

300617



# UNIVERSIDAD LA SALLE

ESCUELA DE INGENIERIA  
INCORPORADA A LA U.N.A.M.

79  
2ej

## DESARROLLO DE UNA INTERFAZ SCSI <-> IBUS-III

TESIS PROFESIONAL  
QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
ESPECIALIDAD EN ELECTRONICA Y  
COMUNICACIONES  
P R E S E N T A :  
RICARDO VILLALOBOS MURO

Director de Tesis: Ing. Guillermo Aranda Pérez

MEXICO, D. F.

TESIS CON  
MESA DE ORIENTACION

199



Universidad Nacional  
Autónoma de México



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## INDICE

Introducción	1
1. Sistema de Adquisición y Control	4
1.1 Descripción del sistema de adquisición y control	4
1.2 Canal de interconexión IBUS-III	5
1.2.1 Introducción	5
1.2.2 Descripción funcional	6
1.2.3 Espacio de direccionamiento	6
1.2.4 Compatibilidad de sistemas de 8 y 16 bits	7
1.2.5 Sistema multimaestro	7
1.2.6 Estructura del canal	7
1.2.7 Función de las líneas del canal	8
1.2.8 Características de las señales del canal	11
1.2.9 Distribución topológica de las líneas	12
1.2.10 Características de los tipos de direccionamiento	14
1.2.11 Empleo de los dispositivos de 8 y 16 bits	15
1.2.12 Manejo de las líneas de interrupción	16
2. Canal SCSI	17
2.1 Introducción	17
2.2 Características físicas	17
2.2.1 Requerimientos de cables	17
2.2.2 Requerimientos de conectores	17
2.2.3 Características de las señales de salida	21
2.2.4 Características de las señales de entrada	21
2.2.5 Terminal opcional de poder	21
2.2.6 Señales del canal	24
2.2.7 Valores de las señales	25
2.2.8 Fuentes de las señales	26
2.3 Características lógicas	27
2.3.1 Fases del canal	27
2.3.2 Condiciones del canal	29
2.3.3 Secuencia de fases del canal	30
2.3.4 Apuntadores	31
2.3.5 Protocolo de mensajes	32
3. Propuesta y análisis de la posible solución	34

4.	Descripción de circuitos	39
4.1	Microprocesador 80C188	39
4.1.1	Características generales	39
4.1.2	Interrupciones	39
4.1.3	Condiciones Iniciales del circuito	42
4.1.4	Interfaz con periféricos interno	42
4.1.5	Selección de circuitos de memoria	45
4.1.6	Selección de circuitos periféricos	48
4.1.7	Generación de <u>READY</u>	49
4.1.8	Canales DMA	50
4.1.9	Controlador de interrupciones	54
4.2	Controlador de canal SCSI	62
4.2.1	Descripción general	62
4.2.2	Descripción de terminales	63
4.2.3	Registros internos	66
4.2.4	Soporte de <u>hardware</u> en el circuito	76
4.2.5	Interrupciones	76
4.2.6	Condiciones de inicio	81
4.2.7	Transferencias de datos	82
4.3	Memoria RAM doble puerto	85
4.3.1	Descripción general	85
4.3.2	Descripción de terminales	86
4.3.3	Descripción funcional	89
4.3.4	Arbitraje de contención	90
5.	Tarjeta SAC-SCSI	92
5.1	Descripción general	92
5.2	Descripción de la interfaz a IBUS-III	94
5.3	Direccionamiento en IBUS-III	98
5.4	Descripción de la sección de procesamiento	98
5.5	Descripción de la interfaz al canal SCSI	105
5.6	Descripción de canales de comunicación serie	108
6.	Programación	111
6.1	Rutina de inicialización de microprocesador 80C188	111
6.2	Rutina para tarjeta interfaz IBUS-III <-> SCSI	112
6.3	Rutina para la tarjeta que simula un dispositivo SCSI	113
	Resultados y conclusiones	122
	Bibliografía	126
	Apéndice Diagramas eléctricos de la tarjeta	127

## INTRODUCCION

El objetivo del Instituto de Investigaciones Eléctricas (IIE) es promover y apoyar el desarrollo y la innovación tecnológica en el sector y la industria eléctricos, mediante la investigación aplicada, la transferencia de tecnología y los servicios técnicos especializados.

