlett Packard y aprobada por la IEEE en 1975 (con una revisión en 1978). Se le conoce bajo diferentes nombres: Bus HP, IEEE-488, pero el nombre más usado es el de IEEE-488 (General Purpose Interface Bus).

Las especificaciones generales de la norma incluyen reglas para definir circuitos, cables, conectores, las señales de control y particularmente la descripción del repertorio de mensajes, que aseguran una transferencia de información sin ambigüedades.

El IEEE-488 cumple los siguientes objetivos básicos³:

- 1.- Especifica un sistema de fácil uso, pero que tiene toda la terminología y definiciones relacionadas exactamente con el sistema, esto significa que todos los dispositivos usan el mismo lenguaje cuando se comunican por el IEEE-488.
- 2.- Define todos los requerimientos mecánicos, eléctricos y funcionales de la interfaz en un sistema, sin definir ningún aspecto del dispositivo.
- 3.- Permite un amplio rango de capacidades de instrumentos y periféricos del computador para usarlos en un sistema con la mayor eficiencia.
- 4.- Permite que equipos manufacturados por diferentes fabricantes puedan ser conectados y trabajar juntos en el mismo bus.
- 5.- Define un sistema adecuado para distancias de

³ MICROSYSTEM COMPONENTS HANDBOOK VOLUME II, INTEL, pg. 7-321

interconexión limitadas.

- 6.- Define un sistema con mínimas restricciones en el funcionamiento de los dispositivos.
- 7.- Define un bus que permite comunicación asincrónica con un amplio rango de velocidades de transmisión de datos.
- 8.- Define un sistema de bajo costo que no requiere una interfaz lógica extensa, ni elaborada, para instrumentos de bajo costo y provee grandes capacidades para los instrumentos de alto costo si se desea.
- 9.- Permite la existencia de sistemas sin necesidad de un controlador.

1.2.2 CARACTERISTICAS DEL BUS

Para que un sistema esté dentro del alcance de esta norma debe cumplir con las siguientes restricciones:

- 1.- El número de dispositivos interconectados por un sólo bus continuo no debe exceder de 15.
- 2.- La longitud total del cable de enlace máximo puede ser 20 metros.
- 3.- Dos dispositivos cualesquiera no deben estar a distancias mayores de 4 metros entre sí.
- 5.- La máxima velocidad de transferencia de información

es de 1Mbyte/s.

5.- Intercambio solamente de datos digitales.

Las especificaciones de la interfaz no pueden ser usadas en aplicaciones que requieran grandes distancias, un número mayor de dispositivos, o incremento en la inmunidad de ruido.

1.2.3 ESTRUCTURA DEL BUS

El IEEE-488 tiene una estructura del bus de líneas compartidas, indicada en la Figura 1.14, con un máximo de 15 elementos de cuatro tipos conectados al bus. Todos los dispositivos se conectan en paralelo.

Los cuatro tipos de dispositivos que pueden conectarse al bus son los siguientes:

- a) Un controlador : que es la unidad central de control.
- b) Los dispositivos oyentes ("listener"), que intervienen únicamente en la recepción de información, es decir sólo escuchan.
- c) Los dispositivos Locutores ("talker"), que intervienen únicamente en la emisión de información, es decir sólo hablan. Pero pueden ser receptores de las órdenes.

- d) Los dispositivos a la vez oyente y locutores, que hablan o escuchan dependiendo de cómo estén programados.

El controlador maneja el flujo de información en el IEEE-488 enviando comandos a todos los elementos. Usualmente direcciona a un locutor y a un oyente antes de que el primero pueda enviar mensajes al segundo, y después de la transmisión del mensaje los desactiva.

Algunas configuraciones del bus no requieren un controlador, por ejemplo en el caso en que un dispositivo sea siempre el locutor y los otros dispositivos siempre escuchen. Sin embargo, si la cadena es más compleja o es necesario cambiar al locutor y a los oyentes, el controlador dirigirá los intercambios activando los dispositivos y controlando el proceso.

Aunque puede haber múltiples controladores, uno sólo debe estar activo en un instante dado, o ser el controlador activo CIC (Controller In Charge).

El control puede transferirse desde el actual CIC a un controlador pasivo (en standby). Solamente el controlador del sistema puede autoactivarse.

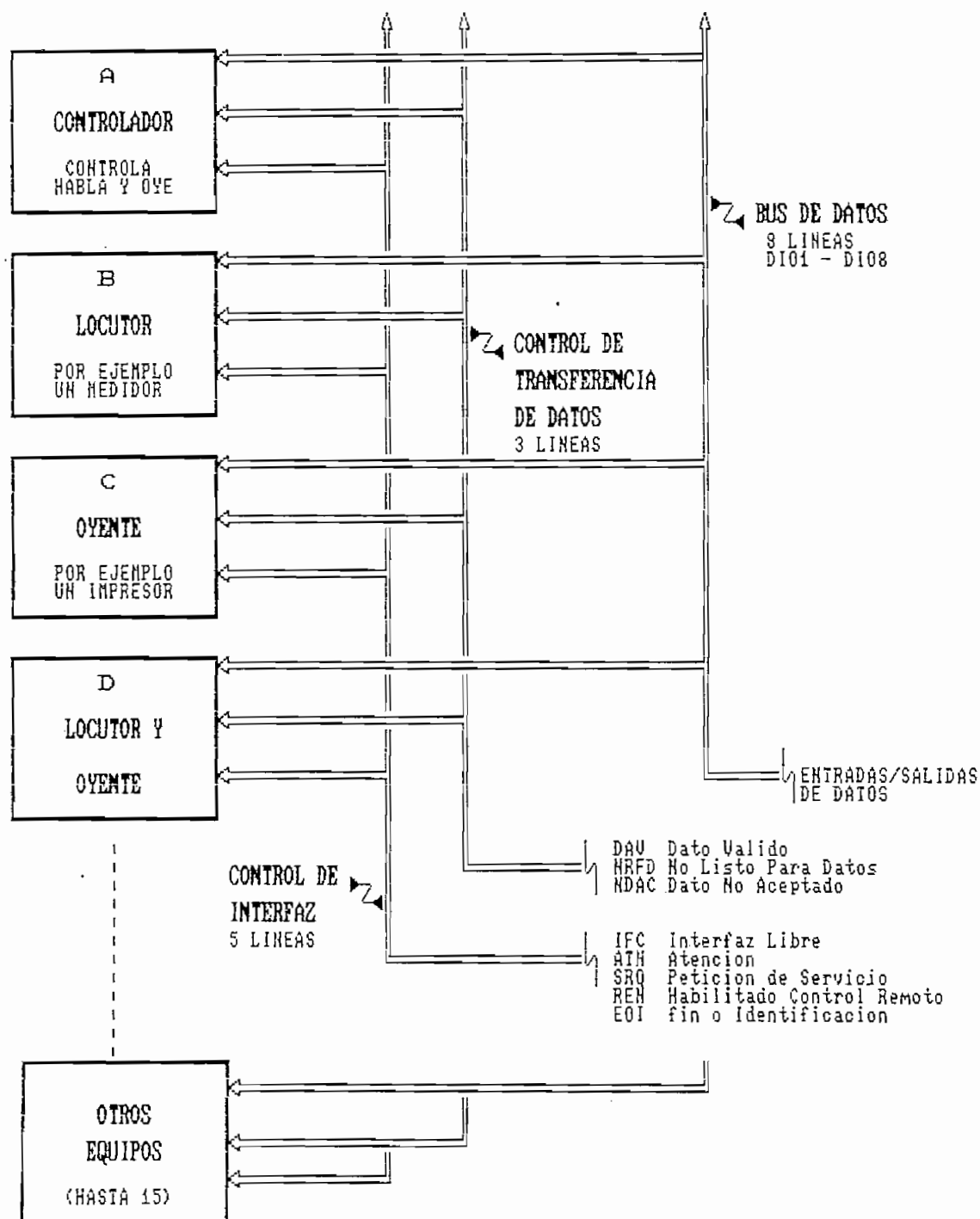


FIGURA 1.14 Estructura del IEEE-488

El locutor en cambio envía mensajes de datos a uno o más oyentes. Muchos oyentes pueden funcionar simultáneamente pero en un instante cualquiera sólo un locutor debe estar activo. Una cadena mínima la constituye un locutor y un oyente.

1.2.4 LINEAS Y SEÑALES DEL BUS

El IEEE-488 conecta el controlador a los dispositivos de la forma indicada en la Figura 1.15.

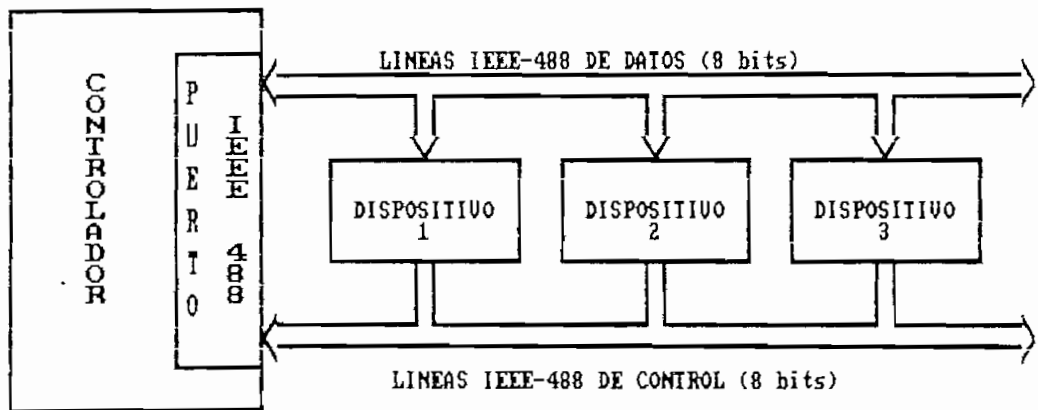


FIGURA 1.15 Características del IEEE-488

El bus consiste de dos grupos de líneas, un grupo con 8 líneas de datos y otro grupo con 8 líneas de control.

LINEAS DE DATOS _ Las ocho líneas de datos DIO1 a DIO8,

permiten transmitir por el bus mensajes de byte en byte. Estos mensajes pueden ser datos del dispositivo, direcciones, comandos normalizados y palabras de estado. Todos los comandos y la mayoría de datos usan el código ASCII de 7 bits o código ISO, en ambos casos el octavo bit DIO8 no es usado o se lo usa para la paridad.

LINEAS DE CONTROL.- Las 8 líneas de control forman dos subgrupos de líneas llamadas: a) líneas de control de la transferencia de datos (handshake) y b) líneas de control de la interfaz (General interface management).

a) LINEAS DE CONTROL DE LA TRANSFERENCIA DE DATOS (HANDSHAKE)

Estas tres líneas controlan asincrónicamente la transferencia de bytes de mensaje entre dispositivos y permiten que el byte de mensaje en las líneas de datos se envíe y reciba sin error en la transmisión.

NRFD : Not Ready For Data (No listo para datos).- Indica si el dispositivo está listo o no para recibir un byte de mensaje. La línea es manejada por todos los dispositivos cuando reciben comandos y por los oyentes direccionados cuando reciben datos, cada dispositivo a su tiempo libera esta línea. Sin embargo, la línea no retorna al estado

inactivo hasta que el elemento que responde más lentamente lo libere.

NDAC : Not Data Accepted (Dato no aceptado).- Indica si los dispositivos direccionados han aceptado o no un byte de mensaje. La línea es manejada por todos los dispositivos cuando reciben comandos y por los oyentes direccionados cuando reciben datos.. Se mantiene en el estado activo hasta que todos los dispositivos direccionados acepten el byte.

DAV : Data Valid (Dato válido).- Indica cuándo las señales en las líneas de datos están estables, es decir son válidas y pueden ser aceptadas con seguridad por los dispositivos. El controlador maneja la línea cuando envía comandos y el locutor la maneja cuando envía datos.

b) LINEAS DE CONTROL DE LA INTERFAZ (GENERAL INTERFACE MANAGEMENT)

ATN : Attention (Atención).- La emplea el controlador del bus para indicar que está enviando un mensaje de interés general. El controlador activa la línea ATN cuando usa las líneas de datos para enviar comandos y la desactiva cuando permite que un locutor envíe datos.

IFC : Interface Clear (Limpiar la interfaz).- Esta línea lleva una señal de reset que puede ser activada solamente por el controlador, con esto pone al bus en un estado conocido.

SRQ : Service Request (Petición de servicio).- Es un tipo de línea para interrupción que puede ser activada por cualquiera dispositivo conectado al bus que necesite servicio del controlador. Estas necesidades de servicio pueden darse cuando un dispositivo quiere utilizar el bus para efectuar una transferencia de datos, cuando un dispositivo ha detectado un error interno, etc.

REN : Remote Enable (Habilitado control remoto).- Activado por el controlador, indica a los dispositivos direccionados que ignoren el control local para obedecer al control remoto recibido a través del bus.

EOI : End or Identify (Fin o identificación).- Esta línea tiene dos propósitos: a) la activa el locutor durante la transferencia del último byte de datos en una cadena de datos para señalar el fin de la cadena y b) la activa el controlador simultáneamente con la línea ATN para preguntar el estado de los dispositivos en una búsqueda en paralelo (ver numeral 1.2.13).

Estas 16 líneas con sus señales asociadas de retorno hacen el total de conexiones físicas entre el controlador y cualquiera de los 15 dispositivos externos conectados al bus. Las líneas DAV, NDAC, NREFD, ATN, IFC y SRQ disponen de su tierra lógica individual, existiendo además la tierra general y el blindaje general. El elevado número de líneas de tierra se explica por la necesidad de evitar interacciones eléctricas entre las principales líneas de control.

1.2.5 ESPECIFICACIONES MECANICAS

El cable de enlace constituye el bus propiamente dicho, es un cable pasivo por el cual la información se transmite en los dos sentidos. Se compone de dieciséis líneas de señal, una tierra lógica, seis blindajes parciales y un blindaje general. Con una longitud de 0.5 a 4 metros el cable está provisto en cada extremo de un conector macho/hembra. En cada dispositivo debe haber un conector hembra. Se recomienda conectores de forma trapezoidal de 24 contactos tipo rack y panel (MICRORIBBON o CHAMP). La configuración del conector se indica en la Figura 1.16.

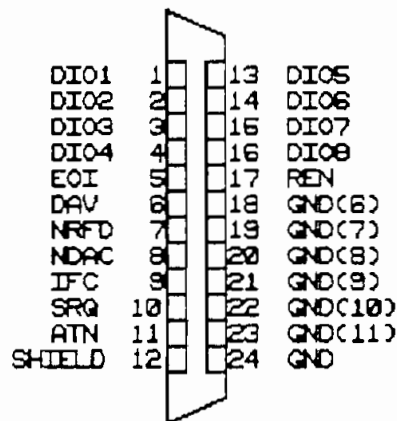


FIGURA 1.16 Configuración del Conector IEEE-488

Las señales se distribuyen en los 24 contactos del conector de la forma indicada en la Tabla 1.3.

Los equipos en el bus de interfaz IEEE-488 pueden conectarse en cualquier tipo de configuración, pero las configuraciones más usadas son: a) Configuración en estrella b) en cadena y c) Cualquier combinación de las dos anteriores, indicadas en la Figura 1.17.

1.2.6 FUNCIONES DE INTERFAZ

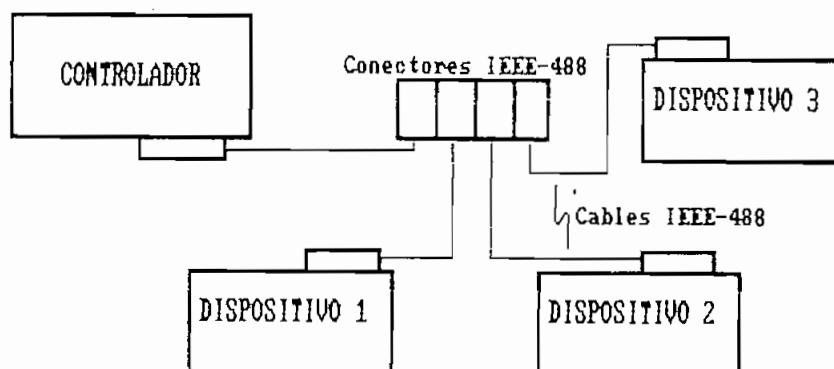
Cada dispositivo está compuesto de dos grupos de funciones: las funciones de dispositivo y las funciones de interfaz. Esto se indica en la Figura 1.18. El primer grupo controla la operación del dispositivo y no está

definido por la norma. El segundo grupo en cambio, realiza el procesamiento de los mensajes de interfaz y está definido por la norma.

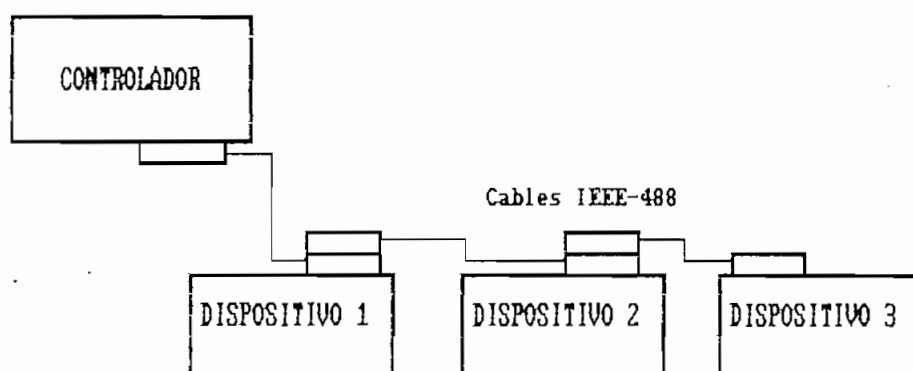
Contacto	Señal en el pin
1	DIO1
2	DIO2
3	DIO3
4	DIO4
5	EOI
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	SHIELD
13	DIO5
14	DIO6
15	DIO7
16	DIO8
17	REN
18	Gnd (6)
19	Gnd (7)
20	Gnd (8)
21	Gnd (9)
22	Gnd (10)
23	Gnd (11)
24	Gnd (datos)

TABLA 1.3 Distribución de Contactos en el Conector

a) Configuración en estrella



b) Configuración en cadena



c) Configuración estrella - cadena

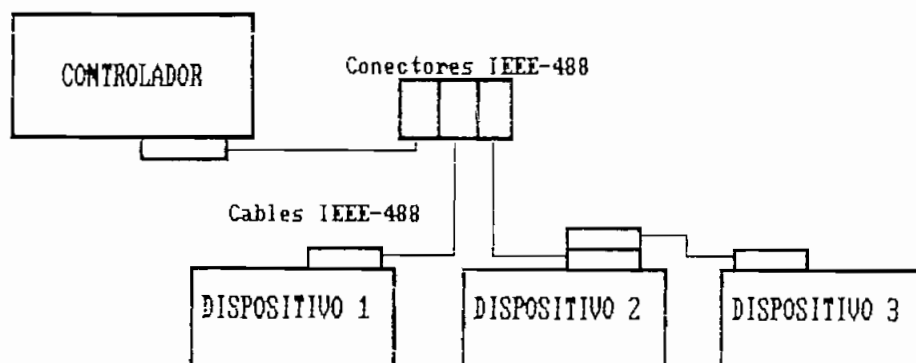


FIGURA 1.17 Configuraciones del Bus IEEE-488

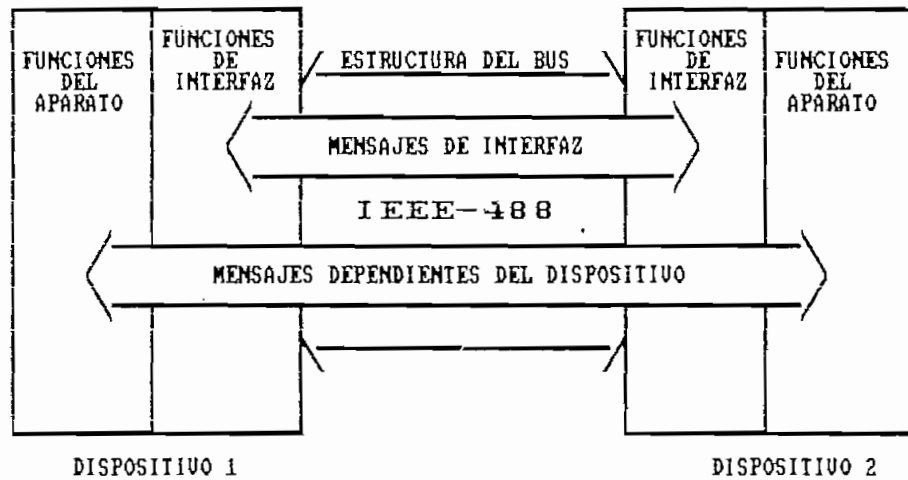


FIGURA 1.18 Estructura de las Funciones de un Dispositivo

La norma IEEE-488 define un total de diez funciones de interfaz que se indican en la Tabla 1.44. Un dispositivo puede tener todas las funciones de interfaz o solamente algunas de ellas.

Para que un dispositivo sea capaz de interpretar los mensajes enviados a través del bus es preciso que se le haya dotado de las funciones necesarias para analizarlos y responder en función de dichos mensajes.

⁴IEEE-488 STANDARD DIGITAL INTERFACE FOR PROGRAMMABLE INSTRUMENTATION, IEEE, pg. 56

FUNCIONES INTERFAZ	SIMBOLO
Handshake Fuente (Source Handshake)	SH
Handshake Destino (Acceptor Handshake)	AH
Locutor (Talker) o Locutor Extendido	T TE
Oyente (Listen) u Oyente Extendido	L LE
Petición de Servicio (Service Request)	SR
Control Remoto/Local (Remote/Local)	RL
Búsqueda en paralelo (Parallel Poll)	PP
Limpiar Dispositivo (Device Clear)	DC
Disparar Dispositivo (Device Trigger)	DT
Controlador (Controller)	C

TABLA 1.4 Funciones de interfaz de la norma IEEE-488

Las funciones que puede realizar un dispositivo conectado al bus se indican en la Figura 1.19.

Las características de las diez funciones de interfaz definidas por la norma son las siguientes:

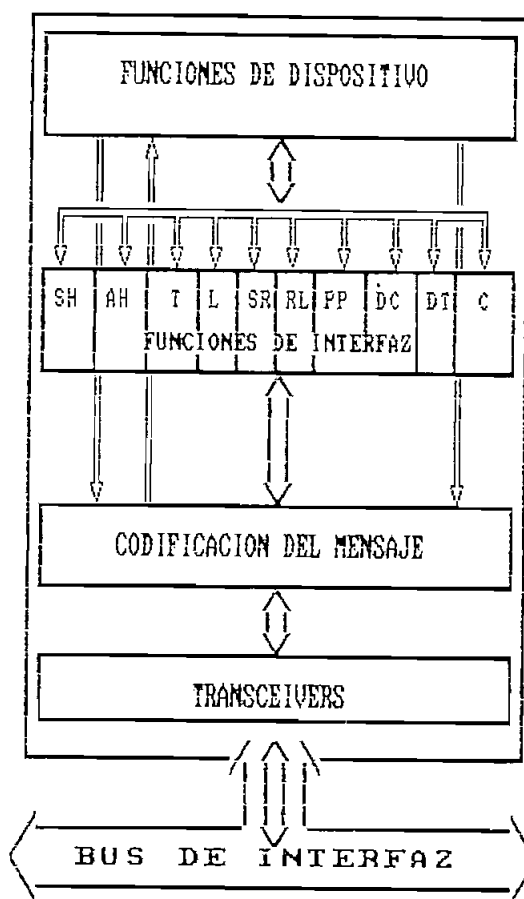


FIGURA 1.19 Funciones de un dispositivo conectado al
IEEE-488

a) PROTOCOLO FUENTE (Source Handshake) (SH)

Esta función permite al dispositivo realizar una apropiada transferencia de información por el bus, controlando el inicio, terminación y transferencia de los mensajes. Utiliza las líneas NDAC, NRFD y DAV para efectuar cada transferencia.

b) PROTOCOLO RECEPTOR (Acceptor Handshake) (AH)

Esta función permite al dispositivo realizar una apropiada recepción de mensajes. Dado que la transferencia de datos es asincrónica, y puede haber más de un dispositivo recibiendo el mensaje, el protocolo receptor puede retardar el inicio de la transmisión, o el final de la transferencia del mensaje. Utiliza las líneas DAV, NRFD y NDAC para efectuar cada transferencia.

c) LOCUTOR (T)

Esta función permite que el dispositivo envíe datos o estado a otros dispositivos conectados al bus. Un dispositivo con esta capacidad sólo podrá usarla cuando haya sido direccionado por el controlador para hablar.

Hay dos versiones alternativas de la función: La primera es la función T normal que usa un byte de dirección. La segunda tiene direccionamiento extendido y usa dos bytes de dirección, se le llama locutor extendido TE. Solamente una de las dos funciones debe tener el dispositivo.

d) OYENTE (L)

Esta función permite que el dispositivo reciba datos desde otros dispositivos conectados al bus. Un dispositivo con esta capacidad sólo podrá usarla cuando haya sido

direccionado por el controlador para oír.

Tiene dos versiones alternativas: La normal L que usa un byte de dirección. Y la que tiene direccionamiento extendido que usa dos bytes para la dirección y se le llama locutor extendido LE. Solamente una de las dos funciones debe incluirse en el dispositivo.

e) PETICION DE SERVICIO (SR)

Esta función capacita al dispositivo para pedir servicio de modo asincrónico al controlador activo. Para esto, el dispositivo activa la línea SRQ del bus, y la mantiene en este estado hasta que el controlador indique que ha recibido la petición.

f) CONTROL REMOTO/LOCAL (RL)

Permite al dispositivo seleccionar entre dos fuentes de información de entrada. La función indica si el dispositivo debe usar la información desde los controles del panel frontal (local) o la correspondiente información desde la interfaz (remota).

g) BUSQUEDA EN PARALELO (PP)

Capacita al dispositivo para responder a una búsqueda en paralelo presentando un mensaje FPR al controlador activo. Las líneas DIO1 a DIO8 llevan los bits de estado

durante la búsqueda en paralelo. Antes de responder a una búsqueda el dispositivo debió haber sido configurado asignándole una línea DIOi mediante el controlador o por un mensaje local.

h) DEVICE CLEAR (DC)

Permite que el dispositivo pueda ser borrado o inicializado individualmente o como parte de un grupo. Normalmente ésta función del bus pone a los dispositivos en el estado inicial de "power on".

La diferencia entre el comando IFC y el Device Clear es que el primero reinicializa al bus IEEE-488 y los circuitos de interfaz del bus de todos los dispositivos conectados a él sin afectar las funciones internas de éstos, y el segundo reinicializa las funciones internas del dispositivo a los valores definidos por defecto.

i) DEVICE TRIGGER (DT)

Permite que el dispositivo pueda ser iniciado en su operación básica individualmente o como parte de un grupo.

j) CONTROLADOR (C)

Esta función proporciona al dispositivo la capacidad de enviar comandos universales y comandos de direccionamiento a otros dispositivos por medio de la interfaz.

Además puede conducir búsquedas en paralelo o en serie para determinar el estado de los dispositivos.

El controlador analiza las peticiones de servicio, averigua el dispositivo que ha realizado la petición y le autoriza el uso del bus por orden de prioridad, según los criterios con los que se haya programado el controlador. Tiene además la capacidad de enviar órdenes a todos los dispositivos conectados al bus, modificando el estado interno de estos.

1.2.7 CARACTERISTICAS ELECTRICAS

Las siguientes especificaciones⁵ se definen para sistemas de interfaz que son usados en ambientes donde:

- 1.- La distancia física entre dispositivos es corta.
- 2.- El ruido eléctrico es relativamente bajo.

Las especificaciones eléctricas están basadas en el uso de tecnología TTL en lógica negativa. La relación entre los estados lógicos de los mensajes y los niveles eléctricos presentes en las líneas de señal se indican en la Tabla 1.5:

⁵ IEEE-488 STANDARD DIGITAL INTERFACE FOR PROGRAMMABLE INSTRUMENTATION, IEEE, pg. 17

ESTADO	NIVEL
0	Corresponde a +2.0V llamado estado alto
1	Corresponde a +0.8V llamado estado bajo

TABLA 1.5 Niveles Eléctricos de los Estados Lógicos

Deben usarse drivers de colector abierto para manejar las líneas de señal SRQ, NREFD y NDAC y drivers de colector abierto o drivers tres estados para manejar las líneas de señal DIO1-8, DAV, IFC, ATN, REN y EOI. Con una excepción: DIO1-8 deben usar drivers de colector abierto para aplicaciones de búsqueda en paralelo. Los drivers tres estados son usados donde se requiere operación a alta velocidad.

Las especificaciones para los drivers tres estados son las siguientes:

estado bajo: voltaje de salida (drivers tres estados o colector abierto) $< +0.5V$ en +48mA de corriente.

estado alto: voltaje de salida (tres estados) $\geq +2.4V$ en 5.2mA

Y debe ser capaz de absorber 48mA de corriente continua-

mente.

Requerimientos del receptor: Las especificaciones para el receptor con inmunidad al ruido nominal deben ser las siguientes:

estado bajo: voltaje de entrada $\leq +0.8V$

estado alto: voltaje de entrada $\geq +2.0V$

Requerimientos de carga: Cada una de las líneas de señal (esté o no conectada) debe terminar dentro del dispositivo con una carga resistiva cuyo propósito principal es establecer un voltaje de estado cuando todos los drivers de una línea están en alta impedancia y mantener una impedancia de dispositivo uniforme en la línea para proveer inmunidad al ruido.

Configuración de un circuito típico: La Figura 1.20 muestra la configuración de un circuito típico para una línea de señal de entrada/salida.

Cada una de las líneas de señal DAV, NRFD, NDAC, IFC, ATN, EOI, REN y SRQ deben ser entorchadas con su tierra lógica correspondiente, o aisladas usando un esquema equivalente para minimizar el ruido de cross talk.

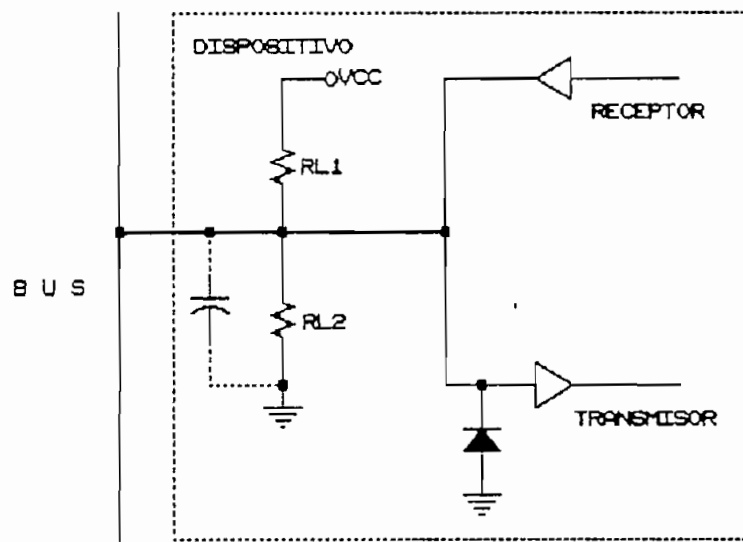


FIGURA 1.20 Impedancia Terminal de las Líneas del Bus.

1.2.8 TEMPORIZACION PARA LA TRANSICION DE ESTADOS

Para asegurar la máxima compatibilidad entre los dispositivos interconectados la norma IEEE-488 especifica la relación de tiempo entre las señales críticas de entrada y salida en los dispositivos. La Tabla 1.6 indica los valores definidos y su descripción⁶. Los valores de tiempo T1, T6, T7, T8 y T9 indican el retardo de propagación normal permitido en los enlaces de transmisión y en los circuitos de retardo dentro del dispositivo.

⁶ IEEE-488 STANDARD DIGITAL INTERFACE FOR PROGRAMMABLE INSTRUMENTATION, IEEE, pg. 52

TIEMPO	FUNCION A LA QUE SE APLICA	DESCRIPCION	VALOR
T1	SH	Tiempo de seteo de mensajes multilínea	$\geq 2 \text{ s}$
t2	SH,AH,T,L	Respuesta a ATN	$\leq 200\text{ns}$
T3	AH	Tiempo de aceptación mensajes de interfaz	> 0
t4	T,L,,C,RL	Respuesta a IFC o REN falsos	$< 100 \text{ s}$
t5	PP	Respuesta a ATN y EOI	$\leq 200\text{ns}$
T6	C	Tiempo de ejecución búsqueda en paralelo	$\geq 2 \text{ s}$
T7	C	Retardo para que el locutor vea ATN	$\geq 500\text{ns}$
T8	C	Longitud de IFC y REN falsos	$> 100 \text{ s}$
T9	C	Retardo para EOI	$\geq 1.5 \text{ s}$

TABLA 1.6 Relaciones de Tiempo para las Señales de
Entrada/Salida

1.2.9 TIPOS DE MENSAJES

La comunicación entre elementos interconectados al bus se lleva a cabo enviando mensajes a través del sistema de interfaz. En el IEEE-488 existen dos tipos de mensajes: a) mensajes dependientes del dispositivo y b) mensajes de interfaz⁷.

⁷ IEEE-488 STANDARD DIGITAL INTERFACE FOR PROGRAMMABLE INSTRUMENTATION, IEEE, pg. 55

a) Mensajes dependientes del dispositivo .- Contienen información específica de los dispositivos tales como instrucciones de programación, resultados de mediciones, estado de la máquina, archivos de datos, etc.

Después de que un locutor y uno o más oyentes han sido direccionados por medio de los mensajes de interfaz, cualquier código conocido comúnmente, como BCD, binario, o alfanumérico puede ser usado.

Los códigos alfanuméricos (código ISO) son preferidos para comunicación de los mensajes dependientes de los dispositivos donde sea posible. El bit 1 al 7 del código ISO corresponden a DIO1 - DIO7 (Ver Anexo B).

Cuando otros códigos son usados (por ejemplo el binario) el bit más significativo debe ser puesto en la línea DIO8. El código específico de los mensajes dependientes del dispositivo no ha sido especificado por esta norma.

b) Mensajes de la interfaz.- Son mensajes que manejan el bus, realizan funciones como: inicialización el bus, direccionamiento y desactivación de los dispositivos, programación local o remota, etc. Se emplea el código ISO para su codificación.

Para diferenciar los mensajes de interfaz de los mensajes dependientes del dispositivo se utiliza la línea ATN. Cuando la línea ATN está activada se envían o reciben mensajes de interfaz. En cambio cuando la línea ATN está desactivada se intercambian mensajes dependientes del dispositivo.

1.2.10 TRANSFERENCIA Y CODIFICACION DE MENSAJES REMOTOS

Los mensajes remotos que pueden ser transmitidos por el bus se indican en la Tabla 1.7:^a

TABLA 1.7 Listado de Mensajes Remotos

SIMB.	NOMBRE DEL MENSAJE	LINEAS DE DATOS								LINEAS DEL BUS							
		1	2	3	4	5	6	7	8	D	N	N	A	E	S	I	R
										A	R	D	T	O	R	F	E
										V	F	A	N	I	Q	C	N
ACG	Grupo de órdenes selectivas	x	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
ATN	Atención	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x
DAB	Byte de datos	D	D	D	D	D	D	D	D	x	x	x	x	x	x	x	x
DAC	Dato aceptado	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x
DAV	Dato válido	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x	x
DCL	Limpiar dispos.	x	0	0	1	0	1	0	0	x	x	x	x	x	x	x	x
END	Fin	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x
EOS	Fin de cadena	E	E	E	E	E	E	E	E	x	x	x	x	x	x	x	x
GET	Disparo de grupo	x	0	0	0	1	0	0	0	x	x	x	x	x	x	x	x
GTL	Pasar a local	x	0	0	0	0	0	0	1	x	x	x	x	x	x	x	x
IDY	Identificación	x	x	x	x	x	x	x	x	x	x	x	1	1	x	x	x
IFC	Limpiar interfaz	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	x
LLO	Bloqueo local	x	0	0	1	0	0	0	1	x	x	x	x	x	x	x	x

^a IEEE-488 STANDARD DIGITAL INTERFACE FOR PROGRAMMABLE INSTRUMENTATION, IEEE, pgs. 50, 51

LAG	Grupo de direcc. de oyente	x 0 1	x x x x x	x x x x x x x x	
MLA	Mi dirección de oyente	x 0 1	L L L L L	x x x x x x x x	
			5 4 3 2 1		
MTA	Mi dirección de locutor	x 1 0	T T T T T	x x x x x x x x	
			5 4 3 2 1		
MSA	Mi dirección secundaria	x 1 1	S S S S S	x x x x x x x x	
			5 4 3 2 1		
NUL	Byte nulo	0 0 0	0 0 0 0 0	x x x x x x x x	
OSA	Otra dirección secundaria		(OSA=SCG \wedge \overline{MSA})		
OTA	Otra dirección de locutor		(OTA=TAG \wedge \overline{MTA})		
PCG	Grupo de órdenes primarias		(PCG=ACG \vee UCG \vee LAG \vee TAG)		
PPC	Montar encuesta paralelo	x 0 0 0 0 1 0 1		x x x x x x x x	
PPE	Validar encuesta paralelo	x 0 0 0 0 1 0 1		x x x x x x x x	
PPD	Invalidar encuesta paralelo	x 1 1 1	D D D D	x x x x x x x x	
			4 3 2 1		
PPR1	Respuesta a encuesta paralelo1	x x x x x x x 1		x x x x x x x x	
PPR2	Respuesta a encuesta paralelo2	x x x x x x 1 x		x x x x x x x x	
PPR3	Respuesta a encuesta paralelo3	x x x x x 1 x x		x x x x x x x x	
PPR4	Respuesta a encuesta paralelo4	x x x x 1 x x x		x x x x x x x x	
PPR5	Respuesta a encuesta paralelo5	x x x 1 x x x x		x x x x x x x x	
PPR6	Respuesta a encuesta paralelo6	x x 1 x x x x x		x x x x x x x x	
PPR7	Respuesta a encuesta paralelo7	x 1 x x x x x x		x x x x x x x x	
PPR8	Respuesta a encuesta paralelo8	1 x x x x x x x		x x x x x x x x	
PPU	Desmontar encuesta paralelo	x 0 0 1 0 1 0 1		x x x x x x x x	
REN	Validar control remoto	x x x x x x x x		x x x x x x x 1	
RFD	Listo para dato	x x x x x x x x		x 0 x x x x x x	
RQS	Servicio pedido	x 1 x x x x x x		x x x x x x x x	
SCG	Grupo de órdenes secundarias	x 1 1 x x x x x		x x x x x x x x	
SDC	Limpiar dispos. seleccionado	x 0 0 0 0 1 0 0		x x x x x x x x	
SPD	Fin de encuesta serie	x 0 0 1 1 0 0 1		x x x x x x x x	
SRQ	Pedido de servicio	x x x x x x x x		x x x x x 1 x x	
STB	Byte de estado	S x S S S S S S		x x x x x x x x	

TCT	Tome el control	x 0 0 0 1 0 0 1	x x x x x x x x
TAG	Direcc. de locutor	x 1 0 x x x x x	x x x x x x x x
UCG	Orden universal	x 0 0 1 x x x x	x x x x x x x x
UNL	No oír	x 0 1 1 1 1 1 1	x x x x x x x x
UNT	No hablar	x 1 0 1 1 1 1 1	x x x x x x x x

Cada mensaje remoto se envía a través de una o más líneas de señal. Existen dos clases de mensajes remotos: mensajes unilínea y mensajes multilínea.

Los mensajes enviados como el estado lógico de una línea de señal son llamados mensajes unilínea por ejemplo ATN. Y los mensajes enviados como una combinación lógica de estados de dos o más líneas son llamados mensajes multilínea por ejemplo DCL. Este tipo de mensajes pueden ser definidos como una combinación lógica AND, OR o NOT de otros mensajes por ejemplo OTA.

FINALIZACION DE LA COMUNICACION:

Un tipo especial de mensaje es el que indica la finalización de la transferencia de una cadena de datos. La IEEE-488 define dos métodos de identificación del último byte de una cadena.

a) MENSAJE END.- En este método, el locutor activa la señal EOI (End o Identify) simultáneamente con la transmisión del último byte de la cadena de datos. Por

diseño, el oyente debe dejar de leer cuando detecta un byte acompañado por EOI, prescindiendo del valor del byte.

b) **CARACTER DE FIN DE SECUENCIA.**— En este caso, el locutor usa un caracter especial al final de su cadena de datos. Por previo arreglo el oyente deja de recibir datos cuando lo detecta. Un caracter ASCII de 7 bits o un byte binario de 8 bits puede usarse como caracter de fin de secuencia (forma ISO o ASCII).

También es posible usar una combinación de los dos métodos; pero lo importante es configurar adecuadamente al locutor y al oyente para evitar errores al finalizar la comunicación.

1.2.11 VELOCIDAD DE TRANSFERENCIA DE MENSAJES

Debido a que los intercambios que se realizan son asincrónicos no todos los dispositivos aceptan los datos a la misma velocidad, por lo tanto la velocidad de transmisión es fijada por el dispositivo más lento. Para determinar la velocidad a la cual el dispositivo transmitirá los mensajes, deben considerarse las siguientes recomendaciones de la norma⁹:

⁹ IEEE-488 STANDARD DIGITAL INTERFACE FOR PROGRAMMABLE INSTRUMENTATION, IEEE, pg. 59

a) El bus de interfaz operará en distancias de hasta 20 metros, a un máximo de 250.000 bytes/s, con un equivalente normalizado de carga por cada 2 metros de cable usando drivers de colector abierto de 48mA.

b) El bus de interfaz también operará en distancias de hasta 20 metros a un máximo de 500.000 bytes/s, con una carga equivalente normalizada por cada 2 metros usando drivers tres estados de 48mA.

c) Operación a alta velocidad: Para conseguir en un sistema la máxima velocidad posible de transferencia de datos (nominalmente de hasta 1 Mbyte/s), deben observarse las siguientes condiciones:

- (1) Todos los dispositivos que van a hablar a altas velocidades deben usar un valor mínimo de T_1 de 350ns.
- (2) Todos los dispositivos que van a operar a altas velocidades deben usar drivers tres estados de 48 mA.
- (3) La capacitancia del dispositivo en cada línea (excepto REN e IFC) debe ser menor que 50pF por cada carga resistiva equivalente en el sistema.
- (4) Todos los equipos en el sistema deben estar encendidos.
- (5) Los cables de interconexión deben ser tan cortos como

sea posible hasta un máximo de 15 metros de longitud total por sistema con al menos una carga equivalente por cada metro de cable.

1.2.12 DIRECCIONAMIENTO

Todo dispositivo conectado al IEEE-488 debe tener al menos una dirección por la cual el controlador activo del bus pueda configurarlo para hablar, oír o enviar su estado. El sistema permite 31 direcciones primarias y otro tanto de direcciones secundarias.

A todos los dispositivos se les debe asignar una dirección primaria en el rango de 00H a 1EH. La dirección listen se forma sumando 20H a la dirección primaria y la dirección talk se forma sumando 40h a la dirección primaria.

Cada dispositivo debe disponer de un juego de cinco dipswitches para ingresar por medio de estos la dirección del dispositivo, adicionalmente deben haber dos dipswitches más para seleccionar los modos "oyente solamente" y "locutor solamente".

El controlador activo es el encargado de direccionar los dispositivos enviando los comandos de

direccionamiento. El byte que emite puede tener algunos o todos los siguientes elementos:

- Un bit de paridad (P)
- Un código de dos bits que define el modo
- Cinco bits de dirección

El código de dos bits es el siguiente:

- 01 indica una dirección primaria de oyente
- 10 indica una dirección primaria de locutor
- 11 indica una dirección secundaria
- 00 indica un comando del bus

Los dispositivos conectado al bus detectan por los bits 6 y 7 de los datos que tipo de mensaje de dirección ha enviado el controlador. La dirección propiamente dicha la constituyen los bits del 1 al 5 de los datos.

En el IEEE-488 existen tres tipos de direccionamiento talk only/listen only (Ton/Lon), primario y secundario.

Ton/Lon es un método donde la habilidad de la interfaz IEEE-488 para hablar u oír está determinada por el dispositivo y no por el controlador IEEE-488. Con este método funciones fijas pueden asignarse fácilmente en sistemas simples donde no es necesaria un controlador.

Esto es conveniente para ciertas aplicaciones.

En cambio cuando es necesaria una reasignación el controlador direcciona los dispositivos mediante tres comandos: MTA (My Talk Address), MLA (My Listen Address), y MSA (My Secondary Address). La dirección del dispositivo se envía por medio de un comando. El dispositivo cuya dirección es igual a la enviada en el comando se habilita.

Cuando el controlador envía al bus una dirección de locutor, el dispositivo al que corresponde esa dirección queda seleccionado como locutor para futuras comunicaciones a través del bus. En cambio el dispositivo que estaba actuando como locutor al detectar esa dirección deja de actuar como tal. Por lo tanto un mismo mensaje de dirección de locutor es interpretado de dos formas distintas: por el dispositivo que actuaba de locutor como OTA (Otra dirección de locutor) y por el dueño de la dirección transmitida como MTA (Mi dirección de locutor).

En el direccionamiento primario, un dispositivo está habilitado para hablar u oír recibiendo el mensaje MTA o MLA respectivamente. El direccionamiento secundario extiende el campo de dirección de 5 a 10 bits, permitiendo un byte adicional. El byte adicional es enviado mediante el mensaje MSA. El direccionamiento secundario se usa para

separar lógicamente los dispositivos en varios subgrupos y se aplica solamente a los dispositivos cuya dirección primaria los precedió inmediatamente, en caso contrario se toma como el mensaje OSA (Otra Dirección Secundaria). El controlador puede enviar una dirección primaria y a continuación una serie de direcciones secundarias, seleccionando de esta forma a varios dispositivos.

1.2.13 PETICION DE SERVICIO DEL BUS

La norma IEEE-488 dispone de dos métodos para que el controlador pueda dar servicio a los dispositivos conectados al bus: la búsqueda en serie y la búsqueda en paralelo.

1.- BUSQUEDA EN SERIE

Cuando el controlador realiza una búsqueda en serie, cada dispositivo devuelve al controlador un byte de estado indicando en el sexto bit si requiere o no servicio. La norma define únicamente el bit de petición de servicio, por lo que los siete bits restantes son definidos por el usuario, y se usan para indicar qué tipo de servicio se requiere.

Si el dispositivo necesita servicio activa la línea

datos de estado desde múltiples dispositivos al mismo tiempo, en tanto que en la búsqueda en serie se recoge secuencialmente el estado de cada dispositivo.

En el desarrollo de la búsqueda en paralelo se produce la siguiente secuencia de mensajes:

1) Se selecciona los ocho dispositivos que deben responder a la búsqueda en paralelo. Para esto el controlador direcciona a los dispositivos y les envía el comando PPC (Parallel Poll Configure), y la configuración del bit asignado a cada dispositivo con el comando PPCi.

2) Se realiza la búsqueda, preguntando a los ocho dispositivos su estado. El controlador activa simultáneamente las líneas ATN y EOI (mensaje IDY). Los dispositivos seleccionados responden a la pregunta al mismo tiempo, en el bit asignado del byte de datos.

El controlador interpreta el mensaje y da servicio de acuerdo a la prioridad de cada dispositivo.

3) Si se quiere cambiar alguno de los dispositivos seleccionados se utiliza el comando PPD. Y para acabar la búsqueda en paralelo se envía el comando PPU (Parallel Poll Unconfigure).

1.2.14 DIALOGO

La transferencia de datos se realiza a través de las líneas DIOi. Todo el bus puede ser visualizado como un enlace simple de comunicación entre un locutor y al menos un oyente como se indica en la Figura 1.21.

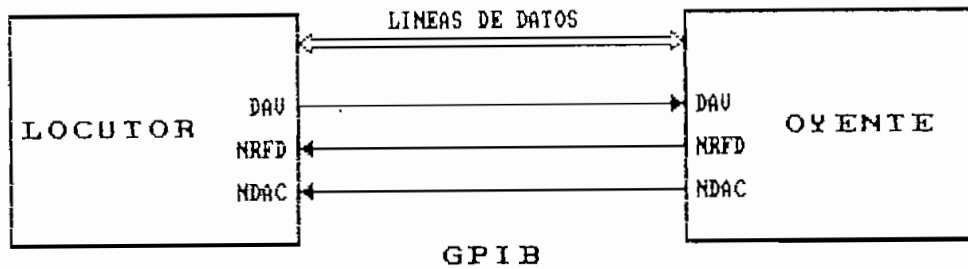


FIGURA 1.21 Enlace de Comunicación Simple IEEE-488.

Para comunicarse el locutor debe dar a conocer al oyente cuándo el dato está disponible en el bus y el oyente debe reconocer este hecho. Además deberán estar de acuerdo en que elemento activará las líneas de control.

Las tres líneas de control que hacen la sincronización entre el emisor y el receptor son DAV, NRFD y NDAC.

La línea DAV corresponde al locutor, e identifica un dato válido. Las dos líneas restantes NRFD y NDAC

corresponden a los oyentes. Los oyentes usan NRFD para indicar si están listos o no para aceptar el dato. Los oyentes usan NDAC para indicar que han recibido y aceptado el dato.

El controlador inicializa al sistema enviando el mensaje IFC (Interface Clear) para luego direccionar a los dispositivos que intervendrán en el diálogo.

El diagrama de tiempo de la Figura 1.22 ilustra el proceso del diálogo.

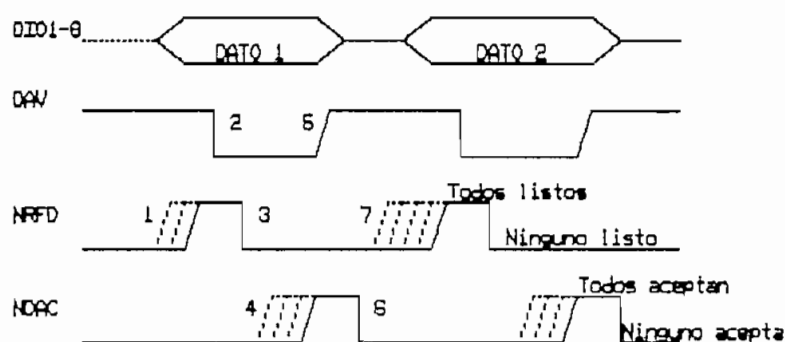


FIGURA 1.22 Diagrama de Tiempo de las Señales de Handshake

Asumiendo que la información de direccionamiento ya ha sido transferida por el bus. Oyente y locutor saben ya que han sido seleccionados.

El oyente sube la línea NRFD ① (en la Figura 1.22) cuando está listo para aceptar el dato que el locutor le enviará por el bus. El locutor pone entonces el dato en las líneas DIO del IEEE-488. Después de dejar el suficiente tiempo para que el dato se establezca en las líneas, el locutor baja su línea DAV ② indicando que el dato está disponible.

Al reconocer el nivel bajo en DAV los oyentes contestan poniendo un bajo en NRFD ③ para indicar que no están listos para aceptar otro dato. Una vez que los oyentes almacenan el dato en el buffer interno, alzan la línea NDAC ④ indicando que el dato ha sido aceptado.

El locutor reconoce que la línea NDAC está en alto y baja la línea DAV ⑤ para indicar que el dato en el bus ya no es válido. Cuando el oyente detecta que la línea DAV está en alto, nuevamente baja NDAC ⑥ reconociendo que el dato se está retirando del bus. El oyente pone entonces NRFD a alto ⑦ indicando que está listo para que el próximo byte de datos se envíe por el bus.

La secuencia se completa de esta manera. El oyente espera por el próximo dato y el locutor procesa el próximo dato antes de ponerlo en el bus. Esta secuencia es la esencia de toda comunicación a través del bus.

Cuando hay más de dos oyentes cada dispositivo participa en el mismo ciclo de handshake. La principal diferencia está en que cada paso del ciclo de handshake no se completa hasta que todos los dispositivos acaben el proceso.

El mensaje RFD no lo ve el locutor hasta que todos los oyentes están listos para datos, y el mensaje DAC no es visto por el locutor hasta que el último oyente ha aceptado el dato.

Debido a la estructura del bus y al uso de drivers de colector abierto las líneas NRFD y NDAC, forman una función lógica AND (wired OR).

El diagrama de flujo de la Figura 1.23 indica los pasos que siguen el locutor y el oyente para realizar la transferencia de datos en el bus IEEE-488 de una manera similar a la explicada con el diagrama de tiempo de la Figura 1.22.

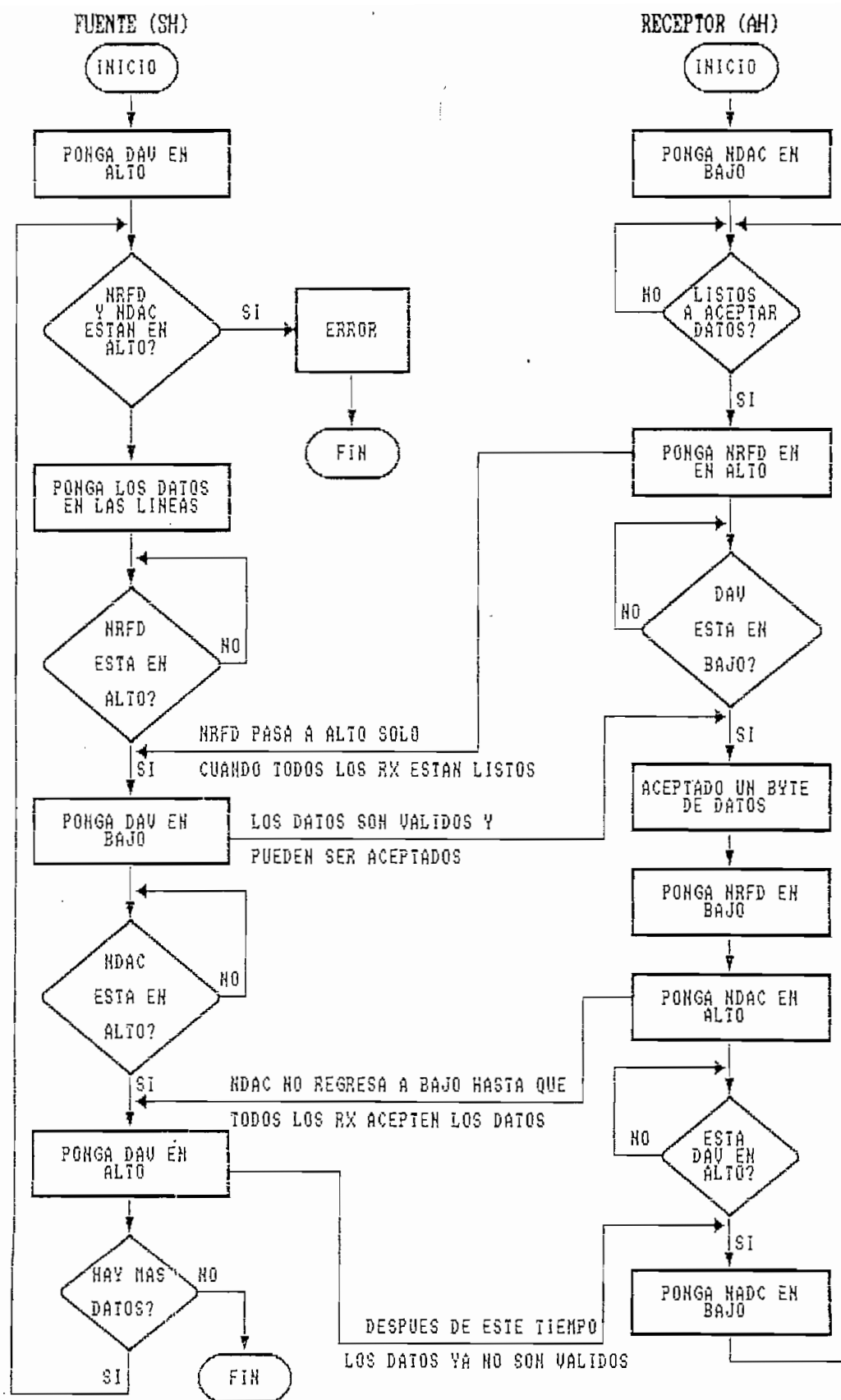


FIGURA 1.23 Procedimiento del Diálogo.

1.3 DEFINICION DE LAS CARACTERISTICAS DEL SISTEMA

Las características que se tuvieron en cuenta en el diseño para cada una de las interfaces se indican en este apartado. Como se vio en los apartados anteriores, cada una de las normas tiene diversas características que pueden ser implementadas en el diseño; sin embargo, no todas esas características se tomaron en cuenta en el diseño del sistema y para cada interfaz se definieron las características que se debía cumplir.

INTERFAZ EIA-RS232C

La interfaz EIA-RS232C, permite una comunicación sincrónica o asincrónica, con diversas formas de conexión que ya fueron explicadas, en el diseño sin embargo, se escogió únicamente la comunicación asincrónica, con ocho bits de datos, un bit de parada y sin paridad.

Debido al amplio uso de la conexión null modem se escogió ese tipo de conexión en el diseño de la parte de la interfaz serial del sistema.

En cuanto al ritmo binario, el sistema puede comunicarse a 19200, 9600, 4800, 2400, 1200, 600, 300 y 150 bits/s, este ritmo binario será detectado

automáticamente por el sistema mediante software.

INTERFAZ IEEE-488

Para la interfaz IEEE-488, el sistema tiene las capacidades de controlador, locutor; oyente, puede realizar y responder a los comandos SDC, DT, REN, LOC y a las búsquedas en paralelo y en serie.

Como se vio en el apartado 1.2 la interfaz paralela es conocida con diversos nombres; sin embargo, en la presente Tesis, solamente se le designará con el nombre de norma o interfaz IEEE-488.

CAPITULO II

DESARROLLO DEL HARDWARE

CAPITULO II

DESARROLLO DEL HARDWARE

2.1. INTRODUCCION

El sistema de conversión de interfaces serie - paralelo se diseñó para permitir la interconexión de dispositivos compatibles con la norma IEEE-488 con dispositivos compatibles con la norma EIA-RS232C.

El sistema manejado por un computador personal a través de una interfaz EIA-RS232C puede controlar dispositivos con salida IEEE-488 trabajando en ese caso como controlador del bus, ya que están implementadas todas las funciones de control del IEEE-488. Puede también conectar equipo con interfaz EIA-RS232C a un controlador IEEE-488 funcionando en ese caso como un dispositivo direccionable a través del bus o como un dispositivo "locutor solamente" u "oyente solamente" en un sistema sin controlador.

Antes de iniciar la comunicación es necesario determinar si el sistema va o no a trabajar como el controlador del bus seleccionando esa función a través de un switch externo incluido en el equipo.

Cuando el sistema no va a realizar las funciones de controlador del bus, el usuario debe seleccionar adicionalmente la dirección IEEE-488 primaria del dispositivo por la cual podrá ser direccionado por el controlador del bus o en otro caso su funcionamiento como "oyente solamente" o "locutor solamente" por medio de dip-switchs incluidos en el equipo.

Básicamente el sistema consta de tres bloques principales que son: a) el bloque EIA-RS232C, b) el bloque IEEE-488 y c) el bloque de control. Como se ilustra en la Figura 2.1.

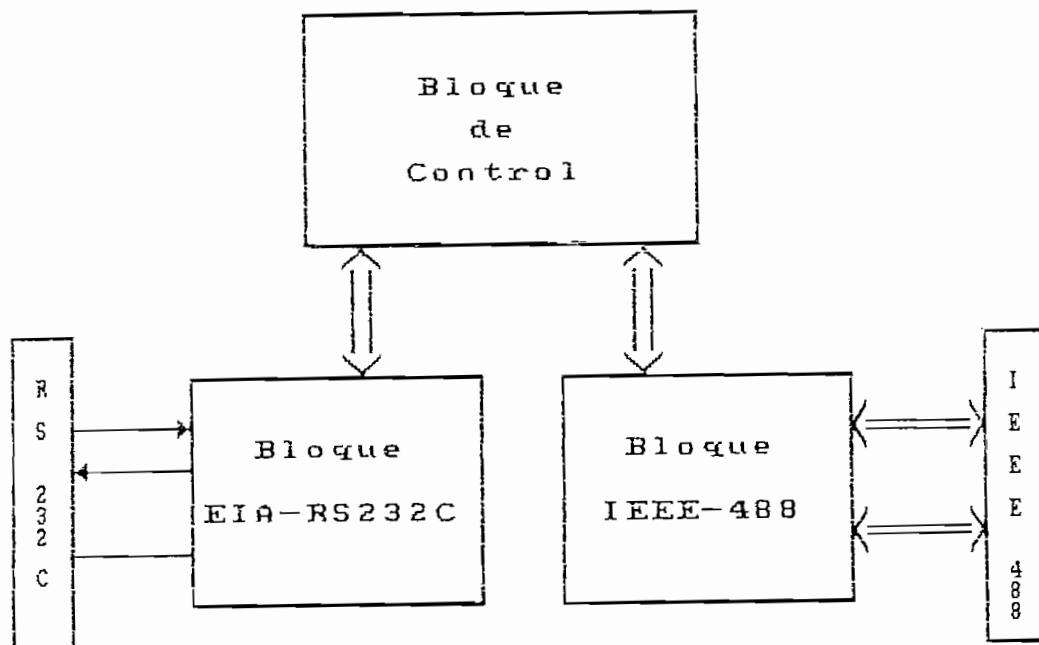


FIGURA 2.1 Partes principales del sistema de conversión

Las funciones que realizan cada uno de los bloques se indican brevemente a continuación:

BLOQUE EIA-RS232C

El bloque EIA-RS232C tiene la función básica de convertir los niveles de voltaje del microcontrolador i8751H (+5V) a los niveles de voltaje de la interfaz EIA-RS232C ($\pm 9V$), tanto en las señales de entrada como para las señales de salida del microcontrolador.

BLOQUE IEEE-488

El bloque IEEE-488 permite implementar las funciones definidas en la norma IEEE-488, el protocolo de comunicación del bus IEEE-488 y la interfaz eléctrica con dicho bus. De manera que el sistema pueda funcionar como el controlador del bus o como un dispositivo controlado a través del bus.

BLOQUE DE CONTROL

El bloque de control es la parte central del sistema, que se encarga de programar la operación del sistema, controlar su funcionamiento y realizar la comunicación de datos entre el bloque EIA-RS232C y el bloque IEEE-488.

2.2 DIAGRAMA DE BLOQUES GENERAL DEL SISTEMA

La Figura 2.2 es un diagrama de bloques detallado de la estructura del sistema de conversión. Las características generales de cada elemento constitutivo del diagrama de bloques del sistema se detallan brevemente a continuación:

C.I. i8293

El i8293 contiene los transceivers bidireccionales que permiten realizar la interfaz eléctrica del bus IEEE-488 de manera que se cumplen las especificaciones eléctricas de la norma IEEE-488 y adicionalmente puede conectarse a los circuitos integrados i8291A e i8292.

C.I. i8291A

Este circuito integrado realiza las funciones de la norma IEEE-488 que no tienen que ver con el control del bus y maneja la transferencia de mensajes entre el microcontrolador i8751H y el bus IEEE-488.

C.I. i8292

El circuito integrado i8292 realiza las funciones de control del bus IEEE-488, de modo que se puede implementar un sistema controlador con un C.I. i8292 para las funciones de control, un C.I. i8291A para la transferencia de

mensajes y dos C.I. i8293 para establecer las características eléctricas que cumple totalmente la norma IEEE-488.

C.I. i8751H

Es la parte más importante del bloque de control y todo el sistema funciona en base al programa escrito para el microcontrolador, configura la operación de los demás circuitos integrados y controla la transferencia de mensajes de un bloque a otro.

C.I. 74LS373

Puesto que el microcontrolador i8751H tiene multiplexados los buses de dirección y de datos en tiempo, el circuito integrado 74LS373 realiza la separación del bus de direcciones y de datos por medio de este latch externo, controlado por señales provenientes del microcontrolador i8751H.

C.I. MAX232

El circuito integrado MAX232 se encarga de recibir los datos de la interfaz EIA-RS232C en niveles de voltaje de $\pm 9V$ y los envía en niveles TTL al microcontrolador. De igual manera recibe los datos del microcontrolador en niveles TTL y los envía convertidos en niveles EIA-RS232C.

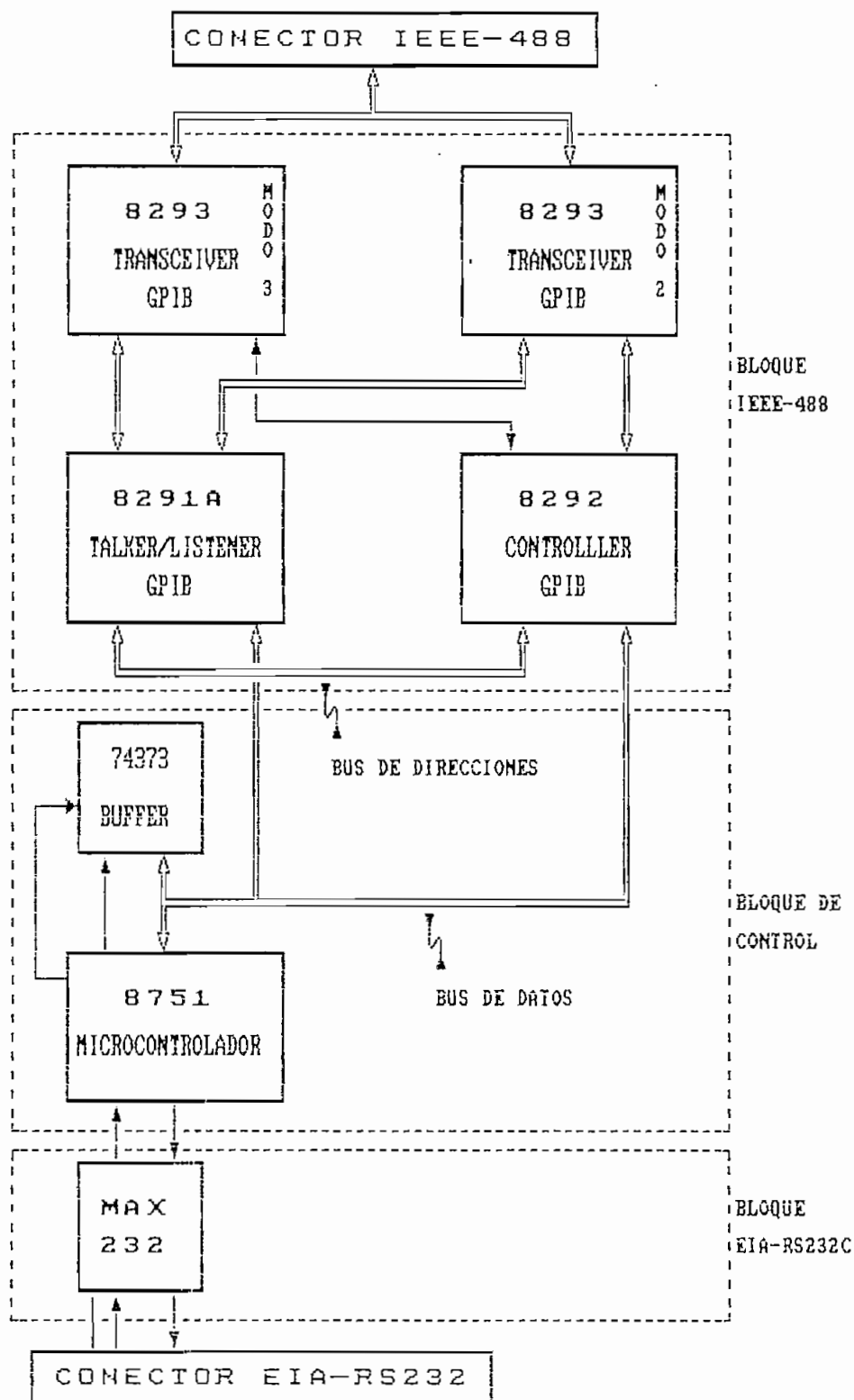


FIGURA 2.2 Diagrama de Bloques General del Sistema

2.3 DESCRIPCION Y DISEÑO DE LOS BLOQUES INDIVIDUALES

2.3.1 BLOQUE EIA-RS232C

La función principal del bloque consiste en adaptar los niveles de voltaje de la interfaz EIA-RS232C ($\pm 9V$) a los niveles TTL (5V) que usa el microcontrolador, tanto para las señales de entrada como para las señales de salida. Utilizando el circuito integrado MAX232.

El circuito integrado MAX232 que se indica en la Figura 2.3 consta de dos transmisores y dos receptores del bus serial de la norma EIA-RS232C. Y permite generar los voltajes de la interfaz serial ($\pm 9V$) a través de capacitores externos con una sola fuente de alimentación de 5V. Esta característica es bastante útil en el diseño, porque el sistema total necesita solamente una fuente.

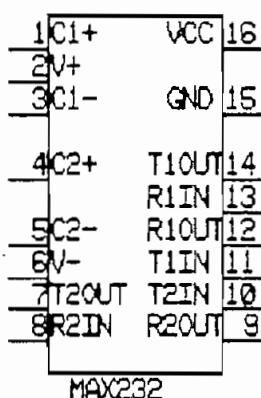


FIGURA 2.3 Circuito Integrado MAX232

Los capacitores externos necesarios para el funcionamiento del circuito integrado MAX232 son los siguientes:

- 1) Entre Vcc y GND un capacitor de $10\mu\text{F}$ (C_0).
- 2) Entre C1+ y C1- un capacitor de $4.7\mu\text{F}$ a 6.w (C_1).
- 3) Entre C2+ y C2- un capacitor de $4.7\mu\text{F}$ a 10V (C_2).
- 4) Entre V- y GND un capacitor de $10\mu\text{F}$ a 6.3V (C_3).
- 5) Entre V+ y Vcc un capacitor de $10\mu\text{F}$ a 10V (C_4)

Debido a la amplia utilización de las conexiones de tres líneas o Null Modem; en el diseño del bloque EIA-RS232C solamente se utilizaron en la comunicación las señales TXD (pin 2), RXD (pin 3) y GND (pin 7), prescindiendo de las demás señales de control y de sincronismo que incluye la interfaz por lo tanto no es necesario más que un circuito integrado MAX232 para cumplir con todas las conexiones.

Estas conexiones se ilustran en la Fig. 2.4.

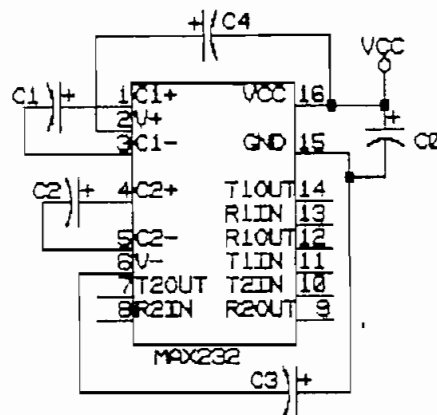


FIGURA 2.4 Conexiones del bloque EIA-RS232C

2.3.2 BLOQUE IEEE-488

Este bloque implementa la interfaz IEEE-488 como se define en la norma. Realiza el protocolo de comunicación, las funciones de interfaz y provee la interfaz eléctrica del sistema con el bus IEEE-488, de manera que se cumplen todas las especificaciones eléctricas.

Tiene dos modos de operación que pueden elegirse mediante un switch externo: a) Como un dispositivo direccionable a través del bus o b) Como el controlador del bus. En el último caso tiene capacidad para realizar todas las funciones de control y transferencia de información.

El bloque consta de un circuito integrado i8292

(IEEE-488 Controller), un circuito integrado i8291A (IEEE-48-Talker/Listener), dos circuitos integrados i8293 (IEEE-488 Transceivers) y el microcontrolador i8751H.

i8291A (IEEE-488 TALKER/LISTENER)

Realiza las funciones no controladoras de la norma IEEE-488 y maneja la comunicación entre el microcontrolador i8751H y el bus IEEE-488.

Puede decodificar la información de entrada distinguiendo entre comandos y datos, por medio del estado de la línea ATN. Si llega una dirección la verifica y reconoce cuando ha sido direccionado para oír o hablar. El circuito integrado i8291A se presenta en la Figura 2.5.

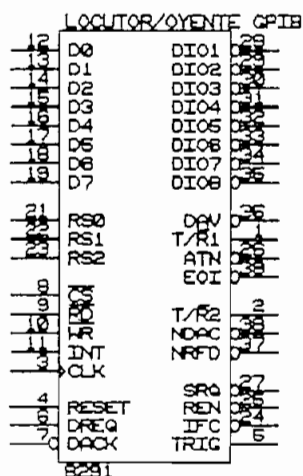


FIGURA 2.5 Circuito Integrado i8291A

Sus características principales son:

- Compatibilidad con las familias de los microcontroladores i8048/49, i8051, i8080/85 e i8086/88.
- Velocidad de transferencia de datos programable.
- Funciones: Handshake Fuente (SH), Handshake Aceptor (AH), Locutor (T), Locutor Extendido (TE), Oyente (L), Oyente Extendido (LE), Petición de servicio (SRQ), Remoto/Local (RL), Búsqueda en serie , Búsqueda en paralelo (PP2), Limpiar dispositivo (DCL), Disparar dispositivo (DT) incluidas en el circuito integrado.
- Interrupciones seleccionables.
- Reconocimiento de la dirección en el circuito integrado.
- Direccionamiento y protocolo de handshake manejado automáticamente.
- Características adicionales por software, como por ejemplo responder a una búsqueda en paralelo.
- Conecta directamente a transceivers externos no invertidos.
- Provee tres modos de direccionamiento.
- Handshake en DMA para permitir transferencias sin la intervención de el microcontrolador.
- Pin para salida de disparo.
- Reconocimiento del mensaje EOS en el circuito in-

tegrado, lo que facilita el manejo de transferencias de bytes múltiples.

Tiene dieciséis registros, ocho para escritura y ocho para lectura, de los cuales dos se utilizan en la transferencia de datos, los restantes registros controlan varias características del circuito integrado y del bus e información de estado.

Los registros son accesibles usando los pines de selección de registros RS0, RS1, RS2 y los pines de control \overline{CS} , \overline{RD} , \overline{WR} de acuerdo a lo especificado en la Tabla 2.1:

REGISTROS	\overline{CS}	\overline{RD}	\overline{WR}	RS0	RS1	RS2
Registros lectura	0	0	1	C	C	C
Registros escritura	0	1	0	C	C	C
Alta impedancia	1	X	X	X	X	X

TABLA 2.1 Pines de selección de registros

Los pines RS0, RS1 y RS2 llevan la dirección del registro seleccionado y el microcontrolador i8751H activa las señales \overline{RD} y \overline{WR} para leer y escribir en el registro.

Los tres pines de selección de registros RS0, RS1 y

RS2 se conectan a las líneas A0, A1 y A2 del microcontrolador i8751H previamente demultiplexadas a través del circuito integrado 74LS373 evitando de esta manera conflictos entre datos y direcciones y la línea de control \overline{CS} se conecta a la línea de dirección A4.

Los registros del i8291A se indican en las Figuras 2.6 y 2.7.

DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
-----	-----	-----	-----	-----	-----	-----	-----

REGISTRO DATA OUT

CPT	APT	GET	END	DEC	ERR	BO	BI
-----	-----	-----	-----	-----	-----	----	----

REGISTRO INTERRUPT ENABLE 1

0	0	DMAO	DMAI	SPC	LLOC	REMC	ADSC
---	---	------	------	-----	------	------	------

REGISTRO INTERRUPT ENABLE 2

S8	rs	S6	S5	S4	S3	S2	S1
----	----	----	----	----	----	----	----

REGISTRO SERIAL POLL MODE

TO	LO	0	0	0	0	ADM1	ADM0
----	----	---	---	---	---	------	------

REGISTRO ADDRESS MODE

FIGURA 2.6 Registros de Escritura del C.I. i8291A

CNT2	CNT1	CNT0	CM4	CM3	CM2	CM1	CM0
------	------	------	-----	-----	-----	-----	-----

REGISTRO AUX MODE

ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
-----	----	----	-----	-----	-----	-----	-----

REGISTRO ADDRESS 0/1

EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
-----	-----	-----	-----	-----	-----	-----	-----

REGISTRO EOS

FIGURA 2.6 Continuación

DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
-----	-----	-----	-----	-----	-----	-----	-----

REGISTRO DATA IN

CPT	APT	GET	END	DEC	ERR	BO	BI
-----	-----	-----	-----	-----	-----	----	----

REGISTRO INTERRUPT STATUS 1

INT	SPAS	LLO	REM	SPC	LLOC	REMC	ADSC
-----	------	-----	-----	-----	------	------	------

REGISTRO INTERRUPT STATUS 2

S8	SRQS	S6	S5	S4	S3	S2	S1
----	------	----	----	----	----	----	----

REGISTRO SERIAL POLL STATUS

FIGURA 2.7 Registros de Lectura del C.I. i8291A

ton	lon	EOI	LPAS	TPAS	LA	TA	MJMN
-----	-----	-----	------	------	----	----	------

REGISTRO ADDRESS STATUS

CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0
------	------	------	------	------	------	------	------

REGISTRO COMMAND PASS TROUGH

INT	DT0	DL0	AD50	AD40	AD30	AD20	AD10
-----	-----	-----	------	------	------	------	------

REGISTRO ADDRESS 0

X	DT1	DL1	AD51	AD41	AD31	AD21	AD11
---	-----	-----	------	------	------	------	------

REGISTRO ADDRESS 1

FIGURA 2.7 Continuación

El control del circuito integrado i8291A por parte del microcontrolador se realiza leyendo y escribiendo a los dieciséis registros del circuito integrado.

Para la transferencia de mensajes con el bus se utilizan los registros DATA IN (lectura) y DATA OUT (escritura). El primero permite ingresar datos desde el IEEE-488 al microcontrolador o hacia memoria cuando el i8291A está recibiendo datos. El segundo en cambio permite que el microcontrolador envíe datos al bus.

Los Registros Interrupt Enable e Interrupt Status permiten definir la manera en que el microcontrolador establecerá la comunicación con el i8291A.

Cuando se requiere que la comunicación se establezca mediante interrupciones, los bits de los Registros Interrupt Enable 1 y 2 deben habilitarse previamente al inicio de la comunicación. Los bits de los Registros Interrupt Status 1 y 2 tienen su respectivo bit de habilitación en los Registros Interrupt Enable 1 y 2.

Las condiciones que pueden generar interrupción se indican en la Tabla 2.2:

Al ocurrir una interrupción el microcontrolador debe leer los registros de estado para determinar qué condición la produjo y realizar la rutina correspondiente. Los bits activados se borran al leer el registro de estado respectivo.

El pin de interrupción (11) del i8291A normalmente se activa en alto y se borra cuando el microcontrolador lee el registro de estado. Pero puede ser configurado por software para activarse en bajo como requiere el i8751H habilitando esta característica en el i8291A.

CONDICION	DESCRIPCION
CPT	Recibido comando indefinido
APT	Pasar dirección secundaria
GET	Ocurrió comando GET
END	Recibido mensaje EOI y/o EOS
DEC	Estado activo de Device Clear
ERR	Error no hay oyentes activos
EO	Espera byte de salida
BI	Léase byte de entrada
SPC	Completa búsqueda serial
LLOC	Cambio en estado del Local Lock
REMC	Cambio estado de Remoto/Local
ADSC	Cambio en el direccionamiento

* Nota: Todas las condiciones se activan con 1L.

TABLA 2.2 Condiciones que Producen Interrupción
en el i8291A.

Si no se requiere una comunicación mediante interrupciones todos los bits que pueden producirlas deben deshabilitarse y el microcontrolador deberá leer secuencialmente los Registros Interrupt Status 1 y 2 para determinar si uno o más bits están seteados y realizar la rutina correspondiente.

El i8291A puede operar en diversos modos de direccionamiento que se eligen a través del Registro Address

Mode. El modo indican la manera en que el circuito integrado será direccionado.

A través de los bits AD1 y AD2 se eligen cuatro modos de operación:

MODO 1: Tiene dos direcciones primarias separadas, una dirección de locutor oyente mayor y una dirección de locutor oyente menor. En aplicaciones donde sólo una dirección es necesaria se usa la dirección mayor y se deshabilita la menor. Las direcciones se cargan en el Registro Address 0/1 una a continuación de la otra.

MODO 2: Este modo en cambio permite al usuario tener una sola dirección locutor oyente pero de dos bytes, es decir una dirección primaria y una dirección secundaria. Ambas direcciones deben recibirse para habilitar al dispositivo. Con este modo se consiguen las funciones de locutor extendido y oyente extendido como se definen en la norma. Las dos direcciones se cargan en el Registro Address 0/1.

MODO 3: Se parece al Modo 1, pero en este caso las dos direcciones primarias tienen una dirección secundaria, que necesita ser verificada por el microcontrolador.

Adicionalmente a esos cuatro modos de

direccionamiento el i8291A puede programarse en los modos TON (talk only) o LON (listen only) habilitados a través de los bits TO y LO del Registro Address Mode.

En el modo TON el dispositivo puede operar como un locutor en un sistema sin controlador y se lo usa también cuando el i8291A funciona con el i8292 para enviar comandos o datos al bus porque en ese caso no puede ser direccionado a través del IEEE-488. En el modo LON el dispositivo opera como un oyente en un sistema sin controlador y también cuando funciona con el i8292 para recibir datos y estado del IEEE-488.

Otro registro del i8291A es el Registro Command Pass Through necesario para los comandos de la norma que el i8291A no reconoce. Y se lo usa también en sistemas que emplean direccionamiento secundario, ya que en el modo 3 de direccionamiento la dirección secundaria debe pasarse al microcontrolador para su verificación.

Un registro importante en el i8291A es el Registro Aux Mode. Con este registro es posible realizar las siguientes funciones:

- a) Cargar los registros auxiliares escondidos A y B.
- b) Enviar comandos desde el microcontrolador al i8291A.

- c) Presetear un contador interno usado para generar el retardo T1 en la función Handshake Fuente, como la define el IEEE-488.

Estas funciones se realizan en el circuito integrado de acuerdo a la Tabla 2.3:

BITS CONTROL	BITS COMANDOS	FUNCION
000	0CCCC	Ejecuta comando auxiliar CCCC
001	0DDDD	Carga contador interno a frecuencia DDDD
100	DDDDD	Escribe al Registro Auxiliar A comando DDDDD
101	DDDDD	Escribe al Registro Auxiliar B comando DDDDD
011	USP ₃ P ₂ P ₁	Habilita o no la búsqueda en paralelo

TABLA 2.3 Funciones del Registro Aux Mode.

Registros Auxiliares:

El i8751H tiene dos registros auxiliares escondidos que habilitan algunas características del circuito integrado y a los que puede escribirse a través del Registro Aux Mode

El Registro Auxiliar A permite habilitar caracterís-

ticas como: Setear el bit END en el Registro Interrupt Status 1 cuando se ha recibido el caracter EOS en una lectura, o activar la línea EOI cuando se envía el caracter EOS en escritura, entre otras características.

El Registro Auxiliar B permite características como: habilitar el manejo por software de comandos indefinidos, alta velocidad en transferencia de datos, permitir que el pín 11 de interrupción se active en bajo, etc.

Comandos Auxiliares:

Los comandos auxiliares que el microcontrolador puede enviar al i8291A permiten entre otras cosas inicializar al circuito integrado, liberarlo del estado de reset, enviar EOI con el último byte de datos, reconocer la dirección secundaria en el Modo 3 de direccionamiento, etc.

Contador Interno:

Para cargar el contador interno debe enviarse un byte de la forma 0010DDDD (binario), en donde DDDD puede ir de 0001 (1 MHz) a 1000 (8 MHz), de acuerdo a la frecuencia de la señal conectada en el pín 3 (Clock). Dicha señal se usa para determinar el tiempo de retardo permitido para que se establezcan los datos en las líneas DIO, definido como T1.

DAV se activa T1 segundos después de que se envió los datos a las líneas DIO.

Consecuentemente T1 es un factor importante en la determinación de la velocidad de transferencia de datos del i8291A hacia el IEEE-488. Cuando se usan transmisores de colector abierto para la conexión al bus, T1 se define como 2µs. Con los transmisores tres estados, el IEEE-488 permite la velocidad de transferencia más alta. Para el diseño se escogió una frecuencia de 1 MHZ como reloj externo por lo que se carga el contador interno con 21H.

FUNCIONAMIENTO:

Antes de operar normalmente el i8291A debe ser inicializado enviando una señal de reset al pin 4 (RST), o escribiendo el comando Reset al Registro Aux Mode. Con lo que en el circuito integrado se producen los siguientes eventos:

- Se mantiene el mensaje local "pon" como está definido en la norma hasta que el circuito integrado sea liberado del estado de reset.
- Los Registros Interrupt Status 1 y 2, Serial Poll Mode y Auxiliares A y B, se borran.
- Se borran el bit de búsqueda en paralelo y el bit EOI

en el Registro Address Status.

- El contador interno se inicializa a 8 MHz.
- Es enviado el mensaje local rdy.

En este estado el i8291A puede ser programado por el microcontrolador, escribiendo en sus registros las características requeridas en el sistema. Como: habilitación de interrupciones, dirección del dispositivo, cargar el caracter EOS si es necesario, etc.

Para liberar al circuito integrado del estado de reset el microcontrolador envía el comando Immediate Execute Pon al Registro Aux Mode.

Antes de escribir datos al bus, el microcontrolador debe esperar que el bit BO en el Registro Interrupt Status 1 del i8291A se setee. Esto sucede cuando ha sido detectado el mensaje RFD lo que indica que debe enviarse un dato al IEEE-488. Se escribe el dato al Registro Data Out del i8291A, y este circuito integrado inicia y completa el handshake mientras envía el dato al bus.

Cuando quiere leer un dato, el microcontrolador debe esperar que el bit BI en el Registro Interrupt Status 1 del i8291A se setee, lo que indica que hay un dato disponible en el Registro Data In, para que el i8751 lo

lea. El mensaje RFD se mantiene falso mientras no se lee el registro. El i8291A completa automáticamente el handshake.

i8292 IEEE-488 CONTROLLER

El circuito integrado i8292 (Fig.2.8) es un microcontrolador Intel 8041A programado para realizar las funciones de controlador IEEE-488 usado con un i8291A y dos i8293. En este caso se comporta como un procesador esclavo inteligente para el microcontrolador maestro. En la Figura 2.9 se presenta el diagrama de bloques del i8041A, éste contiene procesador, memoria RAM, memoria ROM (Conteniendo el programa controlador IEEE-488), líneas de entrada/salida y un contador/temporizador. Y en la Figura 2.10 se presenta el Modelo de Programación del i8041A.

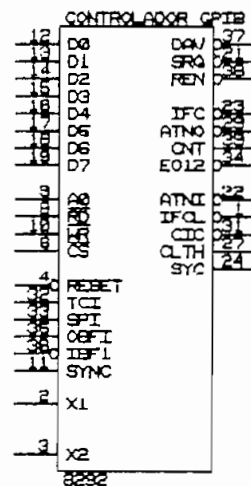


FIGURA 2.8 Circuito Integrado i8292.

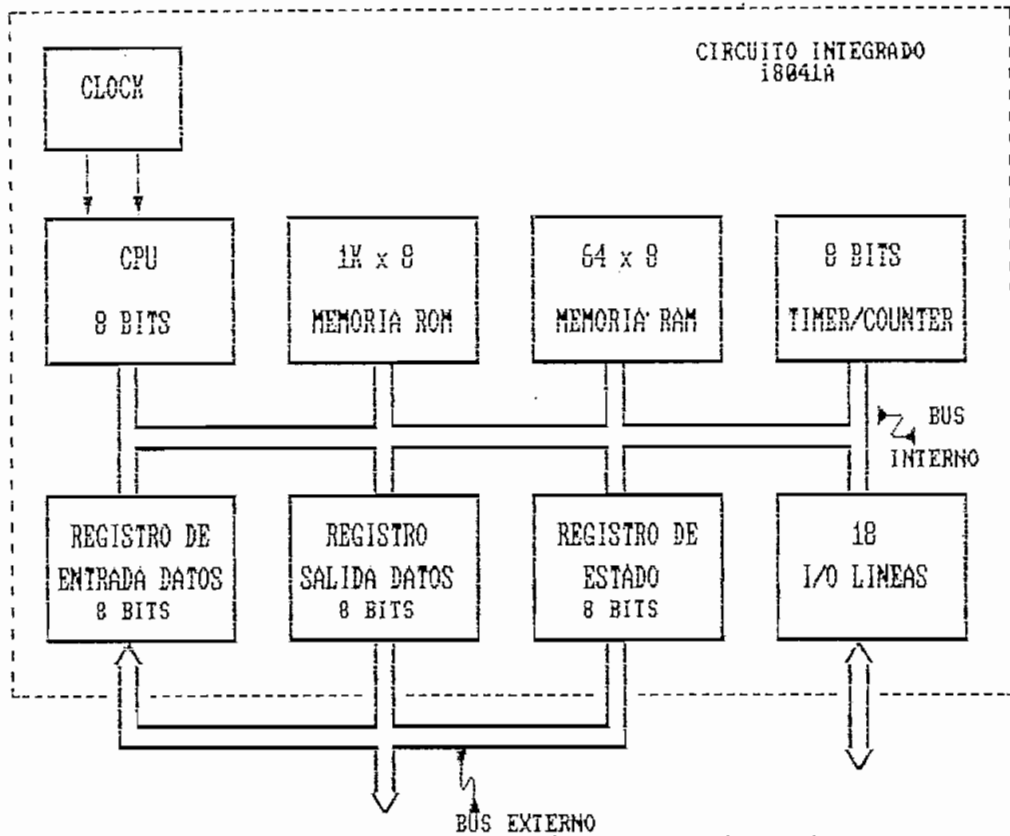


FIGURA 2.9 Diagrama de Bloques del C.I. 8041A

Las capacidades del circuito integrado son las siguientes:

- Control e inicialización del bus a través del manejo de la línea IFC.
- Atención a peticiones de servicio.
- Envío del mensaje REN.
- Protocolo completo de control de la transferencia.
- Toma de control sincrónica para prevenir la destrucción de cualquier transmisión de datos en progreso.
- Envío de mensajes de interfaz.

- Toma y entrega de control del bus.
- Búsquedas en paralelo.

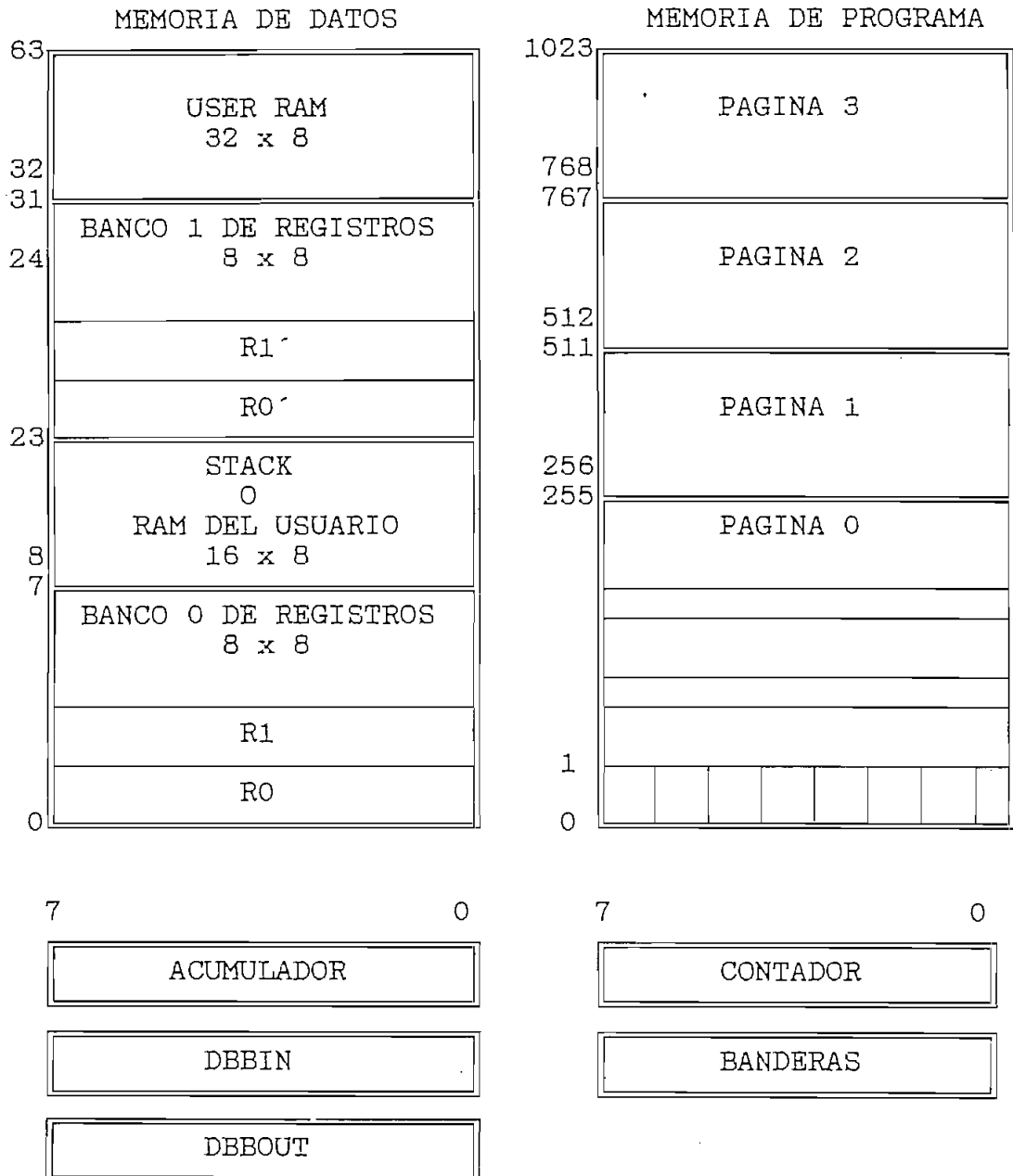


FIGURA 2.10 Modelo de Programación del i8041A

El contador interno se utiliza para funciones de timeout o para contar los bytes de datos transferidos. La RAM interna se usa como un banco de registros de propósito especial. Aunque el 8041A tiene solamente dos registros de lectura y uno de escritura, a través del firmware el i8292 presenta seis registros de lectura y cinco de escritura.

Para acceder a los registros se utilizan las líneas de control \overline{CS} , \overline{RD} y \overline{WR} y la línea de dirección A0 de la manera que se indica en la Tabla 2.4. La línea de dirección A0 se usa para seleccionar entre el bus de datos y el Registro de Estado durante operaciones de lectura y para distinguir entre datos y comandos en operaciones de escritura.

REGISTROS	\overline{CS}	\overline{RD}	\overline{WR}	A0
Registros lectura	0	0	1	0
Leer estado	0	0	1	1
Escribir comandos	0	1	0	0
Registros escritura	0	1	0	1
Deshabilitados	1	X	X	X

TABLA 2.4 Líneas de Selección del C.I. i8292.

Las Figuras 2.11 y 2.12 presentan los registros disponibles para lectura y escritura en el i8292 respectivamente. La línea de control CS fue conectada a la línea

de dirección A5 del microcontrolador i8751.

SYC	ERR	SRQ	EV	X	IFCR	IBF	OBF
-----	-----	-----	----	---	------	-----	-----

REGISTRO INTERRUPT STATUS

X	X	USER	X	X	TOUT3	TOUT2	TOUT1
---	---	------	---	---	-------	-------	-------

REGISTRO ERROR FLAG

CSBS	CA	X	X	SYCS	IFCC	REN	SRQ
------	----	---	---	------	------	-----	-----

REGISTRO CONTROLLER STATUS

REN	DAV	EOI	X	SYC	IFC	ANTI	SRQ
-----	-----	-----	---	-----	-----	------	-----

REGISTRO GPIB (BUS) STATUS

D	D	D	D	D	D	D	D
---	---	---	---	---	---	---	---

REGISTRO EVENT COUNTER STATUS

D	D	D	D	D	D	D	D
---	---	---	---	---	---	---	---

REGISTRO TIME OUT STATUS

FIGURA 2.11 Registros de Lectura del C.I. i8292.

1	SPI	TCI	SYC	OBFI	IBFI	0	SRQ
---	-----	-----	-----	------	------	---	-----

REGISTRO INTERRUPT MASK

0	0	USER	0	0	TOUT3	TOUT2	TOUT1
---	---	------	---	---	-------	-------	-------

REGISTRO ERROR MASK

1	1	1	1	OP	C	C	C
---	---	---	---	----	---	---	---

REGISTRO COMMAND FIELD

D	D	D	D	D	D	D	D
---	---	---	---	---	---	---	---

REGISTRO EVENT COUNTER

D	D	D	D	D	D	D	D
---	---	---	---	---	---	---	---

REGISTRO TIME OUT

FIGURA 2.12 Registros de Escritura del C.I. i8292.

Los Registro Interrupt Mask (IM) y Error Mask (EM), pueden ser escritos directamente con la línea de dirección A0 igual a cero, el firmware del circuito integrado usa el bit más significativo para diferenciar un registro de otro.

En cambio para lectura el Registro Interrupt Status

es el único que puede leerse directamente. Los restantes registros se seleccionan a través de comandos del i8292 que se escriben en el Registro Command Field.

El i8292 puede realizar dos grupos de comandos que se distinguen por el valor del bit OP en el Registro Command Field y son: 1) los comandos de operación y 2) los comandos utilitarios. Las dos clases de comandos se escriben en el i8292 con la línea A0 en alto.

1) COMANDOS DE OPERACION

Inician alguna acción en el bus de interfaz IEEE-488. Usando estos comandos se realizan funciones de control como búsqueda en paralelo, pasar y recibir el control, tomar el control y otras funciones del control del sistema. Para estos comandos el bit OP es igual a 1L.

2) COMANDOS UTILITARIOS

Son utilizados para leer y escribir a los registros del i8292 que no son accesibles directamente. Para realizar esto, se escribe al Registro Command Field el comando utilitario correspondiente, y a continuación se escribe o lee la información en el registro del i8292 seleccionado. En este caso el bit OP es igual a 0L.

El microcontrolador puede escribir comandos al buffer del bus de datos tan pronto como el comando anterior ha sido leído, comprobando el estado de la interrupción $\overline{\text{IBFI}}$. El microcontrolador no debe escribir al i8292 mientras $\overline{\text{IBFI}}$ esté en 1L porque la información se perderá.

El circuito integrado i8292 tiene cuatro pines de interrupción que son los siguientes:

$\overline{\text{IBFI}}$	Buffer de entrada vacío: usado para interrumpir al microcontrolador mientras el buffer de entrada del i8292 está vacío.
OBFI	Buffer de salida lleno: usado para indicar que el buffer de salida está lleno.
SPI	Interrupción en eventos especiales: usada en eventos no iniciados por el microcontrolador.
TCI	Interrupción en tarea completa: interrumpe al microcontrolador para indicar que la tarea requerida se completó en el i8292 y la información pedida está lista en el buffer del bus de datos.

Los eventos especiales que provocan interrupción. SPI son los siguientes:

IFCR	IFC recibida
---------------	-----------------------

EV	Interrupción del Contador de Eventos
SRQ	Petición de servicio
ERR	Ocurrió un error
SYC	Cambió el estado del switch controlador.

Al detectar una interrupción SPI, el microcontrolador debe leer el Registro Interrupt Acknowledge para que el i8292 borre la interrupción y los bits correspondiente en el Registro Interrupt Status. Puede reconocer cualquier combinación de las cinco interrupciones SPI: SYC, ERR, SRQ, EV e IFC.

La interrupción TCI es útil cuando el microcontrolador envía comandos al i8292, TCI se borrará con cada nuevo comando escrito al circuito integrado.

Los comandos enviados al i8292 pueden dividirse en dos grupos:

- a) Comandos que requieren respuesta del i8292 al microcontrolador, por ejemplo la lectura de un registro y
- b) Comandos que inician alguna acción o habilitan características, pero no requieren respuesta del i8292 por ejemplo la habilitación de las interrupciones.

Con el primer grupo, la interrupción TCI se usa para indicar que la respuesta requerida está lista en el buffer y el microcontrolador puede leerla. Con el segundo grupo, se usa para indicar que se completó la tarea y el microcontrolador puede enviar nuevos comandos.

El i8292 puede comunicarse con el microcontrolador mediante interrupciones, o por lectura secuencial de los registros de estado.

Cuando la comunicación no se realiza a través de interrupciones sino por lectura secuencial de los registros de estado, todas las interrupciones deben deshabilitarse, el microcontrolador revisará si algún bit está seteado y realizará la rutina correspondiente.

FUNCIONAMIENTO:

El i8292 puede reinicializarse de tres modos diferentes: a) cuando se enciende el equipo, b) con una señal de reset externa aplicada al pin 4 (RST), o c) enviando el comando RST a su Registro Command Field. En este estado se produce la siguiente secuencia de eventos:

- 1) Todas las salidas del IEEE-488 van a alto \overline{SRQ} , \overline{ATNI} , \overline{SYC} , \overline{CLTH} , \overline{ATNO} , \overline{CIC} , \overline{TCI} , \overline{SPI} , \overline{EOI} , \overline{OBFI} , \overline{IBFI} , \overline{DAV} ,

$\overline{\text{REN}}$.

- 2) Las cuatro salidas de interrupción (TCI, SPI, $\overline{\text{IBFI}}$ y OBFI) y la salida CLTH se ponen en bajo.
- 3) Los registros de Interrupt Status, Interrupt Mask, Error Flag, Error Mask, Time Out y Event Counter se borran.
- 4) Si el i8292 es el controlador del sistema, ejecuta un comando ABORT y toma el control del bus.

Para su correcto funcionamiento el i8292 necesita una señal de reloj, que puede generarla internamente ya que tiene integrado un oscilador al que debe conectarse un cristal externamente en los pines 2 y 3 como se indica en la Figura 2.13.

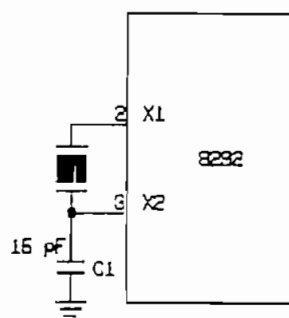


FIGURA 2.13 Circuito de Reloj del C.I. i8292.

i8293 IEEE-488 TRANSCEIVER

El i8293 es un circuito integrado multiuso que contiene los transceivers bidireccionales IEEE-488 no invertidos, diseñados para realizar la interfaz eléctrica con el IEEE-488 (Fig. 2.14). Contiene la lógica adicional para que el i8291A, i8292 e i8293 trabajen juntos.

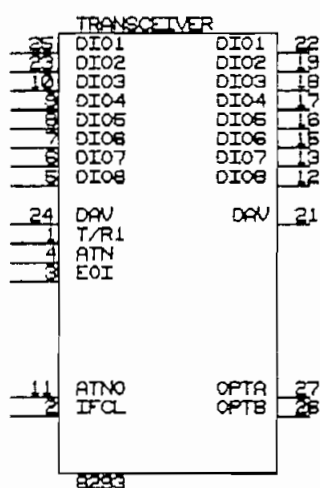


FIGURA 2.14 Circuito integrado i8293.

El circuito integrado i8293 tiene las siguientes características:

- Nueve drivers con salida seleccionable de colector abierto o tres estados de acuerdo a la configuración del usuario.
- Drivers de línea con capacidad de corriente de hasta 48 mA.

- Nueve receptores tipo Schmitt Trigger.
- Drivers con capacidad de manejo de cargas capacitivas altas.
- Fuente única de 5V.
- Tecnología HMOS de baja potencia.
- Decodificador interno del modo de operación.
- Protección del estado del bus en encendido y apagado.
- Conectado con el i8291A y el i8292 forma una interfaz IEEE-488 Locutor/Oyente/Controlador sin componentes adicionales.
- Se requiere dos circuitos integrados i8293 por cada interfaz.
- Terminaciones del bus de acuerdo a la norma IEEE-488.

El i8293 cumple y en algunos casos excede todas las especificaciones eléctricas definidas en la norma IEEE-488 1978.

Puede ser programado por hardware para funcionar en uno de sus cuatro modos de operación. Estos modos le permiten ser configurado y funcionar en ambientes Locutor/Oyente/Controlador y Locutor/Oyente. Adicionalmente puede ser usado como transceiver del bus con salida tres estados (pull/push) o colector abierto de propósito general con nueve drivers/receivers.

Los modos de operación del i8293 se escogen a través de los pines OPTA (27) y OPTB (26) de acuerdo a la Tabla 2.5. Según el modo escogido se configura la lógica interna del circuito integrado.

OPTA	OPTB	MODO
0	0	0
0	1	1
1	0	2
1	1	3

TABLA 2.5 Modos de operación del circuito integrado
i8293

MODO 0: Este modo de operación configura al i8293 para manejar las líneas de control de un sistema Talker/Listener, IFC, REN, ATN como entradas al sistema; SRQ como salida del sistema y las líneas EOI, NRFD y NDAC como líneas bidireccionales.

MODO 1: En el modo 1 en cambio el i8293 se configura para manejar las líneas de datos del bus IEEE-488 DIO1 - DIO8 y la líneas de control DAV como líneas bidireccionales en un sistema Talker/Listener.

MODO 2: Este modo y el modo 3 se utilizan en sistemas con-

figurados como Talker/Listener/Controller. El circuito integrado i8293 en el modo 2 maneja las líneas de control IFC, REN, ATN como salidas del sistema; SRQ como entrada al sistema y las líneas NDAC, NRFD y EOI como líneas bidireccionales.

MODO 3: El i8293 en el modo 3 se encarga de manejar las líneas de datos DIO1 - DIO8 del bus IEEE-488 y la línea de control DAV funcionando en un sistema Controlador. Con la característica adicional de que en este caso las salidas de los drivers de las líneas de datos se seleccionan como salidas tres estados para enviar comandos y datos y como salidas de colector abierto cuando se realiza una búsqueda en paralelo.

Cada interfaz IEEE-488 requiere dos circuitos integrados i8293 configurados en dos modos diferentes porque uno de ellos lleva los datos y la señal DAV hacia el bus IEEE-488 y el otro lleva las restantes señales de control.

Si el sistema va a trabajar como Locutor/Oyente se utilizan los Modos 0 y 1. Si adicionalmente va a trabajar como Controlador se utiliza los modos 2 y 3. Estas configuraciones se indican en las Figuras 2.14 y 2.15.

El sistema realizado en el diseño tiene una con-

figuración Locutor/Oyente/Controlador por lo tanto los dos circuitos integrados i8293 funcionarán en los Modos 2 y 3 respectivamente.

Las señales en las líneas pueden ser tanto entradas como salidas, el i8293 tiene dos pines que controlan la dirección de las líneas, y son el pin 1 ($T/\bar{R}1$) y el pin 2 ($T/\bar{R}2$) manejadas por el i8291A y el i8292.

$T/\bar{R}1$: Controla la dirección de las líneas NDAC, NRFD, DAV y DIO1 -DIO8, cuando está en alto NRFD y NDAC se reciben y las demás señales se transmiten.

$T/\bar{R}2$: Controla la dirección de EOI.

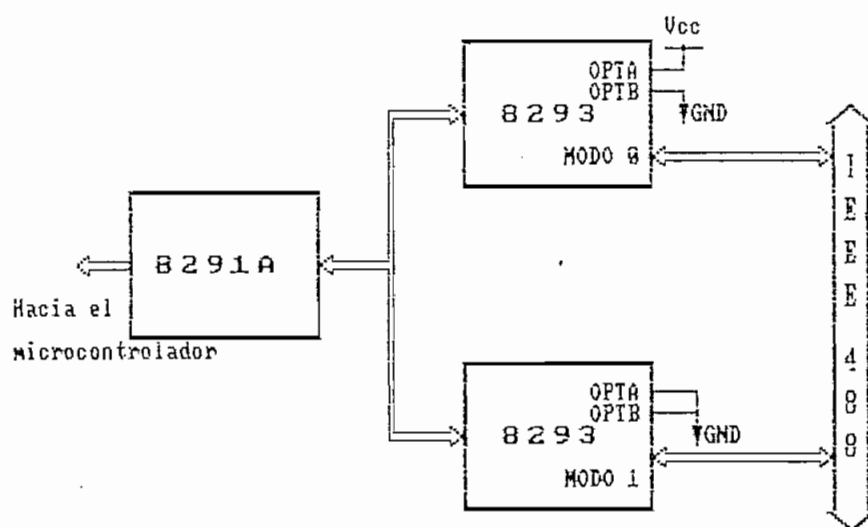


FIGURA 2.15 Configuración Locutor/Oyente.

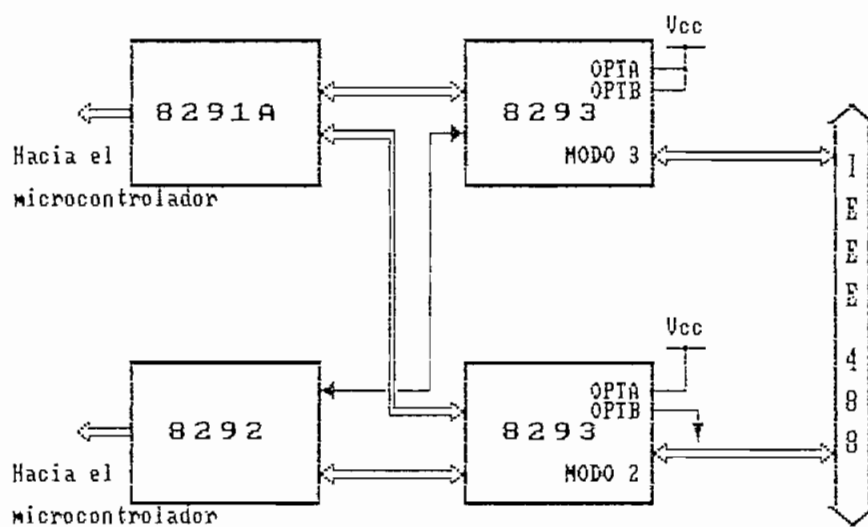


FIGURA 2.16 Configuración Locutor/Oyente/Controlador.