El Departamento de Electrónica del IIE ha desarrollado diversos equipos electrónicos con los que se integran sistemas de Control Distribuido y Control Supervisorio para las plantas de generación y distribución de la energía eléctrica del país.

El propósito de los equipos diseñados y construídos por este departamento, es proporcionar la máxima autonomía a los usuarios de manera que aprovechen óptimamente la capacidad del equipo, sin que dependan excesivamente del proveedor en cuanto a modificaciones o expansiones que ellos mismos pueden realizar.

Este es el caso de la línea SAC (Sistema de Adquisición y Control), que es un conjunto de módulos electrónicos que pueden aplicarse en el control de procesos industriales.

La principal característica de SAC es poder monitorear y manejar dispositivos del mundo real. El concepto básico de SAC es trasladar las computadoras del centro de control al campo, junto a los dispositivos a controlar; por lo mismo, sus características físicas le permiten soportar condiciones ambientales difíciles.

Los diferentes módulos de este sistema están interconectados a través de un canal (bus) trasplano (Norma IBUS-III), que fue desarrollado por el mismo departamento, con las características necesarias para los diseños realizados. La transferencia de información a dispositivos externos se realiza por medio de las tarjetas de comunicación (SAC-821), por canal serie (RS-232 ó RS-485), lo que provoca, hasta cierto punto, que el sistema sea lento.

La tarjeta de interfaz IBUS-III <-> canal SCSI (Small Computer System Interface) surge como una necesidad para conectar sistemas de almacenamiento masivo de manera rápida con los sistemas ya existentes de control y adquisición de datos, debido a que este es un canal paralelo de 8 bits, lo que permitirá que la transferencia de información sea por lo menos 8 veces más rápida que con los sistemas actuales; otra ventaja que ofrece este canal, es la posibilidad de conectar varios dispositivos (hasta 8), estableciéndose la comunicación sólo entre 2 a un mismo tiempo, además de que este canal es bastante independiente de los dispositivos que le sean conectados, es decir que en el mismo canal se pueden conectar tanto discos rígidos como discos ópticos, cintas magnéticas o impresoras, sin la necesidad de un manejador especial para cada uno de esos dispositivos.

El canal SCSI está basado en una interfaz para manejador de disco, desarrollada por Shugart Associates, conocida como Shugart Associates System Interface (SASI), por ser un canal paralelo comercial muy rápido que se está usando en muchos dispositivos periféricos y sistemas.

Este canal se presenta en dos modos: diferencial y single-ended, con longitudes de hasta 25 y 6 m respectivamente. Asimismo ofrece dos modos de transferencia: síncrona y asíncrona, los que permite velocidades de transferencia de 4 y 1.5 Mbytes/s respectivamente.

El canal SCSI según la norma ANSI X3.131-1986, conocida comúnmente como SCSI-1, encontró pocos adeptos entre los fabricantes de sistemas de alta tecnología, ya que los rangos de transferencia de datos estaban atrasados respecto a los obtenibles con los semiconductores, especialmente en los dispositivos de muy alta escala de integración (VLSI). Pero la existencia de un estándar fue el incentivo para renovar esfuerzos por el comité ANSI X3T9.2 y mejorar el SCSI-1. La cuarta versión SCSI-2 es el resultado final y es completamente compatible con los estándares anteriores.

La versión más avanzada de este canal y dispositivos más sofisticados, han logrado que en la actualidad se cuente con sistemas cuya velocidad de transferencia es de 80 Mbytes/s.