En el i8293 las salidas de los drivers pueden configurarse como salidas de colector abierto o como salidas tres estados. Esta característica es útil para cumplir con las especificaciones de la norma que permite el uso de drivers con salidas tres estados en las líneas DIO1 -DIO8 salvo en el caso en que se realice una búsqueda en paralelo en la cual las salidas deben ser de colector abierto.

En el Modo 3 del i8293 las líneas EOI y ATN están internamente conectadas a una compuerta AND, esto determina cuando las salidas de los drivers son tres estados o de colector abierto. Cuando alguna de las señales está en alto, las salidas son tres estados, pero si las dos señales están en bajo simultáneamente el controlador está realizando una búsqueda en paralelo, por lo que las salidas de los drivers se configuran como salidas de colector abierto.

2.3.3 BLOQUE DE CONTROL

El bloque de control consta del microcontrolador i8751H, el latch 74LS373 y lógica adicional. Es la parte más importante del sistema, recibe y envía los datos, programa la función de los circuitos integrados y controla todas las tareas que se realizan.

El i8751H incorpora un puerto serial full duplex para manejar las comunicaciones seriales con dos buffers separados, uno para la transmisión y otro para la recepción. El caracter entrante llega al microcontrolador a través del pin 10 y el caracter saliente se envía por el pin 11. La información llega serialmente desde el MAX232 y el i8751H la convierte internamente en un dato paralelo, por lo que la parte serial no necesita elementos adicionales para la conversión serie/paralelo. La Figura 2.17 muestra el C.I. i8751.

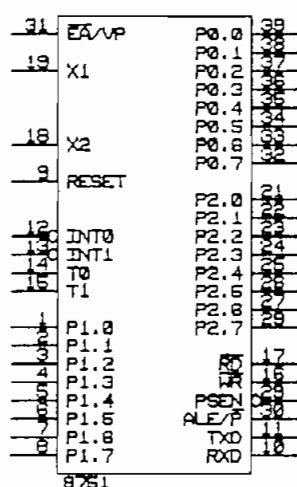


FIGURA 2.17 Microcontrolador i8751H

Con el IEEE-488, el microcontrolador i8751H programa al i8291A y al i8292, leyendo y enviando datos al bus. Estos circuitos integrados son tratados como memoria de datos externa usando los pines RD y WR para leer o

escribir en ellos, y habilitándolos a través de las líneas de dirección. El i8751H controla el intercambio de información y atiende las interrupciones.

Puesto que el i8751H tiene los buses de datos y direcciones multiplexados en tiempo, la separación del bus de direcciones se realiza por medio del latch externo 74LS373, controlado por el i8751H a través de la señal ALE (pin 30). De esta manera se obtiene el byte menos significativo del bus de direcciones (A0 - A7), con la configuración indicada en la Figura 2.18.

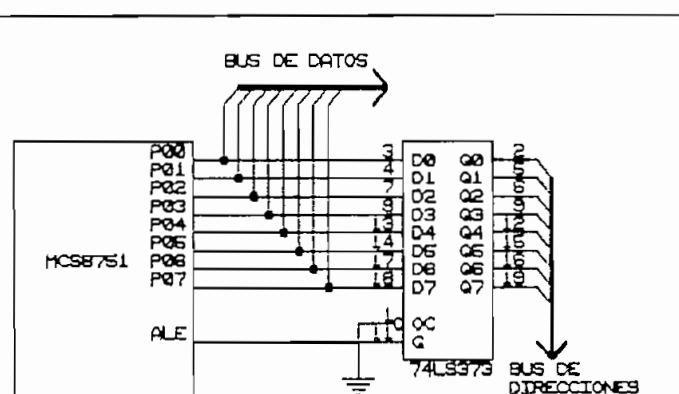


FIGURA 2.18 Multiplexación del bus de datos/direcciones del C.I. i8751H

A través del puerto 2 del i8751H se ingresan los parámetros de operación del IEEE-488 mediante ocho dip-switches, en los que se selecciona la dirección primaria

del dispositivo y otras características adicionales como las funciones "locutor solamente", "oyente solamente".

El procesador genera la señal de reloj de trabajo a través de un oscilador integrado al que se conecta externamente un cristal de frecuencia 7.3728 MHz que permite obtener por software todas las velocidades de la interfaz EIA-RS232C.

2.3.4 FUNCIONAMIENTO DEL SISTEMA

Como se señaló anteriormente es posible seleccionar el funcionamiento del equipo para que trabaje como el controlador del bus o para que trabaje como un dispositivo direccionado a través del bus. Esta selección se la realiza a través de un switch externo conectado al pin 24 (SYC) del i8292. En la Tabla 2.6 se indica la posición del switch y el respectivo estado del sistema.

SW	ESTADO
0	No controlador
1	Controlador del sistema

TABLA 2.6 Operación del equipo

Si se elige que funcione como un dispositivo no

controlador del bus, el circuito integrado i8292 no realiza ninguna función y permanece en un estado de reposo (idle). El equipo puede entonces ser direccionado a través del bus por el controlador o funcionar como un dispositivo "locutor solamente" u "oyente solamente" en un sistema sin controlador, estas características se escogen a través de dipswitches, en los que también se ingresa la dirección primaria del equipo.

Los primeros cinco switches se usan para ingresar la dirección a la cual responderá el equipo, y los dos siguientes para escoger los modos "oyente solamente", "locutor solamente", de acuerdo a la Tabla 2.7.

TON	LON	ESTADO
0	0	Direccionado
0	1	Modo LON
1	0	Modo TON
1	1	Deshabilitado

TABLA 2.7 Características del equipo.

La dirección del dispositivo pueden ser cualquier número de 0 a 30, la dirección 31 en el IEEE-488 se toma como el comando UNL (Nadie Oye).

Operación como controlador del bus

Si el sistema va a ser el controlador del bus el i8291A oye y habla por medio de las líneas de datos y de handshake (NRFD, NDAC y DAV) y el i8292 controla cuatro de las cinco líneas de manejo del bus (IFC, SRQ, ATN y REN). La quinta línea EOI es manejada por el i8291A en transferencias de datos y por el i8292 cuando se realiza una búsqueda en paralelo.

Es necesario inicializar los dos circuitos integrados del siguiente modo:

Para el i8291A

- 1.- Pulsar la entrada RESET o escribir 02H (Comando Reset) al Registro Aux Mode.
- 2.- Escribir a los Registros Interrupt Enable 1 y 2, habilitando o deshabilitando los bits de interrupción según el caso.
- 3.- Escribir al Registro Address Mode para elegir el modo TON para poder enviar los comandos al bus.
- 4.- Deshabilitar las direcciones en el Registro Address 0/1.
- 5.- Escribir el caracter de fin de secuencia EOS si se va a utilizar en la comunicación.

- 6.- Escribir en el Registro Aux Mode las características como enviar y reconocer EOI en una escritura o lectura y para poner el tiempo T1 en el contador interno.
- 7.- Escribir 00H al Registro Aux Mode para liberar al circuito integrado del estado de reset y dejarlo en estado de espera.

Para el i8292 en cambio:

- 1.- Pulsar la señal de reset o enviar el comando RST al Registro Command Field.
- 2.- Habilitar las interrupciones en el Registro Máscara de Error especialmente las interrupciones TCI y SPI.

El i8292 toma el control del bus activando la línea ATN y se autoactiva como controlador, envía entonces la señal IFC por al menos 100 μ s como se define en la norma.

Como el i8291A está en el modo TON (Talk Only), el bit BO en el Registro Interrupt Status 1 se activa tan pronto como se libera al circuito integrado del estado de reset. El microcontrolador escribe el comando al Registro Data Out y espera que se active nuevamente el bit BO para enviar el siguiente comando.

La línea ATN permanece en estado activo hasta que se envíen todos los comandos y el i8292 es puesto en estado de reposo escribiendo el comando GTSB (Go To Standby) en el Registro Command Field. Al reconocer este comando la línea ATN se desactiva y en este momento se puede enviar o recibir datos.

El i8291A se encarga de enviar los comandos y direcciones al bus mientras ATN es verdadera y de enviar y recibir los datos cuando ATN es falsa.

En una transferencia de datos el i8751H controla al i8291A programándole en los modos TON y LON de acuerdo a su participación en la transferencia, porque en estas condiciones no puede ser direccionado a través del bus.

Operación como no controlador

Si el sistema no va a ser el controlador del bus al empezar la comunicación es necesario realizar los siguientes pasos:

El microcontrolador debe leer el puerto 2 para determinar las características como la dirección del equipo, o si se han elegido la funciones "locutor solamente", "oyente solamente" en las que el equipo

funcionará en un sistema sin controlador y por lo tanto no será direccionado a través del bus.

Si el equipo va a ser direccionado a través del bus, entonces el i8751H debe leer la dirección primaria e inicializar al i8291A de la siguiente manera:

- 1.- Pulsar la entrada de RESET o escribir 02H (Comando Reset) al Registro Aux Mode.
- 2.- Escribir a los Registros 1 y 2 de Habilitación de las Interrupciones, habilitando o deshabilitando los bits de interrupción según el caso.
- 3.- Escribir al Registro Address Mode para elegir el Modo 1 de direccionamiento.
- 4.- Escribir al Registro Address 0/1 la dirección primaria del dispositivo que se leyó previamente.
- 5.- Escribir el caracter de fin de secuencia EOS si es necesario.
- 6.- Escribir en el Registro Aux Mode las características requeridas en la comunicación, como cargar el contador interno, enviar el caracter EOS en una escritura, si es necesario, etc.
- 7.- Escribir 00H al Registro Aux Mode para liberar al circuito integrado del estado de reset y lo deja en estado de espera.

El i8291A permanecerá en estado de reposo hasta que el controlador inicie alguna actividad enviando ATN. El controlador direcciona al sistema activando ATN y poniendo el comando MLA (My Listen Address) o MTA (My Talk Address) en el bus. Si los cinco bits más bajos del comando se igualan a la dirección programada en el Registro Address 0/1 del i8291A, el sistema es direccionado para oír o hablar. El bit ADSC en el Registro Interrupt Status 2 indica si el i8291A ha sido direccionado o desactivado. Y los bits TA y LA del Registro Address Mode indican cuándo el sistema es direccionado para oír o hablar.

Si es direccionado para oír, el microcontrolador puede leer el Registro Data In cuando BI se setee en el Registro Interrupt Status 1. En cambio, si se le direccionó para hablar debe esperar que se setee el bit BO en ese registro de estado antes de escribir el dato al Registro Data Out. Si el bit END del Registro Interrupt Status 1 se ha seteado EOI o el byte EOS ha sido recibido y la comunicación debe finalizar.

En cambio si se ha escogido las funciones talk only o listen only el i8291A debe escoger en el Registro Address Mode las funciones ton, lon respectivamente.

2.3.5 DIAGRAMA DE CONEXIONES DEL SISTEMA CONVERTOR DE INTERFACES SERIE - PARALELO.

El diagrama de las conexiones del sistema diseñado se presenta en la Figura 2.18.

CAPITULO III

DESARROLLO DEL SOFTWARE

CAPITULO III

DESARROLLO DEL SOFTWARE

3.1. DESCRIPCION GENERAL DEL SOFTWARE

La operación del sistema de conversión serie - paralelo es controlada por un programa que se ejecuta en el microcontrolador i8751 que se encarga de controlar el flujo de información de cada transferencia de datos y realizar los pasos necesarios para establecer la comunicación.

El diagrama de flujo de la Figura 3.1 presenta el procedimiento que seguirá el programa general del microcontrolador. El programa configura los valores iniciales de la interfaz serial y determina el ritmo binario de la comunicación, para la operación del puerto EIA-RS232C, luego configura las condiciones de operación del bloque IEEE-488.

3.2 MODULO PARA LA PARTE EIA-RS232C

Para el bloque EIA-RS232C se realizaron las subrutinas de inicialización del puerto serial del i8751 en su modo de operación, para transmitir y recibir datos desde la interfaz serial y para establecer el ritmo binario de

la comunicación.

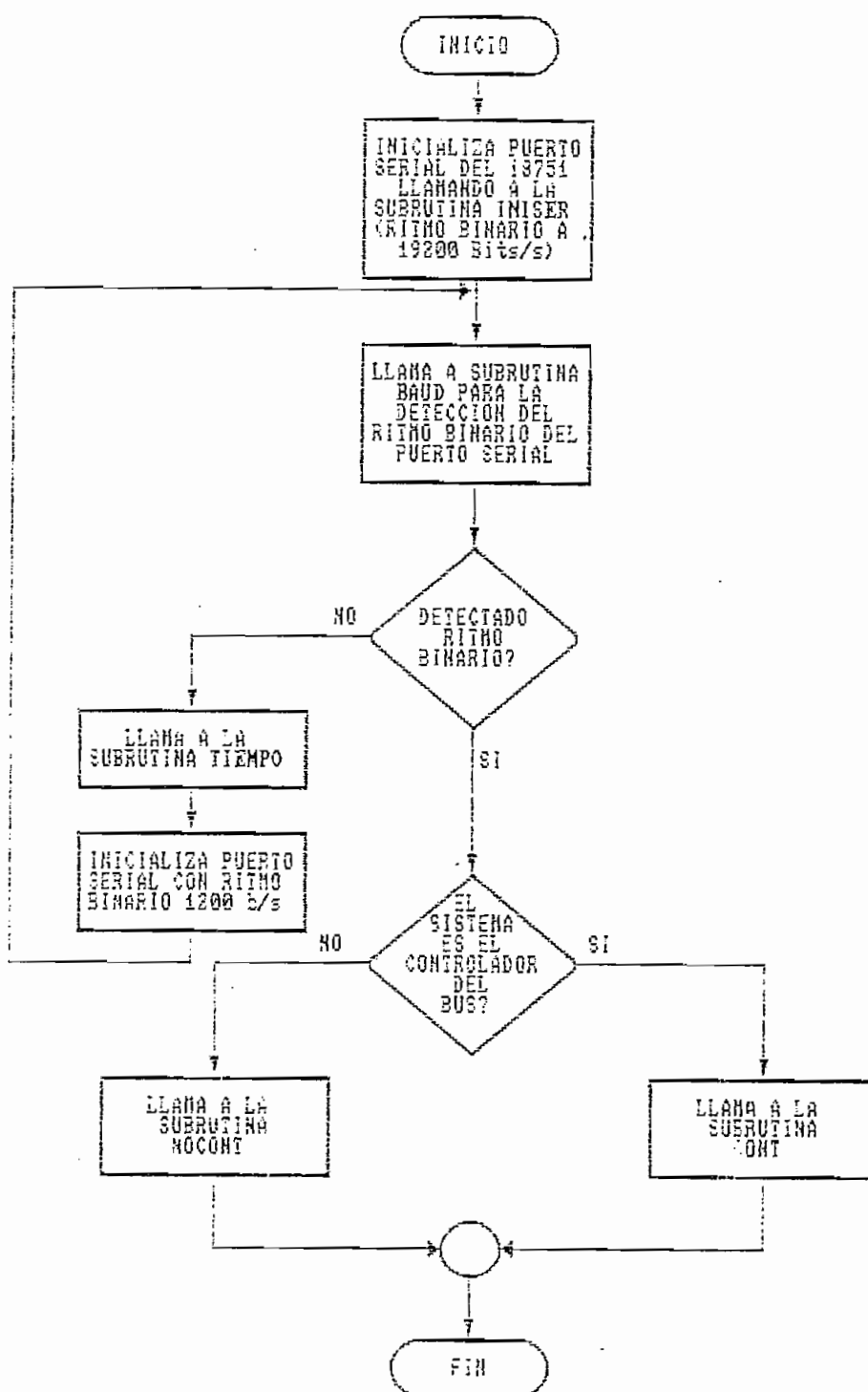


FIGURA 3.1 Diagrama de Flujo del Programa General

3.2.1 INICIALIZACION DEL PUERTO SERIAL

INFORMACION GENERAL:

Antes de recibir o transmitir datos por el puerto serial el microcontrolador debe inicializar los registros que controlan ese puerto, configurándolo a un ritmo binario y en un modo de trabajo determinados. Estos registros de control son: TMOD, SCON, TCON y el Timer 1, los cuales se indican en la Figura 3.2.

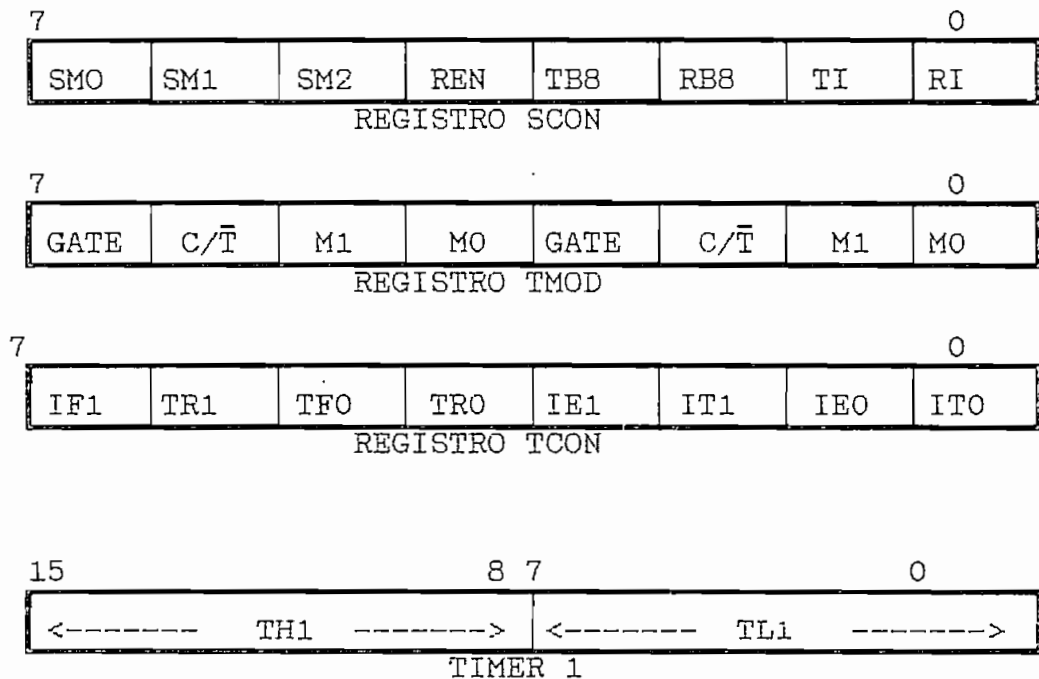


FIGURA 3.2 Registros de Control de la Interfaz Serial

El i8751 tiene cuatro modos de operación para el puerto serial, seleccionados en el Registro SCON (Control del Puerto Serial), este registro se indica en la Figura 3.2.

Donde los bits del registro tienen las siguientes funciones:

SM0 y SM1: Eligen el modo de operación del puerto serial de acuerdo a la especificación de la Tabla 3.1.

SM0	SM1	MODO
0	0	0
0	1	1
1	0	2
1	1	3

TABLA 3.1 Modos de Operación del Puerto Serial del i8751

SM2: Habilita la característica de comunicación de multiprocesador.

REN: Habilita la recepción serial cuando se pone en 1L.

TI y RI: Son las banderas de interrupción en transmisión y recepción respectivamente.

TB8 y RB8: Sirven para almacenar y enviar un bit adicional programable para los modos 2 y 3.

Las características de los modos de operación del puerto serial se indican a continuación:

MODO 0 (Interfaz Serial)

En este modo el puerto serial recibe o transmite un caracter de 8 bits de datos, el bit menos significativo primero y a un ritmo binario fijo a $1/12$ de la frecuencia del oscilador.

MODO 1 (Interfaz Serial)

Cuando el puerto serial trabaja en el modo 1 transmite o recibe un caracter de 10 bits. Un bit de inicio (0), 8 bits de datos y un bit de parada (1). A un ritmo binario variable, generado a través del Timer 1.

MODO 2 (Interfaz Serial)

En el modo 2 el puerto serial transmite o recibe un caracter de 11 bits. Un bit de inicio, 8 bits de datos, un bit programable para la paridad y un bit de parada. El ritmo binario es programable a $1/32$ o $1/64$ de la frecuencia del oscilador.

MODO 3 (Interfaz Serial)

En este modo el puerto serial funciona de manera parecida al Modo 2; sin embargo, en este caso el ritmo binario es variable y generado a través del Timer 1.

El i8751 tiene dos registros Timer/Counter de 16 bits: el Timer 1 y el Timer 0. Estos registros pueden funcionar como temporizadores o como contadores de eventos. En cada Timer pueden direccionarse separadamente los ocho bits menos significativos como un registro de ocho bits TL1 o TL0 y los ocho bits más significativos del Timer como el registro TH1 o TH0. El Timer 1 es usado para generar el ritmo binario de la interfaz serial.

La operación del Timer 1 y del Timer 0 se configura a través del Registro TMOD (Modo del Timer), indicado en la Figura 3.2.

Los cuatro bits menos significativos del Registro TMOD controlan la operación del Timer 0 y los cuatro bits más significativos la operación del Timer 1.

Cada Timer puede ser configurado para operar como Timer o como Counter con el bit C/ \bar{T} que le corresponde en el Registro TMOD. Cuando se escribe un 1L en ese bit, el Timer funciona como Counter y cuando se escribe un 0L fun-

ciona como Timer.

Los Timers pueden operar en cuatro modos diferentes. Los bits M0 y M1 del Registro TMOD (Modo del Timer) permiten seleccionar el modo en que funcionará cada registro de acuerdo a la manera que indica la Tabla 3.2.

SM0	SM1	MODO
0	0	0
0	1	1
1	0	2
1	1	3

TABLA 3.2 Modos de Operación del Timer del i8751

Las características de los modos de operación de los Timer son las siguientes:

MODO 0 (Timer/Counter)

En este modo el Timer 1 y el Timer 0 son configurados como registros de 13 bits. Cuando en la cuenta del Timer se produce un sobreflujo, la bandera de interrupción del Timer se activa. Este registro de 13 bits consiste de los 8 bits de TH y los 5 bits menos significativos de TL, los tres bits más significativos de TL se ignoran.

MODO 1 (Timer/Counter)

El modo 1 funciona de manera parecida al Modo 0; sin embargo, en este caso el Timer se configura como un registro de 16 bits con TH y TL completos.

MODO 2 (Timer/Counter)

El modo 2 configura al Timer 1 o al Timer 0, como un contador de 8 bits (TL) con copia automática de un valor inicial cargado en TH, cuando la cuenta en TL alcanza 255. El sobreflujo de TL no solamente activa la bandera de interrupción del Timer, sino que además TL se carga con el contenido de TH, que debió haber sido inicializado por software.

MODO 3 (Timer/Counter)

El Timer 0 funciona de una manera diferente al Timer 1 cuando se selecciona el modo 3. El Timer 1 en este modo se halla deshabilitado, de una manera similar al caso en que no se activa el bit TR1 en el Registro TCON. En cambio en el Timer 0 TLO y TH0 se configuran como dos contadores separados. TLO usa los bits de control del Timer 0 en el Registro TCON. Y TH0 usa los bits de interrupción y de habilitación del Timer 1 en el Registro TCON.

Cuando se utiliza el Timer 1 para generar el ritmo binario, éste se determina por la relación de sobreflujo

del Timer 1. Si se elige el Modo 2 del Timer la relación de sobreflujo está dada por la Ecuación 3.1. Donde TH1 es el valor que se debe cargar en la parte alta del Timer 1 (TH1).

$$TH1 = 256 - \frac{2^{SMOD} * F_{osc}}{384 * VELOCIDAD} \quad EC 3.1$$

Si SMOD = 0 y una frecuencia del oscilador de 7.3728 MHz se puede obtener la gama de ritmos binario mostradas en la Tabla 3.3.

RITMO BINARIO (bits/s)	VALOR EN TH1 (Hexadecimal)
19200	FF
9600	FE
4800	FC
2400	F8
1200	F0
600	E0
300	C0
150	80

TABLA 3.3 Valores de Autoinicio para el Timer 1

El registro TCON (Control del Timer) se indica en la

Figura 3.2. Este registro controla al Timer 1 con los bits indicados con el subíndice 1 y al Timer 0 con los bits indicados con el subíndice 0.

Los bits del Registro TCON tienen las siguientes funciones:

TFi: Es la bandera de sobreflujo del Timer i. Activada por hardware al producirse el sobreflujo y desactivada por hardware cuando el procesador atiende la interrupción.

TRi: Control de cuenta del Timer i. Si se activa se habilita al Timer para contar.

IEi: Bandera de interrupción en señal externa para el Timer i. Activada por hardware y desactivada cuando la interrupción se procesa.

ITi: Este bit determina si la interrupción externa se generará con un nivel bajo en la señal externa o con una transición de alto a bajo en dicha señal.

* Nota: Todos los bits anteriores son activados con 1L y desactivados con 0L.

DISEÑO DE LA SUBROUTINA DE INICIALIZACION DE LA INTERFAZ SERIAL

La subrutina de inicialización del puerto serial (INISER) cuyo diagrama de flujo se indica en la Figura 3.3, programa la operación de la interfaz serial, para esto realiza los siguientes pasos: escribe en el Registro SCON 50H configurando el puerto serial en el Modo 1 de operación (Un bit de inicio, ocho bits de datos, un bit de parada y un ritmo binario variable generado a través del Timer 1), escribe en el Registro TMOD 20H para configurar al Timer 1 en su función de Timer operando en el Modo 2, escribe en los Registro TH1 y TL1 el número 0FFH como valor inicial de carga del Timer 1 para obtener un ritmo binario de 19200 bits/s, y por último la subrutina activa la cuenta del Timer 1 seteando el bit TR1 en el Registro TCON (Registro de Control del Timer). Para esta aplicación debe deshabilitarse la interrupción de sobreflujo de Timer 1.

3.2.2 DISEÑO DE LA SUBROUTINA PARA TRANSMISION DE DATOS:

La transmisión de datos en el i8751 se inicia con cualquier instrucción que utilice el Registro SBUF como registro de destino. El dato se escribe en el Registro SEUF y el microcontrolador lo envía serialmente por el pin

11. La subrutina envía un dato y espera que se active la interrupción de transmisión TI en el Registro SCON antes de enviar el siguiente dato, esto garantiza que no se enviará un nuevo dato antes de acabar de transmitir el dato anterior. El diagrama de flujo de la subrutina se indica en la Figura 3.4.



FIGURA 3.3 Diagrama de Flujo de la Subrutina INISER

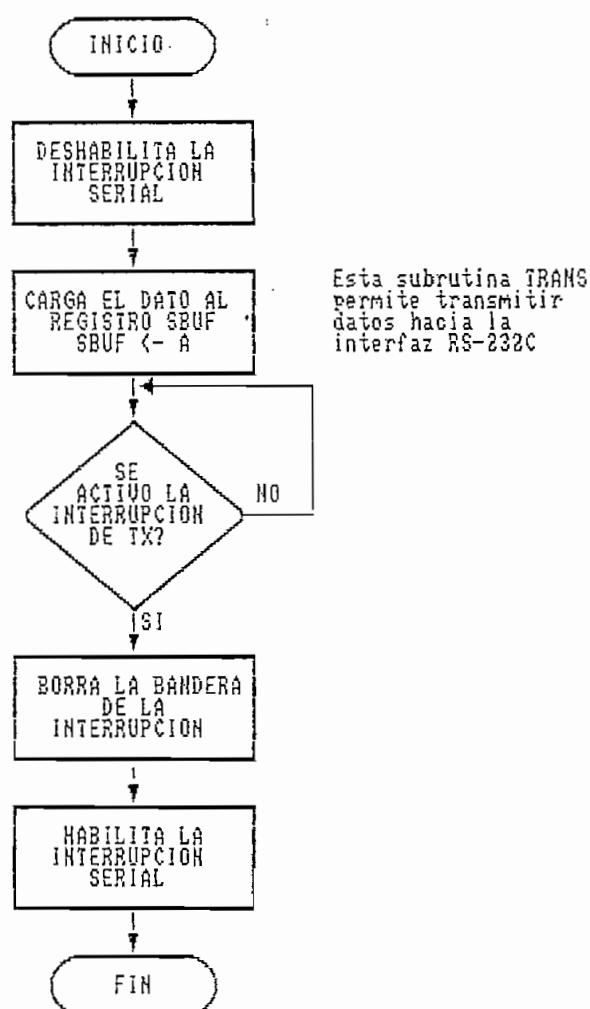


FIGURA 3.4 Diagrama de Flujo de la Subrutina TRANS

3.2.3 DISEÑO DE LA SUBROUTINA PARA RECEPCION DE DATOS:

La recepción en el i8751 se inicia al detectar una transición de 1 a 0 en la línea RXD (pin 10). La subrutina de recepción espera que el dato se haya acabado de recibir antes de leer ese dato desde el Registro SBUF. El

microcontrolador activa el bit de interrupción en recepción RI en el Registro SCON cuando acaba de recibir el dato. El diagrama de flujo de la subrutina se indica en la Figura 3.5.

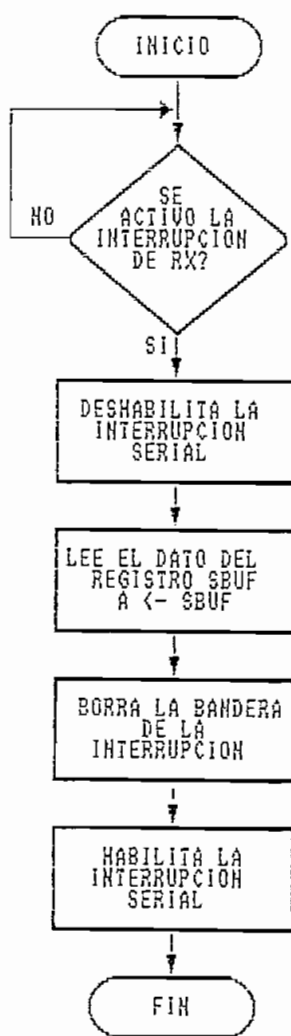


FIGURA 3.5 Diagrama de Flujo de la Subrutina RECEP.

3.2.4 DISEÑO DE LA SUBROUTINA DE DETECCION AUTOMATICA DEL RITMO BINARIO (BAUD RATE).

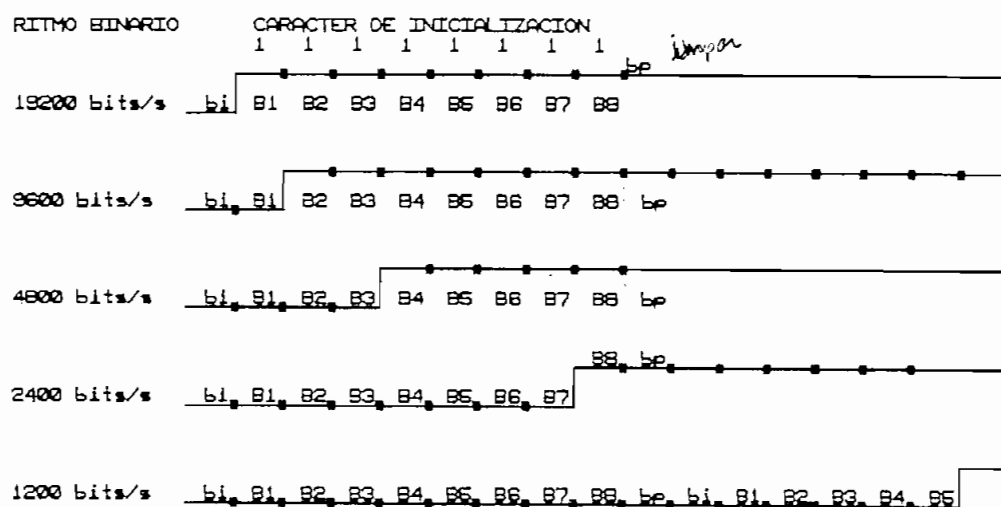
Esta subrutina permite que el microcontrolador detecte automáticamente el ritmo binario de la interfaz serial a partir de un dato enviado al iniciar la operación del sistema por el dispositivo conectado en el otro extremo de la interfaz EIA-RS232C.

Inicialmente se asume que la comunicación se realizará con el ritmo binario más alto posible para el sistema, en este caso se escogió 19200 baudios. La subrutina espera recibir el caracter de inicialización desde la interfaz EIA-RS232C. Si el dispositivo envía el dato a 19200 bits/s el caracter es recibido correctamente; pero si el ritmo binario es menor que el asumido, el caracter se recibe alargado.

Al muestrear el microcontrolador el dato a un ritmo binario mayor que el verdadero, parte del bit de inicio que es cero, se lee como si fueran bits de datos. El número de bits cero que se leen como datos depende del ritmo binario.

Para distinguir cuando se acaba el bit de inicio (0L) y comienzan los bits de datos, es necesario que el bit

menos significativo del carácter recibido sea 1L o lo que es lo mismo que el carácter sea impar ya que el primer bit recibido es el bit menos significativo y adicionalmente para ésta subrutina el carácter enviado debe tener todos sus bits en 1L, es decir el carácter debe ser FFH. La Figura 3.6 indica un diagrama de lo que ocurre con un carácter cuando se recibe a diferentes ritmos binarios.



*Nota: bi - bit de inicio
 bp - bit de parada
 Bj - bit de datos ($1 < j < 8$)

FIGURA 3.6 Recepción del Carácter de Inicialización a Diferentes Ritmos Binarios

En la Tabla 3.4 se indica la relación entre el ritmo binario y el dato recibido a diferentes velocidades a

partir del envío del caracter de inicialización (FFH) y adicionalmente el valor de autoinicio que se debe escribir en el TH1 para conseguir ese ritmo binario.

Por los valores de la Tabla 3.4 se puede observar que existe un problema en la detección del ritmo binario cuando éste es igual o menor que 1200 bits/s, ya que en esos casos el dato recibido es igual a cero. La subrutina realiza dos procesos: el uno para cuando el ritmo binario está entre 19200 bits/s y 2400 bits/s y el otro cuando está entre 1200 bits/s y 150 bits/s. Determinando cual de los procesos se realiza al detectar si el dato recibido es o no igual a cero.

RITMO BINARIO (bits/s)	DATO RX (DE FFH)	VALOR EN TH1 (Hexadecimal)
19200	FF	FF
9600	FE	FE
4800	F8	FC
2400	80	F8
1200	00	F0
600	00	E0
300	00	C0
150	00	80

TABLA 3.4 Valores para la Subrutina BAUD_RATE

En el programa general se asume un ritmo binario de 19200 para la interfaz serial. La subrutina BAUD_RATE cuyo diagrama de flujo se indica en la Figura 3.7, se encarga de la detección automática del ritmo binario, realizando los siguientes pasos: espera la llegada de un caracter por el puerto serial, cuando ha recibido el caracter comprueba si es diferente de cero, si eso se cumple, compara éste dato con el valor FFH si no es igual la velocidad es diferente de 19200 bits/s por lo que salta a comparar el dato recibido con los diferentes valores indicados en la Tabla 3.4 y de acuerdo a esto cargar en TH1 el valor correspondiente de autoinicio.

En cambio si el caracter recibido es igual a cero sale de la subrutina. En éste caso el programa sabe que el ritmo binario es menor que 2400 bits/s; por lo tanto espera el tiempo suficiente para que se limpie el puerto y lo reinicializa a un ritmo binario de 1200 bits/s, y luego llama nuevamente a ésta subrutina BAUD_RATE que espera la llegada de un nuevo caracter. Esta última parte del programa necesita que el equipo conectado a la interfaz EIA-RS232C envíe nuevamente el caracter de inicialización para su adecuado funcionamiento.

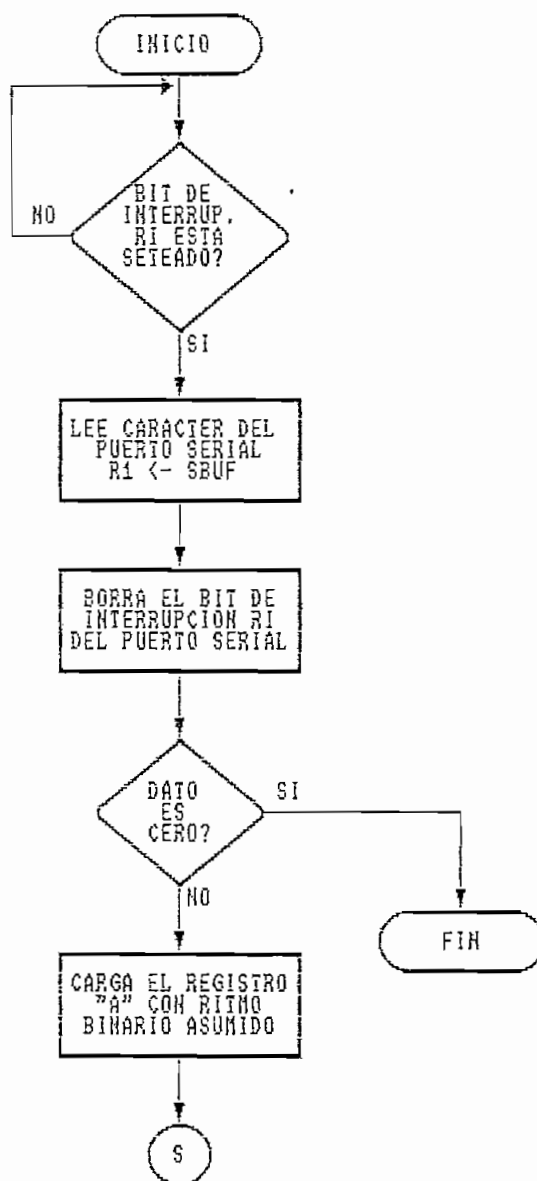


FIGURA 3.7 Diagrama de Flujo de la Subrutina BAUD_RATE

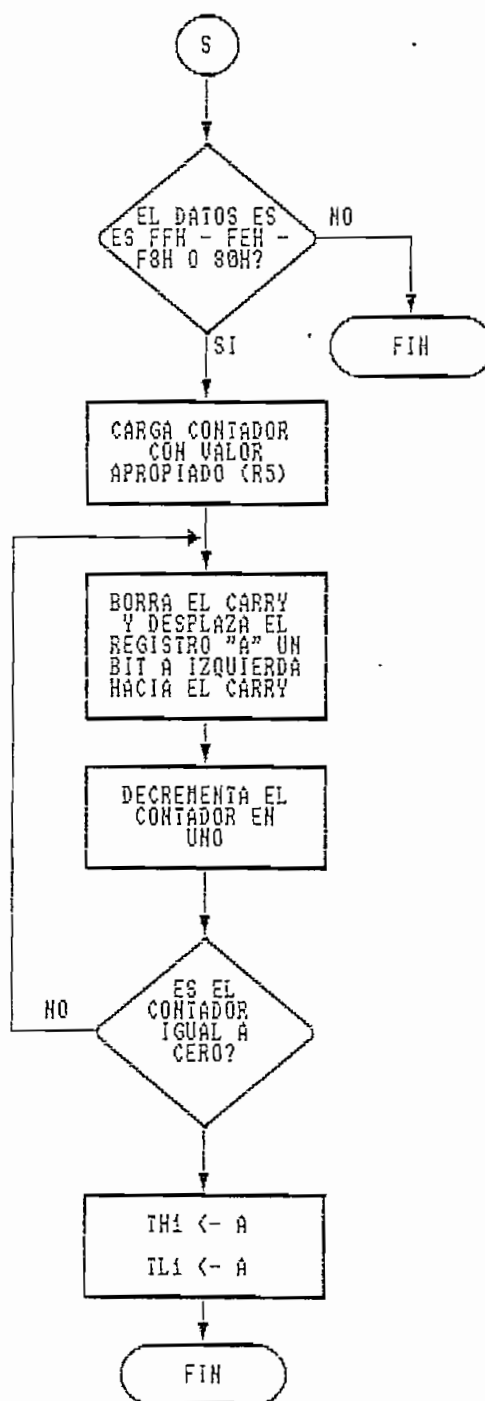


FIGURA 3.7 Continuación

3.3 MODULO PARA EL BLOQUE IEEE-488

Para el bloque IEEE-488 se realizaron dos grupos de subrutinas:

- a) Subrutinas para cuando el sistema no es el controlador del bus.
- b) Subrutinas para cuando el sistema es el controlador del bus.

Para el primer caso se pueden dar también dos formas de funcionamiento:

- i) Existe controlador conectado al bus por lo que el dispositivo espera ser direccionado para hablar, oír, y responder a una búsqueda en serie o en paralelo.
- ii) No existe controlador en el bus y el dispositivo puede funcionar como locutor solamente o como oyente solamente.

Para el segundo caso el sistema se encarga de enviar comandos y direcciones a los dispositivos en el bus. Y puede también realizar las funciones de locutor, oyente; además de realizar otras funciones propias de control del bus.

Los dos grupos de programas se agrupan en las subrutinas NO_CONT y CONT.

3.3.1 DISEÑO DE LA SUBROUTINA NO_CONT

Esta subrutina tiene el diagrama de bloques indicado en la Figura 3.8 y se ejecuta cuando el equipo no es el controlador del bus. La subrutina inicializa al circuito integrado i8291A, escribiendo el comando de reset (02H) en el Registro Aux Mode. Luego lee el puerto 2 del microcontrolador i8751 en el que se ingresan las características de operación del sistema como son: la dirección primaria del equipo y el estado de los pines de selección de las funciones "Locutor solamente/oyente solamente".

Cuando se han elegido las funciones "locutor solamente" u "oyente solamente" la subrutina no toma en cuenta la dirección primaria del sistema ya que en ese caso no será direccionado por un controlador.

Al elegir la función "Locutor solamente", la subrutina programa al i8291A en el modo TON (Talk Only) de direccionamiento escribiendo 80H en el Registro Address Mode y espera hasta que haya un oyente activo antes de comenzar la comunicación. El bit ERR del Registro "Inte-

rrupt Status 1" del circuito integrado i8291A en 1L indica que no hay oyentes activos. Una vez que se ha desactiva ese bit la subrutina esperar que se active el bit B0 del Registro "Interrupt Status 1" que indica que ya puede escribir el dato al Registro Data Out que se encarga de enviarlo al bus. Si ya se han acabado de enviar todos los datos sale de la subrutina, en caso contrario debe volver a esperar la interrupción B0 para enviar el nuevo dato.

Si se escoge la función "oyente solamente", el i8291A debe ser programado en el modo LON (Listen Only) escribiendo 40H al Registro Address Mode. La subrutina espera que se active el bit BI en el Registro Interrupt Status 1, que le indica la llegada de un dato que debe leer desde el Registro DATA IN, cuando se detecta el mensaje END, la subrutina acaba la recepción de datos.

Si se eligen al mismo tiempo las funciones "locutor solamente" y "oyente solamente" el sistema indicará la ocurrencia de un error, porque en ese caso sería necesaria la presencia de un controlador para determinar qué función le toca realizar en un momento cualquiera, como esa opción ya está considerada en el funcionamiento del sistema con un controlador, ésta opción no es válida y produce un error.

Cuando el sistema va a ser conectado a un controlador, la subrutina inicializa los registros internos del i8291A configurándolo al Modo 1 de Direcccionamiento cargando en el Registro Address Mode el número 01H, escribe en el Registro Address 0/1, la dirección primaria obtenida en el puerto 2 del i8751 y después habilita en el Registro Auxiliar B la recepción de comandos indefinidos, ésto es necesario para responder a una búsqueda en paralelo.

El sistema espera que se active un bit en los Registros "Interrupt Status 1" o "Interrupt Status 2" y determina la función que va a realizar. De manera que si se activa el bit ADD en el Registro "Interrupt Status 2" el sistema sabe que ha sido direccionado por el controlador del bus, por lo que debe determinar si se le ha seleccionado como locutor o como oyente mediante los comandos MTA (My Talk Address) o MLA (My Listen Address) y en ese caso debe realizar las funciones de enviar o recibir datos.

Los otros bits de los registros pueden activarse cuando el controlador envía un comando como SDC (Select Device Clear) que le indica que el sistema debe reinicializarse, esto es volver a un estado inicial determinado, puede ser puesto en estado remoto si llega su

dirección de oyente y luego el comando REN (Habilitado control remoto), o volver al estado local al recibir el comando LOC (Control local). Finalmente se detecta también si se ha recibido un comando indefinido para responder a una búsqueda en paralelo.

3.3.2 DISEÑO DE LA SUBROUTINA CONT

En la Figura 3.9 se indica el diagrama de flujo de la subrutina CONT, ejecutada cuando el sistema va a operar como el controlador del bus, en ese caso la subrutina realiza los siguientes pasos: inicializa los circuitos integrados i8291A e i8292, enviando los comandos de reset al Registro Aux Mode del i8291A y al Registro Command Field del i8292, inicializa los Registros "Interrupt Status 1" e "Interrupt Status 2" del C.I. i8291A y programa su funcionamiento en el modo TON (Talk Only) escribiendo 80H en el Registro Address Mode, carga en el contador interno del i8291A la frecuencia de reloj externo de 1 Mhz escribiendo 21H en el Registro Aux Mode. Habilita la recepción de comandos indefinidos escribiendo en el Registro Auxiliar B el número 0A1H y por último escribe 00H en el Registro Aux Mode para liberarlo del estado de reset y dejarlo en estado de espera.

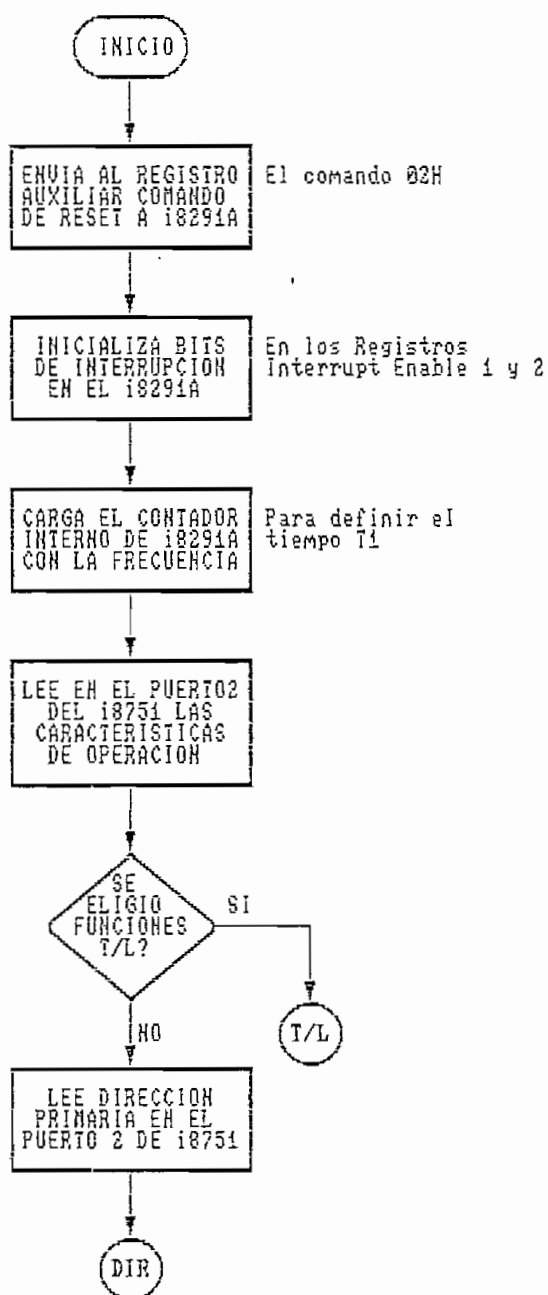


FIGURA 3.8 Diagrama de Flujo de la Subrutina NOCONT

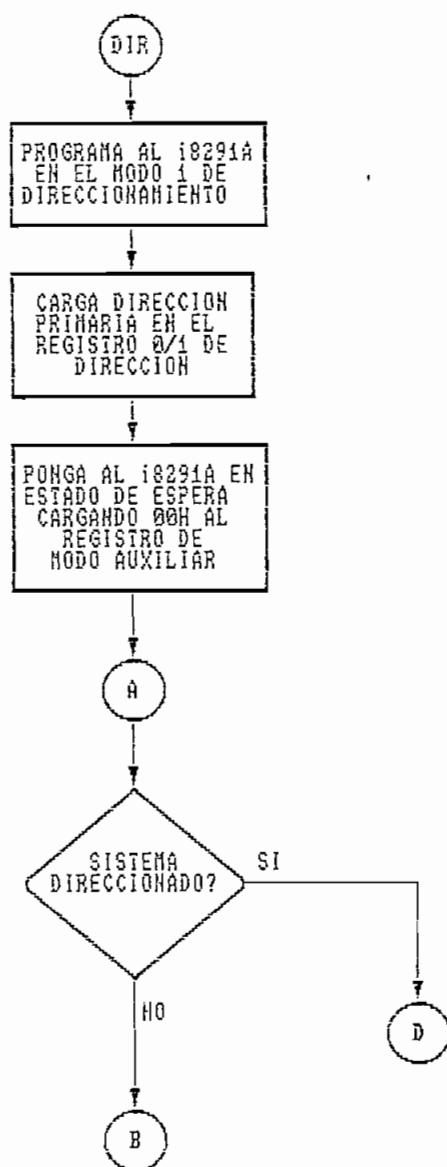


FIGURA 3.8 Continuación

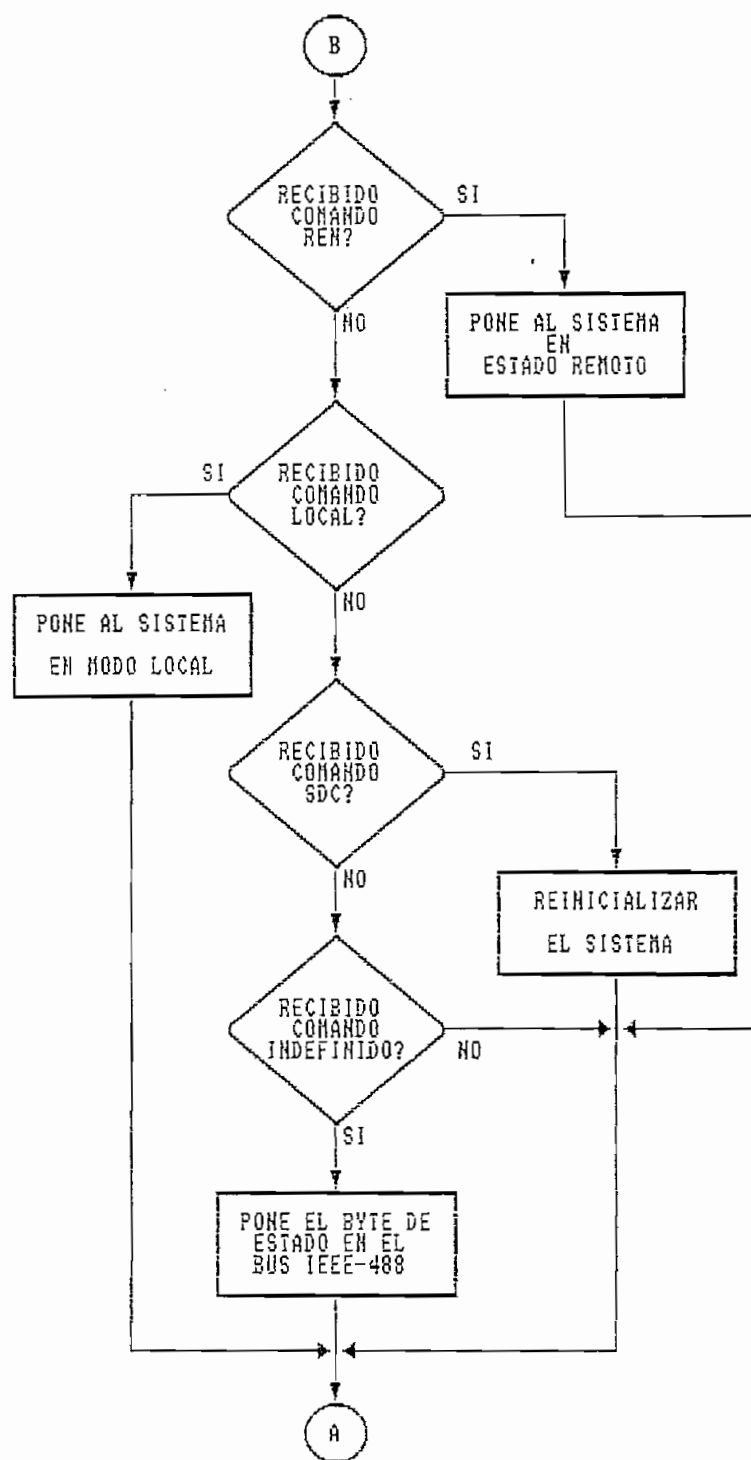


FIGURA 3.8 Continuación

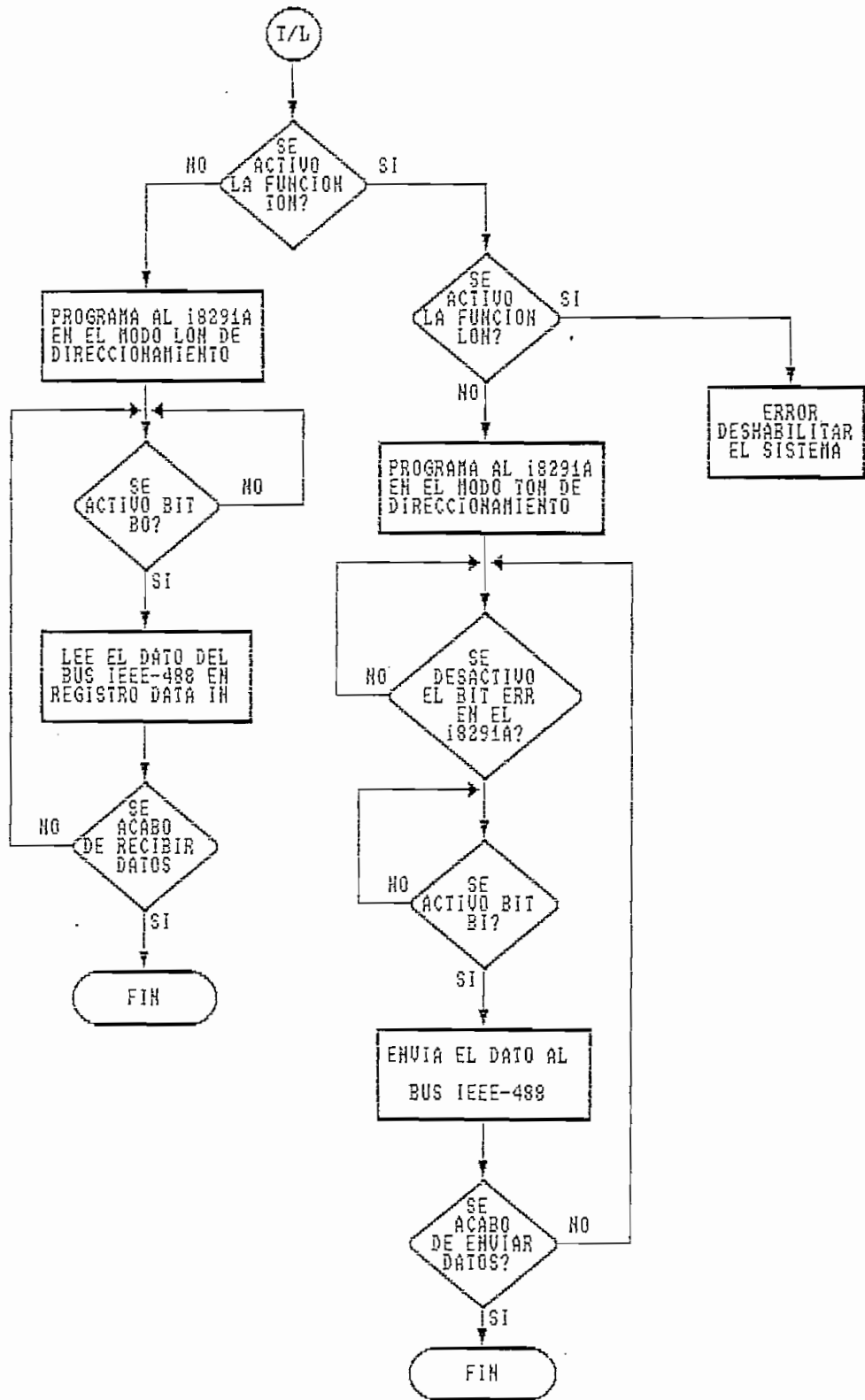


FIGURA 3.8 Continuación

Después del proceso de inicialización la subrutina espera un caracter desde la interfaz EIA-RS232C que le indicará si la comunicación va a usar el caracter EOS para finalizar la transferencia de una cadena de datos.

Si la comunicación va a finalizar con el caracter EOS, entonces la subrutina espera recibir ese caracter desde el puerto serial y lo escribe en el Registro EOS del i8291A, además habilita en el circuito integrado características como: activar la línea EOI al enviar el caracter EOS al final de una cadena de datos cuando el sistema es el locutor y activar el bit END en el Registro Interrupt Status 1 al recibir el caracter EOS al final de una cadena de datos cuando el sistema es el oyente, escribiendo el número 8CH en el Registro Auxiliar A.

En cambio cuando no se va a utilizar el caracter EOS el final de una comunicación se indicará -activando la línea EOI que es el mensaje END.

Una vez determinadas las características de la comunicación la subrutina espera recibir un caracter desde la interfaz EIA-RS232C indicándole la función del bus que se debe realizar.



FIGURA 3.9 Diagrama de Flujo de la Subrutina CONT

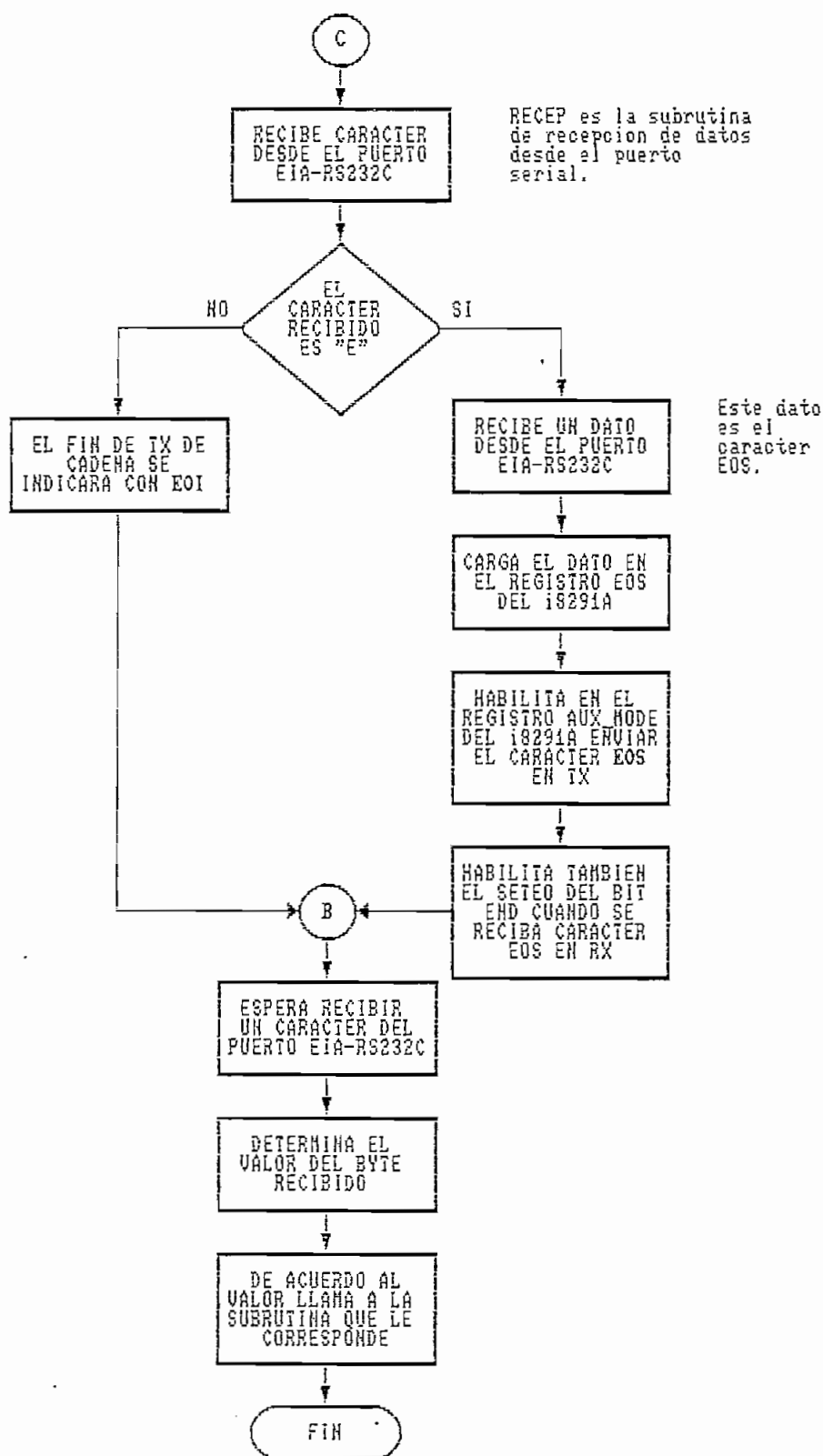
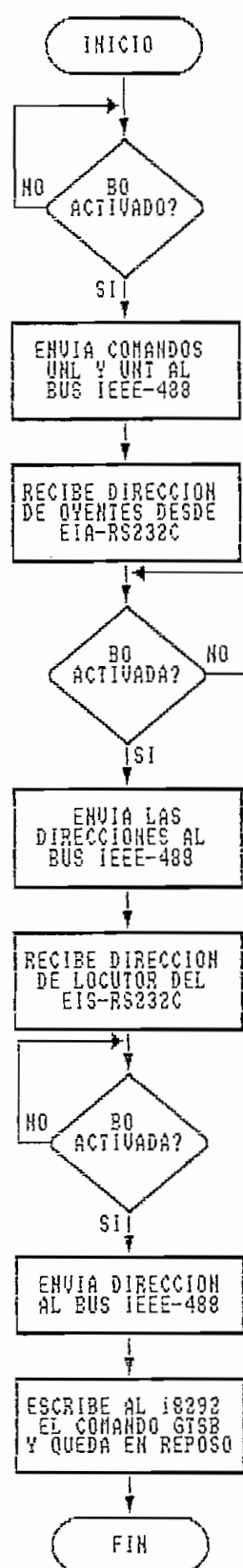


FIGURA 3.9 Continuación

DISEÑO DE LA SUBROUTINA DIRECC

Antes de comenzar la comunicación, el controlador debe enviar los comandos de direccionamiento a los dispositivos conectados al bus, ésta información se envía con la subrutina DIRECC cuyo diagrama de flujo se indica en la Figura 3.10. La subrutina espera que el equipo conectado a la interfaz EIA-RS232C le envíe las direcciones del locutor y de los oyentes, entonces el i8291A se encarga de enviar esas direcciones al bus mientras el i8292 mantiene activada la línea ATN.

La subrutina realiza los siguientes pasos: Envía a la interfaz IEEE-488 los comandos UNL (Unlisten) y UNT (Untalk) para deshabilitar al locutor y a los oyentes anteriores. Recibe desde la interfaz EIA-RS232C la dirección del locutor, espera que se active el bit B0 en el Registro "Interrupt Status 1" del i8291A y luego envía la dirección al bus IEEE-488. Espera las direcciones de los oyentes y luego de comprobar el estado del bit B0 las envía al bus. Y por último escribe en el Registro Command Field del i8292 el comando GTSB (Go To Standby) para liberar la línea ATN dejando al integrado en reposo. La subrutina DIRECC es utilizada en las siguientes subrutinas: ENVIAR, RECIBIR y TRANSF.



Con esto es posible
enviar datos en
lugar de comandos
al GPIB

FIGURA 3.10 Diagrama de Flujo de la Subrutina DIRECC

Las funciones que puede realizar el sistema controlador del bus se describen a continuación:

a) DISEÑO DE LA SUBROUTINA ENVIAR

El diagrama de flujo de la subrutina ENVIAR se indica en la Figura 3.11. Esta función permite que el sistema envíe datos al bus, actuando en este caso como locutor del bus IEEE-488.

Antes de comenzar a enviar datos al bus deben direccionarse los dispositivos que intervendrán en la comunicación para lo que se llama inicialmente a la subrutina DIRECC. Después de enviar los comandos de dirección, la subrutina pone al i8292 en estado de reposo para que libere la línea ATN y el i8291A puede enviar los datos. La subrutina ENVIAR escribe los datos al bus, estos datos llegan al equipo a través de la interfaz serial y son enviados al bus por el Registro DATA OUT del i8291A, la subrutina acaba la transmisión de datos cuando recibe desde la interfaz EIA-RS232C el caracter ESC.

Cuando acaba la transferencia de datos la subrutina escribe en el Registro Command Field del i8292 el comando TCSY para que éste tome el control del bus y active la línea ATN y envía al bus IEEE-488 los comandos UNT y UNL.

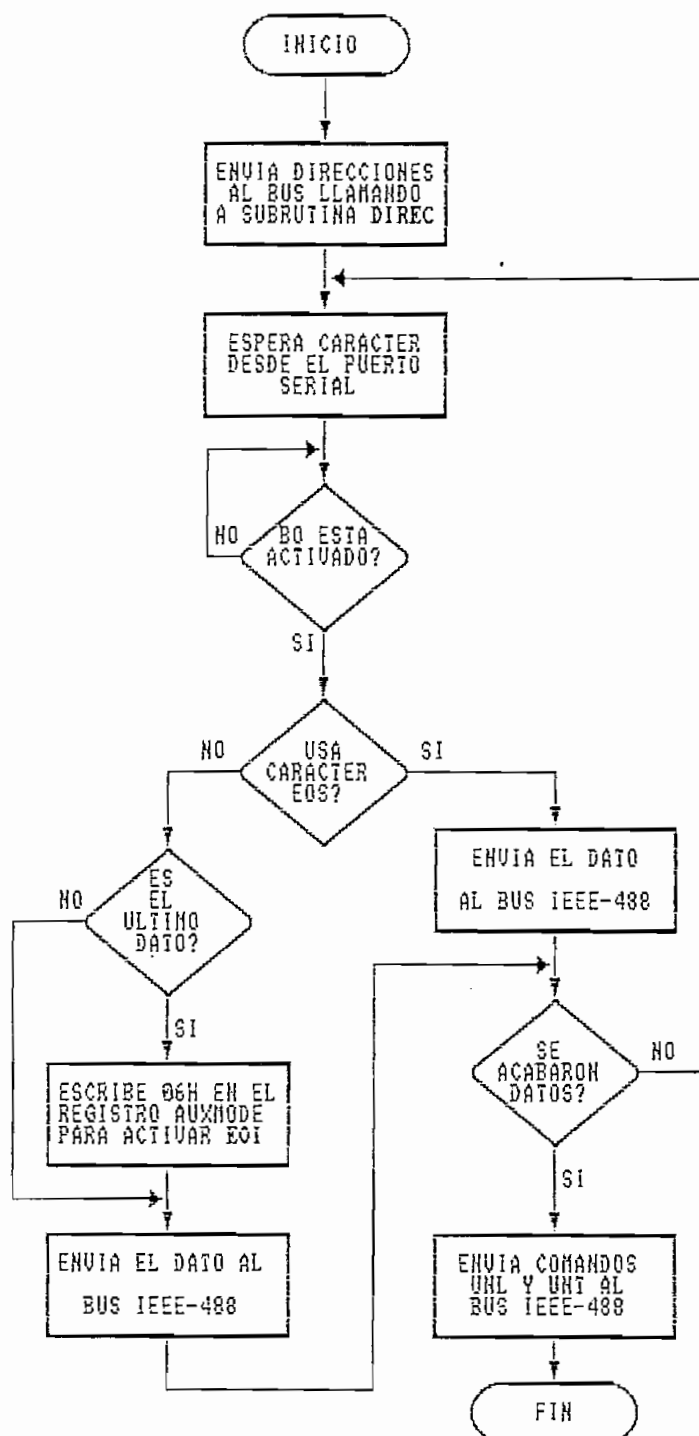


FIGURA 3.11 Diagrama de Flujo de la Subrutina ENVIAR

b) DISEÑO DE LA SUBROUTINA RECIBIR

La Figura 3.12 presenta el diagrama de flujo de la subrutina RECIBIR. Esta función se utiliza para leer datos desde el bus IEEE-488 hacia la interfaz EIA-RS232C. En este caso el sistema es uno de los oyentes o el único oyente. El controlador envía la dirección del locutor y las direcciones de los oyentes al bus, esta información se transfiere con la subrutina DIRECC.

Una vez que el controlador ha enviado los comandos de direccionamiento, debe liberar la línea ATN antes de recibir datos desde el bus, por lo que pone al i8292 en estado de reposo para que desactive la línea ATN, escribiendo el comando GTSB en el Registro Command Field. El i8291A. recibirá los datos desde el Registro Data In al detectar que el bit BI en el Registro Interrupt Status 1 está activado. La comunicación termina al recibir el caracter EOS o al detectar la activación de la línea EOI desde el bus IEEE-488.

El microcontrolador i8751 escribe el comando TCSY en el Registro Command Field del i8292, para que tome el control y active la línea ATN cuando ha finalizado la comunicación. El C.I.i8291A es puesto en el modo TON para enviar los comando UNL y UNT al IEEE-488.

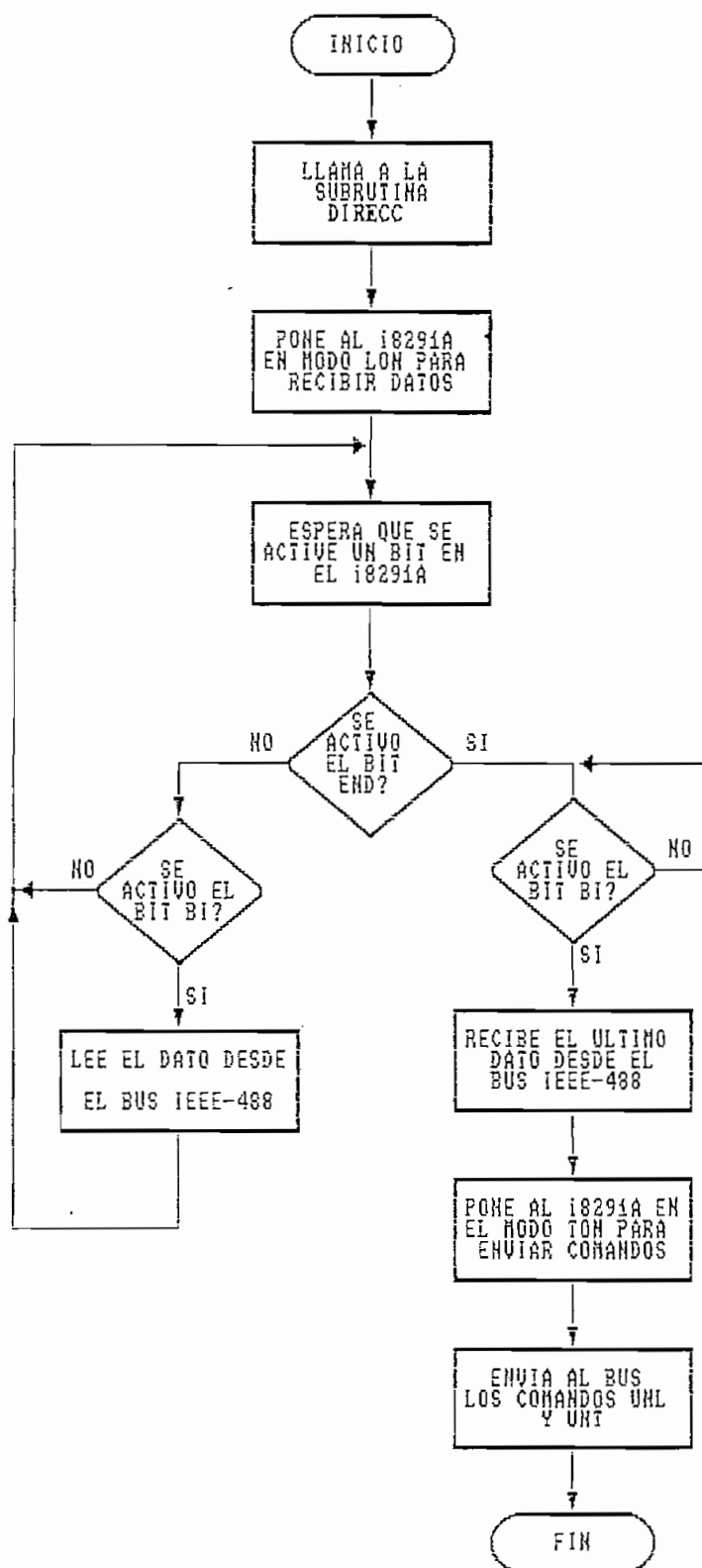


FIGURA 3.12 Diagrama de Flujo de la Subrutina RECIBIR

c) DISEÑO DE LA SUBROUTINA TRANSFERIR

El diagrama de flujo de la subrutina TRANSF se indica en la Figura 3.13. Esta función permite realizar una comunicación en la que no participa el controlador como locutor ni como oyente. El controlador envía la dirección del locutor y las direcciones de los oyentes con la subrutina DIRECC al bus IEEE-488 y se pone en estado de espera hasta que la comunicación se termine. El i8292 va a standby al recibir el comando GTSB al acabar de enviar los comandos de direccionamiento.

Con la subrutina TRANSFERIR se configura al C.I. 8751A al modo Lon (Listen Only) en un modo de "ciclo de handshake acceptor continuo" escribiendo 83H en el Registro Auxiliar A, esta característica permite que el i8291A no participe en la transferencia de información, pero pueda detectar el final de la comunicación al reconocer el caracter EOS o la activación de la línea EOI.

Cuando el controlador detecta que la transferencia de datos finalizó, escribe en el Registro Command Field del i8292 el comando TCSY para que éste tome el control del bus y active la línea ATN. Entonces la subrutina programa al circuito integrado i8291A en el modo TON escribiendo 80H en el Registro Address Mode y envía los comandos UNL y

UNT al bus.

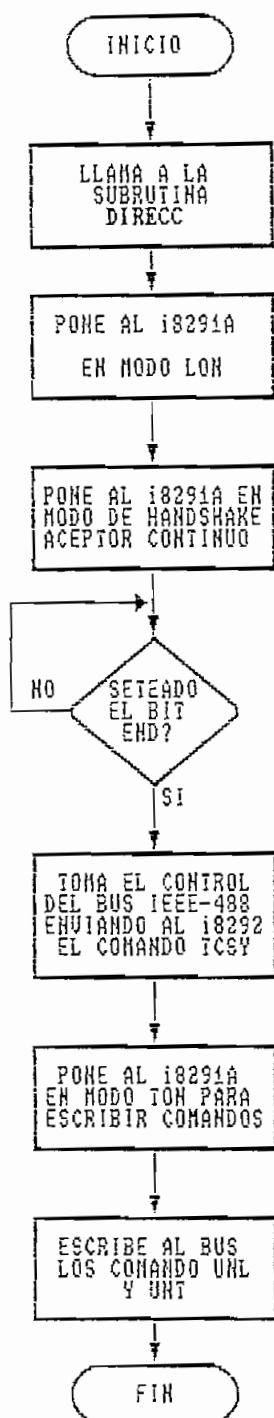


FIGURA 3.13 Diagrama de Flujo de la Subrutina TRANSF

d) DISEÑO DE LAS SUBROUTINAS PARA LA BÚSQUEDA EN PARALELO

Antes de realizar una búsqueda en paralelo el controlador debe configurar los dispositivos para responder a la petición de estado en una línea de datos DIOi. La subrutina CONFIG realiza esta configuración y debe ser llamada antes que se ejecute la subrutina que inicia la búsqueda en paralelo PPOLL.

- DISEÑO DE LA SUBROUTINA CONFIG

El diagrama de Flujo para esta subrutina se indica en la Figura 3.14. Para la configuración de los dispositivos el controlador debe tener el control del bus, es decir el i8292 debe estar en estado activo para que la línea ATN también este activa. El sistema envía el comando PPC (Configure Búsqueda en Paralelo) luego direcciona un dispositivo como oyente y le envía el comando PPEi, donde i es el número de la línea de datos que usará el dispositivo para responder a una búsqueda en paralelo. Direcciona al siguiente dispositivo y continúa el proceso para configurar hasta a 8 dispositivos en el bus a la vez.

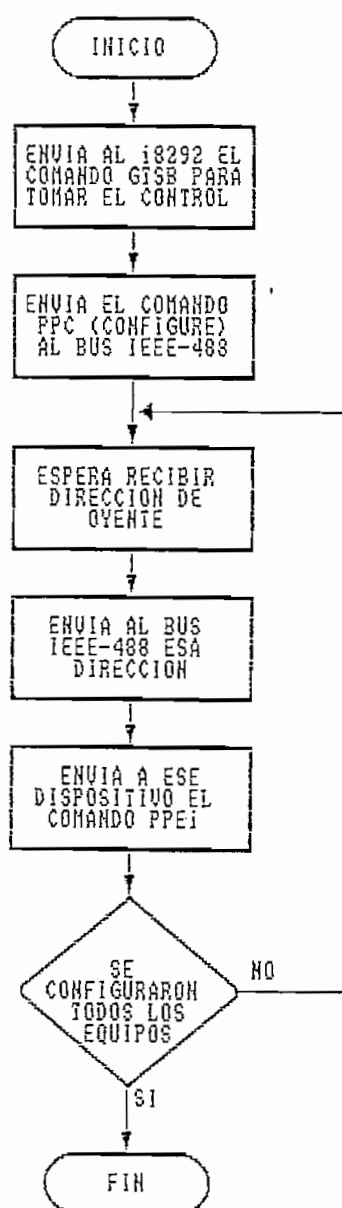


FIGURA 3.14 Diagrama de Flujo de la Subrutina CONFIG

- DISEÑO DE LA SUBROUTINA PPOLL

La subrutina PPOLL cuyo diagrama de flujo se indica en la Figura 2.15, asume que la configuración de los

dispositivos ya fue realizada. El microcontrolador pone al i8291A en el Modo LON escribiendo en el Registro Address Mode 40H, y envía al Registro Command Field del i8292 el comando EXPP (Realice Búsqueda en Paralelo), lo que hace que las líneas ATN y EOI se activen simultáneamente indicando a los dispositivos que deben poner en la línea de datos correspondiente su estado. El i8291A leerá ese dato desde el Registro DATA OUT. Para acabar la búsqueda el sistema envía el comando PPU (Desconfigure Búsqueda en Paralelo).

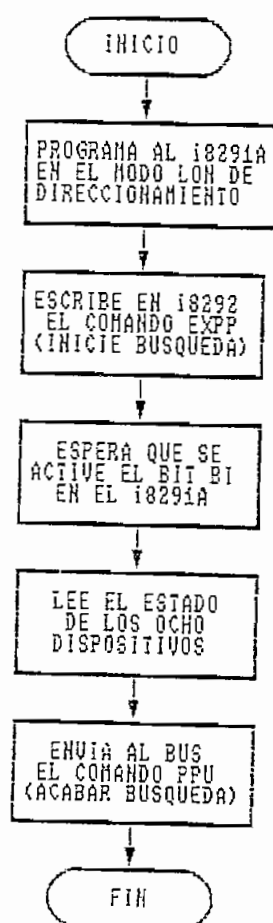


FIGURA 3.15 Diagrama de Flujo de la Subrutina PPOLL

e) DISEÑO DE LA SUBROUTINA TCONTROL

Esta función permite que el sistema se convierta en el controlador activo del bus, transfiriendo el control desde el actual controlador activo hacia el sistema. La Figura 3.16 presenta el diagrama de flujo de la subrutina TCONTROL. Para que la transferencia de control pueda realizarse el sistema no debe ser el controlador activo del bus, ya que no tiene sentido que se transfiera el control a sí mismo.

Para proceder a transferir el control, el actual controlador activo envía el comando TCT y el sistema esperar ser direccionado para hablar por parte del actual controlador activo, al detectar este direccionamiento, el i8291A es configurado al modo TON escribiendo 80H al Registro Address Mode, se deshabilita la dirección primaria y se deja al circuito integrado en estado de espera.

El C.I. i8292 que hasta ese momento no estuvo funcionando se activa para que tome el control del bus IEEE-488, el i8751 escribe el comando TCNTR en el Registro Command Field del i8292 informándole que debe tomar el control del bus. El C.I. i8292 espera a que el actual controlador activo libere la línea ATN para poder tomar el control, si esto no sucede la transferencia del control no

puede producirse.

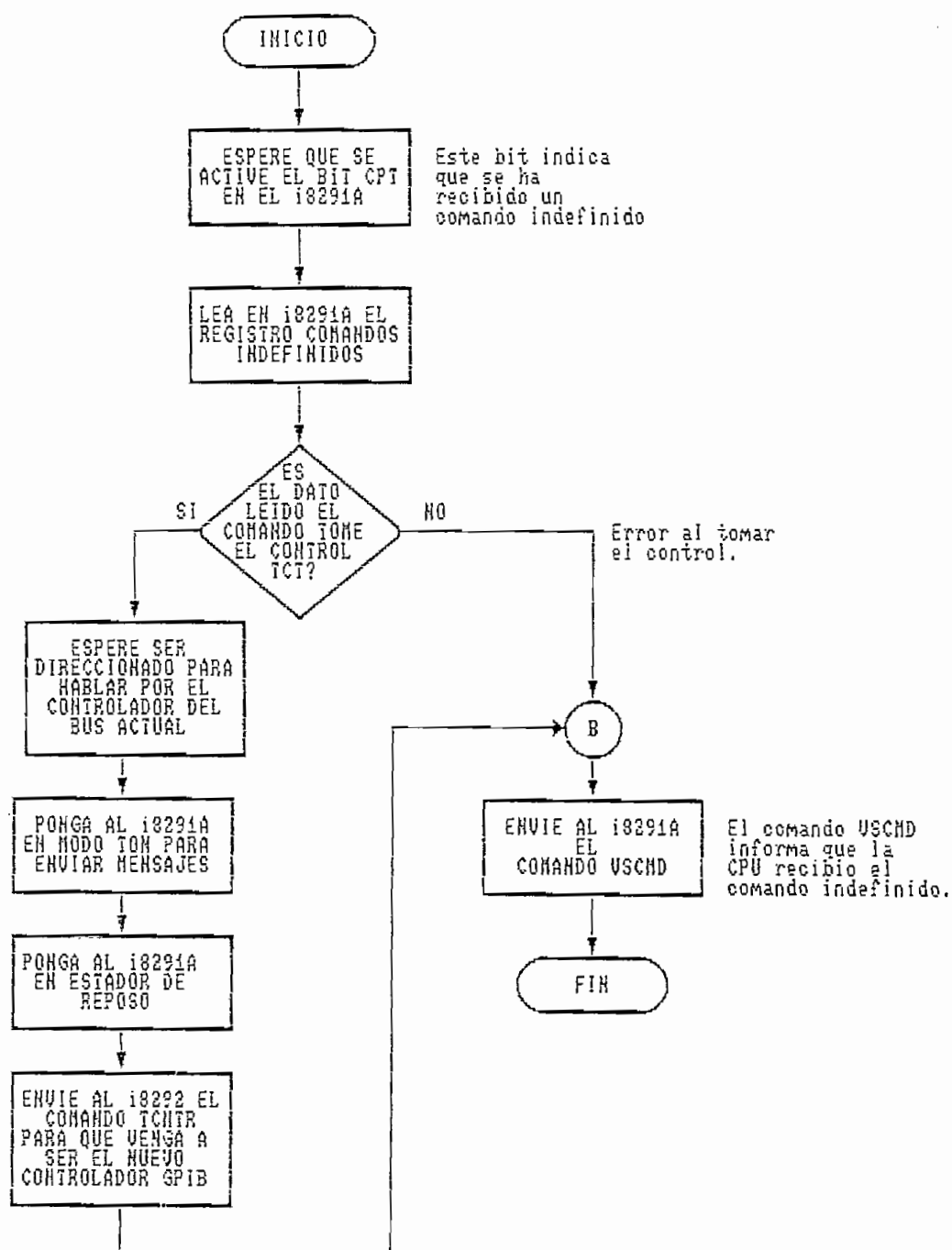


FIGURA 3.16 Diagrama de Flujo de la Subrutina
TCONTROL

f) DISEÑO DE LA SUBROUTINA PCONTROL

Esta función permite que el sistema pueda transferir el control del bus a otro dispositivo conectado al bus IEEE-488 que tiene incorporadas las funciones de control del bus. Esta subrutina solamente puede ejecutarse si el sistema es el controlador activo. La Figura 3.17 presenta el diagrama de flujo de la subrutina PCONTROL.

Para que se realice la transferencia del control, el sistema debe enviar al dispositivo que tomará el control el comando TCT y lo direccionará como locutor. En tanto se programa al i8291A en el Modo 1 de direccionamiento escribiendo en el Registro Address Mode el número 01H, y en el Registro Address 0/1 la dirección primaria del sistema y se dejará al circuito integrado en estado de espera.

La subrutina envía al i8292 el comando GIDL (Go To Idle) para que libere la línea ATN, con lo que el dispositivo al que se transfiere el control tomará el control del bus IEEE-488 y vendrá a ser el nuevo controlador activo.



FIGURA 3.17 Diagrama de Flujo de la Subrutina PCONTROL

g) DISEÑO DE LA SUBROUTINA DISPD

La función DISPD permite enviar el comando GET (disparo) a un grupo de dispositivos a los que se ha direccionado como oyentes. Para esto el controlador envía las direcciones de los oyentes y a continuación el comando GET todo esto con la línea ATN activada. El diagrama de flujo de la subrutina DISP se indica en la Figura 3.18.

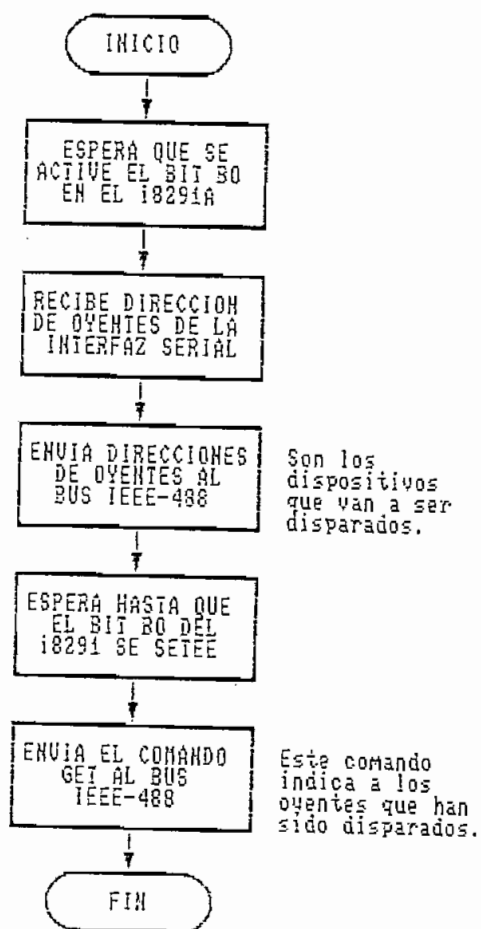


FIGURA 3.18 Diagrama de Flujo de la Subrutina DISPD

h) DISEÑO DE LA SUBROUTINA LIMDS

Permite enviar el comando SDC (reinicializar dispositivos seleccionados) a un grupo de dispositivos direccionados como oyentes. La subrutina envía las direcciones de los oyentes y luego el comando SDC. El diagrama de flujo de la subrutina se indica en la Figura 3.19.

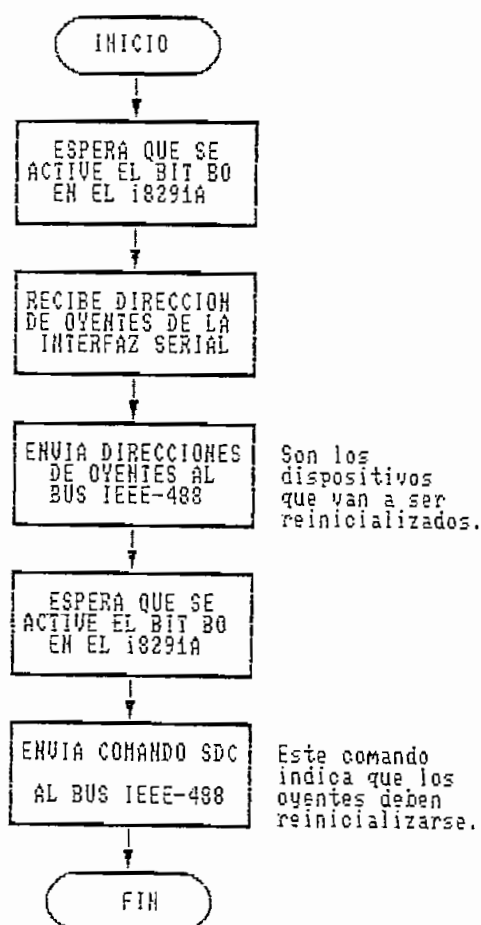


FIGURA 3.19 Diagrama de Flujo de la Subrutina LIMDIS

i) DISEÑO DE LA SUBROUTINA IFCL

Esta función activa la línea IFCL por al menos 100 μ s como está definido en la norma IEEE-488. Esto provoca que la interfaz IEEE-488 vaya a un estado conocido o de inicialización. La Figura 3.20 indica el diagrama de flujo de la subrutina IFCL.

El controlador activo se encarga de enviar este mensaje al iniciar la operación del bus IEEE-488. Esta función se realiza escribiendo al Registro Command Field del i8292 el comando ABORT.

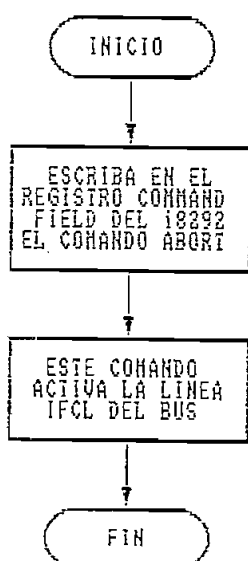


FIGURA 3.20 Diagrama de Flujo de IFCL

j) DISEÑO DE LA SUBROUTINA REM

Con esta función se habilita el control remoto de los dispositivos que previamente fueron direccionados como oyentes, activando la línea REN de la interfaz IEEE-488.

Para esto el controlador envía las direcciones de oyente de los dispositivos en los que quiere habilitar el control remoto y luego escribe el comando SREN en el Registro Command Field del i8292 para que active la línea REN. La Figura 3.21 indica en diagrama de flujo de la subrutina REM.

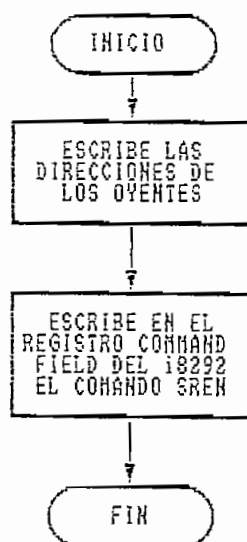


FIGURA 3.21 Diagrama de Flujo de la Subrutina REM

k) DISEÑO DE LA SUBROUTINA LOCAL

La Figura 3.22 presenta el diagrama de flujo de la subrutina LOCAL. Esta función habilita el control local en los dispositivos previamente direccionados como oyentes, desactivando la línea REN. El controlador envía la dirección de los oyentes y luego desactiva la línea REN escribiendo en el Registro Command Field del i8292 el comando SLOC.

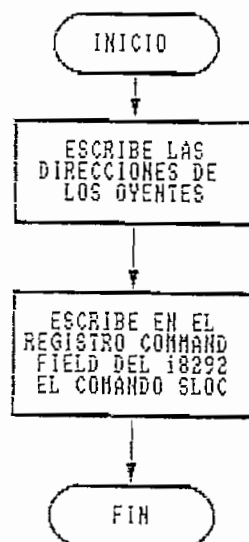


FIGURA 3.22 Diagrama de Flujo de la Subrutina LOCAL

1) DISEÑO DE LA SUBROUTINA P SERIAL

El controlador realiza una búsqueda en serie cuando detecta que la línea SRQ ha sido activada, esto indica que uno o más dispositivos conectados al bus requiere servicio. El sistema debe tener el control del bus para realizar una búsqueda en serie, por la subrutina escribe en el i8292 el comando TCSY para que tome el control del bus y active la línea ATN. Envía al bus el comando SPE y la dirección de locutor del primer dispositivo al que se va a pedir el estado, se deja al i8292 en estado de standby escribiendo en su Registro Command Field el comando GTSB para que libere la línea ATN y el i8291A lee el estado del dispositivo.

El i8292 toma nuevamente el control del bus y si hay más dispositivos en el bus envía la dirección del siguiente locutor y recibe su estado. Para acabar la búsqueda en serie el sistema envía el comando SPD al bus y escribe en el Registro Command Field del i8292 el número 2BH para indicar el reconocimiento de la interrupción y borrar el bit de interrupción activado. El diagrama de flujo de la subrutina P SERIAL se presenta en la Figura 3.23.

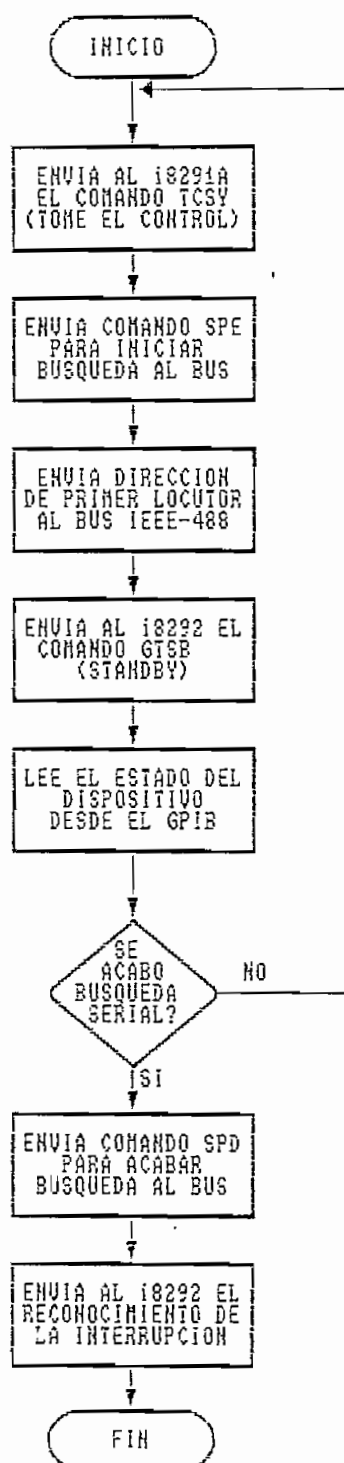


FIGURA 3.23 Diagrama de Flujo de la Subrutina PSERIAL

3.4 SOFTWARE PARA PRUEBAS DEL SISTEMA.

3.4.1 DISEÑO DE PROGRAMAS PARA LA PARTE EIA-RS232C.

El programa que maneja la operación del sistema desde la interfaz EIA-RS232C fue desarrollado en QUICKBASIC, y se encarga de realizar la comunicación desde la interfaz serial con el sistema. El diagrama de flujo general del programa se indica en la Figura 3.23.

Antes de empezar la comunicación deben inicializarse las características que se van a utilizar en dicha comunicación, como: el ritmo binario de la interfaz serial y si el sistema va o no a ser el controlador del bus IEEE-488. En caso de que vaya a realizar las funciones de controlador, deben ingresarse la dirección del sistema y el caracter EOS si se va a utilizar en la transferencia de datos. En cambio si no va a operar como el controlador del bus debe indicarse si el sistema va a funcionar direccionado por un controlador o en sus funciones de "solamente locutor" o "solamente oyente".

El programa presenta un menú con las opciones principales, de manera que al comenzar el funcionamiento debe escogerse la opción de inicialización en la que se ingresarán las características antes mencionadas, o la

opción para salir del programa, antes de seleccionar la opción "FUNCIONES". Las opciones pueden seleccionarse presionando la tecla de la letra más iluminada o mediante el movimiento del cursor y la presión de la tecla ENTER.

Si se escoge la opción "INICIALIZACION" el programa llama a la subrutina CARAC cuyo diagrama de flujo se indica en la Figura 3.24, en la cual se presenta una pantalla con las características que deben inicializarse. Y adicionalmente dos opciones: presionar la tecla F1 para ingresar o editar los parámetros y presionar la tecla F2 para salir al menú principal.

Si se presiona la tecla F2 para salir sin ingresar las características se presenta el mensaje de que no se ha inicializado la comunicación y sale al menú anteriormente descrito.

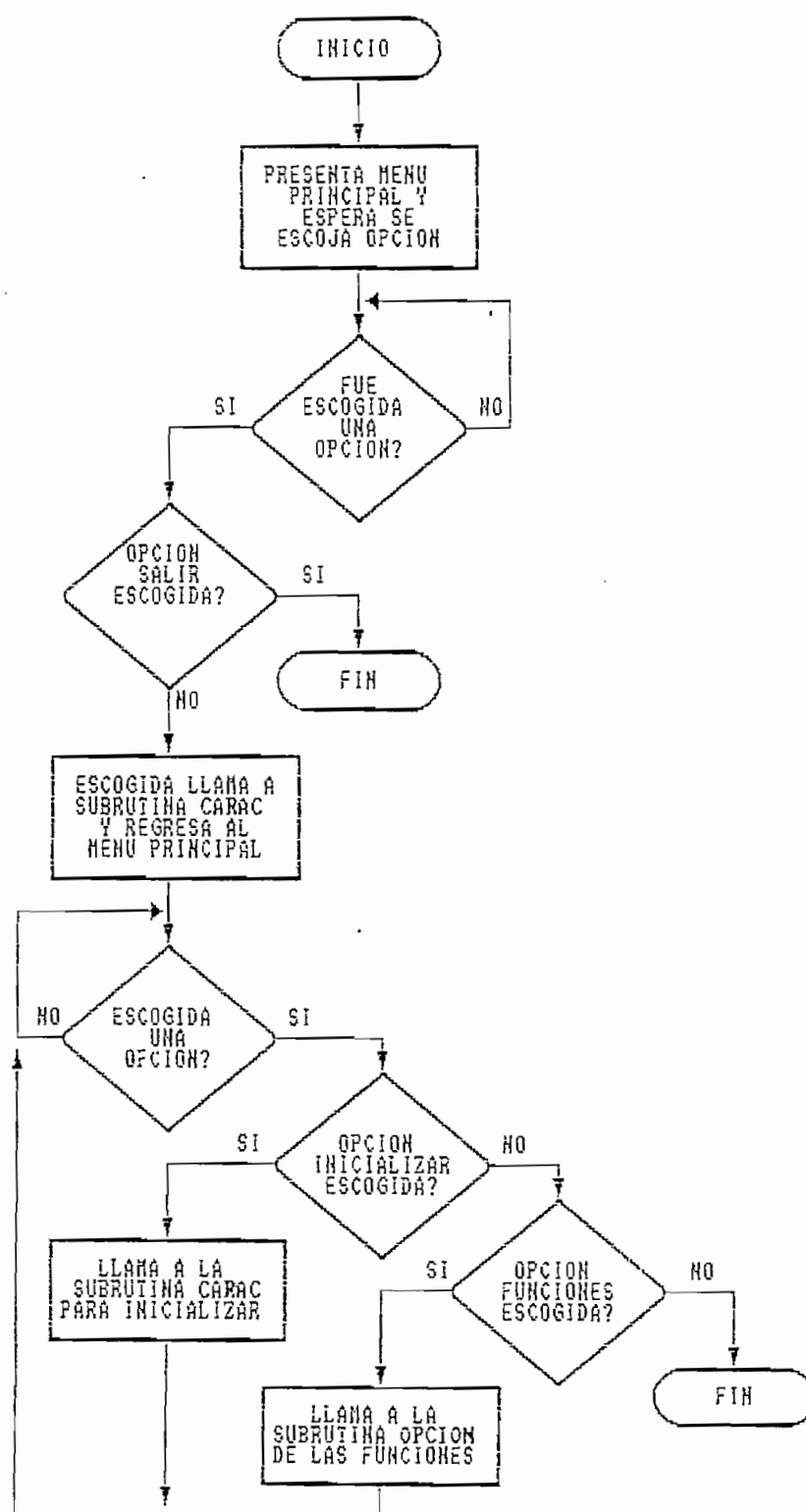


FIGURA 3.23 Diagrama de Flujo General

Si ya se ha realizado la inicialización, antes de salir de éste submenú la subrutina abre el puerto de comunicación serial y envía al sistema el caracter de inicialización FFH para que éste detecte el ritmo binario y espera recibir el caracter ACK (06H) que le indica que la comunicación con el sistema se ha establecido con el ritmo binario adecuado, si dentro de cierto tiempo no recibe el caracter ACK (06H), envía nuevamente el caracter de inicialización FFH y de nuevo espera recibir el caracter ACK (06H), esto lo hace cuando el ritmo binario es menor o igual que 1200 bits/s. Si no se recibió el caracter ACK el programa muestra una indicación de error de comunicación no establecida.

Para ingresar las características se debe presionar la tecla F1, en éste caso el programa llama a la subrutina EDITAR que tiene el diagrama de flujo de la Figura 3.25, que permite ingresar los valores para cada característica, determinando si son o no correctos con los límites esperados y si no lo son, espera el ingreso de nuevos valores.

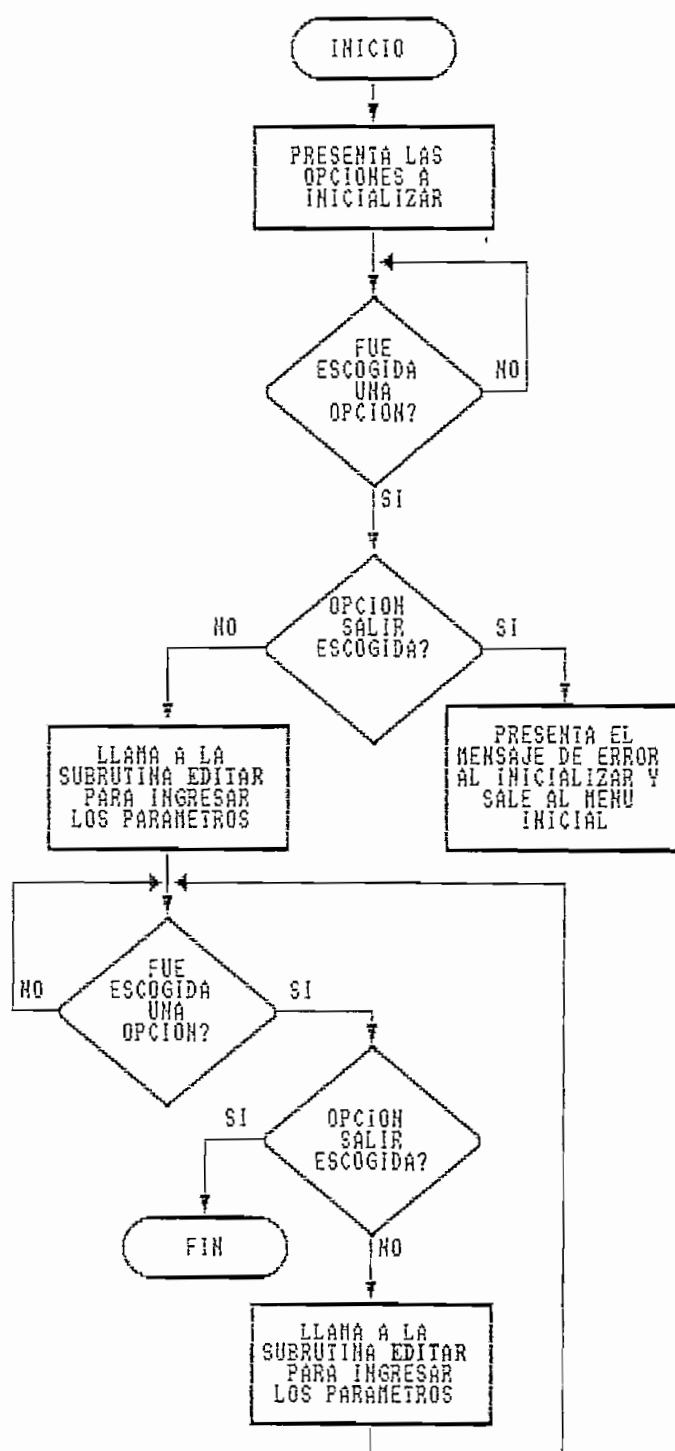


FIGURA 3.24 Diagrama de Flujo de la Subrutina CARAC

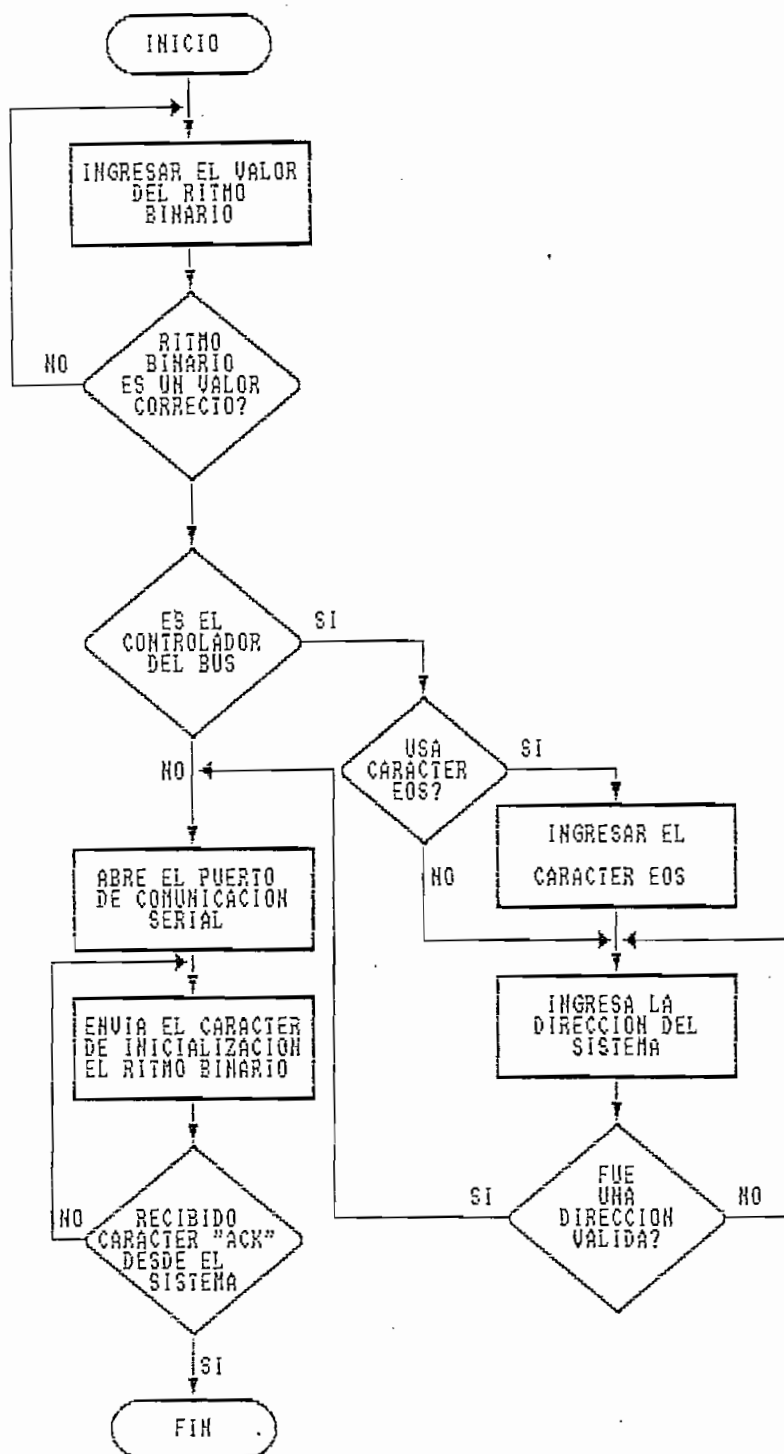


FIGURA 3.25 Diagrama de Flujo de la Subrutina EDITAR

a) Operación del sistema como controlador del bus IEEE-488.

Si el sistema va a operar como el controlador del bus IEEE-488 al escoger la opción "FUNCIONES" en el menú principal, se ingresa a un submenú de las funciones que puede realizar un sistema controlador. El programa llama a la subrutina OPCION, cuyo diagrama de flujo se indica en la Figura 3.26 en donde las funciones se escogen mediante el movimiento del cursor y presionando la tecla ENTER.

Al escoger una de las funciones el programa ejecuta la rutina correspondiente a la función elegida. Las doce funciones que pueden seleccionarse son: ENVIAR, RECIBIR, TRANSFERIR, BUSQUEDA SERIAL, BUSQUEDA PARALELO, PASE EL CONTROL, TOME EL CONTROL, REMOTO, LOCAL, REINICIALIZAR, DISPARAR, IFCL.

a) RUTINA ENVIAR

Esta rutina permite al sistema enviar datos al bus IEEE-488, en primer lugar indica al sistema la función que debe realizar. Luego envía al sistema la dirección de locutor del sistema, y las direcciones de los oyentes que van a participar en la transferencia de datos, verificando si las direcciones ingresadas son válidas es decir si

están entre 0 y 30, en caso contrario, indica el error y regresa a esperar que el usuario ingrese una nueva dirección válida. Una vez realizado el direccionamiento, espera el ingreso del mensaje por el teclado y lo envía al sistema por la interfaz serial para que sea transferido al bus IEEE-488, al detectar que la tecla ESC fue presionada acaba de enviar datos y sale de la rutina al submenú de funciones.

b) RUTINA RECIBIR

La rutina RECIBIR permite al sistema recibir datos desde el bus IEEE-488. Envía al sistema la dirección del locutor, la dirección de oyente del sistema, y las direcciones de los demás oyentes que van a participar en la transferencia de datos, verificando si las direcciones ingresadas son válidas, en caso contrario, indica el error y regresa a esperar que el usuario ingrese una nueva dirección válida. Espera recibir por la interfaz serial los datos que le enviará el sistema provenientes de la interfaz IEEE-488 al acabar de recibir los datos sale de la rutina al submenú de funciones.