El objetivo de este trabajo es realizar una interfaz entre IBUS-III y el canal SCSI con dos características principales:

a) modo single-ended, ya que se considera que la tarjeta será usada dentro de un gabinete.

b) velocidad de transferencia mínimo de 1.0 Mbyte/s, considerando que en la norma ANSI X3.131-1986 se especifica 1.5 Mbytes/s como velocidad para transferencias asíncronas.

Debido a que esta interfaz es poco conocida y a que la implantación del software para la norma SCSI es muy compleja, todo el software desarrollado será exclusivamente para pruebas, no para soportar transferencias con dispositivos periféricos comerciales de almacenamiento masivo, como discos duros, cintas magnéticas y discos ópticos.

El trabajo se presenta en 6 capítulos, el primero es una descripción de los sistemas desarrollados en el IIE, como son el IBUS-III y la línea SAC; el capítulo dos describe el canal SCSI según la norma ANSI X3.131-1986, en el capítulo tres se describen las posibles soluciones y cuál sería la más conveniente; el capítulo cuatro describe las características de los componentes que se usaron por vez primera en la elaboración de una tarjeta en el IIE, estos son: procesador 80C188, controlador de canal SCSI NCR-53C80 y memoria estática de doble puerto CY-131; el capítulo cinco es el diseño y desarrollo de la tarjeta, aquí se explica como se conectaron los circuitos y algunas condiciones necesarias para activar señales; el capítulo seis describe el software desarrollado para realizar pruebas al sistema; por último, los resultados obtenidos y las conclusiones. Además se tiene un apéndice con los diagramas de conexión de todo el sistema.

## 1. DESCRIPCION GENERAL.

### 1.1 Descripción del Sistema de Adquisición y Control (SAC).

La "inteligencia" del SAC se basa en la tarjeta maestra o procesadora, que puede ser programada por el usuario a través de una terminal o una computadora personal para llevar a cabo tareas complejas.

Algunas tarjetas, como las de entradas digitales, le permiten sensar interruptores; otras, como las de entradas analógicas le permiten monitorear variables físicas, como temperatura, presión, etc.

Tarjetas como las de salidas digitales, pueden encender o apagar diodos emisores de luz (led's), indicadores luminosos u otros dispositivos externos y otras, como las de salidas analógicas, pueden controlar variables continuas.

El sistema está constituido además, por una tarjeta de propósito múltiple y una de comunicaciones. La tarjeta de propósito múltiple tiene una área para alambrado, utilizable por el usuario, dependiendo de la aplicación que se requiera. Dicha tarjeta se puede utilizar, por ejemplo, para controlar un teclado y un despliegue (display) a fin de que el usuario pueda ordenar a la procesadora realizar diferentes operaciones.

La tarjeta de comunicaciones sirve para que el sistema se comunique con otro sistema similar, con una computadora externa o con un monitor por medio de la interfaz serial.

Las tarjetas que componen el sistema SAC son:

#### Procesadoras:

SAC 1126 Maestra  
SAC 1887 Maestra

#### Expansión de memoria:

SAC 929 Expansión de memoria 96 Kb RAM EPROM EEPROM  
SAC 930 Expansión de memoria 393 Kb RAM EPROM

#### Salidas Digitales:

SAC 158 Salidas Digitales con relevador  
SAC 612 Salidas Digitales con transistor

- SAC 613 Salidas Digitales con transistor con protección de sobrecorriente  
 SAC 617 32 Salidas Digitales

**Entradas Digitales:**

- SAC 415 Entradas digitales optoacopladas  
 SAC 417 32 Entradas digitales  
 SAC 421 Entradas digitales con interrupción  
 SAC 460 32 Entradas/Salidas digitales

**Entradas Analógicas:**

- SAC 700 Controladora de entradas analógicas  
 SAC 701 Controladora de entradas analógicas con conversión A/D rápida  
 SAC 720 Entradas Analógicas

**Salidas Analógicas:**

- SAC 512 Salidas analógicas

**Comunicación:**

- SAC 821 Comunicación serie RS-232, RS-485, bit a bit

**Otras:**

- SAC 398 PID  
 SAC 300/310 Expansoras de BUS (maestra/esclava)

Todas estas tarjetas se comunican entre sí a través del canal trasplano (IBUS-III).