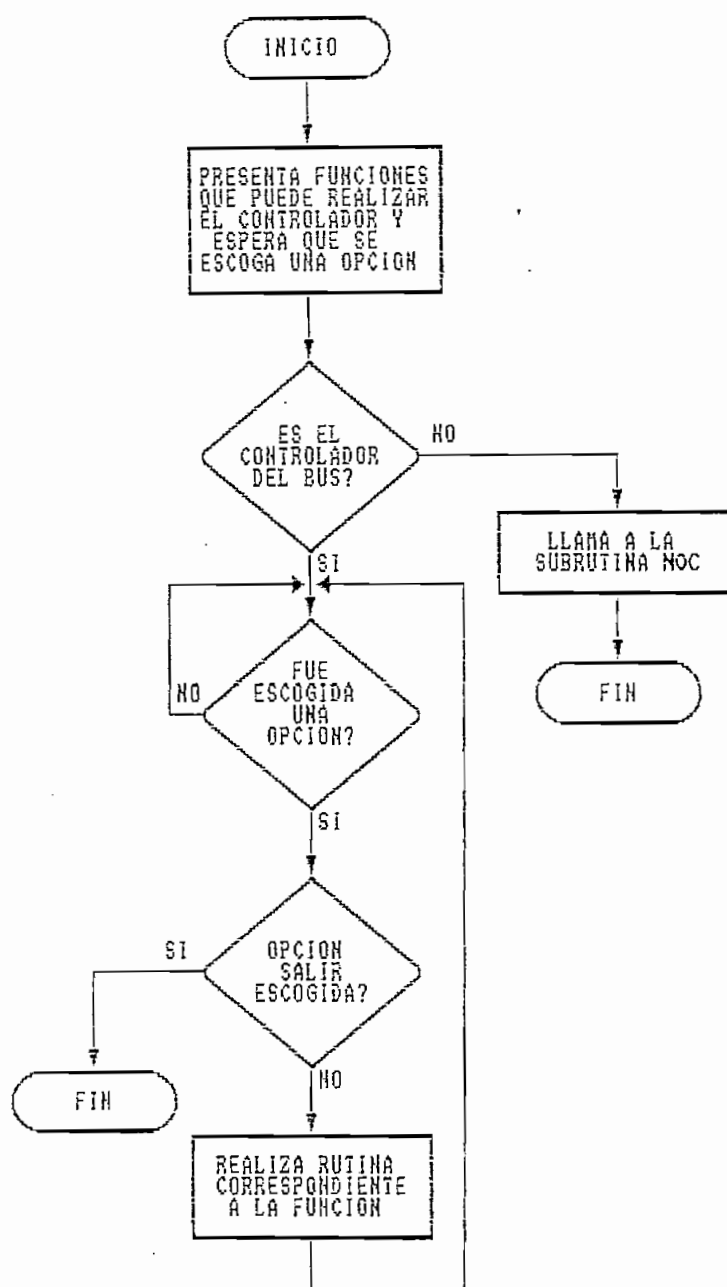


FIGURA 3.26 Diagrama de Flujo de la Subrutina OPCION

c) RUTINA TRANSFERIR

Envía la dirección del locutor, y las direcciones de los oyentes, verificando si las direcciones ingresadas son válidas. Una vez realizado el direccionamiento, espera recibir desde el sistema la indicación de que la comunicación ha finalizado, ya que en éste caso el sistema no va a intervenir en la comunicación y luego sale de la rutina al submenú de funciones.

d) RUTINA BUSQUEDA EN PARALELO

La rutina configura los dispositivos para la búsqueda en paralelo para luego poder realizarla, en ese caso espera que el sistema envíe los bytes de estado de los dispositivos, después la rutina sale al submenú de funciones.

e) RUTINA PASE CONTROL

La rutina permite pasar el control a otro dispositivo conectado al bus IEEE-488 para esto el programa envía la dirección de locutor del controlador que va a recibir el control. Y espera que el sistema le indique si se realizó correctamente la transferencia del control o se produjo algún error, luego sale al submenú de funciones.

f) RUTINA TOME CONTROL

La rutina permite tomar el control desde otro dispositivo conectado al bus IEEE-488 el programa indica al sistema la función que debe realizar y luego espera la indicación del sistema de si se realizó correctamente la transferencia del control o se produjo algún error, finalmente sale al submenú de funciones.

g) RUTINA REMOTO

Esta rutina pone a los dispositivos direccionados como oyentes en control remoto. Envía al sistema las direcciones de los oyentes que van a ser puestos en modo remoto verificando si las direcciones ingresadas son válidas, en caso contrario, indica el error y regresa a esperar que el usuario ingrese una nueva dirección válida.

h) RUTINA LOCAL

Esta rutina retorna a los dispositivos direccionados como oyentes al control local. Envía al sistema las direcciones de los oyentes que van a ser puestos en modo local verificando si las direcciones ingresadas son válidas, en caso contrario, indica el error y regresa a esperar que el usuario ingrese una nueva dirección válida.

i) RUTINA REINICIALIZAR

Esta rutina reinicializa a los dispositivos direccionados como oyentes. Envía las direcciones de los oyentes que van a ser reinicializados verificando si son válidas, en caso contrario, indica el error y regresa a esperar que el usuario ingrese una nueva dirección válida.

j) RUTINA DISPARAR

Esta rutina envía el comando de disparo a los dispositivos direccionados como oyentes. Envía las direcciones de los oyentes que van a ser disparados verificando si son válidas, en caso contrario, indica el error y regresa a esperar que el usuario ingrese una nueva dirección válida.

k) RUTINA IFCL

Esta rutina retorna al bus IEEE-488 a un estado conocido de reinicialización.

l) RUTINA BUSQUEDA EN SERIE

La rutina envía la dirección de locutores de los dispositivos a los que se va a pedir estado verificando si

las direcciones ingresadas son válidas, en caso contrario, indica el error y regresa a esperar que el usuario ingrese una nueva dirección. Espera que el sistema envíe los bytes de estado de los dispositivos, después la rutina sale al submenú de funciones.

b) Operación del sistema como no controlador del bus
IEEE-488

Si se eligió la opción de que el sistema no sea el controlador del bus, el programa realiza las funciones de escribir y leer datos desde o hacia el bus IEEE-488, pedir servicio al controlador y poner estado, con la subrutina NCON cuyo diagrama de flujo de esta rutina se indica en la Figura 3.27.

Se realiza un lazo de espera hasta que el sistema envíe por la interfaz serial un caracter indicando que el sistema envió un comando y debe realizarse una de las funciones. El caracter "L" (Locutor) indica que el sistema fue direccionado para hablar por el controlador del bus, por lo que se envia los datos ingresados desde el teclado. El caracter "O" (Oyente) indica que el sistema fue direccionado para oír, en ese caso se espera recibir los datos leídos por el sistema.

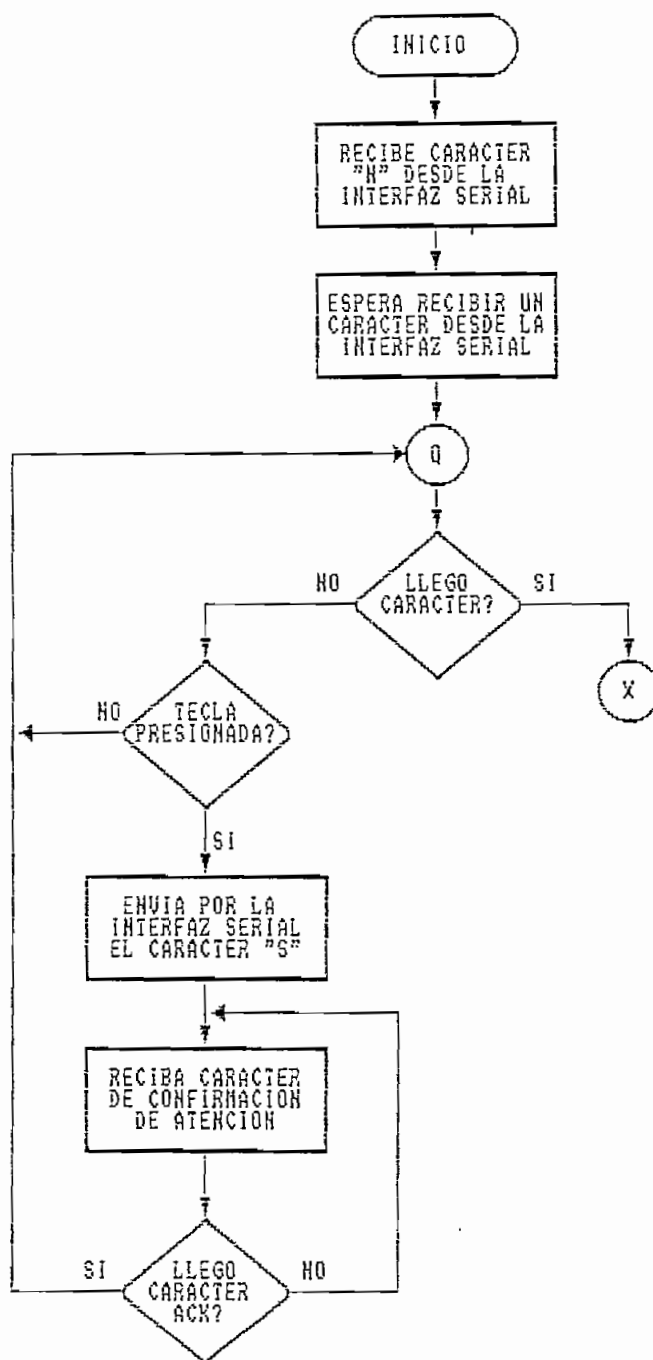


FIGURA 3.27 Diagrama de Flujo de la Rutina NCON

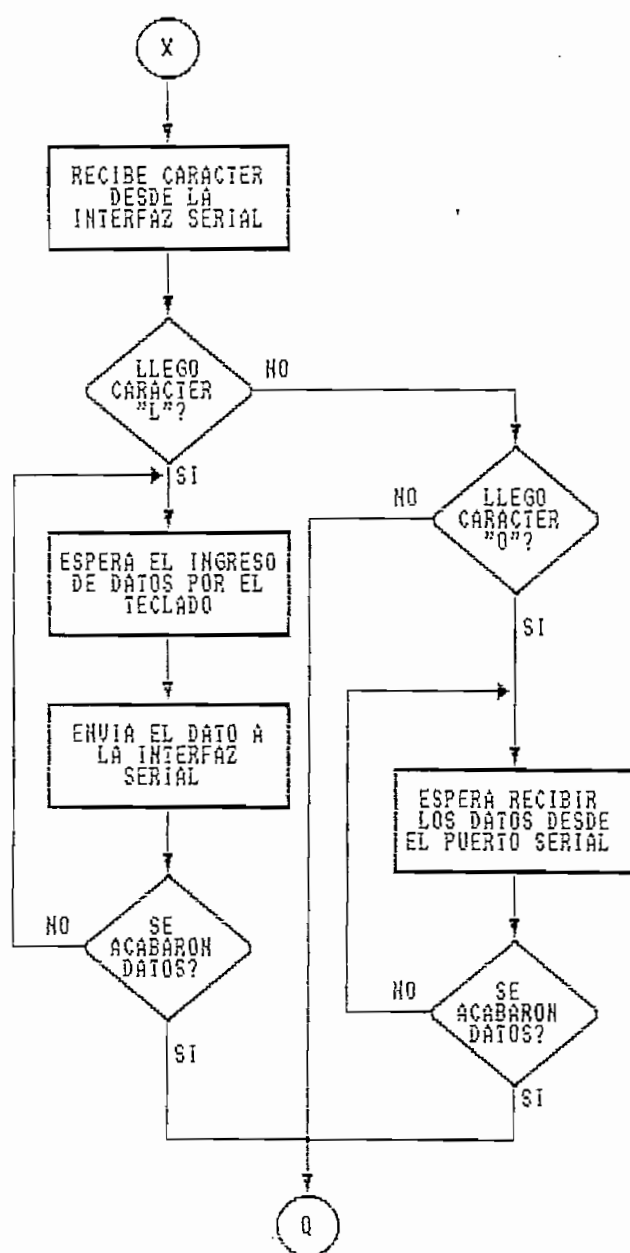


FIGURA 3.27 Continuación

Si se requiere que el sistema pida servicio al bus el usuario debe ingresar el caracter "S" desde el teclado con lo que se envía por el puerto serial la indicación correspondiente al sistema y se espera la indicación de que el controlador realizó la búsqueda en paralelo. Se puede también habilitar al sistema para responder a una búsqueda en paralelo.

3.4.2 DISEÑO DE PROGRAMAS PARA LA PARTE IEEE-488.

El programa que maneja la operación del sistema desde la interfaz IEEE-488 fue desarrollado en GWBASIC, y se encarga de realizar la comunicación desde la interfaz IEEE-488 con el sistema. El diagrama de flujo general del programa se indica en la Figura 3.28. Las rutinas de la tarjeta GPIB-PCIIA fueron desarrolladas en GWBASIC por lo que para poder usarlas se debe programar en ese lenguaje.

Antes de iniciar la comunicación debe escogerse los parámetros que van a utilizarse en dicha comunicación, en el paquete IBCONF de la biblioteca de rutinas de la tarjeta GPIB-PCIIA.

Al comenzar el funcionamiento se debe escoger la operación de la tarjeta como controlador o no controlador del bus de acuerdo a lo seleccionado en IBCONF.

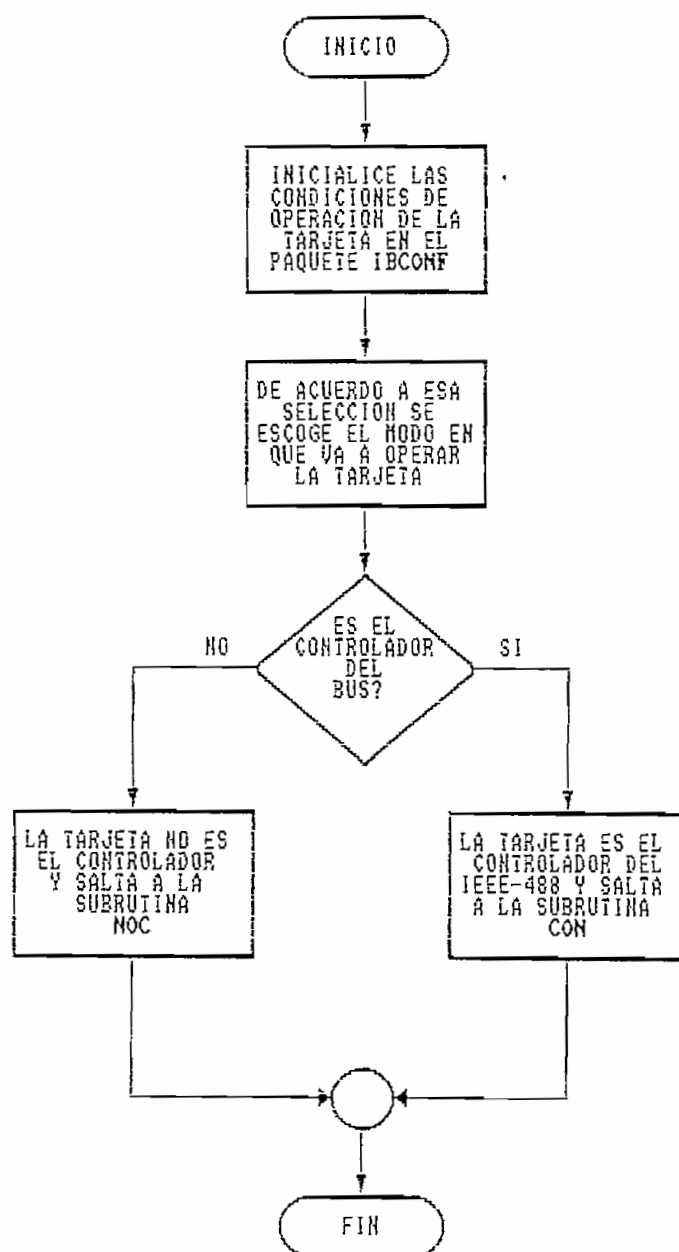


FIGURA 3.28 Diagrama de Flujo General

Si se escoge la opción "NO CONTROLADOR" el programa salta a la subrutina NOC, en la cual se presenta una pantalla con las funciones que el sistema puede realizar, como son: Leer datos, escribir datos, reinicializar la tarjeta, pedir servicio, habilitar la respuesta de una búsqueda en paralelo. El diagrama de flujo de esta parte se indica en la Figura 3.29. En éste caso el programa debe esperar que se le direcciona para hablar, oír, o para enviar estado, o recibir los comandos para ponerse en estado de control remoto, local, reinicializarse, o el comando de disparo. Al recibir un comando la rutina determina de que comando se trata y realiza la función correspondiente.

Cuando se elige la opción "CONTROLADOR" también se presenta la pantalla con las funciones que puede realizar el controlador del bus. Al escoger una de las funciones el programa ejecuta la rutina correspondiente a la función elegida. Las funciones que pueden seleccionarse son: ENVIAR, RECIBIR, BUSQUEDA SERIAL, BUSQUEDA PARALELO, PASE EL CONTROL, TOME EL CONTROL, REMOTO, LOCAL, REINICIALIZAR, DISPARAR, IFCL.

El Diagrama de flujo de esta rutina se indica en la Figura 3.30.

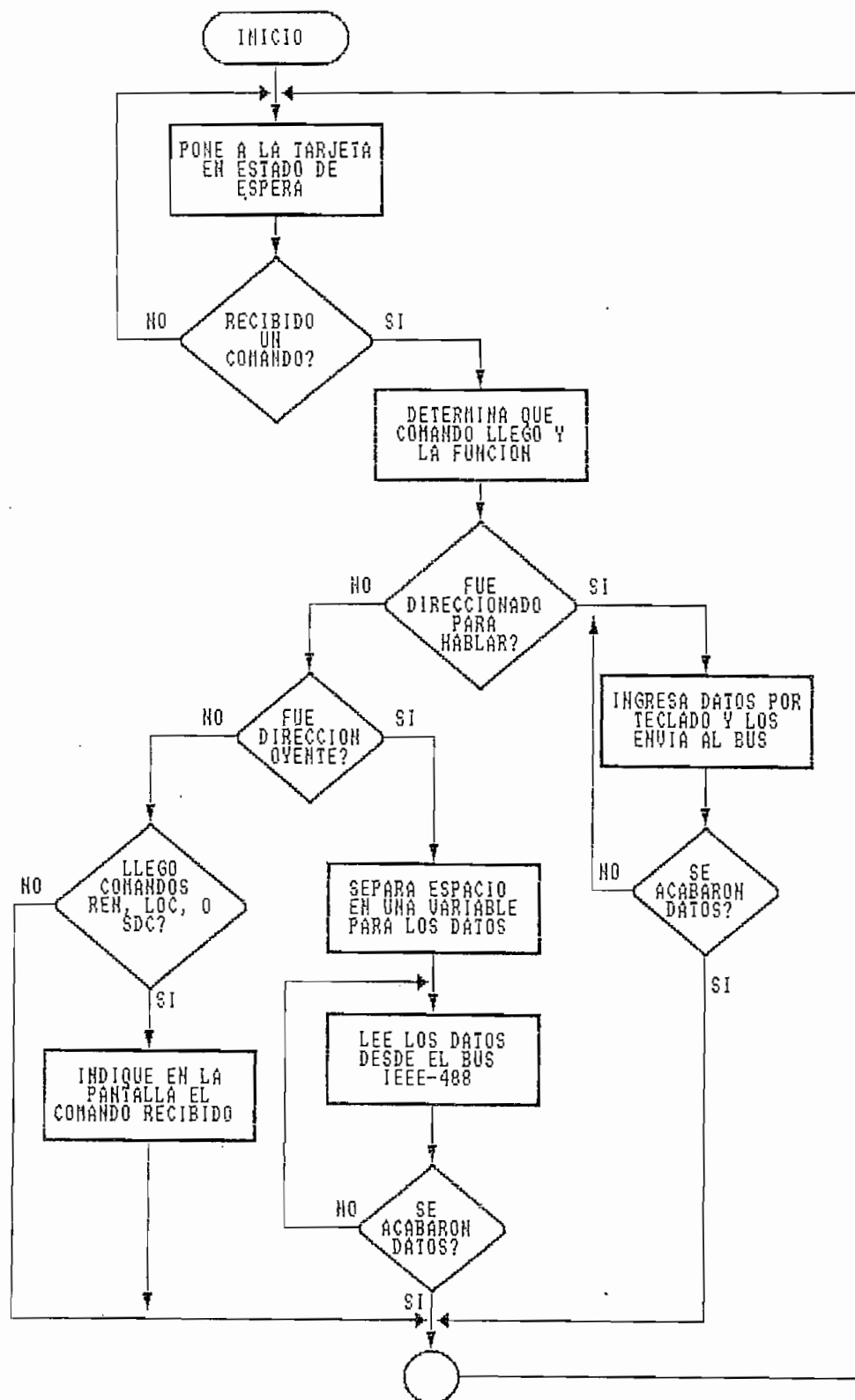


FIGURA 3.29 Diagrama de Flujo para la Rutina NOC

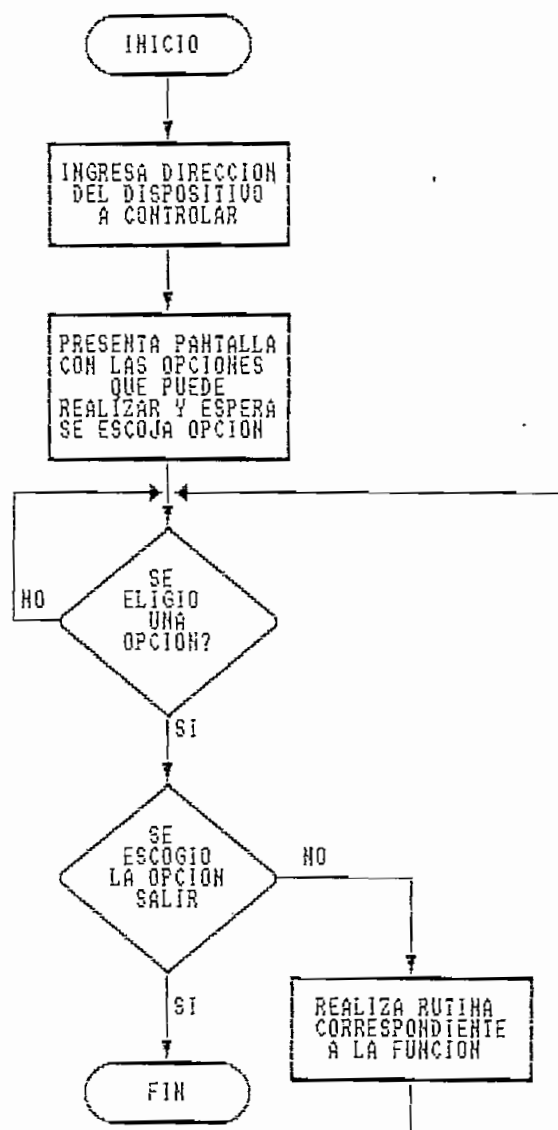


FIGURA 3.30 Diagrama de Flujo de la Rutina CON

CAPITULO IV

**RESULTADOS EXPERIMENTALES
CONCLUSIONES Y RECOMENDACIONES**

CAPITULO IV

RESULTADOS EXPERIMENTALES CONCLUSIONES Y RECOMENDACIONES

4.1 RESULTADOS EXPERIMENTALES

El sistema de conversión diseñado en la presente Tesis funciona mediante un programa realizado para el microcontrolador i8751 que se encarga de todas las funciones de inicialización, transferencia de comandos y datos y del control del sistema.

Dicho programa fue probado con el paquete AVSIM51, comprobando que a nivel de simulación las subrutinas trabajaran adecuadamente. Las pruebas realizadas con el simulador no aseguran que el hardware funcione adecuadamente, ya que el AVSIM51 solamente puede simular la operación del i8751 y no puede determinar la operación de los circuitos integrados asociados a éste en el diseño. Por lo tanto la simulación es especialmente útil para detectar los errores que se presentan en el programa como: saltos incorrectos, lazos sin fin, etc.

4.1.1 EQUIPO DE PRUEBAS

Las pruebas en el hardware se realizaron utilizando

la configuración de equipo que se indica en la Figura 4.1, compuesta por: el sistema de conversión de interfaces serial - paralelo, un computador dotado de una tarjeta para comunicación serial EIA-RS232C y el un computador provisto de la tarjeta GPIB-PCIIA disponible en el laboratorio. Se realizó un programa para el computador que contiene la tarjeta de comunicación serial y otro programa para el computador que contiene la tarjeta GPIB-PCIIA.

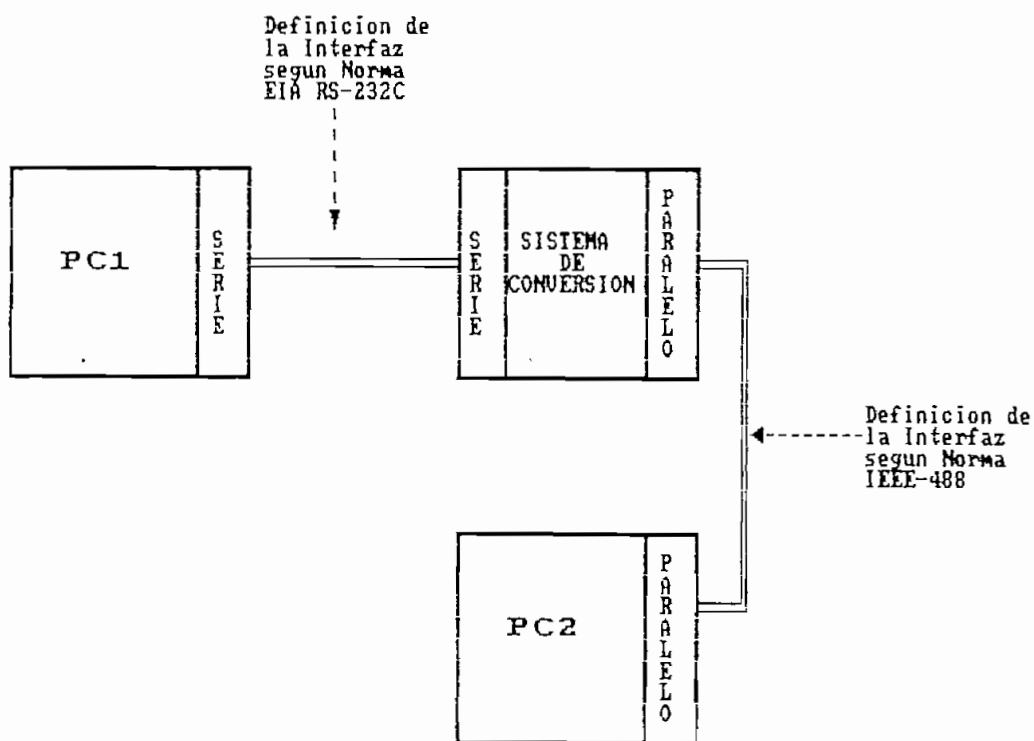


FIGURA 4.1 Diagrama del Equipo de Pruebas

4.1.2 PROTOCOLO DE PRUEBAS

a) Sistema de conversión como Controlador del bus IEEE-488

El computador conectado a la interfaz serial a través del sistema de conversión es el encargado de direccionar al computador que contiene la tarjeta GPIB-PCIIA, y de enviarle comandos de reinicialización, disparo, etc, éste último reconoce el comando que el controlador le ha enviado y el programa realiza la tarea apropiada.

b) Computador provisto de la tarjeta GPIB-PCIIA como Controlador del bus

En este caso el sistema de conversión opera como el dispositivo a controlar. El controlador envía los comando al sistema y éste reconoce cuando ha sido direccionado, reinicilizado, disparado, etc y envía la indicación al computador conectado a la interfaz serial, de acuerdo a lo cual el programa de dicho computador realiza la tarea correspondiente

c) Sistema de comunicación sin Controlador

En este caso el sistema de conversión puede ser

configurado ya sea como "locutor solamente" u "oyente solamente". Si es configurado como locutor, el computador dotado de la tarjeta GPIB-PCIIA desempeñará las funciones de oyente y viceversa.

En el caso de que el sistema de conversión sea el locutor los datos serán enviados a través del computador dotado de la tarjeta de comunicación serial EIA-RS232C y en el caso de que el sistema sea el oyente los datos serán enviados por el computador provisto de la tarjeta GPIB-PCIIA.

El resultado de ésta prueba no es plenamente satisfactorio en virtud de que la tarjeta GPIB-PCIIA no ha sido diseñado para operar en un ambiente sin controlador y requiere necesariamente de ser direccionada.

4.2 CONCLUSIONES Y RECOMENDACIONES

- a) La disponibilidad de circuitos integrados VLSI específicos para la norma IEEE-488 simplifica considerablemente la tarea de diseño de la interfaz, permitiendo que se pueda realizar interfaces para ésta norma mucho más flexibles, con mayor capacidad de funcionamiento, con un menor número de circuitos integrados y por lo tanto en un espacio más reducido,

que lo que se obtendría usando circuitos integrados de menor escala de integración. Tres de esos circuitos integrados son el i8291A, i8292 e i8293 desarrollados por la INTEL.

Ha sido posible implementar un sistema que cumple con las especificaciones de la norma IEEE-488 incluyendo las características eléctricas.

- b) Aunque una interfaz serial llena los requerimientos de comunicación de muchos sistemas y es más fácil de implementar, una interfaz paralela sobre distancias pequeñas provee velocidades de comunicación mayores, aunque requiera mayor número de conectores y de circuitos complejos. Lo que para algunas aplicaciones es de especial importancia.
- c) El diseño del sistema incluye la detección automática del ritmo binario de la comunicación serial a través de software. Para ésto es necesario que el equipo EIA-RS232C envíe un caracter de inicialización al sistema de conversión que permite detectar automáticamente el ritmo binario de acuerdo al caracter recibido. Esto hace posible que los dispositivos conectados a la interfaz serial puedan comunicarse en cualquiera de los siguientes ritmos

binarios: 19200, 9600, 4800, 2400, 1200, 600, 300 y 150 Bits/s.

- d) La interfaz EIA-RS232C se la realiza mediante el puerto serial del microcontrolador i8751 con 8 bits de datos, 1 bit de parada y sin paridad y el circuito integrado MAX232, que permite adaptar los niveles de voltaje TTL que utiliza el microcontrolador a niveles de la interfaz serial y viceversa, la utilización de éste circuito permite el uso de una sola fuente de 5V.
- e) Una de las aplicaciones del sistema de conversión de interfaces serial - paralela que merece mención especial es la capacidad de controlar remotamente dispositivos IEEE-488 con un computador central, eliminando la necesidad de un controlador independiente en sitios remotos. Con ésto se consigue aumentar la extensión del bus IEEE-488, usando modems conectados a la interfaz EIA-RS232C que permiten distancias mayores que los límites máximos de cada norma.
- f) La norma IEEE-488 cubre las características eléctricas, mecánicas y funcionales de la interfaz, pero no cubre las funciones de dispositivo, es decir

aquellos mensajes que intercambian los dispositivos una vez que se ha realizado el direccionamiento. Por lo que en general para que los dispositivos puedan comunicarse entre sí es necesario que sus mensajes de dispositivo sean también compatibles.

- g) Las funciones del bus IEEE-488 se implementan de acuerdo a la norma en todos los dispositivos; sin embargo, no es necesario que los dispositivos incluyan todas las funciones de la norma, sino únicamente las útiles para su operación. Por lo tanto antes de que el controlador envíe un comando al bus se debe comprobar que el dispositivo está habilitado para responder a dicha función.

BIBLIOGRAFIA

- 1.- Gofton P. Mastering Serial Communications, Berckelely, Paris, 1986.
- 2.- Kruglinsky D. Guía de las Comunicaciones del IBM/PC, Mc Graw Hill, España, 1985.
- 3.- Seyer M. RS-232 Made Easy, Prentice Hall, USA, 1984.
- 4.- Mompin J. Interconexión de Periféricos a Microcontroladores, Mundo Electrónico, 2da edición, Marcombo, Barcelona, 1983.
- 5.- Journal of Data Communication, 1988.
- 6.- González N. Comunicaciones y Redes de Procesamiento de Datos, Mc Graw Hill, 1983.
- 7.- Lilen H. Interfaces pour Microprocesseurs et Micro-Ordinatures, Editions Radio, 1985.
- 8.- Andrews M. Programming Microprocessor Interfaces For Control and Instrumentation, Prentice-Hall Inc, 1982.
- 9.- National Instrument, GPIB-PC IEEE-488 Instrumentation Interface.
- 10.- IEEE, IEEE Transactions on Instrumentation and Measurement #3, 1984.
- 11.- Leibson S. The Handbook of Microcomputer Interfacing, Tab Books Inc, USA, 1983.
- 12.- IEEE, IEEE Standard Digital Interface For Programable Instrumentatiion, USA, 1983.

- 13.- INTEL, Microsystem Componets Handbook Volume II, USA, 1984.
- 14.- CCITT, Recomendación V.24, Fascículo VII.I, Ginebra, 1984.
- 15.- CCITT, Recomendación V.23, Fascículo VII.I, Ginebra, 1984.
- 16.- INTEL, INTEL 8-bit Embedded Controller Handbook, 1987
- 17.- Texas Instruments, The TTL Data Book, Volumen2, 1985.

ANEXO A

MANUAL DE USO

**SISTEMA DE CONVERSION DE INTERFACES
SERIAL — PARALELO**

MANUAL DE USO

INTRODUCCION

El programa desarrollado para controlar el funcionamiento del equipo de llama SERIE.EXE y debe ejecutarse para realizar las diferentes funciones del sistema.

OPERACION:

Antes de encender el equipo es necesario que el usuario seleccione las características de operación del sistema de conversión. El sistema puede funcionar: a) como el controlador del bus y b) como un dispositivo no controlador; para escoger entre éstos dos modos de operación debe seleccionarse la posición del switch de CONTROL, en la posición de ON se escoge la operación CONTROLADOR y en la posición OFF la operación NO CONTROLADOR.

Si ésta última opción es activada, adicionalmente deben setearse en los dip-switches la dirección del sistema con los cinco dip-switches menos significativos (SW1, SW2, SW3, SW4, SW5), es la dirección a la que el sistema responderá con los comandos de direccionamiento enviados por el controlador. Los dos dip-switches

siguientes (SW6, SW7) permiten escoger las funciones "locutor solamente" u "oyente solamente" cuando el sistema se conecte a un dispositivo que tenga las mismas funciones, en un bus sin controlador. La selección de las dos opciones simultáneamente deshabilita al sistema de conversión y produce un error.

El último dip-switch (SW8) habilita la característica del uso del carácter 0Ah como carácter EOS en la comunicación. Esta configuración se indica en la Tabla A.1.

# SW	VALOR	FUNCION
1	n	Dirección 2 ⁰
2	n	Dirección 2 ¹
3	n	Dirección 2 ²
4	n	Dirección 2 ³
5	n	Dirección 2 ⁴
6	1 0	"Locutor solamente" Habilitado Deshabilitado
7	1 0	"Oyente solamente" Habilitado Deshabilitado
8	1 0	Carácter EOS (0Ah) Habilitado Deshabilitado

TABLA A.1 Tabla de selección de las características de un dispositivo no controlador

Una vez escogidas las características de operación del sistema conversión y encendido el dispositivo, cualquier cambio en los switches no será tomado en cuenta, por lo que para que se consideren los cambios el sistema debe ser reinicializado mediante el pulsante de RESET o encendido nuevamente.

Como dirección del sistema conversor puede escogerse cualquier valor de 0 a 30, la dirección 31 deshabilitará al equipo. Debe tenerse la precaución de que la dirección escogida no coincida con la de otro dispositivo en el bus.

a) INICIALIZACION

Al ejecutar el programa SERIE.EXE se presenta un menú principal en el cual se escogen las características iniciales de funcionamiento ingresando a la opción INICIALIZACION, aquí se determina el ritmo binario de la comunicación serial y si el sistema es o no el controlador del bus. Esta opción debe estar de acuerdo a lo seleccionado en el hardware. El ritmo binario puede ser escogido entre los siguientes valores: 19200, 9600, 4800, 2400, 1200, 600, 300 y 150 bits/s, un valor diferente producirá un error en el programa.

b) CONTROL

En caso de escoger la operación del sistema como controlador del bus, se determina en las características de inicialización si se va a utilizar un caracter como caracter EOS y además se ingresa la dirección del sistema como controlador, que no debe ser necesariamente la ingresada en los dip-switches para cuando el sistema es un dispositivo no controlador.

Al salir al menú principal puede ingresarse ya en la opción de FUNCIONES, que presenta un submenú con las posibles funciones que el sistema puede realizar. Se elige cualquier función con el movimiento del cursor y presionando la tecla ENTER.

En las funciones en que se debe ingresar las direcciones de los dispositivos participantes en la comunicación pueden elegirse valores de 0 a 30, teniendo en cuenta que el número máximo de dispositivos en el bus IEEE-488 es 15.

c) NO CONTROLADOR

Al elegir la operación del sistema como un dispositivo no controlador del bus, en la opción de

INICIALIZACION debe determinarse si el sistema será direccionado por un controlador, o entre las funciones "locutor solamente" u "oyente solamente".

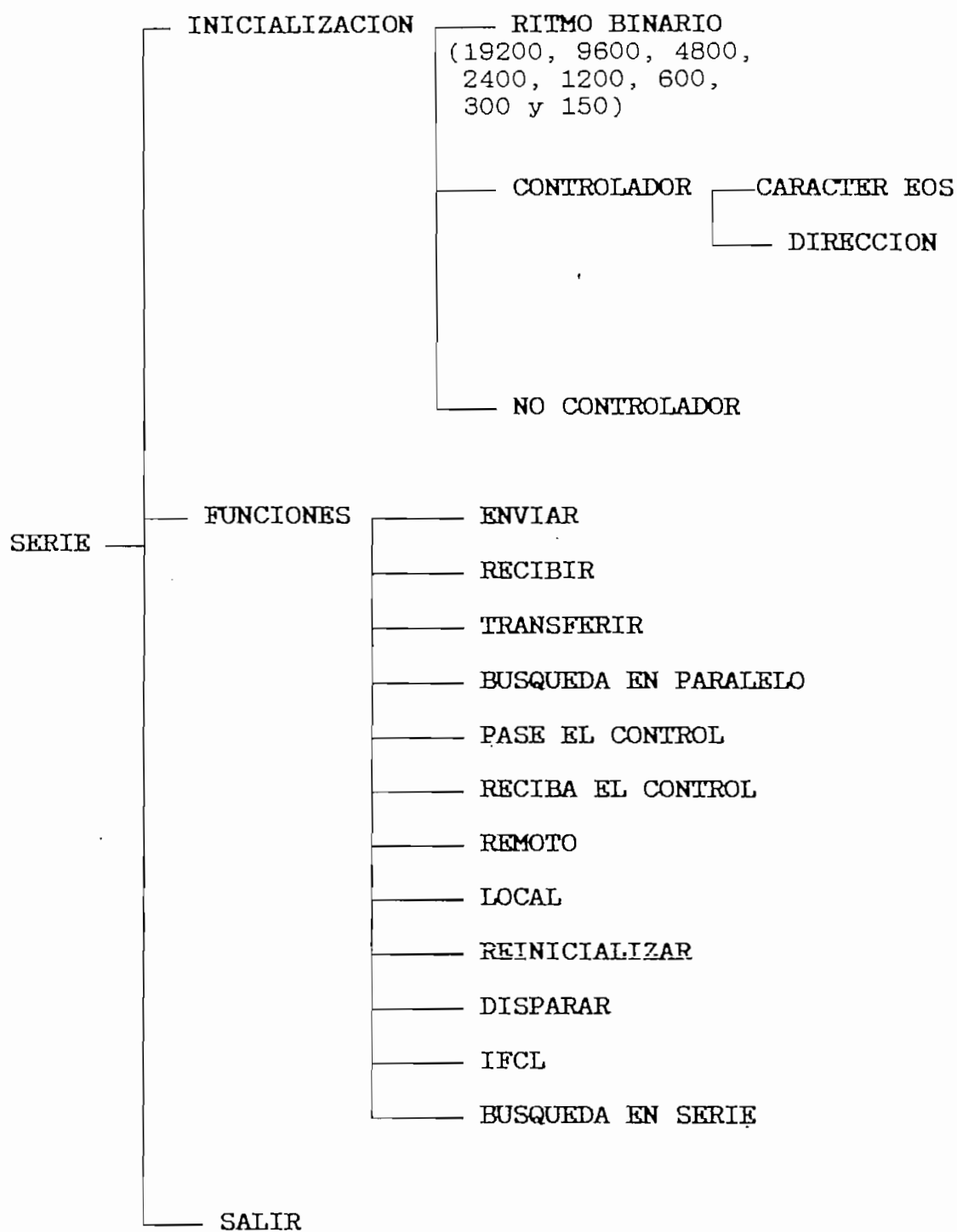
El programa SERIE.EXE, una vez inicializadas las características permite escoger la opción FUNCIONES, en ésta parte el programa espera la indicación enviada desde el sistema conversor que le indicará el comando recibido y en consecuencia la función que deberá realizar.

Para salir del programa se debe escoger la opción SALIR en el menú principal.

ARBOL DE OPCIONES DEL SOFTWARE

La Figura A.1 presenta el árbol de opciones del software a través del cual puede recorrer el usuario del sistema.

* Nota: En el caso de que se escoge en el programa la opción afirmativa al funcionamiento como Controlador la opción FUNCIONES presenta un menú con las funciones que realizará el sistema de acuerdo a lo que escoga el usuario. No se presenta la misma pantalla cuando se ha escogido la opción de No Controlador, en este caso no se presenta ningún menú

FIGURA 4.1 Arbol de opciones del programa `SERIE.EXE`

ANEXO B

REPRESENTACION CODIGO ISO 7-BITS

(SENT AND RECEIVED WITH ATN=1)

[illegible]

NOTES: (1) MSG = INTERFACE MESSAGE
(2) $b_1 = 0101\dots b_7 = 0107$
(3) REQUIRES SECONDARY COMM.
(4) DENSE SUBSET (COLUMN 2 1

Fig E1
Multiline Interface Messages: ISO-7 Bit Code Representation

ANEXO C

INFORMACION TECNICA DE LOS CIRCUITO INTEGRADOS

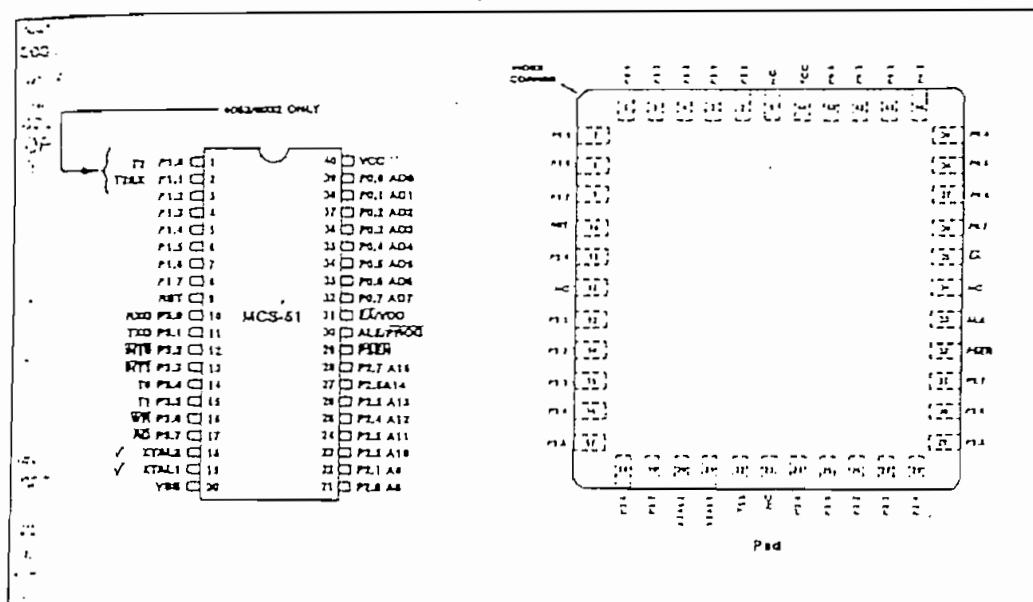


Figure 2. MCS-51 Pin Connections

In the 8032AH and 8052AH, Port 1 pins P1.0 and P1.1 also serve the T2 and T2EX functions, respectively.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL Inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during latches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL Inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternative Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias . . . 0 °C to 70 °C
Storage Temperature -65 °C to +150 °C
Voltage on \overline{EA}/VPP Pin to VSS . . -0.5V to +21.5V
Voltage on Any Other Pin to VSS . . -0.5V to +7V
Power Dissipation 1.5W

*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: ($T_A = 0^\circ\text{C}$ to 70°C ; $V_{CC} = 5\text{V} \pm 10\%$; $V_{SS} = 0\text{V}$)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage (Except \overline{EA} Pin of 8751H, 8751H-12)	-0.5	0.8	V	
V_{IL1}	Input Low Voltage to \overline{EA} Pin of 8751H, 8751H-12	0	0.7	V	
V_{IH}	Input High Voltage (Except XTAL2, RST)	2.0	$V_{CC} + 0.5$	V	
V_{IH1}	Input High Voltage to XTAL2, RST	2.5	$V_{CC} + 0.5$	V	XTAL1 = VSS
V_{OL}	Output Low Voltage (Ports 1, 2, 3)*		0.45	V	IOL = 1.6 mA
V_{OL1}	Output Low Voltage (Port 0, ALE, \overline{PSEN})*				
			0.60	V	IOL = 3.2 mA
	8751H, 8751H-12		0.45	V	IOL = 2.4 mA
	All Others		0.45	V	IOL = 3.2 mA
V_{OH}	Output High Voltage (Ports 1, 2, 3)	2.4		V	IOH = -80 μA
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode, ALE, \overline{PSEN})	2.4		V	IOH = -100 μA
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3) 8032AH, 8052AH All Others		-600 -500	μA	$V_{in} = 0.45\text{V}$ $V_{in} = 0.45\text{V}$
I_{IL1}	Logical 0 Input Current to \overline{EA} Pin of 8751H, 8751H-12 Only		-15	mA	
I_{IL2}	Logical 0 Input Current (XTAL2)		-3.2	mA	$V_{in} = 0.45\text{V}$
I_{LI}	Input Leakage Current (Port 0) 8751H, 8751H-12 All Others		± 100 ± 10	μA	$0.45 < V_{in} < V_{CC}$ $0.45 < V_{in} < V_{CC}$
I_{IH}	Logical 1 Input Current to \overline{EA} Pin of 8751H, 8751H-12		500	μA	
I_{IH1}	Input Current to RST to Activate Reset		500	μA	$V_{in} < (V_{CC} - 1.5\text{V})$
I_{CC}	Power Supply Current: 8031/8051 8031AH/8051AH 8032AH/8052AH 8751H/8751H-12		160 125 175 250	mA	All Outputs Disconnected; $\overline{EA} = V_{CC}$
C_{IO}	Pin Capacitance		10	pF	test freq = 1MHz

*Note: Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the VOLs of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE line may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.

Table 8-1. 8051 Instruction Set Summary (Continued)

INSTRUCTIONS THAT AFFECT FLAG SETTINGS			
INSTRUCTION	FLAG	INSTRUCTION	FLAG
ADD	C OV AC	CLR C	O
ADDC	X X X	CPLC	X
SUBB	X X X	ANLCbit	X
MUL	O X	ANLCbit	X
DIV	O X	ANLCbit	X
DA	X X	ORLCbit	X
RRC	X X	MOVCbit	X
RLC	X X	CNE	X
SETBC	X		

Note that operations on SFR byte address 208 or bit address 20A-215 (i.e., the PSW or bits in the PSW) will also affect flag settings.

Notes on instruction set and addressing modes:
 Rn — Register R7-R0 of the currently selected Register Bank.
 direct — 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 (0-127) addressed indirectly through register Ri or R0.
 #data — 8-bit constant included in instruction.
 #data16 — 16-bit constant included in instruction.
 #addr16 — 16-bit destination address. Used by LCALL and 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 8051AH: SFR-RAH.

LOGICAL OPERATIONS			
Mnemonic	Description	Byte	Oscillator Period
ANL	A, Rn AND register to	1	12
ANL	A, direct AND direct byte	2	12
ANL	A, @Ri AND indirect to Accumulator	1	12
ANL	A, #data AND immediate data to Accumulator	2	12
ANL	direct, A AND Accumulator	2	12
ANL	direct, #data AND immediate data to direct byte	3	24
ORL	A, Rn OR register to	1	12
ORL	A, direct OR direct byte to Accumulator	2	12
ORL	A, @Ri OR indirect RAM	1	12
ORL	A, #data OR immediate data to Accumulator	2	12
ORL	direct, A OR Accumulator	2	12
ORL	direct, #data OR immediate data to direct byte	3	24
XRL	A, Rn Exclusive-OR register to	1	12
XRL	A, direct Exclusive-OR	2	12

LOGICAL OPERATIONS Cont.			
Mnemonic	Description	Byte	Oscillator Period
XRL	A, @Ri Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A, #data Exclusive-OR immediate data to Accumulator	2	12
XRL	direct, A Exclusive-OR Accumulator	2	12
XRL	direct, #data Exclusive-OR immediate data to direct byte	3	24
CLR	A Clear	1	12
CPL	A Complement	1	12
RL	A Rotate	1	12
RLC	A Rotate through the Carry	1	12
RR	A Rotate	1	12
RRC	A Rotate through the Carry	1	12
SWAP	A Swap nibbles within the Accumulator	1	12

All mnemonics copyright Intel Corporation 1979

ARITHMETIC OPERATIONS Cont.			
Mnemonic	Description	Byte	Oscillator Period
SUBB	A, @Ri Subtract indirect RAM from Acc with borrow	1	12
SUBB	A, #data Subtract immediate data from Acc with borrow	2	12
INC	A Increment Accumulator	1	12
INC	Rn Increment register	1	12
INC	direct Increment direct byte	2	12
INC	@Ri Increment indirect RAM	1	12
DEC	A Decrement Accumulator	1	12
DEC	Rn Decrement Register	1	12
DEC	direct Decrement direct byte	2	12
DEC	@Ri Decrement indirect RAM	1	12
INC	OPTR Increment Data Pointer	1	24
MUL	AB Multiply A & B	1	48
DIV	AB Divide A by B	1	48

ARITHMETIC OPERATIONS			
Mnemonic	Description	Byte	Oscillator Period
ADD	A, Rn Add register to Accumulator	1	12
ADD	A, direct Add direct byte to Accumulator	2	12
ADD	A, @Ri Add indirect RAM to Accumulator	1	12
ADD	A, #data Add immediate data to Accumulator	2	12
ADDC	A, Rn Add register to Accumulator with Carry	1	12
ADDC	A, direct Add direct byte to Accumulator with Carry	2	12
ADDC	A, @Ri Add indirect RAM to Accumulator with Carry	1	12
ADDC	A, #data Add immediate data to Accumulator with Carry	2	12
SUBB	A, Rn Subtract register from Acc with borrow	1	12
SCUB	A, direct Subtract direct	2	12

Table 8-1. 8051 Instruction Set Summary (Continued)

Mnemonic	Description	Byte	Period
MOV A,Rn	Move register to Accumulator	1	12
MOV A,direct	Move direct byte to Accumulator	2	12
MOV A,@Ri	Move indirect RAM to Accumulator	1	12
MOV A,data	Move immediate data to Accumulator	2	12
MOV Rn,A	Move Accumulator to register	1	12
MOV Rn,direct	Move direct byte to register	2	24
MOV Rn,@Ri	Move indirect RAM to register	1	12
MOV Rn,data	Move immediate data to register	2	12
MOV direct,A	Move Accumulator to direct byte	2	12
MOV direct,Rn	Move register to direct byte	2	24
MOV direct,direct	Move direct byte to direct byte	3	24
MOV direct,@Ri	Move indirect RAM to direct byte	2	24
MOV direct,data	Move immediate data to direct byte	3	24
MOV @Ri,A	Move Accumulator to indirect RAM	1	12
MOV @Ri,direct	Move direct byte to indirect RAM	2	24
MOV @Ri,data	Move immediate data to indirect RAM	2	12

Mnemonic	Description	Byte	Period
MOV DPTR,data16	Load Data Pointer with 16-bit constant	3	24
MOV A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOV A,@A-PC	Move Code byte relative to PC to Acc	1	24
MOV A,@Ri	Move External RAM (8-bit) to Acc	1	24
MOV A,@DPTR	Move External RAM (16-bit) to Acc	1	24
MOV @Ri,A	Move Acc to External RAM (8-bit)	1	24
MOV @DPTR,A	Move Acc to External RAM (16-bit)	1	24
PUSH direct	Push direct byte onto stack	2	24
POP direct	Pop direct byte from stack	2	24
XCH A,Rn	Exchange register with Accumulator	1	12
XCH A,direct	Exchange direct byte with Accumulator	2	12
XCH A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD A,@Ri	Exchange lower order byte of indirect RAM with Acc	1	12

All instructions copyrighted © Intel Corporation 1984

Mnemonic	Description	Byte	Period
CLR C	Clear Carry	1	12
CLR bit	Clear direct bit	2	12
SETB C	Set Carry	1	12
SETB bit	Set direct bit	2	12
CPL C	Complement Carry	1	12
CPL bit	Complement direct bit	2	12
ANL C,bit	AND direct bit to Carry	2	24
ANL C,reg	AND P complement of direct bit to Carry	2	24
ORL C,bit	OR direct bit to Carry	2	24
ORL C,reg	OR complement of direct bit to Carry	2	24
MOV C,bit	Move direct bit to Carry	2	12
MOV bit,C	Move Carry to direct bit	2	24
JC rel	Jump if Carry is set	3	24
JNC rel	Jump if Carry is not set	3	24
JB bit,rel	Jump if direct bit is set	3	24
JNB bit,rel	Jump if direct bit is not set	3	24
JBC bit,rel	Jump if direct bit is set & clear bit	3	24

Mnemonic	Description	Byte	Period
ACALL addr11	Absolute Subroutine Call	2	24
LCALL addr16	Long Subroutine Call	3	24
RET	Return from Subroutine	1	24

Mnemonic	Description	Byte	Period
RETI	Return from Interrupt	1	24
AJMP addr11	Absolute Jump	2	24
LJMP addr16	Long Jump	3	24
SJMP rel	Short Jump (relative addr)	2	24
JMP @A+DPTR	Jump direct relative to the DPTR	1	24
JZ rel	Jump if Zero	2	24
JNZ rel	Jump if Not Zero	2	24
CJNE A,direct,rel	Compare direct to A; if not equal, jump if not equal	3	24
CJNE A,data,rel	Compare immediate to A; if not equal, jump if not equal	3	24
CJNE Rn,data,rel	Compare immediate to register; if not equal, jump if not equal	3	24
CJNE @Ri,data,rel	Compare immediate to indirect RAM; if not equal, jump if not equal	3	24
DJNZ Rn,rel	Decrement register and jump if not zero	3	24
DJNZ direct,rel	Decrement direct byte and jump if not zero	3	24
NOP	No Operation	1	24

All mnemonics copyrighted © Intel Corporation 1984

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

OCTOBER 1975 - REVISED APRIL 1981

- 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)

'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

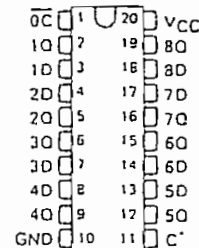
OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

description

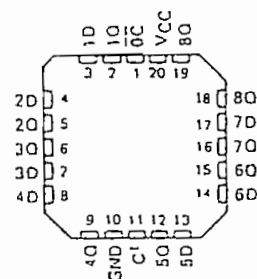
These 8-bit registers feature three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-level drive provide these registers with the capability of being connected directly to and driving the bus lines in 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.

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)



*C for 'LS373 and 'S373, CLK for 'LS374 and 'S374

PRODUCTION DATA
This document contains information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS

POST OFFICE BOX 225012 • DALLAS, TEXAS 75221

3 1021

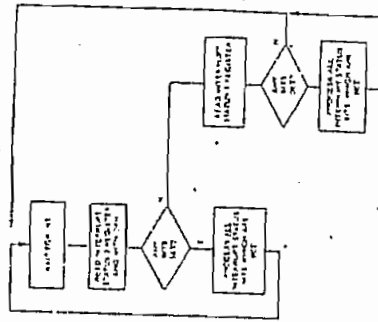
TTL DEVICES

3

8291A FEATURES AND IMPROVEMENTS

The 8291A is an improved design of the 8291 GPIB Talker/Listener. Most of the functions are identical to the 8291, and the pin configuration is unchanged. The 8291A offers the following improvements to the 8291:

1. **EOI** is active with the data as a ninth data bit rather than as a control bit. This is to comply with some additions to the 1975 IEEE-488 Standard incorporated in the 1978 Standard.
2. The **BO** interrupt is not asserted until **RFD** is true. If the Controller asserts **ATN** synchronously, the data is guaranteed to be transmitted. If the Controller asserts **ATN** asynchronously, the SH (Source Handshake) will return to **SIDS** (Source Idle State), and the output data will be cleared. If, if **ATN** is released while the 8291A is addressed to talk, a new **BO** interrupt will be generated. This change fixes 8291 problems which caused data to be lost or repeated and a problem with the **RQS** bit (sometimes cannot be asserted while talking).
- LLOC** and **REMIC** interrupts are setting flipflops rather than toggling flipflops in the interrupt backup register. This ensures that the CPU knows that these state changes have occurred. The actual state can be determined by checking the **LLO** and **REM** status bits in the upper nibble of the Interrupt Status 2 Register.
- DREQ** is cleared by **DACK** (**RD** + **WR**). **DREQ** on the 8291 was cleared only by **DACK** which is not compatible with the 8089 I/O Processor.
- The **INT** bit in Interrupt Status 2 Register and bit 7 of Address 0 Register are duplicates. When software polling is used to check interrupts, poll **INT** in Address 0 Register, instead of Interrupt Status 2 Register. Then, asynchronous status reads and interrupts will not lose interrupts.
- Lockout mechanism prevents all interrupt status to be set in both interrupt status registers. A check-up stores any bits, and launches onto the Interrupt Status Registers after the register with the bits is read.
- An Interrupt Status Register is read, all the interrupt bits should be checked before disregarding the bits. A recommended way to handle this 8291A on **END** is in the flow chart below.
- On the 8291, **BO** could cease to occur upon **IFC** going false if **IFC** occurred asynchronously. On the 8291A, **BO** continues to occur after **IFC** has gone false even if it arrived asynchronously. This can be used to set a flag in the user's software which will permit special routines to be executed for each device. It could be included as part of a normal initialization procedure as the first step after a chip reset.
- User's software can distinguish between the 8291 and the 8291A as follows:
 - a) **pon** (**COH** to register 5)
 - b) **RESET** (**O2H** to register 5)
 - c) Read Interrupt Status 1 Register. If **BO** interrupt is set, the device is the 8291. If **BO** is clear, it is the 8291A.
- To avoid confusion between holdoff on **DAV** versus **RFD** if a device is readdressed from a talker to a listener role or vice-versa during a holdoff, the "Holdoff on Source Handshake" has been eliminated. Only "Holdoff on Acceptor Handshake" is available.
- The **rsv** local message is cleared automatically upon exit from **SPAS** if (**APRS**, **STRS**, **SPAS**) occurred. The automatic resetting of the bit after the serial poll is complete simplifies the service request software.
- The **SPASC** interrupt on the 8291 has been replaced by the **SPC** (Serial Poll Complete) interrupt on the 8291A. **SPC** interrupt is set on exit from **SPAS** if **APRS**, **STRS**, **SPAS** occurred, indicating that the controller has read the bus status byte after the 8291A requested service. This **SPASC** interrupt was ambiguous because a controller could enter **SPAS** and exit **SPAS** generating two **SPASC** interrupts without reading the serial poll status byte. The **SPC** interrupt also simplifies the CPU's software by eliminating the interrupt when the serial poll is half way done.
- The **rtl** Auxiliary Command in the 8291 has been replaced by **Set** and **Clear** **rtl** Commands in the 8291A. Using the new commands, the CPU has the flexibility to extend the length of local mode or leave it as a short pulse as in the 8291.
- A holdoff **RFD** on **GET**, **SDC**, and **DCL** features has been added to prevent additional bus activity while the CPU is responding to any of these commands. The feature is enabled by a new bit (**B₁**) in the Auxiliary Register B.



6. The 8291A's Send **EOI** Auxiliary Command works on any byte including the first byte of a message. The 8291 did not assert **EOI** after this command for a one byte message nor on two consecutive bytes.

7. To avoid confusion between holdoff on **DAV** versus **RFD** if a device is readdressed from a talker to a listener role or vice-versa during a holdoff, the "Holdoff on Source Handshake" has been eliminated. Only "Holdoff on Acceptor Handshake" is available.

8. The **rsv** local message is cleared automatically upon exit from **SPAS** if (**APRS**, **STRS**, **SPAS**) occurred. The automatic resetting of the bit after the serial poll is complete simplifies the service request software.

9. The **SPASC** interrupt on the 8291 has been replaced by the **SPC** (Serial Poll Complete) interrupt on the 8291A. **SPC** interrupt is set on exit from **SPAS** if **APRS**, **STRS**, **SPAS** occurred, indicating that the controller has read the bus status byte after the 8291A requested service. This **SPASC** interrupt was ambiguous because a controller could enter **SPAS** and exit **SPAS** generating two **SPASC** interrupts without reading the serial poll status byte. The **SPC** interrupt also simplifies the CPU's software by eliminating the interrupt when the serial poll is half way done.

10. The **rtl** Auxiliary Command in the 8291 has been replaced by **Set** and **Clear** **rtl** Commands in the 8291A. Using the new commands, the CPU has the flexibility to extend the length of local mode or leave it as a short pulse as in the 8291.

11. A holdoff **RFD** on **GET**, **SDC**, and **DCL** features has been added to prevent additional bus activity while the CPU is responding to any of these commands. The feature is enabled by a new bit (**B₁**) in the Auxiliary Register B.

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
D_0-D_7	12-19	I/O	Data Bus Port: To be connected to microprocessor data bus.
RS_0-RS_7	21-23	I	Register Select: Inputs, to be connected to three non-multiplexed microprocessor address bus lines. Select which of the 8 internal read (write) registers will be read from (written into) with the execution of RD (WR).
\overline{CS}	8	I	Chip Select: When low, enables reading from or writing into the register selected by RS₀-RS₇ .
\overline{RD}	9	I	Read Strobe: When low with CS or DACK low, selected register contents are read.
\overline{WR}	10	I	Write Strobe: When low with CS or DACK low, data is written into the selected register.
INT (INT)	11	O	Interrupt Request: To the microprocessor, set high for request and cleared when the appropriate register is accessed by the CPU. May be software configured to be active low.
DREQ	6	O	Data Request: Normally low, set high to indicate byte output or byte input in data mode.

12. On the 8291, **BO** could cease to occur upon **IFC** going false if **IFC** occurred asynchronously. On the 8291A, **BO** continues to occur after **IFC** has gone false even if it arrived asynchronously. This can be used to set a flag in the user's software which will permit special routines to be executed for each device. It could be included as part of a normal initialization procedure as the first step after a chip reset.

13. User's software can distinguish between the 8291 and the 8291A as follows:

- a) **pon** (**COH** to register 5)
- b) **RESET** (**O2H** to register 5)
- c) Read Interrupt Status 1 Register. If **BO** interrupt is set, the device is the 8291. If **BO** is clear, it is the 8291A.

Symbol	Pin No.	Type	Name and Function
\overline{DACK}	7	I	DMA Acknowledge: When low, resets DREQ and selects data in data register for DMA transfer (actual transfer done by RD , WR pulse). Must be high if DMA is used.
TRIG	5	O	Trigger Output: Normally low; generates a trigger pulse with 1 μ sec minimum width in response to GET bus command or TRIG auxiliary command.
CLOCK	3	I	External Clock: Input used only for T₁ data generator. May be a speed in 1-8 MHz range.
RESET	4	I	Reset Input: When high forces the device into "idle" (initialization) mode. The device will remain "idle" until released by the microprocessor, with the "Immediate Execute" port local message.
$\overline{DIO_0}-\overline{DIO_7}$	28-35	I/O	8-Bit GPIB Data Port: Used for bidirectional data by transfer between 8291 and GPIB via non-inverting external line transceiver.
DAV	36	I/O	Data Valid: GPIB handshake control line. Indicates the availability and validity of information in the DIO₀-DIO₇ and EOI lines.
\overline{NRFD}	37	I/O	Not Ready for Data: GPIB handshake control line. Indicates the condition readiness of device(s) connected to the bus to accept data.
\overline{NDAC}	38	I/O	Not Data Accepted: GPIB handshake control line. Indicates the condition of acceptance of data by device(s) connected to the bus.
ATN	26	I	Attention: GPIB command line. Specifies how data DIO lines are to be interpreted.
IFC	24	I	Interface Clear: GPIB command line. Places the interface functions in known quiescent state.

Table 1. Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
\bar{Q}	27	O	Service Request: GPIB command line. Indicates the need for attention and requests an interruption of the current sequence of events on the GPIB.
\bar{R}	25	I	Remote Enable: GPIB command line. Selects (in conjunction with other messages) remote or local control of the device.
	39	I/O	End or Identify: GPIB command line. Indicates the end of a multiple byte transfer sequence or, in conjunction with ATN, addresses the device during a polling sequence.
1	1	O	External Transceivers Control Line: Set high to indicate output data/signals on the DIO, -DIO, and DAV lines and input signals on the IIRFD and NDAC lines (active source handshake). Set low to indicate input data/signals on the DIO, -DIO, and DAV lines and output signals on the IIRFD and NDAC lines (active acceptor handshake).
2	2	O	External Transceivers Control Line: Set to indicate output signals on the EO line. Set low to indicate expected input signal on the EO line during parallel poll.
	40	P.S.	Positive Power Supply: (SV = 10%).
	20	P.S.	Circuit Ground Potential.

alls on the 8291A pins are specified with positive logic, and IEEE 488 specifies negative logic on its 16 signal lines. The data is inverted once from D₀-D₇ to DIO₀-DIO₇, and external bus transceivers should be used.

GENERAL PURPOSE INTERFACE (GPIB)

General Purpose Interface Bus (GPIB) is defined in the IEEE Standard 488-1978 "Digital Interface for Programmable Instrumentation." Although a knowledge of this standard is assumed,

ence. Also, Tables 2 and 3 reference the interface state mnemonics and the interface messages respectively. Modified state diagrams for the 8291A are presented in Appendix A.

General Description

The 8291A is a microprocessor-controlled device designed to interface microprocessors, e.g., 8048/49, 8051, 8080/85, 8086/88 to the GPIB. It implements all of the interface functions defined in IEEE 488 Standard except for the controller function. If an implementation of the Standard's Controller

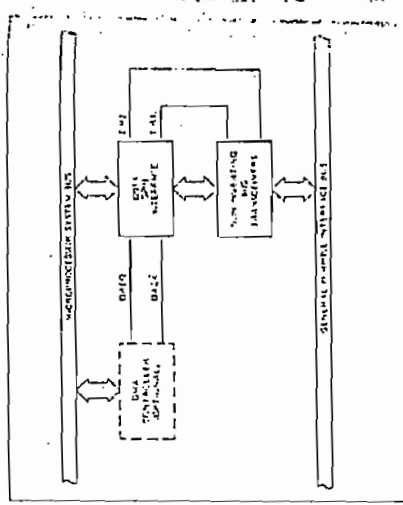


Figure 3. 8291A System Diagram

ler is desired, it can be connected with an interface to form a complete interface.

The 8291A handles communication between microprocessor-controlled device and the GPIB. capabilities include data transfer, handshake protocol, talker/listener addressing, protocol, device clearing and triggering, service request both serial and parallel polling. In most processes it does not disturb the microprocessor unless JEA has arrived (input buffer full) or has to be (output buffer empty).

The 8291A architecture includes 13 registers of these registers may be written into by the processor. The other eight registers may be

write registers is for direct data transfers. The rest of the write registers control the various features of the chip, while the rest of the read registers provide the microprocessor with a monitor of GPIB states, various bus conditions, and device conditions.

GPIB Addressing

Each device connected to the GPIB must have at least one address whereby the controller device in charge of the bus can configure it to talk, listen, or send status. An 8291A implementation of the GPIB offers the user three alternative addressing modes for which the device can be initialized for each application. The first of these modes allows for the device to have two separate primary addresses. The second mode allows the user to implement a single talker/listener with a two byte address (primary address + secondary address). The third mode again allows for two distinct addresses but in this instance, they can each have a ten-bit address (5 low-order bits of each of two bytes). However, this mode requires that the secondary addresses be passed to the microprocessor for verification. These three addressing schemes are described in more detail in the discussion of the Address Registers.

Table 2. IEEE 488 Interface State Mnemonics

Mnemonic	State Represented
ACDS	Accept Data State
ACRS	Accept Ready State
AIDS	Accept Idle State
ANRS	Accept Not Ready State
APRS	Affirmative Poll Response State
AWNS	Accepter Wait for New Cycle State
CACS	Controller Active State
CAOS	Controller Addressed State
CAVS	Controller Active Wait State
CIDS	Controller Idle State
CPPS	Controller Parallel Poll State
CPWS	Controller Parallel Poll Wait State
CSBS	Controller Standby State
CSNS	Controller Service Not Requested State
CSRS	Controller Service Requested State
CSVS	Controller Synchronous Wait State
CTRS	Controller Transfer State
DCAS	Device Clear Active State
DCIS	Device Clear Idle State
DTAS	Device Trigger Active State
DTIS	Device Trigger Idle State
LACS	Listener Active State
LADS	Listener Addressed State
LIDS	Listener Idle State
LOCS	Local State
LPAS	Listener Primary Address State
LPLS	Listener Primary Idle State
LWLS	Local With Lockout State
PPRS	Parallel Poll Ready State

Figure 4. Interface Capabilities and Bus Structure

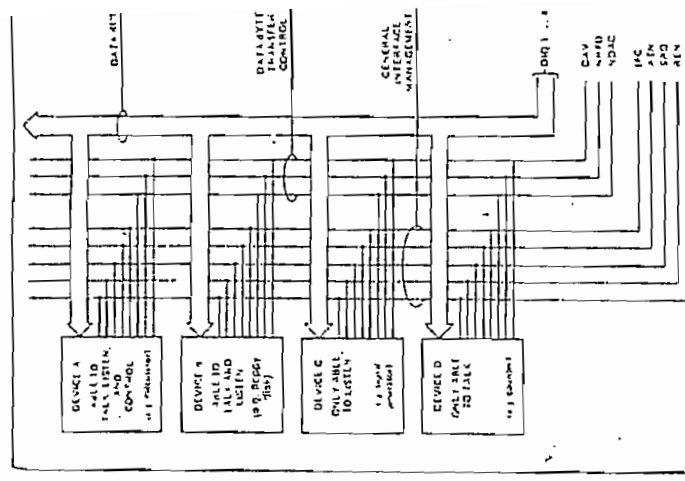


Table 3. IEEE 488 Interface Message Reference List

Memoronic	Message	Interface Function(s)
LOCAL MESSAGES RECEIVED (By Interface Functions)		
'gto	go to standby	C
'lst	listen only	PP
'lon	listen only	L, LE
'lpe	local poll enable	PP
'nba	new byte available	SH
'pon	power on	SH, AH, T, TE, L, LE, SR, RL, PP, C
'rdy	ready	AH
'rpp	request parallel poll	C
'rsc	request system control	C
'rvs	request service	SR
'rl	return to local	RL
'sfc	send interface clear	C
'sre	send remote enable	C
'tca	take control asynchronously	C
'tcs	take control synchronously	AH, C
'lon	talk only	T, TE
REMOTE MESSAGES RECEIVED		
ATN	Attention	SH, AH, T, TE, L, LE, PP, C
DAB	Data Byte	(via L, LE)
DAC	Data Accepted	SH
DAV	Data Valid	AH
DCL	Device Clear	DC
END	End	(via L, LE)
GET	Group Execute Trigger	DT
GTL	Go to Local	RL
IDY	Identify	L, LE, PP
IFC	Interface Clear	T, TE, L, LE, C
LLO	Local Lockout	RL
MLA	My Listen Address	L, LE, RL, T, TE
MISA	My Secondary Address	TE, LE, RL
MTA	My Talk Address	T, TE, L, LE
OSA	Other Secondary Address	TE
OTA	Other Talk Address	T, TE
PCG	Primary Command Group	TE, LE, PP
PPC	Parallel Poll Configure	PP
[PPD]	Parallel Poll Disable	PP
[PPE]	Parallel Poll Enable	PP
[PPR]	Parallel Poll Response N	(via C)
[PPU]	Parallel Poll Unconfigure	PP
REN	Remote Enable	PP
RFD	Ready for Data	SH
RQS	Request Service	(via L, LE)
[SDC]	Select Device Clear	DC
SPO	Serial Poll Disable	T, TE
SPE	Serial Poll Enable	T, TE
[SOR]	Service Request	(via C)
STB	Status Byte	(via L, LE)
[TCT]	Take Control	C
UNL	Unlisten	L, LE

NOTE:
1. These messages are handled only by lat-13, 8292

Table 3. (Cont'd)
IEEE 488 Interface Message Reference List

Memoronic	Message	Interface Function(s)
REMOTE MESSAGES SENT		
ATN	Attention	C
DAB	Data Byte	(via T, TE)
DAC	Data Accepted	AH
DAV	Data Valid	SH
DCL	Device Clear	(via C)
END	End	(via T)
GET	Group Execute Trigger	(via C)
GTL	Go to Local	(via C)
IDY	Identify	C
IFC	Interface Clear	C
LLO	Local Lockout	(via C)
MLA or [MLA]	My Listen Address	(via C)
MISA or [MISA]	My Secondary Address	(via C)
MTA or [MTA]	My Talk Address	(via C)
OSA	Other Secondary Address	(via C)
OTA	Other Talk Address	(via C)
PCG	Primary Command Group	(via C)
PPC	Parallel Poll Configure	(via C)
[PPD]	Parallel Poll Disable	(via C)
[PPE]	Parallel Poll Enable	(via C)
[PPR]	Parallel Poll Response N	PP
[PPU]	Parallel Poll Unconfigure	(via C)
REN	Remote Enable	C
RFD	Ready for Data	AH
RQS	Request Service	T, TE
[SDC]	Select Device Clear	(via C)
SPO	Serial Poll Disable	(via C)
SPE	Serial Poll Enable	(via C)
SRO	Service Request	SR
STB	Status Byte	(via T, TE)
TCT	Take Control	(via C)
UNL	Unlisten	(via C)

NOTE:
3. All Controller messages must be sent via Intel's 8292.

8291A Registers

A bit-by-bit map of the 16 registers on the 8291A is presented in Figure 5. A more detailed explanation of each of these registers and their functions follows. The access of these registers by the microprocessor is accomplished by using the CS, RD, WR, and RS₀-RS₂ pins.

Register	CS	RD	WR	RS ₀ -RS ₂
All Read Registers	0	0	1	CCC

Data Registers

D17	D16	D15	D14	D13	D12	D11	D10
-----	-----	-----	-----	-----	-----	-----	-----

DATA-IN REGISTER (DR)

D07	D06	D05	D04	D03	D02	D01	D00
-----	-----	-----	-----	-----	-----	-----	-----

DATA-OUT REGISTER (OW)

The Data-In Register is used to move data from the GPIB to the microprocessor or to memory when the 8291A is addressed to listen. Incoming information

gister. The RFD (Ready for Data) message is held in the register until the byte is removed from the data in register 91A then completes the handshake automatically. RFD holdoff mode (see Auxiliary Register A), the handshake is not finished until a command is sent which releases the holdoff. In this way, the same byte may be read several times, or an overzealous talker may be held off until all available data has been processed.

When the 8291A is addressed to talk, it uses the Data-Out register to move data onto the GPIB. After BO interrupt is received and a byte is written to the register, the 8291A initiates and completes the handshake while sending the byte out over the bus. The BO interrupt disable mode, the user should wait until BO is active before writing to the register. In DMA mode, this will happen automatically. A flag of the Data-In Register does not destroy the information in the Data-Out Register.

Interrupt Registers

CPT	APT	GET	END	DEC	ERR	BO	BI
-----	-----	-----	-----	-----	-----	----	----

INTERRUPT STATUS 1 (1R)

INT	SPAS	LLOC	REM	SPC	LLOC	REMIC	ADSC
-----	------	------	-----	-----	------	-------	------

INTERRUPT STATUS 2 (2R)

CPT	APT	GET	END	DEC	ERR	BO	BI
-----	-----	-----	-----	-----	-----	----	----

INTERRUPT ENABLE 1 (1W)

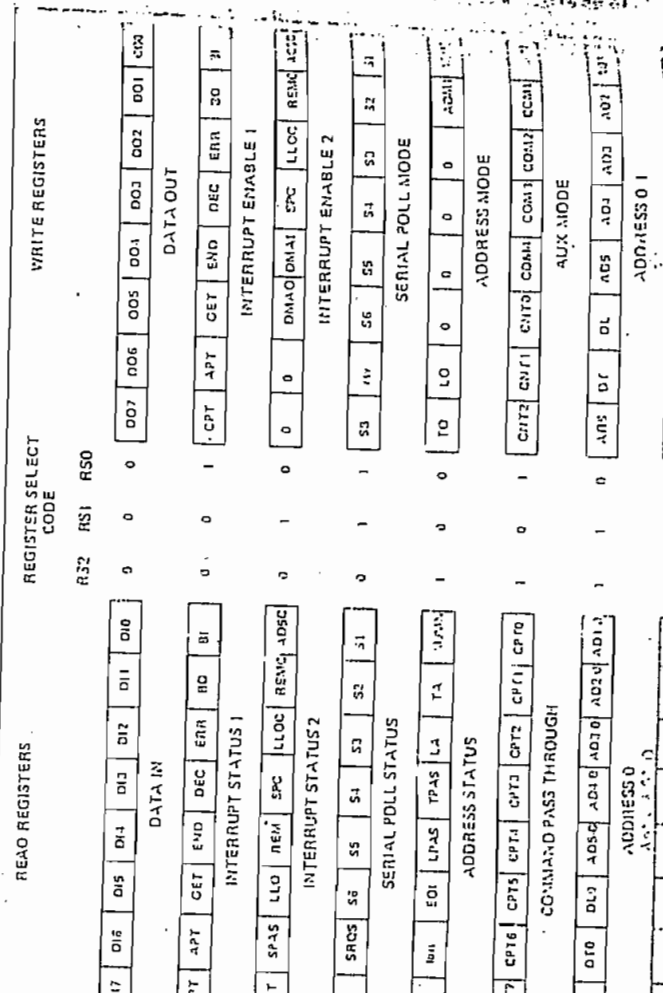
0	0	DMAO	DMAI	SPC	LLOC	REMIC	ADSC
---	---	------	------	-----	------	-------	------

INTERRUPT ENABLE 2 (2W)

INT	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0
-----	-----	-----	-------	-------	-------	-------	-------

ADDRESS 0 REGISTER

Figure 5. 8291A Registers



The 8291A can be configured to generate an interrupt to the microprocessor upon the occurrence of any of 12 conditions or events on the GPIB. Upon receipt of an interrupt, the microprocessor must read the Interrupt Status Registers to determine which event has occurred, and then execute the appropriate service routine (if necessary). Each of the 12 interrupt status bits has a matching enable bit in the interrupt enable registers. These enable bits are used to select the events that will cause the INT pin to be asserted. Writing a logic "1" into any of these bits enables the corresponding interrupt status bits to generate an interrupt. Bits in the Interrupt Status Registers are set regardless of the states of the enable bits. The Interrupt Status Registers are then cleared upon being read or when a local (power-on) message is executed. If an event occurs while one of the Interrupt Status Registers is being read, the event is held until after its register is cleared and then placed in the register.

Table 4. Interrupt Bits

CPT	APT	GET	END	DEC	ERR	BO	BI
Indicates Undefined Commands							
Set by (TPAS + LPAS) - SCG - ACDS - MODE 3							
GET	Set by OTAS						
END	Set by (EOS + EOI) - LACS						
DEC	Set by DCAS						
ERR	Set by TACS - nba - DAC - RFD						
BO	TACS - (SWINS + SGNS)						
BI	Set by LACS - ACDS						
INT	Shows status of the INT pin						
SPAS	The device has been enabled for a serial poll						
LLO	The device is in local lock out state. (LWLS + RWLS)						
REM	The device is in a remote state. (REMIS + RVLS)						
SPC	Serial Poll Complete interrupt.						
LLOC	Local lock out change interrupt.						
REMIC	Remote/Local change interrupt.						
ADSC	Address status change interrupt.						

The mnemonics for each of the bits in these registers and a brief description of their respective functions appears in Table 4. This table also indicates how each of the interrupt bits is set.

NOTE: The INT bit in the Address 0 Register is a duplicate of the INT bit in the Interrupt Status 2 Register. It is only a status bit. It does not generate interrupts and thus does not have a corresponding enable bit.

The BO and BI interrupts enable the user to perform data transfer cycles. BO indicates that a data byte should be written to the Data Out Register. It is set when TACS - (SWINS + SGNS) - RFD. It is reset when data byte is written, ATN is asserted, or the 8291A exits TACS. Data should never be written to the Data Out Register before BO is set. Similarly, BI is set when an input byte is accepted into the 8291A and reset when the microprocessor reads the Data Out Register. BO and BI are also reset by pon (power-on local message) and by a read of the Interrupt Status 2 Register.

An undefined command has been received.
A secondary address must be passed through to the microprocessor for recognition.
A group execute trigger has occurred.
An EOS or EOI message has been received.
Device Clear Active State has occurred.
Interface error has occurred; no listeners are active.
A byte should be output.
A byte has been input.

These are status only. They will not generate interrupts, nor do they have corresponding mask bits.

Serial Poll Complete interrupt.
Local lock out change interrupt.
Remote/Local change interrupt.
Address status change interrupt.

SPAS - SPAS if APRS - STRS - SPAS was true
LLO - LLO
REMIC - Local
ADSC - Addressed Unaddressed

The APT interrupt bit indicates to the processor that a secondary address is available in the CPT register for validation. This interrupt will only occur if Mode 3 addressing is in effect. (Refer to the section on addressing.) In Mode 2, secondary addresses will be recognized automatically on the 8291A. They will be ignored in Mode 1.

The CPT interrupt bit flags the occurrence of an undefined command and of all secondary commands following an undefined command. The Command Pass Through feature is enabled by the B0 bit of Auxiliary Register B. Any message not decoded by the 8291A (not included in the state diagrams in Appendix B) becomes an undefined command. Note that any addressed command is automatically ignored when the 8291A is not addressed.

Undefined commands are read by the CPU from the Command Pass Through register of the 8291A. This register reflects the logic levels present on the data lines at the time it is read. If the CPT feature is enabled, the 8291A will hold off the handshake until this register is read.

An especially useful feature of the 8291A is its ability to generate interrupts from state transitions in the interface functions. In particular, the lower 3 bits of the Interrupt Status 2 Register, if enabled by the corresponding enable bits, will cause an interrupt upon changes in the following states as defined in the IEEE-488 Standard.

- Bit 0 ADSC change in LIDS or TIDS or MJM?
- Bit 1 REMIC change in LOCS or REMIS
- Bit 2 LLOC change in LWLS or RWLS

The upper 4 bits of the Interrupt Status 2 Register are available to the processor as status bits. Thus, if one of the bits 1 and 2 generates an interrupt indicating status change has taken place, the corresponding status bit (bits 4 and 5) may be read to determine what the new state is. To determine the nature of the change in addressed status (bit 0) the Address Status Register is available to be read. The SPC interrupt (bit 3 in Interrupt Status 2) is set upon exit from SPC. If APRS-STRS:SPAS occurred which indicates that the GPIB controller has read the bus serial poll (bit 3).

The controller may read the status byte later, and the byte will contain the last status the 8291A's CPU wrote to the Serial Poll Mode Register, but the SRQS bit will not be set and no interrupt will be generated. Finally, bit 7 monitors the state of the 8291A INT pin. Logically, it is an OR of all enabled interrupt status bits. One should note that bits 4-7 of the Interrupt Status 2 Register do not generate interrupts, but are available only to be read as status bits by the processor. Bit 7 in Interrupt Status 2 is duplicated in Address 0 Register, and the latter should be used when polling for interrupts to avoid losing one of the interrupts in Interrupt Status 2 Register.

Bits 4 and 5 (DMAI, DMAO) of the Interrupt Mask 2 Register are available to enable direct data transfers between memory and the GPIB; DMAI (DMA in) enables the DREQ (DMA request) pin of the 8291A to be asserted upon the occurrence of BI. Similarly, DMAO (DMA out) enables the DREQ pin to be asserted upon the occurrence of BO. One might note that the DREQ pin may be used as a second interrupt output pin, monitoring BI and/or BO and enabled by DMAI and DMAO. One should note that the DREQ pin is not affected by a read of the Interrupt Status 1 Register. It is reset whenever a byte is written to the Data Out Register or read from the Data In Register.

To ensure that an interrupt status bit will not be cleared without being read, and will not remain unclear after being read, the 8291A implements a special interrupt handling procedure. When an enabled interrupt bit is set in either of the Interrupt Status Registers, the input of the registers are blocked until the set bit is read and reset by the microprocessor. Thus, potential problems arise when interrupt status changes while the register is being blocked. However, the 8291A stores all new interrupts in a temporary register and transfers them to the appropriate Interrupt Status Register after the interrupt has been reset. This transfer takes place only if the corresponding bits were read as zeroes.

Serial Poll Registers

S0	SRQS	S6	S5	S4	S3	S2	S1
----	------	----	----	----	----	----	----

SERIAL POLL STATUS (SR)

Address Registers

Ion	Ion	EOI	LPAS	TPAS	LA	TA	NJMN
-----	-----	-----	------	------	----	----	------

ADDRESS STATUS (4R)

INT	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0
-----	-----	-----	-------	-------	-------	-------	-------

ADDRESS 0 (6R)

X	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1
---	-----	-----	-------	-------	-------	-------	-------

ADDRESS 1 (7R)

TO	LO	0	0	0	0	ADN1	ADM0
----	----	---	---	---	---	------	------

ADDRESS MODE (4W)

The Address Mode Register is used to select one of the five modes of addressing available on the 8291A. It determines the way in which the 8291A uses the information in the Address 0 and Address 1 registers.

In Mode 1, the contents of the Address 0 Register constitute the "Major" talker/listener address while the Address 1 Register represents the "Minor" talker/listener address. In applications where only one address is needed, the major talker/listener is used, and the minor talker/listener should be disabled. Loading an address via the Address 0/1 Register into Address Registers 0 and 1 enables the major and minor talker/listener functions respectively.

In Mode 2 the 8291A recognizes two sequential address bytes: a primary followed by a secondary. Both address bytes must be received in order to enable the device to talk or listen. In this manner, Mode 2 addressing implements the extended talker and listener functions as defined in IEEE-188.

Mode 3, the 8291A handles addressing just as it does in Mode 1, except that each Major or Minor address must be followed by a secondary address. All secondary addresses must be verified by the microprocessor when Mode 3 is used. When the 8291A is in TPAS or LPAS (talker/listener primary or listener state), and it does not recognize the byte 0x0F, an APT interrupt is generated (see Section 4.1 on Interrupt Registers) and the byte is available to the CPT (Command Pass Through) Register. The microprocessor must read the CPT Register and write one of the following responses to the Auxiliary Mode Register:

1. implies a non-valid secondary address
2. implies a valid secondary address

The TO bit generates the local ton (talker) message and sets the 8291A to a talk-only mode. This mode allows the device to operate as an interface system without a controller.

Setting the LO bit generates the local ton (talker) message and sets the 8291A to a listen-only mode. This mode allows the device to operate as a listener in an interface system without a controller. The above bits may also be used by a controller in charge to set itself up for remote command or data communication.

The mode of addressing implemented by the 8291A may be selected by writing one of the following bytes to the Address Mode Register.

Register Contents	Mode
10000000	Enable talk only mode (ton)
01000000	Enable listen only mode (lon)
11000000	The 8291 may talk to itself
00000001	Mode 1, (Primary-Primary)
00000010	Mode 2 (Primary-Secondary)
00000011	Mode 3 (Primary/APT-Primary/APT)

The Address Status Register contains information used by the microprocessor to handle its own addressing. This information includes status bits that monitor the address state of each talker/listener, "ton" and "lon" flags which indicate the talk and listen only states, and an EOI bit which, when set, signifies that the END message came with the last data byte. LPAS and TPAS indicate that the listener or talker primary address has been received. The microprocessor can use these bits when the secondary address is passed through to determine whether the 8291A is addressed to talk or listen. The LA (listener addressed) bit will be set when the 8291A is in LACS (Listener Active State) or in LADS (Listener Addressed State). Similarly, the TA (Talker Addressed) bit will be set to indicate TACS or TADS. The MJMN bit is used to determine whether the information in the other bits applies to the Major or Minor talker/listener. It is set to "1" when the Minor talker/listener is addressed. It should be noted that only one talker/listener may be active at any one time. Thus, the MJMN bit will indicate which, if either, of the talker/listeners is addressed or active.

The Address 0/1 Register is used for specifying the device's addresses according to the format selected in the Address Mode Register. Five bit addresses may be loaded into the Address 0 and Address 1 Registers by writing into the Address 0/1 Register. The ARS bit is used to select which of these registers the other seven bits will be loaded into. The DT and DL bits may be used to disable the talker or listener function at the address signified by the other bits.

bits. When Mode 1 addressing is used and only one primary address is desired, both the talker and the listener should be disabled at the Minor address.

As an example of how the Address 0/1 Register might be used, consider an example where two primary addresses are needed in the device. The Major primary address will be selectable only as a talker and the Minor primary address will be selectable only as a listener. This configuration of the 8291A is formed by the following sequence of writes by the microprocessor.

Operation	CS	RD	WR	Data	RS2-RS0
1. Select Addressing Mode 1	0	1	0	00000001	100
2. Load major address into Address 0 Register with listener function disabled.	0	1	0	001AAAAA	110
3. Load minor address into Address 1 Register with talker function disabled.	0	1	0	110ABBBB	110

At this point, the addresses AAAA and BBBB are stored in the Address 0 and Address 1 Registers respectively, and are available to be read by the microprocessor. Thus, it is not necessary to store any address information elsewhere. Also, with the information stored in the Address 0 and Address 1 Registers, processor intervention is not required to recognize addressing by the controller. Only in Mode 3, where secondary addresses are passed through, must the processor intervene in the addressing sequence.

The Address 0 Register contains a copy of bit 7 of the Interrupt Status 2 Register (INT). This is to be used when polling for interrupts. Software should poll register 6 checking for INT (bit 7) to be set. When INT is set, the Interrupt Status Register should be read to determine which interrupt was received.

Command Pass Through Register							
CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0

The Command Pass Through Register is used to transfer undefined 8-bit remote message codes from the GPIB to the microprocessor. When the CPT feature is enabled (bit 50 in Auxiliary Register B), any message not decoded by the 8291A becomes an undefined command. When Mode 3 addressing is used secondary addresses are also passed through

the CPT Register. In either case, the 8291A will hand off the handshake until the microprocessor reads this register and issues the VSCMD auxil command.

The CPT and APT interrupts flag the availability of undefined commands and secondary addresses in the CPT Register. The details of these interrupts explained in the section on Interrupt Registers.

An added feature of the 8291A is its ability to handle undefined secondary commands following undefined primaries. Thus, the number of available commands for future IEEE-488 definition is increased; one undefined primary command followed by a sequence of as many as 32 secondary commands can be processed. The IEEE-188 Standard does not permit users to define their own commands, but upgrades of the standard are thus provided for.

The recommended use of the 8291A's undefined command capabilities is for a controller-configured Parallel Poll. The PPC message is an undefined primary command typically followed by PPE, an undefined secondary command. For details on this procedure, refer to the section on Parallel Poll Protocol.

Auxiliary Mode Register

CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0
------	------	------	------	------	------	------	------

AUX MODE (5V)

CNT0-2:CONTROL BITS
COM0-4:COMMAND BITS

The Auxiliary Mode Register contains a three-bit control field and a five-bit command field. It is used for several purposes on the 8291A:

1. To load "hidden" auxiliary registers on the 8291A.
2. To issue commands from the microprocessor to the 8291A.
3. To preset an internal counter used to generate T1, delay in the Source Handshake function, as defined in IEEE-188.

Table 5 summarizes how these tasks are performed with the Auxiliary Mode Register. Note that the three control bits determine how the five command bits are used.

Table 5

CODE	CONTROL COMMAND BITS	COMMAND
000	00000	Execute auxiliary command
001	00000	CCCC Preset internal counter to match external clock frequency of DDDD MHz
100	00000	DDDD DDDD binary representation of 1 to 8 MHz
101	00000	Write DDDDD into auxiliary register A
011	USP ₃ P ₂ P ₁	Write DDDDD into auxiliary register B Enabled/disabled parallel poll either in response to remote messages (PPC followed by PPS or PPD) or as a local PPS message. (Enable if U = 0, disable if U = 1.)

AUXILIARY COMMANDS

Auxiliary commands are executed by the 8291A whenever 000000000 is written into the Auxiliary Register, where CCCC is the 4-bit command

—Immediate Execute pon: This command sets the 8291A to a power up state (local pon as defined in IEEE-488).

Following conditions constitute the power up state: interrupt status bits are disabled.

8291A is designed to power up in certain states defined in the IEEE-488 state diagrams. Thus, during states are in effect in the power up state, AIDS, TIDS, LIDS, NIDS, LOCS, and

30" pon is an immediate execute command (else). It is also used to release the "initialize" state by either an external reset pulse or a Chip Reset command.

Reset (Initialize): This command has the effect as a pulse applied to the Reset pin. (See section on Reset Procedure.)

Handshake: This command finishes a state that was stopped because of a holdoff after to Auxiliary Register A.)

0100—Trigger: A "Group Execute Trigger" is forced by this command. It has the same effect as a GET command issued by the controller-in-charge of the GPIB, but does not cause a GET interrupt.

0101, 1101—Clear/Set rti: These commands correspond to the local rti message as defined by the IEEE-488. The 8291A will go into local mode when a Set rti Auxiliary Command is received if local lockout is not in effect. The 8291A will exit local mode after receiving a Clear rti Auxiliary Command. If the 8291A is addressed to listen.

0110—Send EOI: The EOI line of the 8291A may be asserted with this command. The command causes EOI to go true with the next byte transmitted. The EOI line is then cleared upon completion of the handshake for that byte.

0111, 1111—Non Valid/Valid Secondary Address or Command (VSCMO): This command informs the 8291A that the secondary address received by the microprocessor was valid or invalid (0111 = invalid, 1111 = valid). If Mode 3 addressing is used, the processor must field each extended address and respond to it, or the GPIB will hang up. Note that the COM3 bit is the invalid/valid flag.

The valid (1111) command is also used to tell the 8291A to continue from the command-pass-through-state, or from RFD holdoff on GET, SET, or DCL.

1000—pon: This command puts the 8291A into the pon (power on) state and holds it there. It is similar to a Chip Reset except none of the Auxiliary Mode Registers are cleared. In this state, the 8291A does not participate in any bus activity. An Immediate Execute pon releases the 8291A from the pon state and permits the device to participate in the bus activity again.

0001, 1001—Parallel Poll Flag (local "ist" message): This command sets (1001) or clears (0001) the parallel poll flag. A "1" is sent over the assigned data bus (P₁R = Parallel Poll Response true) only if the parallel poll flag matches the sense bit from the local message (or indirectly from the PPE message). For more complete description of the Parallel Poll features and procedures refer to the section on Parallel Poll Protocol.

INTERNAL COUNTER

The internal counter determines the delay time allowed for the setting of data on the DIO lines. The delay time is defined as T₁ in IEEE-488 and appears in the Source Handshake state diagrams.

SDYS and STRS. As such, DAV is asserted T₁ after the DIO lines are driven. Consequently, T₁ is a major factor in determining the data-transfer rate of the 8291A over the GPIB (T₁ = TWRDV2-TWRD15).

When open-collector transceivers are used for connection to the GPIB, T₁ is defined by IEEE-488 to be 2μsec. By writing 00100000 into the Auxiliary Mode Register, the counter is preset to match a 1c MHz clock input, where DDDD is the binary representation of N_F (1 ≤ N_F ≤ 8, N_F = (DDDD)₂). When N_F = 1c, a 2μsec T₁ delay will be generated before each DAV is asserted.

$$T_{1(\mu\text{sec})} = \frac{2N_F}{f_c} + t_{\text{sync}}, 1 \leq N_F \leq 8$$

t_{sync} is a synchronization error, greater than zero and smaller than the larger of T clock-high and T clock-low. (For a 50% duty cycle clock, t_{sync} is less than half the clock cycle).

If it is necessary that T₁ be different from 2μsec, N_F may be set to a value other than 1c. In this manner, data transfer rates may be programmed for a given system. In small systems, for example, where transfer rates exceeding GPIB specifications are required, one may set N_F < 1c and decrease T₁.

When tri-state transceivers are used, IEEE-488 allows a higher transfer rate (lower T₁). Use of the 8291A with such transceivers is enabled by setting B₁ in Auxiliary Register B. In this case, setting N_F = 1c causes a T₁ delay of 2μsec to be generated for the first byte transmitted — all subsequent bytes will have a delay of 500 nsec.

$$T_1(\text{High Speed}) \mu\text{sec} = \frac{N_F}{2f_c} + t_{\text{sync}}$$

Thus, the shortest T₁ is achieved by setting N_F = 1 using an 8 MHz clock with a 50% duty cycle clock (t_{sync} < 63 nsec):

$$T_{\text{min}} = \frac{1}{2 \times 8} + 0.063 = 125 \text{ nsec max.}$$

AUXILIARY REGISTER A

Auxiliary Register A is a "hidden" 5-bit register which is used to enable features of the 8291A features

Auxiliary Register, it is loaded with the data A₄A₃A₂A₁A₀. Setting the respective bits to 1 enables the following features.

A₀—RFD Holdoff on all Data: If the 8291A is listening, RFD will not be sent true until the "finish handshake" auxiliary command is issued by the microprocessor. The holdoff will be in effect for each data byte.

A₁—RFD Holdoff on End: This feature enables the holdoff on EOI or EOS (if enabled). However, holdoff will be in effect on any other data bytes.

A₂—End on EOS Received: Whenever the byte in the Data In Register matches the byte in the EOS Register, the END interrupt bit will be set in the Interrupt Status 1 Register.

A₃—Output EOI on EOS Sent: Any occurrence of data in the Data Out Register matching the EOS Register causes the EOI line to be sent true along with the data.

A₄—EOS Binary Compare: Setting this bit causes the EOS Register to function as a full 8-bit word. When it is not set, the EOS Register is a 7-bit word (for ASCII characters).

If A₅ = A₄ = 1, a special "continuous Acceptor Handshake cycling" mode is enabled. This mode should be used only in a controller system configuration where both the 8291A and the 8292 are used. It provides a continuous cycling through the Acceptor Handshake state diagram, requiring no local messages from the microprocessor; the only local message is automatically generated when in ANRS. As such, the 8291A Acceptor Handshake serves as the controller Acceptor Handshake. Thus, the controller cycles through the Acceptor Handshake without data bytes through the data transfer in progress. When the local message is executed, the 8291A should be taken out of the "continuous AH cycling" mode, the GPIB will hang up in ANRS, and a BI interrupt will be generated to indicate that control may be taken. A simpler procedure may be used when a "continuous AH cycling" is executed; the 8291A may stay in "continuous AH cycling". Upon the end of a block (EOI or

PARALLEL POLL PROTOCOL

iliary Register B is a "hidden" 4-bit register which is used to enable some of the features of the A. Whenever a 101 B₃B₂B₁B₀ is written into Auxiliary Mode Register, it is loaded with the B₃B₂B₁B₀. Setting the respective bits to "1" enables the following features:

Enable Undefined Command Pass Through: Feature allows any commands not recognized as 8291A to be handled in software. If enabled, a feature will cause the 8291A to holdoff the handshake when an undefined command is received. The processor must then read the command from Command Pass Through Register and send the D auxiliary command. Until the VSCMD command is sent, the handshake holdoff will be in effect.

and EOI in SPAS: This bit enables EOI to be the status byte; EOI is sent true in Serial (ive State. Otherwise, EOI is sent false in

able High Speed Data Transfer: This feature enabled when tri-state external transceivers . The data transfer rate is limited by T₁ delay erated in the Source Handshake function, defined according to the type of transceiver. When the "High Speed" feature is enabled, icroseconds is generated for the first byte after each true to false transition of ATN. ubsquent bytes, T₁ = 500 nanoseconds. he Internal Counter section for an explanation duration as a function of B₂ and of clock

Active Low Interrupt: Setting this bit a polarity of the INT pin to be reversed, an output signal compatible with Intel's family. Interrupt registers are not affected by

RFD Holdoff on GETor DEC: Setting this RFD to be held false until the "VSCMD" mmand is written after GET, SDC, and ands. This allows the device to hold off til it has completed a clear or trigger n unrecognized command.

Writing a 011 USP₃P₂P₁ into the Auxiliary Mode Register will enable (U=0) or disable (U=1) the 8291A for a parallel poll. When U=0, this command is the "ipe" (local poll enable) local message as defined in IEEE-488. The "S" bit is the sense in which the 8291A is enabled; only if the Parallel Poll Flag ("Ist" local message) matches this bit will the Parallel Poll Response, PPR_W, be sent true (Response = S + Ist). The bits P₃P₂P₁ specify which of the eight data lines PPR_W will be sent over. Thus, once the 8291A has been configured for Parallel Poll, whenever it senses both EOI and ATN true, it will automatically compare its PP flag with the sense bit and send PPR_W true or false according to the comparison.

If a P₂2* implementation is desired, the "Ipe" and "Ist" local messages are all that are needed. Typically, the user will configure the 8291A for Parallel Poll immediately after initialization. During normal operation the microprocessor will set or clear the Parallel Poll Flag (Ist) according to the device's need for service. Consequently the 8291A will be set up to give the proper response to IDY (EOI - ATN) without directly involving the microprocessor.

If a P₁1* implementation is desired, the undefined command features of the 8291A must be used. If P₁1, the 8291A is indirectly configured for Parallel Poll by the active controller on the GPIB. The sequence at the 8291A being enabled or disabled is as follows:

1. The PPC message is received and is loaded into the Command Pass Through Register as an undefined command. A CPT Interrupt is sent to the microprocessor; the handshake is automatically held off.
2. The microprocessor reads the CPT Register and sends VSCMD to the 8291A, releasing the handshake.
3. Having received an undefined primary command, the 8291A is set up to receive an undefined secondary command (the PPE or PPD message). This message is also received into the CPT Register; the handshake is held off, and the CPT Interrupt is generated.

NOTE: As defined in IEEE Standard 488.

4. The microprocessor reads the PPE or PPD message and writes the command into the Auxiliary Mode Register (bit 7 should be cleared first). Finally, the microprocessor sends VSCMD and the handshake is released.

End of Sequence (EOS) Register

EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
-----	-----	-----	-----	-----	-----	-----	-----

EOS REGISTER

The EOS Register and its features offer an alternative to the "Send EOI" auxiliary command. A seven or eight bit byte (ASCII or binary) may be placed in the register to flag the end of a block or read. The type of EOS byte to be used is selected in Auxiliary Register bit A₄.

If the 8291A is a listener, and the "End on EOS Received" is enabled with bit A₄, then an END Interrupt is generated in the Interrupt Status 1 Register, whenever the byte in the Data-In Register matches the byte in the EOS Register.

If the 8291A is a talker, and the "Output EOI on EOS Sent" is enabled with bit A₄, then the EOI line is sent true with the next byte whenever the contents of the Data Out Register match the EOS register.

Reset Procedure

The 8291A is reset to an initialization state either by a pulse applied to its Reset pin, or by a reset auxiliary command (02H) written into the Auxiliary Command Register). The following conditions are caused by a reset pulse (or local reset command):

1. A "pon" local message as defined by IEEE-488 is held true until the initialization state is released.
2. The Interrupt Status Registers are cleared (not Interrupt Enable Registers).
3. Auxiliary Registers A and B are cleared.
4. The Serial Poll Mode Register is cleared.
5. The Parallel Poll Flag is cleared.
6. The EOI bit in the Address Status Register is cleared.
7. N₂ in the Internal Counter is set to 3 MHz. This setting causes the longest possible T₁ delay to be generated in the Source Handshake (16 μsec for 1 MHz clock).
8. The rdy local message is sent.

The initialization state is released by an "Immediate execute pon" command (00H) written into the Auxiliary Command Register).

The suggested initialization sequence is:

1. Apply a reset pulse or send the reset auxiliary command.
2. Set the desired initial conditions by writing into the Interrupt Enable, Serial Poll Mode, Address Mode, Address 0/1, and EOS Registers. Auxiliary Registers A and B, and the internal counter should also be initialized.
3. Send the "immediate execute pon" auxiliary command to release the initialization state.
4. If a P₂2 Parallel Poll implementation is to be used, the "Ipe" local message may be sent, enabling the 8291A for a Parallel Poll Response on an assigned line. (Refer to the section on Parallel Poll Protocol.)

Using DMA

The 8291A may be connected to the Intel[®] 8237 or 8257 DMA Controllers or the 8089 I/O Processor for DMA operation. The 8237 will be used to refer to any DMA controller. The DREQ pin of the 8291A requests a DMA byte transfer from the 8237. It is set by BO, or BI flip flops, enabled by the DMAO and DMAI bits in the Interrupt Enable 2 Register. (After reading, the INT1 register BO and BI interrupts will be cleared but not BO and BI in DREQ equation.)

The \overline{OACK} pin is driven by the 8237 in response to the DMA request. When \overline{OACK} is true (active low) it sets CS = RS0 = RS1 = RS2 = 0 such that the RD and WR signals sent by the 8237 refer to the Data In and Data Out Registers. Also, the DMA request line is reset by \overline{OACK} ($\overline{RD} + \overline{WR}$).

DMA input sequence:

1. A data byte is accepted from the GPIB by the 8291A.
2. A BI interrupt is generated and DREQ is set.
3. \overline{OACK} and \overline{RD} are driven by the 8237, the contents of the Data In Register are transferred to the system bus, and DREQ is reset.
4. The 8291A sends RFD true on the GPIB and proceeds with the Acceptor Handshake protocol.

DMA output sequence:

1. A BO interrupt is generated (indicating that a byte

CLOCK and **WR** are driven by the 8237, a byte is transferred from the MCS bus into the Data Out Register, and **DREQ** is reset.

The 8291A sends **DAV** true on the GPIB and proceeds with the Source Handshake protocol. should be noted that each time the device is addressed (MTA + MLA + lon + lon), the Address Register should be read, and the 8237 should be initialized accordingly. (Refer to the 8237 or 8257 Data Sheets.)