**1.2 Descripción de la norma del canal de interconexión IBUS-III.**

**1.2.1 Introducción.**

IBUS-III es un canal digital de propósito general, diseñado para trabajar simultáneamente con módulos de 8 y 16 bits. A través de sus líneas, los diferentes componentes de un sistema, además de ser alimentados y sincronizados, pueden intercambiar (asíncronamente) información necesaria para la realización de una tarea. Proporciona también herramientas para la implantación de sistemas, tanto del tipo uni-maestro como multi-maestro, ya que dispone de líneas especialmente dedicadas al arbitraje necesario para la asignación del canal.

Dependiendo del tipo de aplicación, se puede escoger de entre dos diferentes tamaños de módulos, los cuales pueden ser procesadores de 8 ó 16 bits, memorias, tarjetas de Entrada/Salida Analógico/Digital, tarjetas para la comunicación, etc.

En el espacio de direccionamiento de 1 Mbyte, los intercambios elementales de información se efectúan en bloques de 8 ó 16 bits.

Además, IBUS-III permite diseñar sistemas modulares y adaptables. La flexibilidad de su concepción da la oportunidad de un eventual crecimiento para poder servir como base a nuevos sistemas o para incluir nuevas posibilidades: métodos más sofisticados para la identificación y procesamiento de las interrupciones, etc. Para lo anterior, se han destinado algunas líneas opcionales para aplicaciones específicas del usuario y otras más se han reservado para permitir futuras expansiones.

### 1.2.2 Descripción funcional.

IBUS-III es un canal digital diseñado para interconectar hasta 18 módulos diferentes (con su respectiva fuente de poder); estos pueden ser de varios tipos: cómputo, comunicaciones, adquisición de señales, etc. IBUS-III permite el intercambio entre palabras de 8 y 16 bits. Todas las transacciones entre los módulos interconectados al sistema a través de IBUS-III, se efectúan mediante un grupo de 80 señales.

La norma especifica un tamaño máximo de 16 espacios para tarjetas No-Maestras, pero además existen trasplanos con espacio para 9 y 5 módulos; estos últimos aceptan un máximo de tarjetas No-Maestras de 8 y 4 respectivamente y ambas dan cabida a un solo maestro. Se debe considerar que aún cuando el número de espacios sea distinto para cada tipo de trasplano, se deben seguir respetando las direcciones relativas para la selección a través de las páginas por posición (PASP/).

### 1.2.3 Espacio de direccionamiento.

El espacio de direccionamiento en IBUS-III es de 1 Mbyte de memoria general (direccionada mediante el uso de la línea PGEN/) y 2 Kbytes de memoria por posición (direccionada mediante el uso de la línea PASP/). La utilización de PGEN/ requiere de la decodificación de las 20 líneas de dirección, mientras que el uso de PASP/ sólo requiere de las 15 líneas menos significativas.

#### 1.2.4 Compatibilidad de sistemas de 8 Y 16 bits.

Dentro de la estructura de IBUS-III, es posible combinar dispositivos de 8 y 16 bits. El enfoque buscado es el de simplificar al máximo los dispositivos de 8 bits, evitando entre otras cosas, que tengan que manejar las 8 líneas más significativas de los datos. Por esta razón, todos los intercambios de información a nivel de bytes, son efectuados a través de los 8 bits menos significativos de las líneas de datos, reservándose los 8 más significativos a la transmisión de palabras completas.

Para lograr este objetivo, es necesario que los módulos de 16 bits (maestros y esclavos) incluyan los medios necesarios para intercomunicar el byte más significativo de las líneas internas de datos con el menos significativo de IBUS-III.

#### 1.2.5 Sistema multi-maestro.

El diseño de IBUS-III incluye los elementos necesarios para que dentro de un sistema determinado, coexistan dos o más maestros efectuando concurrentemente operaciones a través del canal. Es éste el caso de un sistema multi-maestro. Se entiende por maestro, a aquel módulo que tiene la capacidad de ordenar directamente la ejecución de una transacción a través del canal, mientras que el esclavo no tiene esa capacidad y sólo puede tener acceso al canal como una respuesta a una orden del maestro.

#### 1.2.6 Estructura del canal.

Basados en el tipo de función de las líneas, las señales pueden dividirse en 6 categorías:

- Datos. Compuesta de 16 líneas.
- Direcciones. Compuesta de 27 líneas.
- Control. Compuesta de 12 líneas.
- Interrupciones. Compuesta de 3 líneas.
- Alimentación. Compuesta de 18 líneas.
- Reservadas. Compuesta de 3 líneas.

Las líneas de datos son el medio por el cual la información es intercambiada entre los diferentes módulos del sistema, mientras que las de direcciones especifican el módulo y la localidad, dentro de éste, que se desea acceder durante el intercambio de información.

Las líneas de control especifican las funciones que se deben realizar y las de interrupción son utilizadas por esclavos, fuente de poder y, en ciertos casos, algunos maestros, para indicar al maestro del sistema del requerimiento de algún servicio. Las líneas de alimentación tienen como tarea proporcionar los voltajes regulados y la corriente adecuada para todos los módulos.

### 1.2.7 Función de las líneas de IBUS-III.

#### Líneas de datos

**DATO\*...DAT15\***. Son 16 líneas bidireccionales de datos. Se emplean para el intercambio de información entre los diferentes módulos de un sistema. Todas las transacciones a nivel byte se hacen a través de los 8 bits menos significativos (DATO\*...DAT7\*), mientras que las operaciones a nivel palabras completas, utilizan estas líneas en su totalidad. Los 8 bits más significativos (DAT8\*...DAT15\*), pueden ser ignorados por todos los módulos de 8 bits.

#### Líneas de dirección

**DIRO\*...DIR19\***. Son 20 líneas de direccionamiento. Se utilizan para transmitir la dirección de la localidad que se desea acceder, ya sea un direccionamiento por PGEN (direccionamiento a Páginas Generales) o PASP (direccionamiento por Posición). El direccionamiento mediante PGEN requiere de la decodificación de las 20 líneas de dirección, mientras que el de PASP sólo utiliza las 15 líneas menos significativas. En los módulos de 16 bits, DIRO\* activa la conexión del byte más significativo de sus líneas internas de datos con el byte menos significativo de las líneas de datos del sistema.

**BHEN/**. Habilitador del Byte Más Significativo. Empleado para especificar que se desea acceder un byte impar, sea como parte de una palabra completa (DIRO\* directo) o como un solo byte (DIRO\* complementado). En los maestros de 8 bits, siempre se debe cumplir que BHEN/ = DIRO\*. Los esclavos de 8 bits pueden ignorar esta señal.

**PASP/.** Es la línea de selección de Páginas por Posición. Cuando **PASP/** se encuentra activa, indica que se está accediendo algún dispositivo colocado en los espacios destinados físicamente para ello. Dado que sólo utiliza las 15 líneas menos significativas para la decodificación de la localidad, y además 4 de ellas direccionan la posición geográfica a la cual se desea acceder, se tienen 16 espacios disponibles, de los cuales se pueden direccionar en cada uno 2 KBytes.