APPLICATION BRIEF

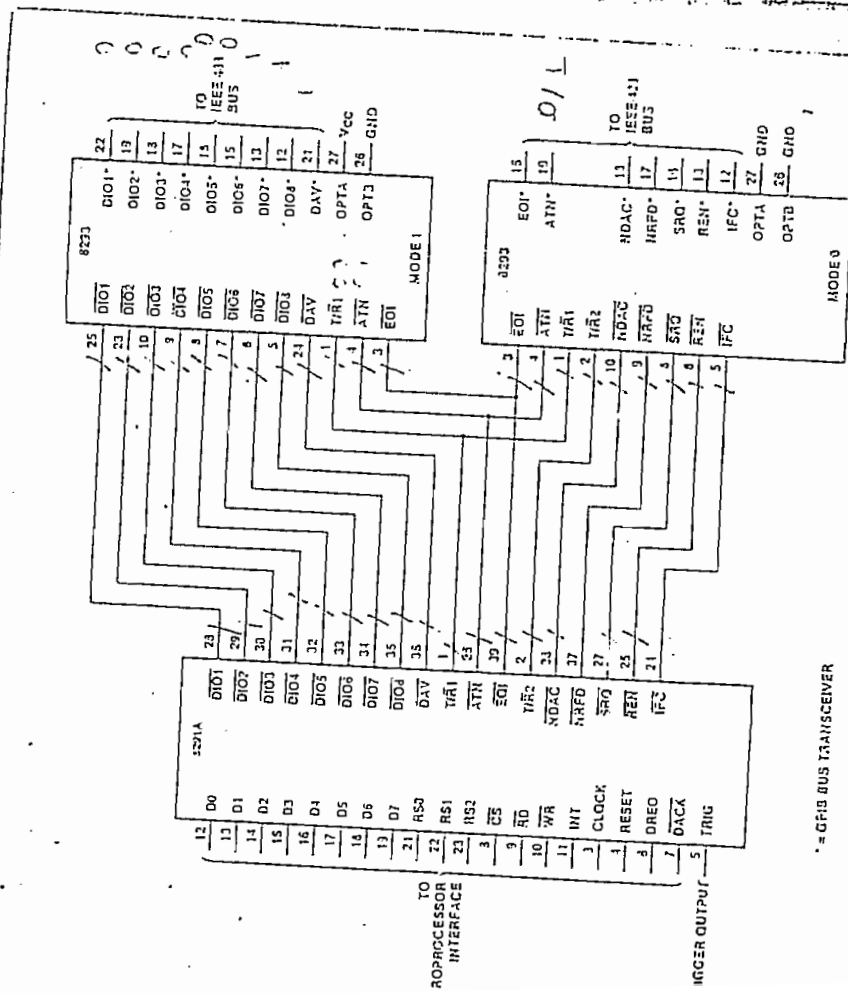
System Configuration

MICROPROCESSOR BUS CONNECTION

8291A is 8048/49, 8051, 8080/85, and 8086/88

EXTERNAL TRANSCEIVERS CONNECTION

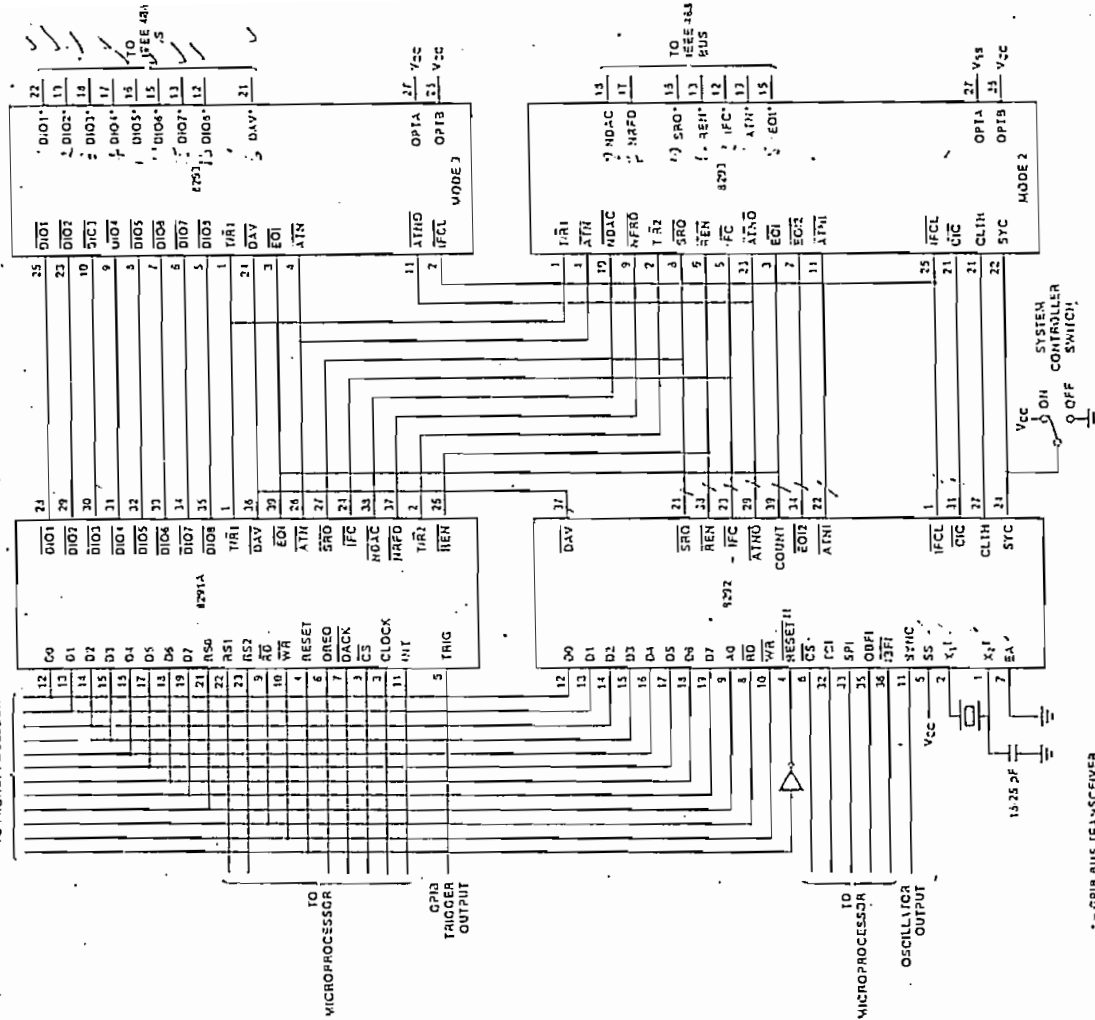
The 8293 GPIB Transceiver interfaces the 8291A directly to the IEEE-488 bus. The 8291A and two 8293's can be configured as a talker/listener (see Figure 6) or with the 8292 as a talker/listener/controller (see Figure 7). Absolutely no active or passive external components are required to comply with the complete IEEE-488 electrical specification.



* = GPIB BUS TRANSCEIVER

compatible. The three address pins (**RS₀**, **RS₁**, **RS₂**) should be connected to the non-multiplexed address bus (for example: **A₀**, **A₁**, **A₂**). In case of 8080, any address lines may be used. If the three lowest address bits are used (**A₀**, **A₁**, **A₂**), then they must be demultiplexed first.

TO MICROPROCESSOR



* = GPIB BUS TRANSCEIVER

† = SEE DATA SHEET FOR ALTERNATE CRYSTAL CONFIGURATIONS

‡ = CAN CONNECT TO SYSTEM RESET SWITCH. SEE 8011A DATA SHEET

Figure 8. 8291A and 8293 System Configuration

true unless the GTSB (Go To Standby) command is sent to the 8292.

ATN has to be false in order to send data rather than commands from the controller. To do this, the following steps are needed:

1. Enable the TCI interrupt if not already enabled.
2. Wait for IBF (Input Buffer Full) in the 8292 Interrupt Status Register to be reset.
3. Write the GTSB (F5H) command to the 8292 Command Field Register.
4. Read the 8282 and wait for TCI to be true.
5. Write the ton (80H) and pon (60H) command to the 8291A Address Mode Register and Auxiliary Mode Registers respectively.
6. Wait for the BO interrupt to be set in the 8291A.
7. Write the data to the 8291A Data-Out Register.

Identically, the user could command the controller to listen rather than talk. To do that, write ton (40H) instead of ton into the Address Mode Register. Then wait for BI rather than BO to go true. Read the data Register.

With the 8292 as the controller-in-charge, it is impossible to address the 8292 via the GPIB. Therefore, the ton or lon modes of the 8291A must be used. To send commands, set the 8291A in the lon mode by writing 80H to the Address Mode Register.

3. Write 26H to the Auxiliary Mode Register to match the T1 data settling time to the 6 MHz clock input.
4. Write an ASCII carriage return (0DH) to the EOS Register.
5. Write 84H to the Auxiliary Mode Register in order to enable "Output EOI on EOS sent" and thus send EOI with the last character.
6. Write 00H—Immediate Execute pon—to the Auxiliary Mode Register to put the 8291A in the idle state.

Communication. Since the 8291A is in the ton mode, a BO interrupt is generated as soon as the Immediate Execute pon command is written. The CPU writes the command into the Data Out Register, and repeats it on BO becoming true for as many commands as necessary. ATN remains continuously

The ADSC bit in the Interrupt Status 2 Register indicates that the 8291A has been addressed unaddressed. The TA and LA bits in the Address Register indicate whether the 8291A is listener (TA=1), listener (LA=1), both (TA=LA=1) or unaddressed (TA=LA=0).

If the 8291A is addressed to listen, the local CPU can read the Data-In Register whenever the BI (Byte In) interrupt occurs in the Interrupt Status 1 Register. The END bit in the same register is also set, either when a data byte matching the pattern in the EOS register has been received.

In the talker mode, the CPU writes data into the Byte-Out Register on BO (Byte Out) true.

TALKER/LISTENER/CONTROLLER SYSTEM

Combined with the Intel 8292, the 8291A executes a complete IEEE-488-1978 controller function. The 8291A talks and listens via the data and handshake lines (NRFD, NDAC and DAV). The 8292 controls four of the five bus management lines (IFC, SRC, ATN and RET). EOI, the fifth line, is shared. The 8291A drives and receives EOI when EOI is used as an end-of-block indicator. The 8292 drives EOI along with ATN during a parallel poll command.

Once again, assume a general system configuration with the following features: (i) Polled system interface; (ii) 8292 as the system controller and controller-in-charge; (iii) ASCII carriage return (CR) as the EOS identifier; (iv) EOI sent with the last character; and, (v) an external buffer (8282) used to monitor the TCI line.

Initialization. In order to send a command across the GPIB, the 8292 has to drive ATN, and the 8291A has to drive the data lines. Both devices therefore need initialization.

To initialize the 8292:

1. Pulse the RESET input. The 8292 will initially drive all outputs high. TCI, SPI, OBF, IBFI and CLTI will then go low. The Interrupt Status, Interrupt Mask, Error Flag, Error Mask and Timeout registers will be cleared. The interrupt counter will be disabled and loaded with 255. The 8292 will monitor the status of the SYNC pin. If high, the 8292 will pulse IFC true for at least 100 ns in compliance with the IEEE-488-1978 standard. It then take control by asserting ATN.

To initialize the 8291A, the following is necessary:

1. Write 00H to Interrupt Enable registers 1 and 2. This disables interrupt and DMA.

The following section describes the steps needed to initialize a typical 8291A system implementing a talker/listener interface and an 8291A/8292 system implementing a talker/listener/controller interface.

TALKER/LISTENER SYSTEM

Assume a general system configuration with the following features: (i) Polled system interface; (ii) Mode 1 addressing; (iii) same address for talker and listener; (iv) ASCII carriage return as the end-of-sequence (EOS) character; (v) EOI sent true with the last byte; and, (vi) 8 MHz clock.

Initialization. Initialization is accomplished with the following steps:

1. Pulse the RESET input or write 02H to the Auxiliary Mode Register.
2. Write 00H to the Interrupt Enable Registers 1 and 2. This disables interrupt and DMA.
3. Write 01H to the Address Mode Register to select Mode 1 addressing.
4. Write 28H to the Auxiliary Mode Register. This loads 8H to the Auxiliary Register. A matching the 8 MHz clock input to the internal T1 delay counter to generate the delay meeting the IEEE spec.
5. Write the talker/listener address to the Address 0-1 register. The three most significant bits are zero.
6. Write an ASCII carriage return (0DH) to the EOS register.
7. Write 88H to the Auxiliary Mode Register to allow EOI to be sent true when the EOS character is sent.
8. Write 00H to the Auxiliary Mode Register. This writes the "Immediate Execute pon" message and takes the 8291A from the initialization state to the idle state. The 8291A will remain idle until a controller initiates some activity by driving ATN true.

Communication. The local CPU now polls the to determine which controller command has been received.

Controller addresses the 8291A by driving ATN. The 8291A (My Listen Address) on the bus and DAV. If the lower five bits of the MLA message address programmed into the Address 0/1 register the 8291A is addressed to listen. It would be addressed to talk if the controller sent the MLTA message instead of MLA.

ABSOLUTE MAXIMUM RATINGS

Ambient Temperature Under Bias 0°C to 70°C
 Storage Temperature -65°C to +150°C
 Voltage on Any Pin -0.5V to +7V
 Power Dissipation 0.65 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 ($V_{CC} = 5V \pm 10\%$, $T_A = 0^\circ C$ to $70^\circ C$ (Commercial))

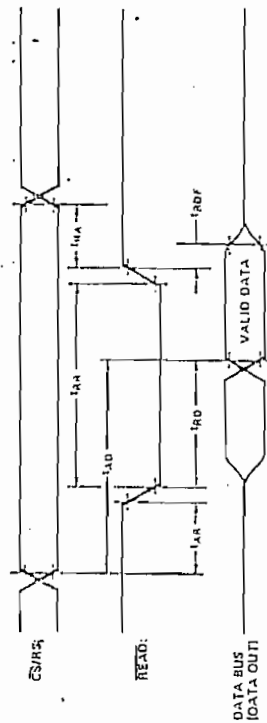
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2	$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage		0.45	V	$I_{OL} = 2mA$ 14mA for TRI pin;
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -400\mu A$ -150 μA for SRO pin;
V_{OH-WR}	Interrupt Output High Voltage	2.4		V	$I_{OH} = -100\mu A$
I_{IL}	Input Leakage	3.5		μA	$I_{IH} = 0V$ to V_{CC}
I_{OFL}	Output Leakage Current		± 10	μA	$V_{OUT} = 0.45V$ V_{CC}
I_{CC}	V_{CC} Supply Current		120	mA	$T_A = 0^\circ C$

A.C. CHARACTERISTICS ($V_{CC} = 5V \pm 10\%$, $T_A = 0^\circ C$ to $70^\circ C$ (Commercial))

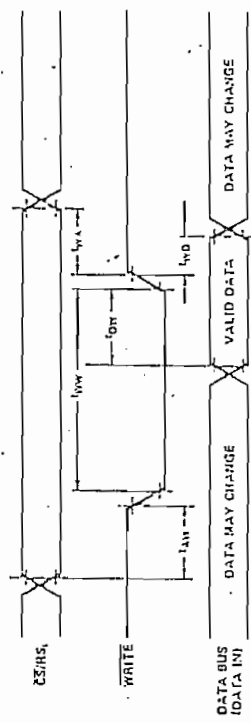
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AN}	Address Stable Before \overline{RD}	0		nsec	
t_{EA}	Address Hold After \overline{RD}	0		nsec	
t_{AR}	\overline{RD} width	140		nsec	
t_{AO}	Address Stable to Data Valid		250	nsec	
t_{RD}	\overline{RD} to Data Valid		100	nsec	
t_{RDF}	Data Float After \overline{RD}	0	60	nsec	
t_{AW}	Address Stable Before \overline{WRITE}	0		nsec	
t_{WA}	Address Hold After \overline{WRITE}	0		nsec	
t_{WR}	\overline{WRITE} width	170		nsec	
t_{OW}	Data Set Up Time to the Trailing Edge of \overline{WRITE}	130		nsec	
t_{DO}	Data Hold Time After \overline{WRITE}	0		nsec	
t_{DOP4}	\overline{RD} or \overline{WR} to \overline{DREQ}		130	nsec	
t_{DCA5}	\overline{RD} to Valid Data (D_3-D_0)		200	nsec	$\overline{DACK1}$ to \overline{RD} 0 $\leq t \leq 50nsec$

WAVEFORMS

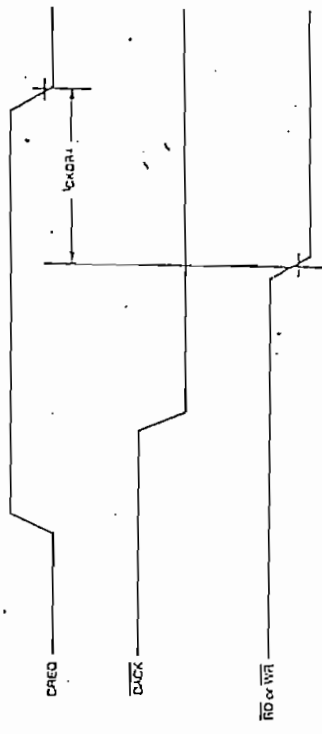
READ



WRITE



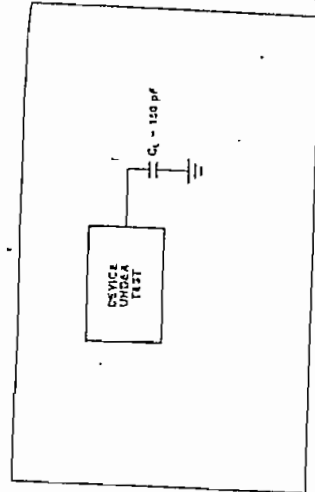
DMA



The diagram shows a timing sequence for a 74181 ALU. The horizontal axis represents time, and the vertical axis represents signal levels. The signals are labeled as INPUT/OUTPUT. The diagram shows two signal traces, one for a logic 1 and one for a logic 0. The timing points are marked as follows:

- For a logic 1: 2.1, 2.0, 0.8
- For a logic 0: 2.2, 0.9

The text below the diagram states: "A.C. TESTING. WAVEFORMS ARE SHOWN AT 2 NS PER A LOGIC '1' AND 0.15V FOR A LOGIC '0'. TIMING MEASUREMENTS ARE MADE AT 75V FOR A LOGIC '1' AND 0.8V FOR A LOGIC '0'."



Symbol	Parameter	Max.	Units	Test Conditions
TEOT13'	EOI ₁ to TR1 ₁	135	nsec	PPSS, ATN=0.45V
TEOD16	EOI ₁ to DIO Valid	155	nsec	PPSS, ATN=0.45V
TEOT12	EOI ₁ to TR1 ₁	155	nsec	PPSS, ATN=0.45V
TATND4	ATN ₁ to NDA ₁	155	nsec	TACS, AIDS
TATT14	ATN ₁ to TR1 ₁	155	nsec	TACS, AIDS
TATT24	ATN ₁ to TR2 ₁	155	nsec	TACS, AIDS
TDVND3-C	DAV ₁ to NDA ₁	650	nsec	AH, CACS
TND0V1	NDA ₁ to DRE ₁	350	nsec	SH, STRS
THRD0R1	NRFO ₁ to DRE ₁	400	nsec	SH
TDVDR3	DAV ₁ to DRE ₁	600	nsec	AH, LACS, ATN=2.4V
TDVND2-C	DAV ₁ to NDA ₁	350	nsec	AH, LACS
TDVNR1-C	DAV ₁ to NRFO ₁	350	nsec	AH, LACS, rdy=True
TDNR0R3	RD ₁ to NRFO ₁	500	nsec	AH, LACS
YWRD15	WR ₁ to DIO Valid	280	nsec	SH, TACS, RS=0.4V
WREO5	WR ₁ to EOI Valid	350	nsec	SH, TACS
WRDV2	WR ₁ to DAV ₁	850 + t _{sync}	nsec	High Speed Transfers Enabled, N _r = t _{o, sync} ≈ 1/2 f _c

55

If GPIB timings are at the pins of the 8291A.

The last number in the symbol for any GPB timing parameter is chosen according to the transition directions of the reference signals. The following table describes the numbering scheme.

1 to 1	1
1 to 1	2
1 to 1	3
1 to 1	4
1 to VALID	5
1 to VALID	6

MODIFIED STATE DIAGRAMS

Figure A-1 presents the interface function state diagrams. It is derived from IEEE Std. state diagrams, with the following changes:

A. The 8291A supports the complete set of IEEE-488 interface functions except for the controller. These include: SH1, AH1, T5, TE5, L3, LE3, SR1, RL1, PP1, DC1, DT1, and CO.

B. Addressing modes included in T.L state diagrams.

Note that in Mode 3, MSA, OSA are generated only after secondary address validity check by the micro-processor (APT interrupt).

C. In these modified state diagrams, the IEEE-488-1978 convention of negative (low true) logic is followed. This should not be confused with the Intel pin- and signal-naming convention based on positive logic. Thus, while the state diagrams below carry very low true logic, the signals described elsewhere in this data sheet are consistent with Intel notation and are based on positive logic.

indicates:

1. When event X occurs, the function returns to state S .
2. X overrides any other transition condition in the function.

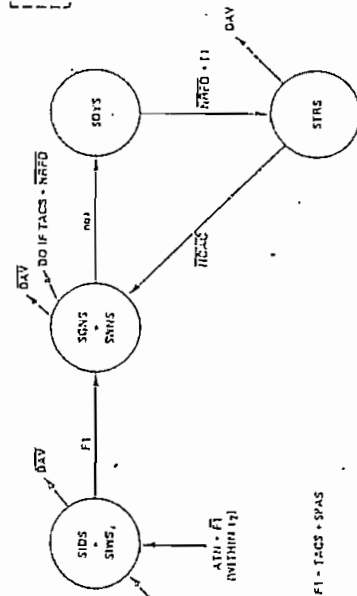
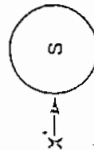
Statement 2 simplifies the diagram, avoiding the explicit use of X to condition all transitions from S to other states.

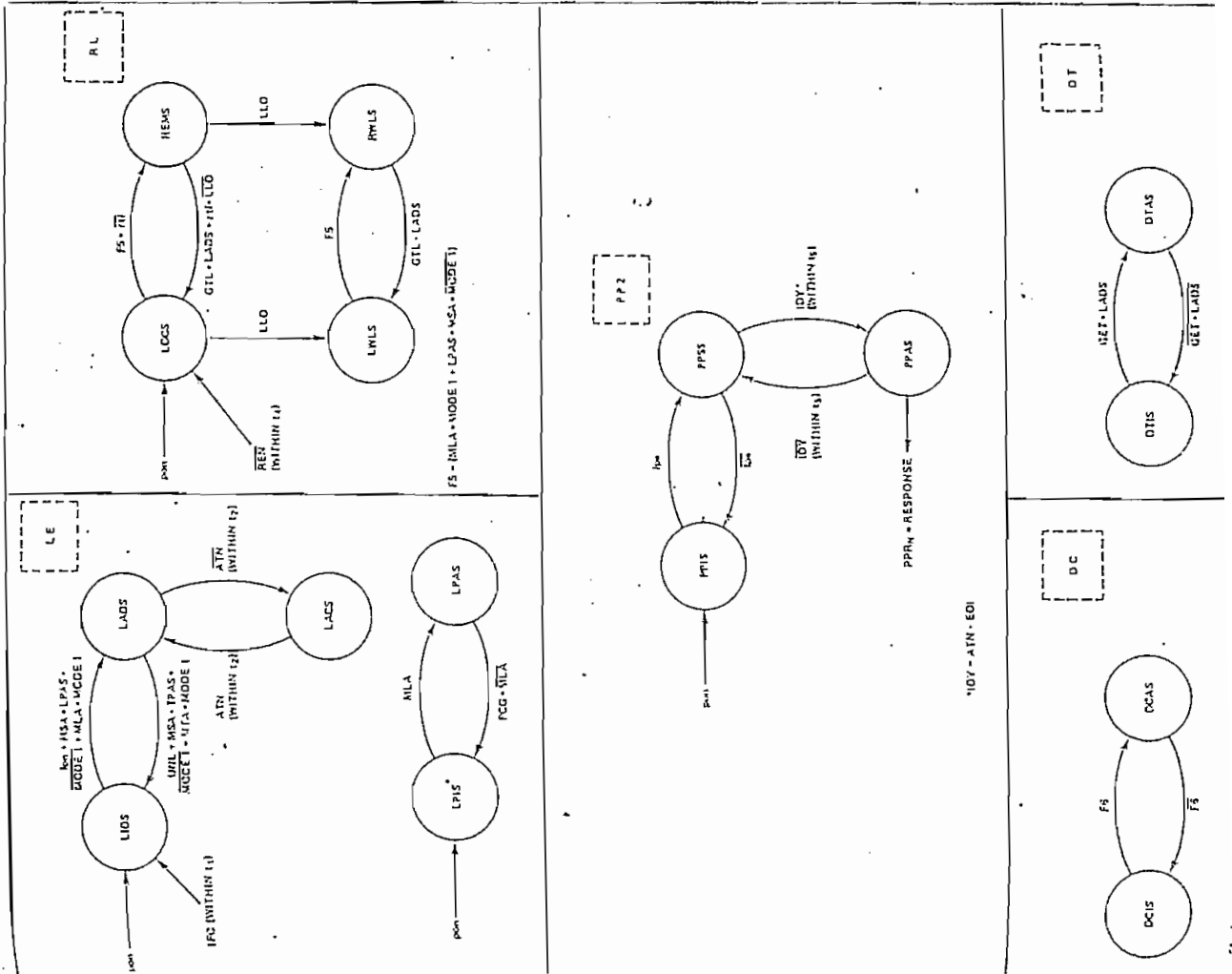
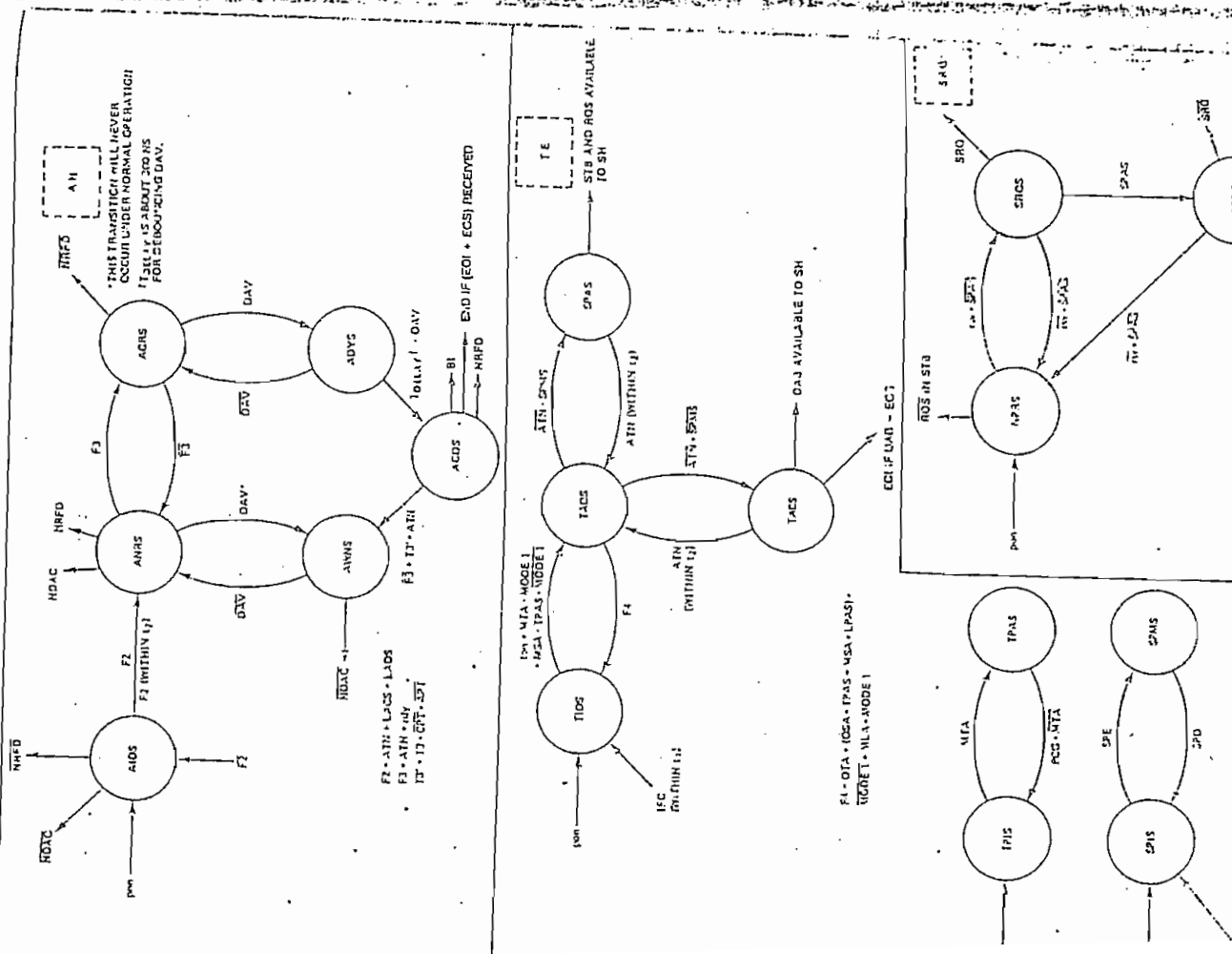
Level	Logic	Conventions	
		IEEE-488	Intel
0	T	DAV	DAV
1	F	DAV	DAV
0	T	NDAC	NDAC
1	F	NDAC	NDAC
0	T	NRFD	NRFD
1	F	NRFD	NRFD

Consider the condition when the Not-Ready-For-Data signal (pin 37) is active. *INRD* indicates this active low signal with the symbol \overline{INRD} ($V_{OUT} \leq V_{OL}$ for AH ; $V_{IN} \leq V_{IL}$ for SH). The IEEE-188-1978 Standard, in its state diagrams, indicates the active state of this signal (True condition) with *NRFD*.

D. All remote multiline messages decoded are conditioned by ACDS. The multiplication by ACDS is not drawn to simplify the diagrams.

E. The symbol





APPENDIX B

Table B-1. IEEE 488 Time Values

Time Value Identifier ¹	Function (Applies to)	Description	Value
T ₁	SH	Settling Time for Multiline Messages	≥ 2μs ²
T ₂	LC, IC, SH, AH, T, L	Response to ATN	≤ 200ns
T ₃	AH	Interface Message Accept Time ³	> 0 ⁴
T ₄	T, TE, L, E, C, OE	Response to IFC or REN False	< 100μs
T ₅	PP	Response to ATN+EOI	≤ 200ns
T ₆	C	Parallel Poll Execution Time	≥ 2μs
T ₇	C	Controller Delay to Allow Current Talker to see ATN Message	≥ 500 ns
T ₈	C	Length of IFC or REN False	> 100μs
T ₉	C	Delay for EOI ⁵	≥ 1.5μs ⁶

NOTES:

¹Time values specified by a lower case *t* indicate the maximum time allowed to make a state transition. Time values specified by an upper case *T* indicate the minimum time that a function must remain in a state before exiting.

²If three-state drivers are used on the \overline{DIO} , \overline{DATA} , and EOI lines, *T₁* may be:

1. ≥ 1100 ns.
2. Or ≥ 700 ns if it is known that within the controller ATN is driven by a three-state driver.
3. Or ≥ 500ns for all subsequent bytes following the first sent after each false transition of ATN (the first byte must be sent in accordance with (1) or (2)).
4. Or ≥ 350ns for all subsequent bytes following the first sent after each false transition of ATN under conditions specified in Section 5.2.3 and warning note. See IEEE Standard 488.

³Time required for interface functions to accept, not necessarily respond to interface messages.

⁴Implementation dependent.

⁵Delay required for EOI, \overline{NDAO} , and \overline{NRFO} signal lines to indicate valid states.

⁶≥ 500 ns for three-state drivers.

APPENDIX C

THE THREE-WIRE HANDSHAKE

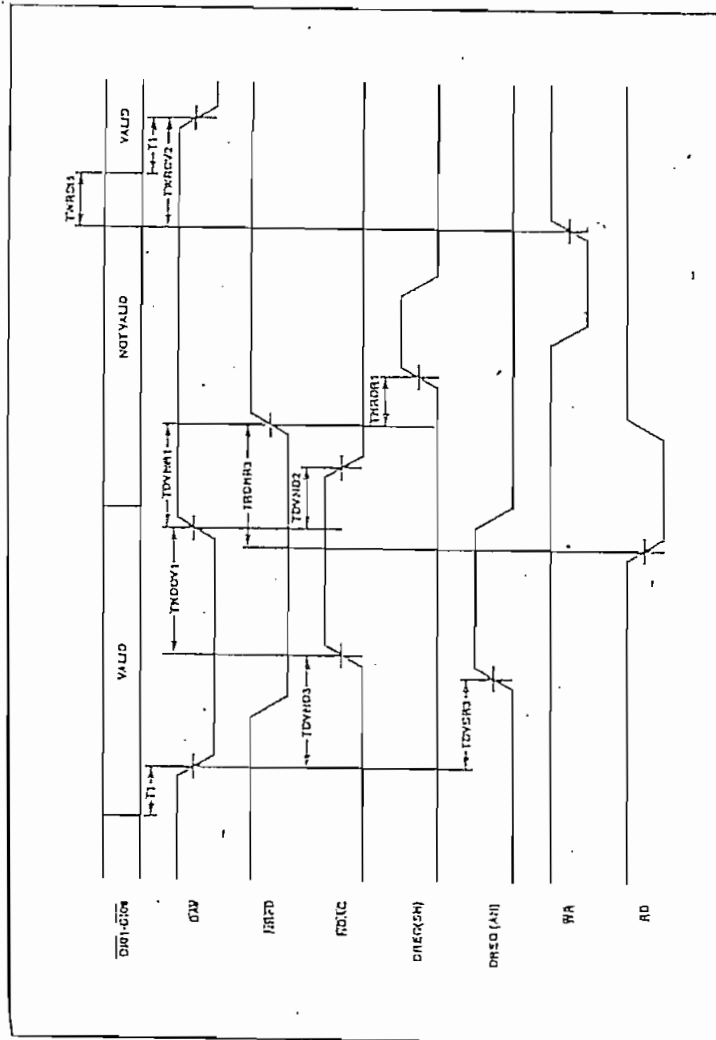


Figure C-1. 3-Wire Handshake Timing at 8291A

8292 GPIB CONTROLLER

- Complete IEEE Standard 488 Controller Function
- Interface Clear (IFC) Sending Capability Allows Seizure of Bus Control and/or Initialization of the Bus
- Responds to Service Requests (SRQ)
- Sends Remote Enable (REN), Allowing Instruments to Switch to Remote Control
- Complete Implementation of Transfer Control Protocol
- Synchronous Control Seizure Prevents the Destruction of Any Data Transmission In Progress
- Connects with the 8291 to Form a Complete IEEE Standard 488 Interface Talker/Listener/Controller

The 8292 GPIB Controller is a microprocessor-controlled chip designed to function with the 8291 GPIB Talker/Listener to implement the full IEEE Standard 488 controller function, including transfer control protocol. The 8292 is a pre-programmed Intel® 8041A.

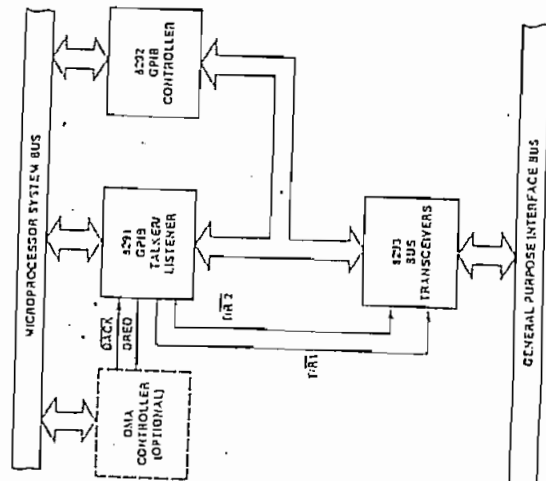


Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
IFC	1	I	IFC Received (Latched): The 8292 monitors the IFC line (when not system controller) through this pin.
X ₁ , X ₂	2, 3	I	Crystal Inputs: Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
RESET	4	I	Reset: Used to initialize the chip to a known state during power on.
CS	5	I	Chip Select Input: Used to select the 8292 from other devices on the common data bus.
RD	8	I	Read Enable: Allows the master CPU to read from the 8292.
A ₀	9	I	Address Line: Used to select between the data bus and the status register during read operations and to distinguish between data and commands written into the 8292 during write operations.
WR	10	I	Write Enable: Allows the master CPU to write to the 8292.
SYNC	11	O	Sync: 8041A instruction cycle synchronization signal; it is an output clock with a frequency of XTAL = 15.
D ₀ -D ₇	12-19	I/O	Data: A bidirectional lines used for communication between the central processor and the 8292's data bus buffers and status register.
V _{SS}	7, 20	P.S.	Ground: Circuit ground potential.
SRQ	21	I	Service Request: One of the IEEE control lines. Sampled by the 8292 when it is controller in charge. If true, SPI interrupt to the master will be generated.
ATN	22	I	Attention In: Used by the 8292 to monitor the GPIB ATN control line. It is used during the transfer control procedure.
IFC	23	I/O	Interface Clear: One of the GPIB management lines, as defined by IEEE Std. 488-1978, places all devices in a known quiescent state.
SYN	24	I	System Controller: Monitors the system controller switch.
CLTH	27	O	Clear Latch: Used to clear the IFC latch after being recognized by the 8292. Usually low (except after hardware Reset). It will be pulsed high when IFC is recognized by the 8292.
ATN	29	O	Attention Out: Controls the ATN control line of the bus through external logic for test and test procedures. (ATN is a GPIB control line, as
Symbol	Pin No.	Type	Name and Function
V _{CC}	5, 26, 40	P.S.	Voltage: +5V supply input ±10%.
COUNT	39	I	Event Count: When enabled by the proper command the internal counter will count external events through this pin. High to low transition will increment the internal counter by one. The pin is sampled once per three internal instruction cycles (7.5 μs at sample period when using 5 MHz XTAL). It can be used for byte counting when connected to DAC, or for block counting when connected to the EOL.
REN	38	O	Remote Enable: The Remote Enable bus signal selects remote or local control of the device on the bus. A GPIB bus management line, as defined by IEEE Std. 488-1978.
DAV	37	I/O	Data Valid: Used during parallel poll to force the 8291 to accept the parallel poll status bits. It is also used during the test procedure.
IBF	36	O	Input Buffer Not Full: Used to interrupt the central processor while the input buffer of the 8292 is empty. This feature is enabled and disabled by the interrupt mask register.
OBFI	35	O	Output Buffer Full: Used as an interrupt to the central processor while the output buffer of the 8292 is full. The feature can be enabled and disabled by the interrupt mask register.
EOI2	34	I/O	End Of Identify: One of the GPIB management lines, as defined by IEEE Std. 488-1978. Used with ATN to identify messages during parallel poll.
SPI	33	O	Special Interrupt: Used as an interrupt on events not initiated by the central processor.
TCI	32	O	Task Complete Interrupt: Interrupt to the central processor used to indicate that the task requested was completed by the 8292 and the information requested is ready in the data bus buffer.
CIC	31	O	Controller In Charge: Controls the SR input of the SRO bus transceiver. It can also be used to indicate that the 8292 is in charge of the GPIB bus.

Event Counter Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

The Event Counter Register contains the initial value for the event counter. The counter can count pulses on pin 39 of the 8292 (COUNT). It can be connected to EOI or NDAO to count blocks or bytes respectively during standby state. A count of zero equals 256. This register cannot be read, and is written using the WEVC command.

Event Counter Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains the current value in the event counter. The event counter counts back from the initial value stored in the Event Counter Register to zero and then generates an Event Counter Interrupt. This register cannot be written and can be read using a REVC command.

Time Out Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

The Time Out Register is used to store the time used for the time out error function. See the individual timeouts (TOUT₁, 2, 3) to determine the units of this counter. This Time Out Register cannot be read, and it is written with the WTOUT command.

Time Out Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains the current value in the time out counter. The time out counter decrements from the original value stored in the Time Out Register. When zero is reached, the appropriate error interrupt is generated. If the register is read while none of the time out functions are active, the register will contain the last value reached the last time a function was active. The Time Out Status Register cannot be written, and it is read with the RTOUT command.

Error Flag Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Four errors are flagged by the 8292 with a bit in the Error Flag Register. Each of these errors can be masked by the Error Mask Register. The Error Flag Register cannot be written, and it is read by the IACK command when the error flag in the Interrupt Status Register is set.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Each of these status bits reflect the current status of the corresponding pin on the 8292.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

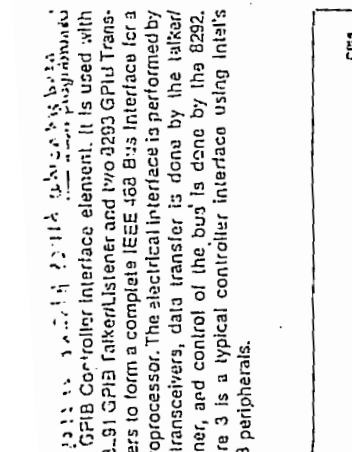
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.

GPB Bus Status Register

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

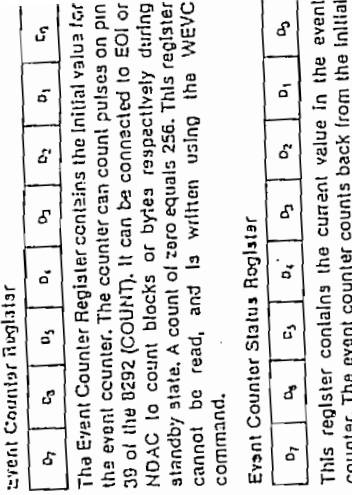
This register contains GPIB bus status information. It can be used by the microprocessor to monitor and manage the bus. The GPIB Bus Register can be read using the RABST command.



Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

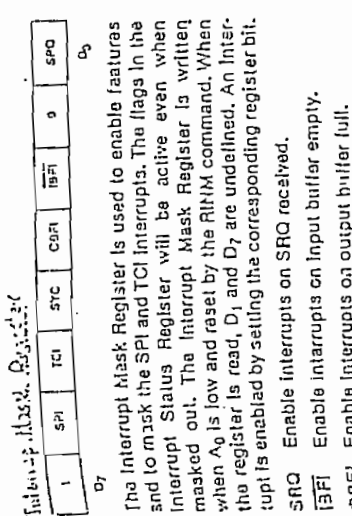
Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.



Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

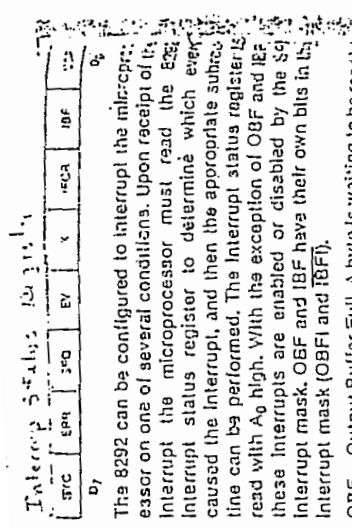
Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.



Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

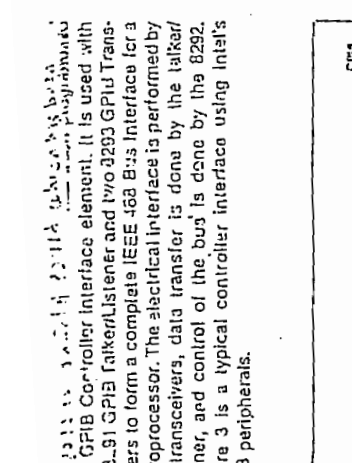
Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.



Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.



Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

Internal RAM in the 8041A is used as a special case register bank for the 8292. Most of these registers (except for the interrupt flag) can be accessed through commands to the 8292. Table 2 identifies the registers used by the 8292 and how they are accessed.

OUT2 — Time Out Error 2 occurs when the transmission between the addressed talker and listener has not started for the time period specified by the Time Out Register. Each count in the Time Out Register is at least 45 μ sec. This feature is only enabled when the controller is in the CSBS state.

OUT3 — Time Out Error 3 occurs when the handshake signals are stuck and the 8292 is not succeeding in taking control synchronously for the time period specified by the Time Out Register. Each count in the Time Out Register is at least 1800 μ sec. The 8292 will continue checking \overline{ATN} until it becomes true or a new command is received. After performing the new command, the 8292 will return to the \overline{ATN} checking loop.

User error occurs when request to assert IFC or REN was received and the 8292 was not the system controller.

Error Mask Register

0	USER	0	0	OUT3	OUT2	OUT1

Error Mask Register is used to mask the interrupt for a particular type of error. Each type of error interrupt is enabled by setting the corresponding bit in the Error Mask Register. This register can be read with the \overline{ATN} command and written with $\overline{A_0}$ low.

Command Register

1	1	OP	C	C	C

Commands are performed by the 8292 whenever a byte is written with $\overline{A_0}$ high. There are two categories of commands distinguished by the OP bit (bit 4). The first category is the operation command (OP = 1). These commands initiate some action on the interface bus. The second category is the utility commands (OP = 0). These commands are used to aid the communication between the processor and the 8292.

UTILITY COMMANDS

Utility commands initiate some action on the GPIB interface bus. It is using these commands that the utility functions such as polling, taking and passing data, and system controller functions are performed.

FE — STCNI — Stop Counter Interrupts

This command disables the internal counter interrupt so that a 8292 will stop interrupting the master on event underflows. However, the counter will continue counting and its contents can still be used.

FD — Go To IDLE

procedure while transferring control to another controller. The 8292 will respond to this command only if it is in the active state. \overline{ATN} will go high, and \overline{CIC} will be high so that this 8292 will no longer be driving the \overline{ATN} line on the GPIB interface bus. TCI will be set upon completion.

F2 — RST — Reset

This command has the same effect as asserting the external reset on the 8292. For details, refer to the reset procedure described later.

F3 — RSTI — Reset Interrupts

This command resets any pending interrupts and clears the error flags. The 8292 will not return to any loop it was in (such as from the time out interrupts).

F4 — QSEC — Go To Standby, Enable Counting

The function causes \overline{ATN} to go high and the counter will be enabled. If the 8292 was not the active controller, this command will exit immediately. If the 8292 is the active controller, the counter will be loaded with the value stored in the Event Counter Register, and the internal interrupt will be enabled so that when the counter reaches zero, the SPI interrupt will be generated. SPI will be generated every 256 counts thereafter until the controller exits the standby state or the SPCNI command is written. An initial count of 256 (zero in the Event Counter Register) will be used if the \overline{WVEC} command is not executed. If the data transmission does not start, a TOUT2 error will be generated.

F5 — EXPP — Execute Parallel Poll

This command initiates a parallel poll by asserting \overline{EOI} when \overline{ATN} is already active. TCI will be set at the end of the command. The 8291 should be previously configured as a listener. Upon detection of \overline{DAV} true, the 8291 enters ACS and latches the parallel poll response (PPR) byte into its data in register. The master will be interrupted by the 8291 BI interrupt when the PPR byte is available. No interrupts except the \overline{BFI} will be generated by the 8292. The 8292 will respond to this command only when it is the active controller.

F6 — GTSB — Go To Standby

If the 8292 is the active controller, \overline{ATN} will go high then TCI will be generated. If the data transmission does not start, a TOUT2 error will be generated.

F7 — SLOC — Set Local Mode

If the 8292 is the system controller, then REN will be asserted false and TCI will be set true. If it is not the system controller, the User Error bit will be set in the Error Flag Register.

F8 — STENI — Set Interface To Remote Control

This command will set REN true and TCI true if this 8292 is the system controller. If not, the User Error bit will be set in the Error Flag Register.

F9 — ABORT — Abort All Operation, Clear Interface
This command will cause IFC to be asserted true for at least 100 μ sec if this 8292 is the system controller. If it is in CIDS, it will take control over the bus (see the TCIEN command).

FA — TCNTR — Take Control

The transfer of control procedure is coordinated by the master with the 8291 and 8292. When the master receives a TCI message from the 8291, it should issue the TCNTR command to the 8292. The following events occur to take control:

1. The 8292 checks to see if it is in CIDS, and if not, it exits.
2. Then \overline{ATN} is checked until it becomes high. If the current controller does not release \overline{ATN} for the time specified by the Time Out Register, then a TOUT1 error is generated. The 8292 will return to this loop after an error or any command except the RST and RSTI commands.
3. After the current controller releases \overline{ATN} , the 8292 will assert \overline{ATN} and \overline{CIC} low.
4. Finally, the TCI interrupt is generated to inform the master that it is in control of the bus.

FC — TCASY — Take Control Asynchronously

TCAS transfers the 8292 from CSBS to CACS independent of the handshake lines. If a bus hangup is detected (by an error flag), this command will force the 8292 to take control (asserting \overline{ATN}) even if the AH function is not in AHRS (Accepter Not Ready State). This command should be used very carefully since it may cause the loss of a data byte. Normally, control should be taken synchronously. After checking the controller function for being in the CSBS (also it will exit immediately), \overline{ATN} will go low, and a TCI interrupt will be generated.

FD — TCSY — Take Control Synchronously

There are two different procedures used to transfer the 8292 from CSBS to CACS depending on the state of the 8291 in the system. If the 8291 is in "continuous AH cycling" mode (Aux. Reg. A0 = A1 = 1), then the following procedure should be followed:

1. The master microprocessor stops the continuous AH cycling mode in the 8291;
2. The master reads the 8291 Interrupt Status Register;
3. If the END bit is set, the master sends the TCSY command to the 8292;
4. If the END bit was not set, the master reads the 8291 Data in Register and then waits for another BI interrupt from the 8291. When it occurs, the master sends the 8292 the TCSY command.

If the 8291 is not in AH cycling mode, then the master just waits for a BI interrupt and then sends the TCSY command. After the TCSY command has been issued, the 8292 checks for CSBS. If CSBS, then it exits the routine. Otherwise, it then checks the \overline{DAV} bit in the GPIB status. When \overline{DAV} becomes false, the 8292 will

wait for at least 1.5 μ sec. (T10) and then \overline{ATN} will go low. If \overline{DAV} does not go low, a TOUT3 error will be generated. If the 8292 successfully takes control, it sets TCI true.

FE — STCNI — Stop Counter Interrupts

This command enables the internal counter interrupt. The counter is enabled by the GSEC command.

UTILITY COMMANDS

All these commands are either Read or Write to registers in the 8292. Note that writing to the Error Mask Register and the Interrupt Mask Register are done directly.

E1 — WTOUT — Write To Time Out Register

The byte written to the data bus buffer (with $\overline{A_0} = 0$) following this command will determine the time used for the line out function. Since this function is implemented in software, this will not be an accurate time measurement. This feature is enable or disable by the Error Mask Register. No interrupts except for the \overline{BFI} will be generated upon completion.

E2 — WVEC — Write To Event Counter

The byte written to the data bus buffer (with $\overline{A_0} = 0$) following this command will be loaded into the Event Counter Register and the Event Counter Status for byte counting or EOI counting. Only \overline{BFI} will indicate completion of this command.

E3 — REVC — Read Event Counter Status

This command transfers the contents of the Event Counter into the data bus buffer. A TCI is generated when the data is available in the data bus buffer.

E4 — RERF — Read Error Flag Register

This command transfers the contents of the Error Flag Register into the data bus buffer. A TCI is generated when the data is available.

E5 — RINM — Read Interrupt Mask Register

This command transfers the contents of the Interrupt Mask Register into the data bus buffer. This register is available to the processor so that it does not need to store this information elsewhere. A TCI is generated when the data is available in the data bus buffer.

E8 — RCST — Read Controller Status Register

This command transfers the contents of the Controller Status Register into the data bus buffer and a TCI interrupt is generated.

E7 — RBST — Read GPIB Bus Status Register

This command transfers the contents of the GPIB Bus Status Register into the data bus buffer, and a TCI interrupt is generated when the data is available.

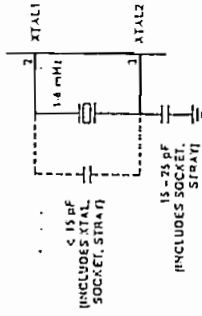
DETAILED TIMINGS (1/2)

Code	Nums	Execution Time	IOEN	TC _{EN}	SP1	INTO	CTC	IFC	REN	EOI	DAV	Comments
21	WTOU	63	24									
22	WEVC	63	24									
23	REVC	71	24	51								
24	REVF	87	24	47								
25	RINH	69	24	48								
26	RCST	97	24	77								
27	RBST	92	24	72								
28												
29	RTOUT	69	24	49								
30	RERM	60	24	49								
31	SPCH	53	24									
32	GIOL	68	24	70		181	181					Count Starts After 20
33	RST	84	21	52								
34	RST	214	24	192	152	1179	1174	1101				Not System Controller System Controller
35	RSTI	81	21									
36	GSEC	125	24	107		183						
37	EXPP	75	24						153	155		
38	G183	118	24	100					159	157		
39	SLOC	73	24	55		181		140				
40	SREI	91	24	73				184				
41	ABORT	155	24	133		1120	1115	142				
42	ICHT	108	24	58		171	168					
43	TCAS	92	24	87		165						
44	TCST	115	24	91		180						
45	STCH	99	24									
46	RESET	29	—	17								Starts Count After 43 Not System Controller
47	JACK	118	—									If Interrupt Pending

Times are multiples of t_{CY} from the 8041A command interrupt. clears after 7 t_{CY} on all commands. indicates a level transition from low to high, 1 indicates a high to low transition.

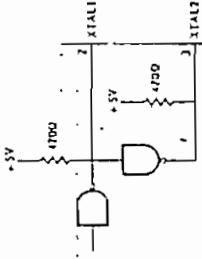
CLOCK DRIVER CIRCUITS

CRYSTAL OSCILLATOR MODE



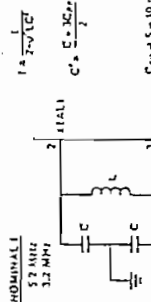
CRYSTAL SERIES RESISTANCE SHOULD BE $< 15 \text{ pF}$ INCLUDES SOCKET STRAY INDUCTANCE

DRIVING FROM EXTERNAL SOURCE



BOTH XTAL1 AND XTAL2 SHOULD BE DRIVEN. RESISTORS TO VCC ARE NEEDED TO ENSURE $V_{DD} = 3.0 \text{ V}$ IF FTL CIRCUIT IS USED.

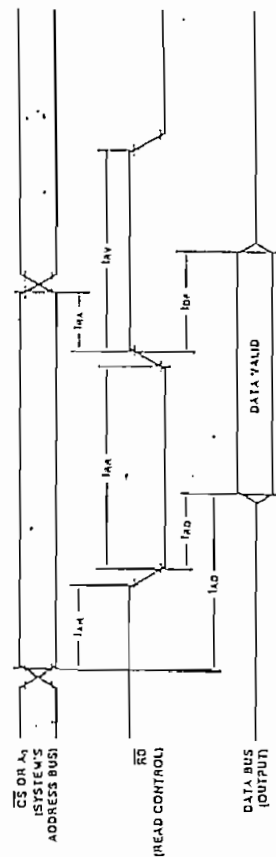
LC OSCILLATOR MODE



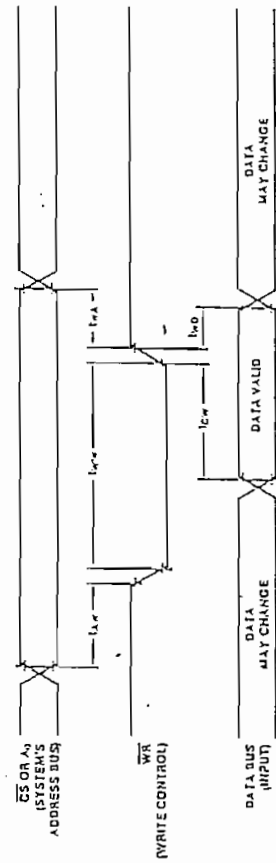
EACH C SHOULD BE APPROXIMATELY 20 pF INCLUDING STRAY CAPACITANCE

WAVEFORMS

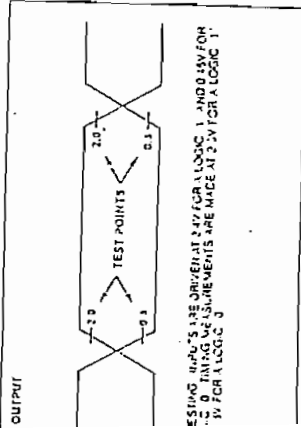
READ OPERATION—DATA BUS BUFFER REGISTER



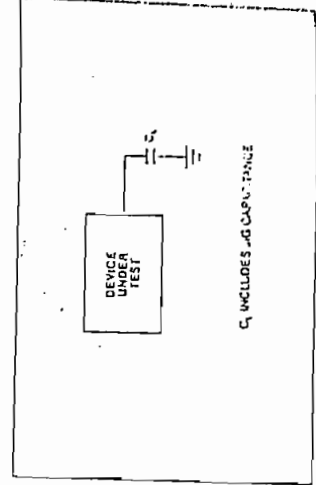
WRITE OPERATION—DATA BUS BUFFER REGISTER



TESTING INPUT, OUTPUT WAVEFORM



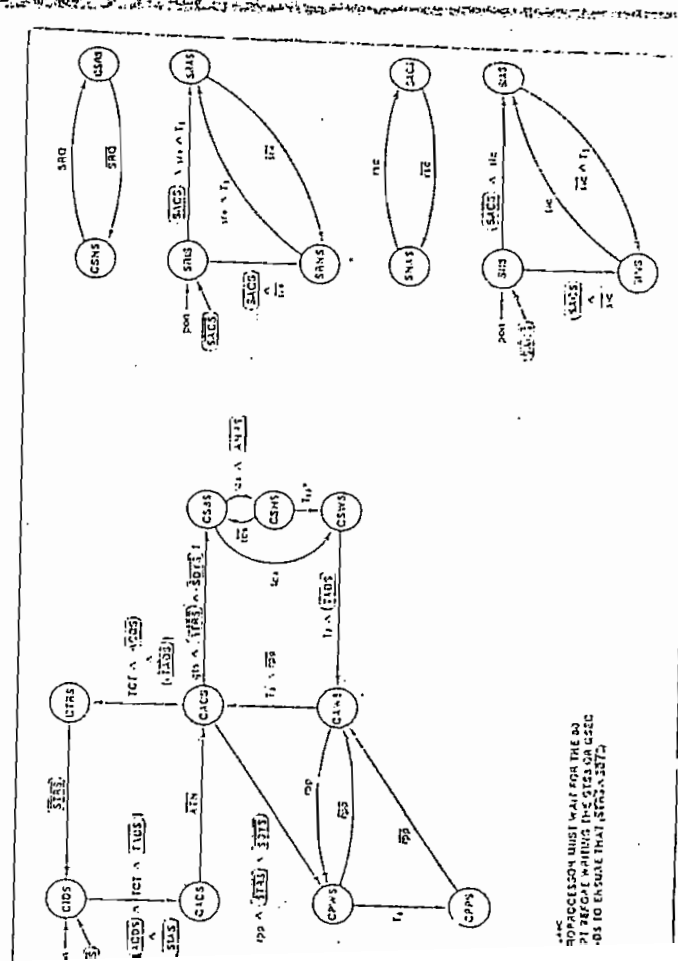
A.C. TESTING LOAD CIRCUIT



Following tables and state diagrams were taken from the IEEE Standard Digital Interface for Program-
 ment is the official standard for the GPIB bus and can be purchased from IEEE, 345 East 47th St., New York, NY 10017.

C MESSAGES

Messages	Interface States
<p>pon = power on</p> <p>rsc = request system control</p> <p>rpp = request parallel poll</p> <p>gis = go to standby</p> <p>lca = take control asynchronously</p> <p>lcs = take control synchronously</p> <p>sic = send interface clear</p> <p>sre = send remote enable</p> <p>lfg = interface clear</p> <p>ATN = attention</p> <p>TCT = take control</p>	<p>CIDS = controller idle state</p> <p>CADS = controller addressed state</p> <p>CTRS = controller transfer state</p> <p>CACS = controller active state</p> <p>CPVS = controller parallel poll wait state</p> <p>CPPS = controller parallel poll state</p> <p>CSBS = controller standby state</p> <p>CSHS = controller standby hold state</p> <p>CAVS = controller active wait state</p> <p>CSWS = controller synchronous wait state</p> <p>CSRS = controller service requested state</p> <p>CSNS = controller service not requested state</p> <p>SNAS = system control not active state</p> <p>SACS = system control active state</p> <p>SRIS = system control remote enable idle state</p> <p>SRNS = system control remote enable not active state</p> <p>SRAS = system control remote enable active state</p> <p>SIIS = system control interface clear idle state</p> <p>SINS = system control interface clear not active state</p> <p>SIAS = system control interface clear active state</p> <p>CAIDS = acceptor data state (AH function)</p> <p>CAIRS = acceptor not ready state (AH function)</p> <p>SOYS = source delay state (SH function)</p> <p>SIRS = source transfer state (SH function)</p> <p>TADS = talker addressed state (T function)</p>



RECESSION MUST WAIT FOR THE 20
 21 BEFORE WRITING THE STATE OR C250
 -05 TO ENSURE THAT STATE IS 3072

REMOTE MESSAGE CODING

		Bus Signal Line(s) and Coding That Assure the True Value of this Message											
Mnemonic	Message Name											T Y P E	C L S
		D	I	0	1	2	3	4	5	6	7		
ACG	Addressed Command Group	Y	0	0	X	X	X	X	X	X	X	M	AC
ATN	Attention	X	X	X	X	X	X	X	X	X	X	U	UC
DAB	Data Byte	D	D	D	D	D	D	D	D	D	D	M	DD
DAC	Data Accepted	X	X	X	X	X	X	X	X	X	X	U	HS
DAV	Data Valid	X	X	X	X	X	X	X	X	X	X	U	HS
DCL	Device Clear	Y	0	0	1	0	1	0	1	0	1	M	UC
END	End	X	X	X	X	X	X	X	X	X	X	U	ST
EOS	End of String	E	E	E	E	E	E	E	E	E	E	M	DD
GET	Group Execute Trigger	Y	0	0	1	0	0	1	0	0	1	M	AC
GTL	Go to Local	Y	0	0	0	0	1	1	1	1	1	M	AC
IDY	Identify	X	X	X	X	X	X	X	X	X	X	U	UC
IFC	Interface Clear	Y	0	1	X	X	X	X	X	X	X	M	UC
LAG	Listen Address Group	Y	0	1	X	X	X	X	X	X	X	M	UC
LLO	Local Lock Out	Y	0	1	0	0	1	1	1	1	1	M	UC
MLA	My Listen Address	Y	0	1	L	L	L	L	L	L	L	M	AD
MTA	My Talk Address	Y	1	0	T	T	T	T	T	T	T	M	AD
MSA	My Secondary Address	Y	1	S	S	S	S	S	S	S	S	M	SE
NUL	Null Byte	0	0	0	0	0	0	0	0	0	0	M	DD
OSA	Other Secondary Address	Y	1	S	S	S	S	S	S	S	S	M	SE
OTA	Other Talk Address	Y	1	S	S	S	S	S	S	S	S	M	SE
PCG	Primary Command Group	Y	0	0	0	1	0	1	0	1	0	M	AC
PPC	Parallel Poll Configure	Y	1	0	S	P	P	P	P	P	P	M	SE
PPE	Parallel Poll Enable	Y	1	1	D	D	D	D	D	D	D	M	SE
PPD	Parallel Poll Disable	X	X	X	X	X	X	X	X	X	X	U	ST
PPR1	Parallel Poll Response 1	X	X	X	X	X	X	X	X	X	X	U	ST
PPR2	Parallel Poll Response 2	X	X	X	X	X	X	X	X	X	X	U	ST
PPR3	Parallel Poll Response 3	X	X	X	X	X	X	X	X	X	X	U	ST
PPR4	Parallel Poll Response 4	X	X	X	X	X	X	X	X	X	X	U	ST
PPR5	Parallel Poll Response 5	X	X	X	X	X	X	X	X	X	X	U	ST
PPR6	Parallel Poll Response 6	X	X	X	X	X	X	X	X	X	X	U	ST
PPR7	Parallel Poll Response 7	X	X	X	X	X	X	X	X	X	X	U	ST
PPR8	Parallel Poll Response 8	X	X	X	X	X	X	X	X	X	X	U	ST
PPU	Parallel Poll Unconfigure	Y	0	1	0	1	0	1	0	1	0	M	UC
REN	Remote Enable	X	X	X	X	X	X	X	X	X	X	U	HS
RFD	Ready for Data	X	X	X	X	X	X	X	X	X	X	U	HS
RQS	Request Service	X	X	X	X	X	X	X	X	X	X	M	SE
SCG	Secondary Command Group	Y	1	X	X	X	X	X	X	X	X	M	AC
SDC	Selected Device Clear	Y	0	0	0	1	0	0	1	0	0	M	AC
SPD	Serial Poll Disable	Y	0	0	1	0	0	1	0	0	1	M	UC
SPE	Serial Poll Enable	Y	0	0	1	0	0	1	0	0	1	M	UC
SRQ	Service Request	X	X	X	X	X	X	X	X	X	X	U	ST
STB	Status Byte	S	X	S	S	S	S	S	S	S	S	M	ST
TCT	Take Control	Y	0	0	1	0	0	1	0	0	1	M	AC
TAG	Talk Address Group	Y	1	0	X	X	X	X	X	X	X	M	AD
UCG	Universal Command Group	Y	0	1	X	X	X	X	X	X	X	M	UC
UNL	Unlisten	Y	0	1	1	1	1	1	1	1	1	M	AD
UNT	Unlink	Y	1	0	1	1	1	1	1	1	1	M	AD

The 100 series of ATMs includes...

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.....0.5V to +7V
Power Dissipation.....1.5 Watt

*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_{SS} = 0V; 8292, V_{CC} = ±5V ± 10%)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V _{IL1}	Input Low Voltage (All Except X ₁ , X ₂ , RESET)	-0.5	0.8	V	
V _{IL2}	Input Low Voltage (X ₁ , X ₂ , RESET)	-0.5	0.6	V	
V _{IH1}	Input High Voltage (All Except X ₁ , X ₂ , RESET)	2.2	V _{CC}	V	
V _{IH2}	Input High Voltage (X ₁ , X ₂ , RESET)	3.8	V _{CC}	V	
V _{OL1}	Output Low Voltage (D _Q -D ₇)		0.45	V	I _{OL} = 2.0 mA
V _{OL2}	Output Low Voltage (All Other Outputs)		0.45	V	I _{OL} = 1.6 mA
V _{OHI}	Output High Voltage (D _Q -D ₇)	2.4		V	I _{OHI} = -400 μA
V _{OHI2}	Output High Voltage (All Other Outputs)	2.4		V	I _{OHI} = -50 μA
I _{IL}	Input Leakage Current (COUNT, IFCL, RD, WR, CS, A ₀)		±10	μA	V _{SS} < V _{IH} < V _{CC}
I _{OZ}	Output Leakage Current (D _Q -D ₇ , High Z State)		±10	μA	V _{SS} + 0.45 < V _{IH} < V _{CC}
I _{I11}	Low Input Load Current (Pins 21-24, 27-38)		0.5	mA	V _{IH} = 0.8V
I _{I2}	Low Input Load Current (RESET)		0.2	mA	V _{IH} = 0.8V
I _{CC}	Total Supply Current		125	mA	Typical = 65 mA
I _{IH}	Input High Leakage Current (Pins 21-24, 27-38)		100	μA	V _{IH} = V _{CC}
C _{IN}	Input Capacitance		10	pF	
C _{IO}	I/O Capacitance		20	pF	

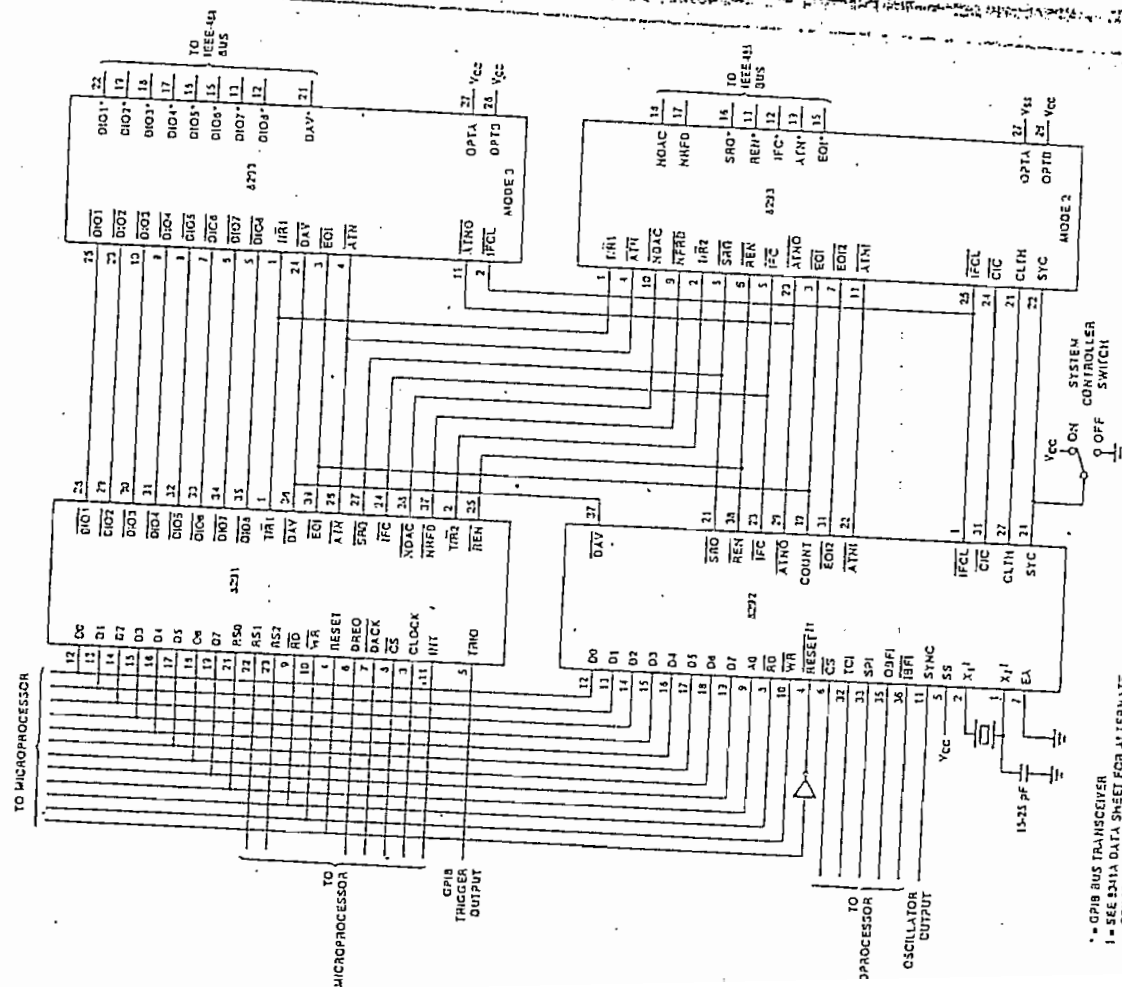
A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{SS} = 0V; 8292, V_{CC} = ±5V ± 10%)

8292 READ

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t _{AR}	CS, A ₀ Setup to RD _i	0		ns	
t _{RA}	CS, A ₀ Hold After RD _i	0		ns	
t _{PR}	RD Pulse Width	250		ns	
t _{AD}	CS, A ₀ to Data Out Delay		225	ns	C _L = 150 pF
t _{RD}	RD _i to Data Out Delay		225	ns	C _L = 150 pF
t _{OF}	RD _i to Data Float Delay		100	ns	
t _{CV}	Cycle Time	2.5	15	μs	

8292 WRITE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t _{AW}	CS, A ₀ Setup to WR _i	0		ns	
t _{WA}	CS, A ₀ Hold After WR _i	0		ns	
t _{WV}	WR Pulse Width	250		ns	
t _{WV}	Data Setup to WR _i	150		ns	



* GPIB BUS TRANSCEIVER
1 - SEE 8211A DATA SHEET FOR ALTERNATE
CRYSTAL CONFIGURATIONS
11 - CAN CONNECT TO SYSTEM RESET SWITCH.
SEE 8211A DATA SHEET

1. D1-C8 specify the device dependent data bits.
2. E1-E3 specify the device dependent code used to indicate the EOS message.
3. L1-L3 specify the device dependent bits of the device's listen address.
4. T1-T5 specify the device dependent bits of the device's talk address.
5. S1-S5 specify the device dependent bits of the device's talk address.
6. S specifies the sense of the PPR.

Response = S#B131

1-P3 specify the PPR message to be sent when a parallel pci is executed.

P3	P2	P1	PPR Message
0	0	0	PPR1
.	.	.	.
.	.	.	.
.	.	.	.
1	1	1	PPR8

D1-D4 specify don't-care bits that shall not be decoded by the receiving device. It is recommended that all zeros be sent.

S1-S6, S8 specify the device dependent status. (DIO7 is used for the RQS message.)

The source of the message on the ATN line is always the C function, whereas the messages on the DIO and ECI lines are enabled by the T function.

The source of the messages on the ATN and ECI lines is always the C function, whereas the source of the messages on the DIO lines is always the PP function.

This code is provided for system use, see 5.2.

8293 GPIO TRANSCEIVER

- Nine Open-collector or Three-state Line Drivers
- 48 mA Sink Current Capability on Each Line Driver
- Nine Schmitt-type Line Receivers
- High Capacitance Load Drive Capability
- Single 5V Power Supply
- 28-Pin Package
- Low Power CMOS Design
- On-chip Decoder for Mode Configuration
- Power Up/Power Down Protection to Prevent Disrupting the IEEE Bus
- Connects with the 8291A and 8292 to Form an IEEE Standard 488 Interface Talker/Listener/Controller with no Additional Components
- Only Two 8293's Required per GPIB Interface
- On-Chip IEEE-488 Bus Terminations

The Intel® 8293 GPIB Transceiver is a high-current, non-inverting buffer chip designed to interface the 8291A GPIB Talker/Listener, or the 8291A/8292 GPIB Talker/Listener/Controller combination, to the IEEE Standard 488-1978 Instrumentation Interface Bus. Each GPIB Interface would contain two 8293 Bus Transceivers. In addition, the 8293 can also be used as a general-purpose bus driver.

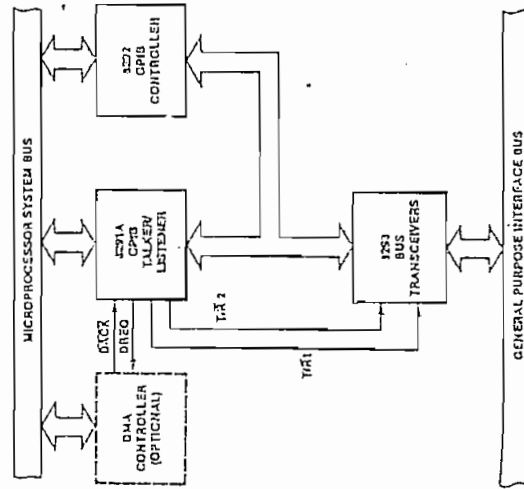


Figure 1. 8291A, 8292, 8293 Block Diagram

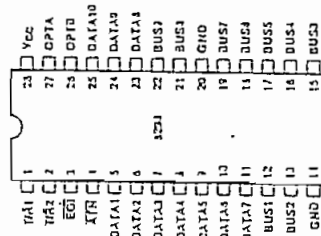


Figure 2. Pin Configuration

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
BUS1- BUS9	12, 13, 15-19, 21, 22	IO	GPB Lines, GPB Side: These are the IEEE-488 bus interface drivers/receivers, or TTL-compatible inputs on the 8291A/8292 side, depending on the mode used. Their use is programmed by the two mode select pins, OPTA and OPTB.
DATA1- DATA10	5-11, 23-25	IO	GPB Lines, 8291A/92 Side: These are the pins to be connected to the 8291A and 8292 to interface with the GPB. Their use is programmed by the two mode select pins, OPTA and OPTB. All these pins are TTL compatible.
TI/R1	1	I	Transmit Receive 1: This pin controls the direction for NOAC, NRFO, DAV, and DIO1-DIO8. Input is TTL compatible.
TI/R2	2	I	Transmit Receive 2: This pin controls the direction for EOI. Input is TTL compatible.
EOI	3	IO	End Of Identify: This pin is the end of a multiple byte transaction in conjunction with ATN, address, and data. It is switched between transmit and receive by TI/R2. This pin is TTL compatible.
ATN	4	O	Attention: This pin is used by 8291A to monitor the GPB Address line. It specifies how data on DIO lines is to be interpreted. The output is TTL compatible.
OPTA OPTB	27 25	I	Mode Select: These two pins control the function of the 8291A and 8292. Various modes are in table 2.
Vcc	28	P.S.	Voltage: Positive power supply (V _{cc} = 10%).
GND	14, 20	P.S.	Ground: Circuit ground.

Table 2. 8293 Mode Selection Pin Mapping

Pin Name	Pin No.	IEEE Implementation Name			
		Mode 0	Mode 1	Mode 2	Mode 3
OPTA	27	0	1	0	1
OPTB	25	0	0	1	1
DATA1	5	IFC	DIO8	IFC	DIO8
BUS1	12	IFC	DIO8	IFC	DIO8
DATA2	6	REN	DIO7	REN	DIO7
BUS2	13	REN	DIO7	REN	DIO7
DATA3	7	NC	DIO6	EOI2	DIO6
BUS3	15	EOI	DIO6	EOI	DIO6
DATA4	8	SRQ	DIO5	SRQ	DIO5
BUS4	16	SRQ	DIO5	SRQ	DIO5
DATA5	9	NRFO	DIO4	NRFO	DIO4
BUS5	17	NRFO	DIO4	NRFO	DIO4
DATA6	10	NDAC	DIO3	NDAC	DIO3
BUS6	18	NDAC	DIO3	NDAC	DIO3
DATA7	11	TI/RIO1	NC	ATN	ATNO
DATA8	23	TI/RIO2	DIO2	ATN	ATNO
BUS7	19	ATN	DIO2	ATN	ATNO
DATA9	24	GIO1	DAV	CIC	DAV
BUS8	21	GIO1	DAV	CLTH	DAV
DATA10	25	GIO2	DIO1	IFCL	DIO1
BUS9	22	GIO2	DIO1	SYC	DIO1
TI/R1	1	TI/R1	TI/R1	TI/R1	TI/R1
TI/R2	2	TI/R2	NC	TI/R2	IFC
EOI	3	EOI	EOI	EOI	EOI
ATN	4	ATN	ATN	ATN	ATN

*Note: These pins are the IEEE-488 bus non-inverting drivers/receivers. They include all the bus terminations required by the Standard and are connected directly to the GPB bus connector.

GENERAL DESCRIPTION

The 8293 is a bidirectional transceiver. It was designed to interface the Intel 8291A GPB Talker/Listener and the Intel 8292 GPB Controller to the IEEE Standard 488-1978 bus. The 8293 GPB Controller also referred to as the GPB. The 8293 GPB Transceiver meets or exceeds all of the electrical specifications defined in the IEEE Standard 488-1978.

The 8293 can be hardware programmed to one of four modes of operation. These modes allow the 8293 to be configured to support both a Talker/Listener/Controller and a Talker/Listener environment. In addition, the 8293 can be used as a general-purpose, three-state (push-pull) or open-collector bus transceiver with two driver/drivers. Two modes each are used to support a Talker/Listener (see Figure 3) and a Talker/Listener/Controller environment (see Figure 4). Mode 1 is used in general-purpose environments.

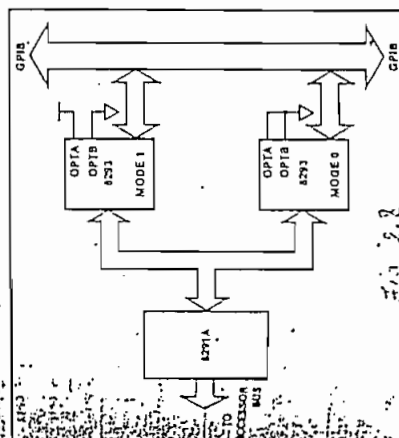


Figure 3. Talker/Listener Configuration

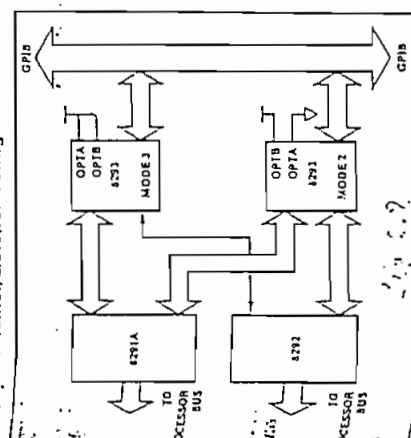


Figure 4. Talker/Listener/Controller Configuration

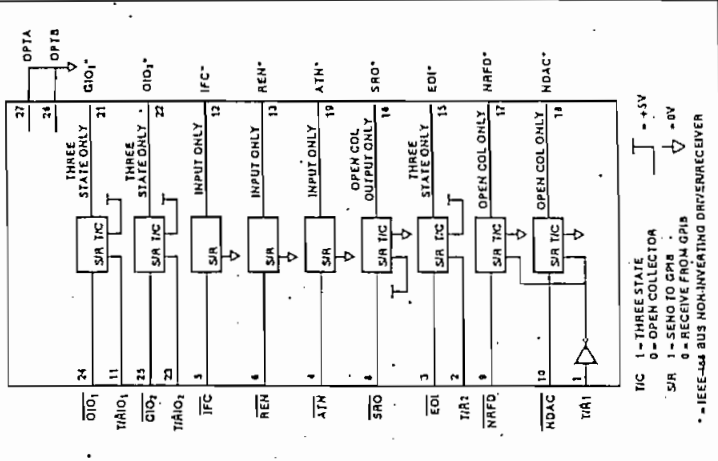


Figure 5. Mode 0 Pin Description

Table 3. Mode 0 Pin Description

Symbol	Pin	Type	Name and Function
TI/R1	1	I	Transmit Receive 1 Direction control for NDAC and NRFO. If TI/R1 is high, then NDAC* and NRFO* are receiving. Input is TTL compatible.
NDAC	10	IO	Not Data Accepted: Processor GPB bus handshake control line; used to indicate the condition of acceptance of data by device(s). It is TTL compatible.
NDAC*	18	IO	Not Data Accepted: IEEE GPB bus handshake control line. When an input, it is a TTL compatible Schmitt-trigger. When an output, it is an open-collector driver with 48 mA sinking capability.
NRFO	9	IO	Not Ready For Data: Processor GPB handshake control line; used to indicate the condition of readiness of device(s) to accept data. This pin is TTL compatible.

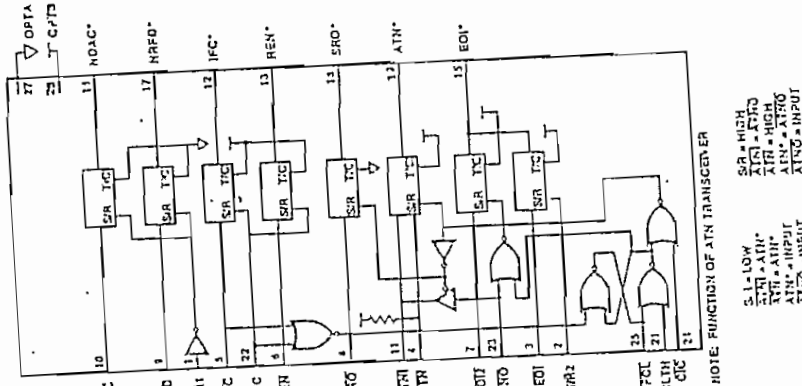
Table 3. Mode 0 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
NRFD*	17	I/O	Not Ready For Data: IEEE GPIB bus handshake control line. When an input, it is a TTL compatible Schmitt-trigger. When an output, it is an open-collector driver with a 49 mA current sinking capability.
TRF2	2	I	Transmit Receive 2: Direction control for EO1. If TRF2 is high, EO1* is sending. Input is TTL compatible.
EO1	3	I/O	End Of Identity: Processor GPIB bus control line; is used by a talker to indicate the end of a multiple byte transfer. This pin is a three-state (push-pull) driver capable of sinking 48 mA and a TTL compatible receiver with hysteresis.
EO1*	15	I/O	End Of Identity: IEEE GPIB bus control line; is used by a talker to indicate the end of a multiple byte transfer. This pin is a three-state (push-pull) driver capable of sinking 48 mA and a TTL compatible receiver with hysteresis.
SREQ	8	I	Service Request: Processor GPIB bus control line; used by a device to indicate the need for service and to request an interruption of the current sequence of events on the GPIB. It is a TTL compatible input.
SRO*	18	O	Service Request: IEEE GPIB bus control line; it is an open collector driver capable of sinking 43 mA.
REN	6	O	Remote Enable: Processor GPIB bus control line; used by a controller (in conjunction with other messages) to select between two alternate sources of device programming data (remote or local control). This output is TTL compatible.
REN*	13	I	Remote Enable: IEEE GPIB bus control line. This input is a TTL compatible Schmitt-trigger.
ATN	4	O	Attention: Processor GPIB bus control line; used by the 8291 to determine how data on the DIO signal lines are to be interpreted. This is a TTL compatible output.
ATN*	19	I	Attention: IEEE GPIB bus control line; this input is a TTL compatible Schmitt-trigger.
IFC	5	O	Interface Clear: Processor GPIB bus control line; used by a controller to place the interface system into a known quiescent state. It is a TTL compatible output.

Table 4. Mode 1 Pin Description

Symbol	Pin No.	Type	Name and Function
TRF1	1	I	Transmit Receive 1: Controls the direction for DAV and the DIO lines. If TRF1 is high, then all these lines are sending information to the IEEE GPIB lines. This input is TTL compatible.
EO1	3	I	End Of Sequence And Attention: Processor GPIB control lines. These two control signals are ANDed together to determine whether all the transceivers in the 8293 are three-state (push-pull) or open-collector. When both signals are low (true), then the controller is performing a parallel poll and the transceivers are all open-collector. These inputs are TTL compatible.
DAV	24	I/O	Data Valid: Processor GPIB bus handshake control line; used to indicate the condition (availability and validity) of information on the DIO lines. It is TTL compatible.
DAV*	21	I/O	Data Valid: IEEE GPIB bus handshake control line. When an input, it is a TTL compatible Schmitt-trigger. When DAV* is an output, it can sink 48 mA.
DIO1*, DIO3	25, 23, 10, 9, 8, 7, 6, 5	I/O	Data Input/Output: Processor GPIB bus data lines; used to carry message and data bytes in a bit-parallel byte-serial form controlled by the three handshake signals. These lines are TTL compatible.
DIO1*, DIO3	22, 19, 18, 17, 16, 15, 13, 12	I/O	Data Input/Output: IEEE GPIB bus data lines. They are TTL compatible Schmitt-triggers when used for input and can sink 43 mA when used for output. See ATN and EO1 description for output mode.

MODE 1



NOTE: FUNCTION OF ATN TRANSCEIVER

S.L. = LOW
S.H. = HIGH
TRF1 = INPUT
TRF2 = INPUT
ATN = INPUT
ATN* = INPUT
ATN* = INPUT
ATN* = INPUT

Figure 7. Talker/Listener/Controller Configuration

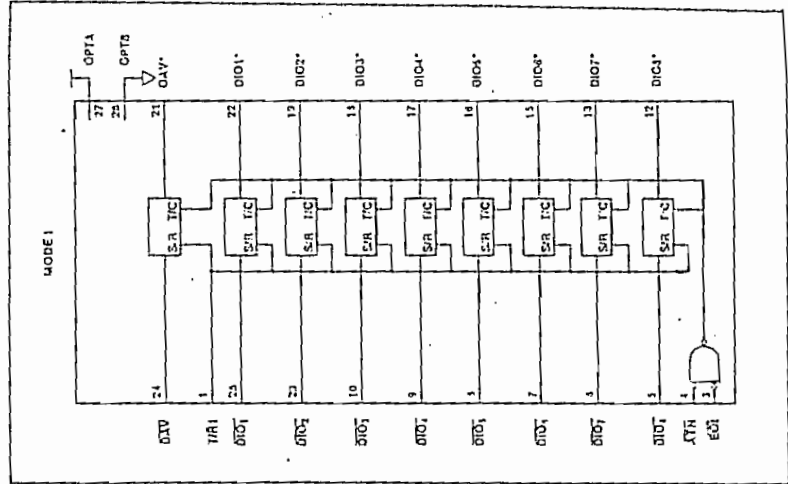


Figure 8. Talker/Listener Data Configuration

Table 3. Mode 2 Pin Description

Symbol	Pin No.	Type	Name and Function
TR1	1	I	Transmit Receive 1: Direction control for NDAC and RRD. If TR1 is high, then NDAC and RRD are receiving. Input is TTL compatible.
NDAC	10	VO	Not Data Accepted: Processor GPIB bus handshake control line; used to indicate the condition of acceptance of data by device(s). This pin is TTL compatible.
NDAC	18	VO	Not Data Accepted: IEEE GPIB bus handshake control line. It is a TTL compatible Schmitt-trigger when used for input and an open-collector driver with a 48 mA current sink capability when used for output.
RRD	9	VO	Not Ready For Data: Processor GPIB bus handshake control line; used to indicate the condition of readiness of device(s) to accept data. This pin is TTL compatible.
RRD	17	VO	Not Ready For Data: IEEE GPIB bus handshake control line. It is a TTL compatible Schmitt-trigger when used for input and an open-collector driver with a 48 mA current sink capability when used for output.
IC	22	I	System Controller: Used to monitor the system controller switch and control the direction for IFC and REN. This pin is a TTL compatible input.
REN	8	VO	Remote Enable: Processor GPIB control line; used by the active controller in conjunction with other messages to select between two alternate sources of device programming data (remote or local control). This pin is TTL compatible.
	13	VO	Remote Enable: IEEE GPIB bus control line. When used as an input, this is a TTL compatible Schmitt-trigger. When an output, it is a three-state driver with a 48 mA current sink capability.
	5	VO	Interface Clear: Processor GPIB bus control line; used by the active controller to place the interface system into a known quiescent state. This pin is TTL compatible.
	12	VO	Interface Clear: IEEE GPIB control line. This is a TTL compatible Schmitt-trigger when used for input and a three-state driver capable of sinking 48 mA current when used for output.
	24	I	Controller In Charge: Used to control the direction of the SRO and to indicate that the S292 is in charge of the bus. CIC is a TTL compatible input.

Table 5. Mode 2 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
EOI	15	VO	End Or Identify: IEEE GPIB bus control line; used by a talker to indicate the end of a multiple byte transfer sequence or, by a controller in conjunction with ATN, to execute a polling sequence. When an output, this pin can sink 48 mA current. When an input, it is a TTL compatible Schmitt-trigger.

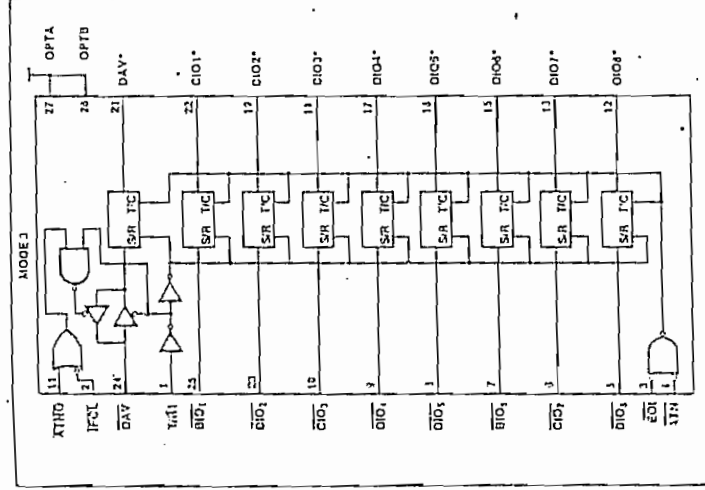


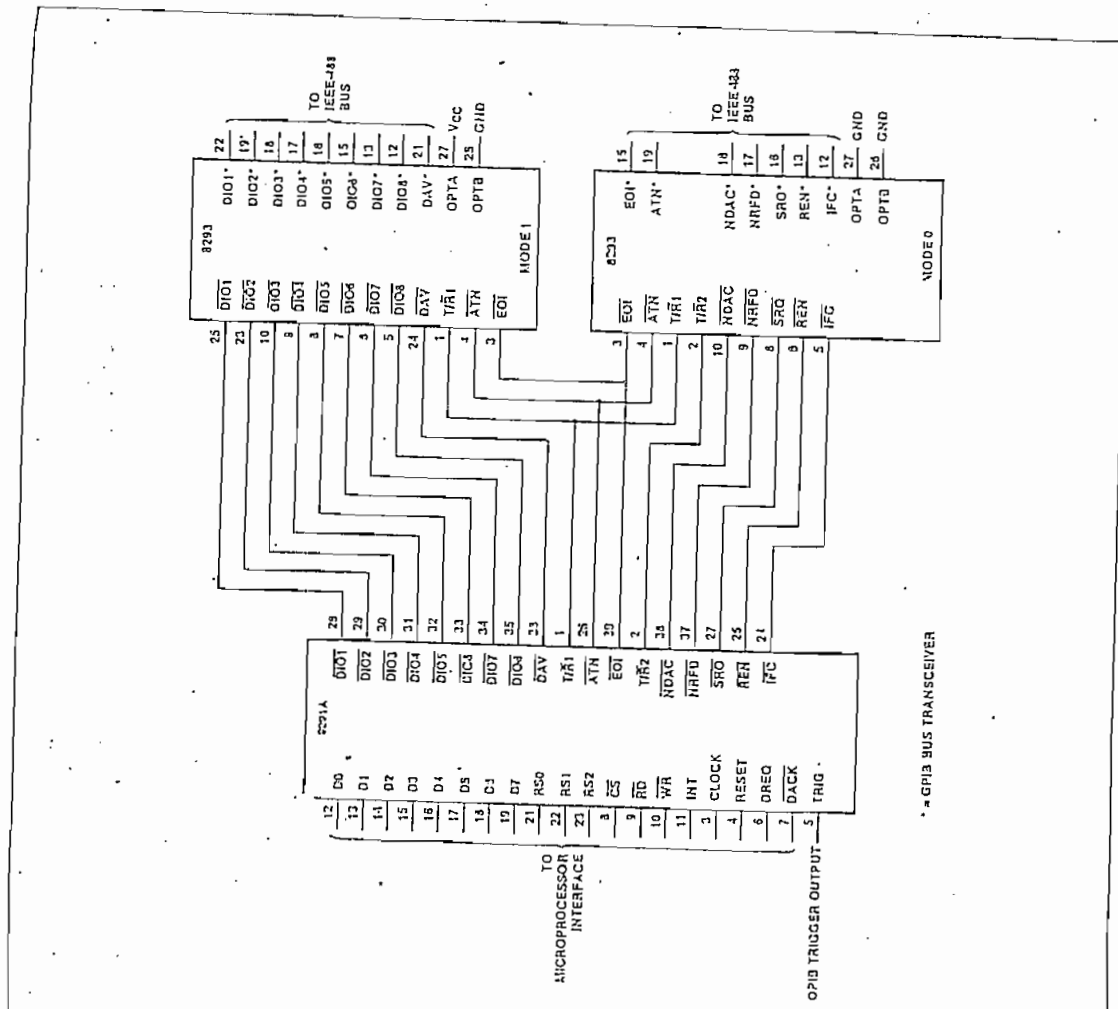
Figure 3. Talker/Listener/Controller Data Configuration

Table 6. Mode 3 Pin Description

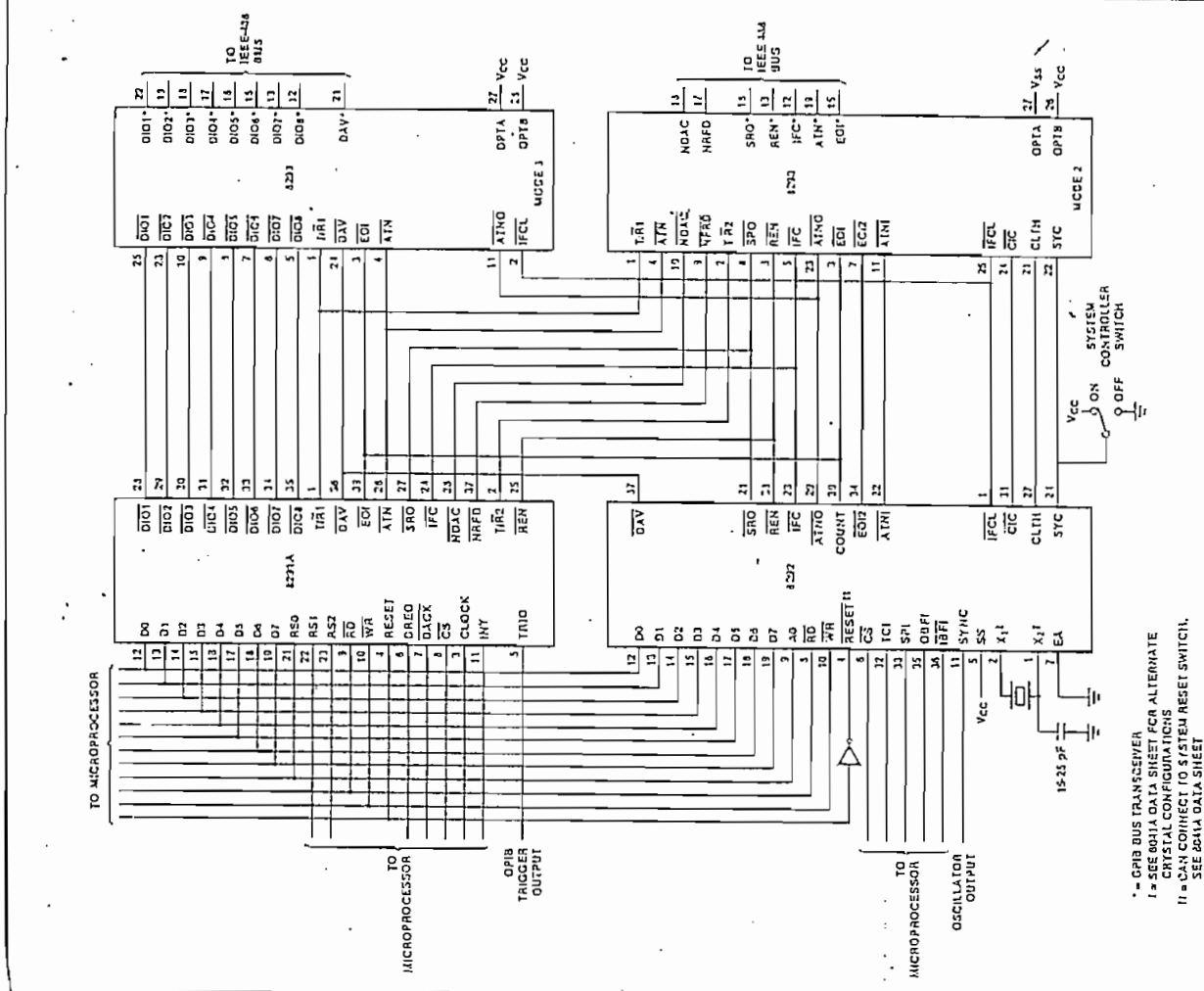
Symbol	Pin No.	Type	Name and Function
TR1	1	I	Transmit Receive 1: Controls the direction for DAV and the DIO lines. If TR1 is high, then all three lines are sending information to the IEEE GPIB bus. This pin is TTL compatible.
EOI	3	I	End Of Sequence and Attention: Processor GPIB control line. These two control lines are used together to determine whether all the transceivers in the S292 are push-out or open-collector. When both signals are low (true), then the controller is performing a parallel poll and the transceivers are all open-collector. These inputs are TTL compatible.
ATN	4	I	Attention Out: Processor GPIB control line; used by the S292 during "take control synchronously" operations. This pin is TTL compatible.
ATN	11	I	Attention Out: Processor GPIB control line; used by the S292 during "take control synchronously" operations. This pin is TTL compatible.
IFCL	2	I	Interface Clear Latched: Used to make DAV received after the system controller asserts IFC. This input is TTL compatible.
DAV	34	VO	Data Valid: Processor GPIB handshake control line; used to indicate the condition (availability and validity) of information on the DIO signals. This pin is TTL compatible.
DAV	21	VO	Data Valid: IEEE GPIB handshake control line. When an input, this pin is a TTL compatible Schmitt-trigger. When DAV is an output, it can sink 48 mA.
DIO1-DIO8	25, 23, 10, 9, 8, 7, 6, 5	VO	Data Input/Output: Processor GPIB bus data lines; used to carry messages and data bytes in a bit-parallel system serial from controller to the three handshake signals. These lines are TTL compatible.
DIO1-DIO3	22, 19, 13, 17, 16, 15, 13, 12	VO	Data Input/Output: IEEE GPIB bus data lines. They are TTL compatible Schmitt-triggers when used for input and can sink 48 mA when used for output.

NOTES:

- V_{DD} is guaranteed at 1.1V even these inputs to recover the high current-sourcing capability of these pins under 1.1V input in Mode 2.



Figuro 9. 8291A and 8293 System Configuration



* - GPIB BUS TRANSCIVER
1 - SEE 8041A DATA SHEET FOR ALTERNATE
CRYSTAL CONFIGURATIONS
1 - CAN CONNECT TO SYSTEM RESET SWITCH.
SEE 8041A DATA SHEET

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.....-1.0V to +7V
Power Dissipation.....1 Watt

*NOTICE:

1. 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.
2. All devices are guaranteed to operate within the minimum and maximum parameter limits specified below. Typical parameters however are not tested and are not guaranteed. Established statistically, they indicate the performance level expected in a typical device at room temperature ($T_A = 25^\circ\text{C}$) and $V_{CC} = 5\text{V}$.

D.C. CHARACTERISTICS ($T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 10\%$, GND = 0V)

Symbol	Parameter	Limits		Units	Test Conditions
		Min.	Typ.		
V_{IL1}	Input Low Voltage (GPIO Bus Pins)		0.8	V	
V_{IL2}	Input Low Voltage (Option Pins)	-0.1	0.1	V	
V_{IL3}^1	Input Low Voltage (All Others)		0.8	V	
V_{IH1}	Input High Voltage (GPIO Bus Pins)	2.0	V_{CC}	V	
V_{IH2}	Input High Voltage (Option Pins)	4.5	V_{CC}	V	
V_{IH3}	Input High Voltage (All Others)	2.0	V_{CC}	V	
V_{IH4}	Receiver Input Hysteresis	400		mV	
V_{OL1}	Output Low Voltage (GPIO Bus Pins)		0.5	V	$I_{OL} = 48\text{ mA}$
V_{OL2}	Output Low Voltage (All Others)		0.5	V	$I_{OL} = 16\text{ mA}$
V_{OH1}	Output High Voltage (GPIO Bus Pins)	2.4		V	$I_{OH} = -5.2\text{ mA}$
V_{OH2}	Output High Voltage (All Others)	2.4		V	$I_{OH} = -800\text{ }\mu\text{A}$
V_{IT}	Receiver Input Threshold	0.8		V	
I_{LC}	Input Load Current (GPIO Pins)	See Bus Load Line Diagram			$V_{CC} = 5.0\text{V} \pm 5\%$
I_{IL}	Input Leakage Current (All Others)		10	μA	$0.45 \leq V_{IN} \leq V_{CC}$
I_{PO}	Bus Power Down Leakage Current		40	μA	$0.45\text{V} \leq V_{BUS} \leq 2.7\text{V}$
I_{CC}	Power Supply Current		110	mA	

NOTES:

$V_{IL3} = 1.1\text{V}$ max on pins 21 and 22 in Mode 2 for the 82C33-10.

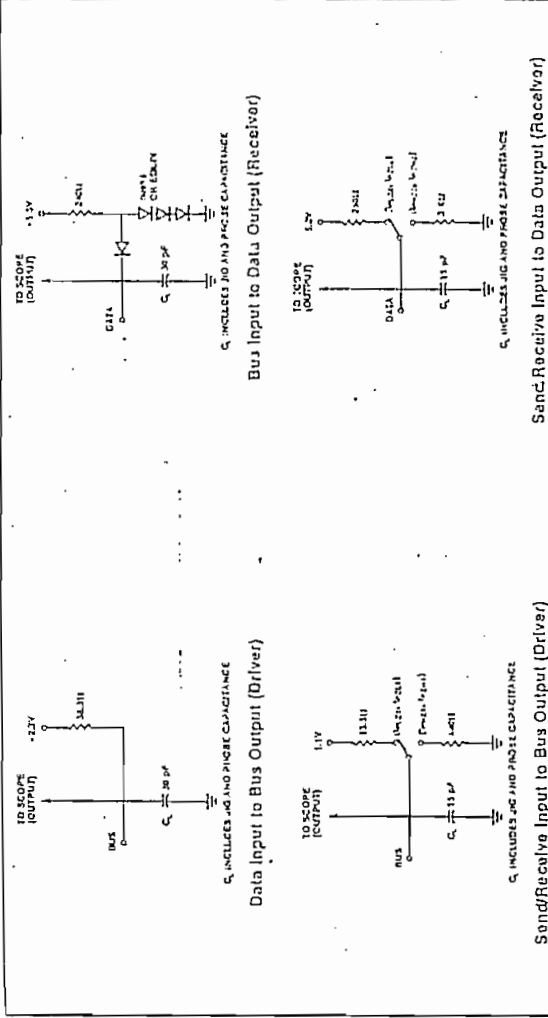
APACITANCE

Symbol	Parameter	Min.	Typ.	Max.	Units	Test Conditions
C_{IO1}	I/O Capacitance (GPIO Side)		50	80	pF	$V_{IN} = V_{CC}$
C_{IO2}	I/O Capacitance (System Side)		35	50	pF	$V_{IN} = V_{CC}$
C_{IR}	Input Capacitance (T.R1, T.R2)		7	10	pF	$V_{IN} = V_{CC}$

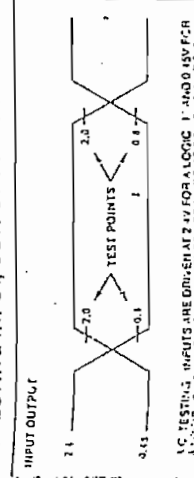
A.C. CHARACTERISTICS ($T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5.0\text{V} \pm 10\%$, GND = 0V)

Symbol	Parameter	Max.	Units
t_{p1}	Transmitter Propagation Delay (All Lines)	30	ns
t_{p2}	Receiver Propagation Delay (EOI, ATN and Handshake Lines)	50	ns
t_{p3}	Receiver Propagation Delay (All Other Lines)	80	ns
t_{PHZ1}	Transmitter Disable Delay (High to 3-State)	40	ns
t_{PLZ1}	Transmitter Enable Delay (3-state to High)	40	ns
t_{PZL1}	Transmitter Enable Delay (3-State to Low)	40	ns
t_{PHZ2}	Receiver Disable Delay (High to 3-State)	40	ns
t_{PLZ2}	Receiver Enable Delay (3-State to High)	40	ns
t_{PZL2}	Receiver Enable Delay (Low to 3-State)	40	ns
t_{HS}	Mode Switch Delay	10	μs

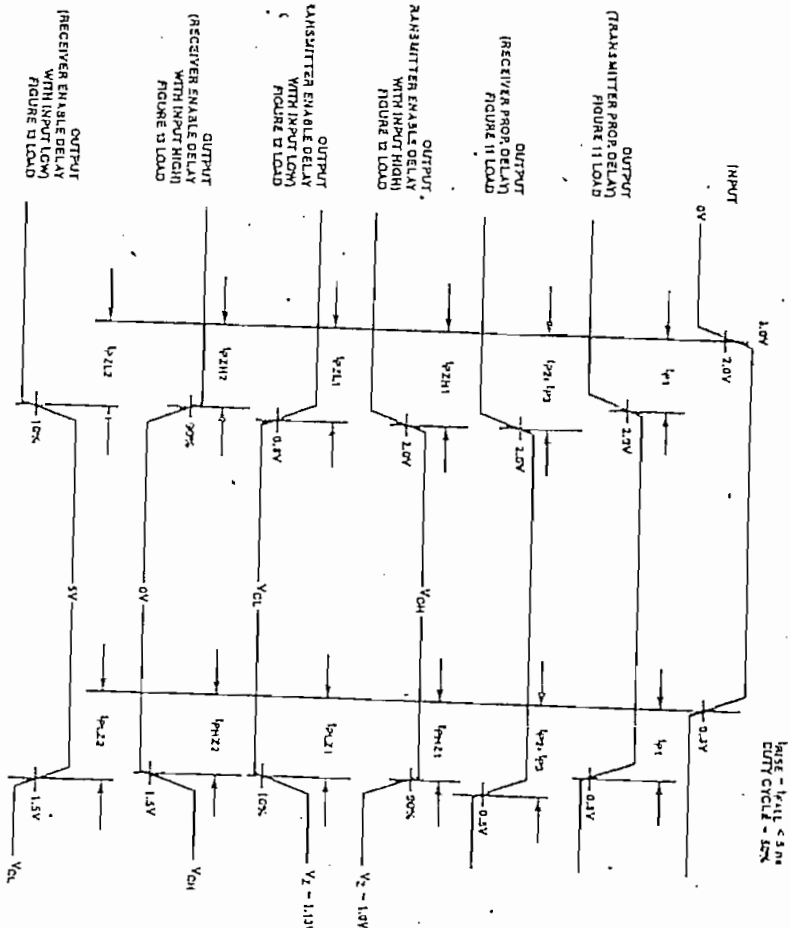
TYPICAL OUTPUT LOADING CIRCUITS



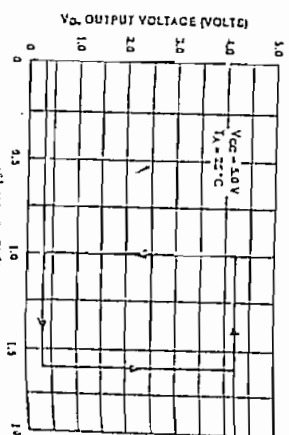
A.C. TESTING INPUT, OUTPUT WAVEFORM



VEFORMAIS



TYPICAL RECEIVER HYSTERESIS CHARACTERISTICS

8294A
DATA ENCRYPTION UNIT

- Certified by National Bureau of Standards
- 400 Byte/Sec Data Conversion Rate
- 64-Bit Data Encryption Using 56-Bit Key
- DMA Interface
- 3 Interrupt Outputs to Aid in Loading and Unloading Data
- 7-Bit User Output Port
- Single 5V $\pm 10\%$ Power Supply
- Fully Compatible with IAPX-86, 88, MCS-85™, MCS-80™, MCS-51™, and MCS-48™ Processors
- Implements Federal Information Processing Data Encryption Standard
- Encrypt and Decrypt Modes Available

The Intel® 8294A Data Encryption Unit (DEU) is a microprocessor peripheral device designed to encrypt and decrypt 64-bit blocks of data using the algorithm specified in the Federal Information Processing Data Encryption Standard. The DEU operates on 64-bit text words using a 56-bit user-specified key to produce 64-bit cipher words. The operation is reversible: if the cipher word is operated upon, the original text word is produced. The algorithm itself is permanently contained in the 8294A; however, the 56-bit key is user-defined and may be changed at any time.

The 56-bit key and 64-bit message data are transferred to and from the 8294A in 8-bit bytes by way of the system data bus. A DMA interface and three interrupt outputs are available to minimize software overhead associated with data transfer. Also, by using the DMA interface two or more DEUs may be operated in parallel to achieve effective system conversion rates which are virtually any multiple of 400 bytes/second. The 8294A also has a 7-bit TTL compatible output port for user-specified functions.

Because the 8294A implements the NBS encryption algorithm it can be used in a variety of Electronic Funds Transfer applications as well as other electronic banking and data handling applications where data must be encrypted.