**PGEN/.** Es la línea de selección de Páginas Generales. No se pueden efectuar direccionamientos a Páginas por Posición y a Páginas Generales simultáneamente, lo que significa que las líneas **PASP/** y **PGEN/** son mutuamente excluyentes. El espacio de direccionamiento por Páginas Generales es de 1 Mbyte. Con **PGEN/** se utilizan las 20 líneas de dirección para decodificar la localidad que se desea acceder. Al igual que para la línea **PASP/**, para **PGEN/** se hace referencia en la manera de utilizar las líneas de dirección y el espacio de direccionamiento cuando **PGEN/** está activa.

**SP0...SP3.** Son las líneas de Selección por Posición. Con estas cuatro líneas se efectúa la selección de los espacios geográficos (Selección por Posición). Estas líneas no están interconectadas entre los diferentes módulos (espacios del canal). Su función es asignar un número de identificación único a cada uno de los 16 espacios disponibles en **IBUS-III**. Estas 16 posiciones son identificadas por las combinaciones entre los distintos valores de **SP0** a **SP3** ("0000", "0001", ..., "1111"). En el cuerpo del trasplano, el "0" lógico corresponde a una conexión a tierra y el "1" a una línea abierta. Es necesario utilizar un pull-up en todas las líneas **SP's**.

### Líneas de Control

**INIC/.** Señal de condición Inicial. Lleva al sistema, en conjunto, a un estado interno conocido cuando se encuentra activa. Esta señal puede ser activada por cualquiera de los módulos conectados al trasplano.

**RD/.** Orden de Lectura. Su activación indica que en las líneas de dirección del sistema se dispone de una dirección válida, y que se desea efectuar la lectura de información (byte o palabra) contenida en una localidad accesada.

**WR/.** Orden de Escritura. Su activación indica que en las líneas de dirección se dispone de una dirección válida, y que se desea escribir en la localidad accesada, la información (byte o palabra) disponible en las líneas de datos en el momento en que **WR/** sea desactivada.

**WAIT/.** Señal de Espera. Sirve para prolongar la duración de una transferencia. Es activada por los esclavos direccionados con el fin de satisfacer sus tiempos de acceso y temporizaciones internas.

**ALE/.** Habilitador del Retenedor (Latch) de Direcciones. Esta señal es activada para indicar que en las líneas de datos (DATO\* a DAT7\*), se encuentran las direcciones más bajas (DIRO\* a DIR7\*), para un acceso mediante PASP/.

**ERR.** Señal de Error. Se utiliza en una operación de transferencia de información para indicar que el esclavo direccionado no se encuentra o para indicar que hubo error en los datos recibidos. Esta línea normalmente permanece activada, y en el caso de que la transferencia se haya efectuado sin errores, debe ser desactivada por el esclavo direccionado. El estado de esta línea sólo es significativo durante el tiempo de la transferencia por lo que debe ser memorizado por el maestro involucrado en ella para su consideración posterior. ERR tiene mayor prioridad que cualquiera de las interrupciones.

**ALPA/.** Alarma de la Fuente de Alimentación. Esta señal es utilizada por la fuente de alimentación para indicar que se ha detectado alguna condición anormal y que se dispone de un mínimo de tiempo antes de que los valores de los voltajes de alimentación caigan por debajo de los límites aceptados.

**BCLK.** Reloj del canal. El frente de bajada de BCLK es empleado para sincronizar los circuitos de asignación del canal. BCLK no requiere estar sincronizado con los relojes empleados por los maestros. Si hay más de un Maestro en el sistema, uno y sólo uno, deberá estar encargado de generar la señal de BCLK. Esta señal se encuentra disponible para cualquier módulo que la requiera. La frecuencia a la cual debe ser generada esta señal es de 9.216 Mhz.

**BUSY/.** Línea que indica que el canal está ocupado. Se activa por el maestro que se adueña del canal e indica la posesión de éste. Ningún otro podrá hacer uso del canal mientras no se desactive esta línea.

### Líneas de Interrupción

**INT0/...INT2/.** Líneas de Interrupción. Son tres líneas de interrupciones paralelas. En cada línea se puede conectar más de un esclavo.

Líneas de Alimentación

+5v Línea de alimentación regulada de 5 volts positivos. Está destinada a los circuitos digitales.

GND Tierra. Retorno para los circuitos digitales.

+VD Línea de alimentación regulada de 18 volts positivos.

-VD Línea de alimentación regulada de 18 volts negativos.

AGND Tierra Analógica. Retorno para los circuito analógicos.

+VS Línea de alimentación regulada de 24 volts positivos.

-VS Línea de retorno para VS.

Línea Reservada

Canal serie. Este ha sido el nombre que se le ha dado a la línea 68, considerada anteriormente como reservada. Su uso tiene el propósito específico de efectuar transferencias de información entre dos módulos en el trasplano mediante UART's conectados a esta línea. Lo anterior es con el fin de no saturar el canal.

## 1.2.8 Características de las señales del canal.

Las características para el grupo de líneas que componen IBUS-III son: el NIVEL para el cual son activas, SENTIDO en el cual se dirigen dentro del canal, ORIGEN de éstas y DESTINO al que son enviadas. A continuación se muestra en la tabla toda la información referente a estas características.

NOMBRE	NIVEL	ORIGEN	SENTIDO	DESTINO
INIC/	B	M, M	<----->	E, M
DIRO*-DIR19*	N	M	----->	E
BHEN/	B	M	----->	E
PASP/	B	M	----->	E
PGEN/	B	M	----->	E
SPO-SP3	P	Canal	----->	E
DATO*-DAT15*	N	M, E	<----->	E, M
RD/	B	M	----->	E
WR/	B	M	----->	E
WAIT/	B	E	----->	M
ALE/	B	M	----->	E
INTO/-INT2/	B	E	----->	M
ERR	A	E	----->	M
ALFA/	B	M, E	----->	E, M
BCLK	-	M, M	----->	E, M
BUSY/	M	M	<----->	M

TABLA 1.1 Características de las señales de IBUS-III.

## NOTAS:

A - Activo Alto

B - Activo Bajo

P - Señal Directa

N - Señal Complementada

-----&gt; Unidireccional

&lt;-----&gt; Bidireccional

M - Maestro

E - Esclavo

Cabe hacer algunas aclaraciones para la tabla anterior. En el caso de que el origen contenga más de una opción, ese mismo número de opciones deberá tener el destino. En este caso, el flujo de información será de la primera opción del origen a la correspondiente del destino.

## 1.2.9 Distribución topológica de las líneas en IBUS-III.

En seguida se muestra la distribución topológica de las líneas que componen IBUS-III. Esta tabla contiene el nombre de la señal e información referente a la asignación de cada una de éstas a las posiciones fijas del canal y el arreglo resistivo correspondiente para la señal en cada caso que sea necesario.

MNEMONICO	TERM RESIST	POSICIÓN	MNEMONICO	TERM RESIST	POSICIÓN
+5V	NO	79	+5V	NO	80
GND	NO	77	GND	NO	78
+VD	NO	75	+VD	NO	76
-VD	NO	73	-VD	NO	74
AGND	NO	71	AGND	NO	72
RESERVADA	SI	69	ALFA/	SI	70
INT2/	SI	67	CANAL SERIE	SI	68
INT1/	SI	65	RESERVADA	SI	66
INT0/	SI	63	ERR	SI	64
WAIT/	SI	61	ALE/	SI	62
PASP/	SI	59	DIR15*	SI	60
SP3	NO	57	DIR14*	SI	58
SP2	NO	55	DIR13*	SI	56
SP1	NO	53	DIR12*	SI	54
SPO	NO	51	DIR11*	SI	52
DIR7*	SI	49	DIR10*	SI	50
DIR6*	SI	47	DIR9*	SI	48
DIR5*	SI	45	DIR8*	SI	46
DIR4*	SI	43	DIR19*	SI	44
DIR3*	SI	41	DIR18*	SI	42
DIR2*	SI	39	DIR17*	SI	40
DIR1*	SI	37	DIR16*	SI	38
DIRO*	SI	35	BHEN/	SI	36
PGEN/	SI	33	RESERVADA	SI	34
RD/	SI	31	WR/	SI	32
DAT7*	SI	29	DAT15*	SI	30
DAT6*	SI	27	DAT14*	SI	28
DAT5*	SI	25	DAT13*	SI	26
DAT4*	SI	23	DAT12*	SI	24
DAT3*	SI	21	DAT11*	SI	22
DAT2*	SI	19	DAT10*	SI	20
DAT1*	SI	17	DAT9*	SI	18
DATO*	SI	15	DAT8*	SI	16
BCLK	SI	9	BUSY/	SI	14
-VS	NO	7	INIC/	SI	10
+VS	NO	5	-VS	NO	8
GND	NO	3	+VS	NO	6
+5V	NO	1	GND	NO	4
			+5V	NO	2

TABLA 1.2 Disposición de las señales de IBUS-III.

## 1.2.10 Características de los tipos de direccionamiento.

Los tipos de direccionamiento en IBUS-III, permiten acceder localidades de manera geográfica en 16 espacios disponibles, con 2 Kbytes de memoria (debido al número de líneas utilizadas en la decodificación) o de memoria en páginas generales con una capacidad de 1 Mbyte.

Las 20 líneas de dirección son utilizadas en la decodificación por PGEN/, por lo que no se requiere de nota alguna para su uso. En el caso de la decodificación de memoria geográfica (mediante PASP/) se muestra a continuación un esquema donde se marcan las líneas utilizables en este tipo de direccionamiento, junto con una tabla donde se muestran las combinaciones posibles para los tipos de accesos.

DIR19..DIR16    DIR15..DIR12    DIR11..DIR8    DIR7..DIR4    DIR3..DIR0  
 XXXX            XNNN            Nddd            dddd            dddd

PASP/	PGEN/	ACCIÓN
0	0	Inválido
0	1	Acceso a Páginas por Posición
1	0	Acceso a Páginas Generales
1	1	No hay acceso a IBUS-III

TABLA 1.3 Tipos de direccionamiento.

NNN N - Número de módulo seleccionado  
 d dddd dddd - Dirección dentro del módulo  
 XXXX X - Valor indiferente

En el caso de la selección del módulo, el número binario que se genere con estas líneas, indicará el slot o el espacio a escoger. La siguiente tabla muestra el módulo seleccionado dada la combinación de las líneas NNNN (equivalentes a las líneas de dirección DIR14\* a DIR11\*).

DIR14	DIR13	DIR12	DIR11	MODULO SELECCIONADO	RANGO DIRECCIONADO
0	0	0	0	0	08000H-087FFH
0	0	0	1	1	08800H-08FFFH
0	0	1	0	2	09000H-097FFH
0	0	1	1	3	09800H-09FFFH
0	1	0	0	4	0A000H-0A7FFH
0	1	0	1	5	0A800H-0AFFFH
0	1	1	0	6	0B000H-0B7FFH
0	1	1	1	7	0B800H-0BFFFH
1	0	0	0	8	0C000H-0C7FFH
1	0	0	1	9	0C800H-0CFFFH
1	0	1	0	A	0D000H-0D7FFH
1	0	1	1	B	0D800H-0DFFFH
1	1	0	0	C	0E000H-0E7FFH
1	1	0	1	D	0E800H-0EFFFH
1	1	1	0	E	0F000H-0F7FFH
1	1	1	1	F	0F800H-0FFFFH

TABLA 1.4 Direccionamiento por PASP

## 1.2.11 Empleo de los dispositivos de 8 Y 16 bits en IBUS-III

Ya que es posible combinar dispositivos de 8 y 16 bits, se requiere compatibilidad en la comunicación de los módulos; para esto se muestran las características que deben cumplir tanto los maestros como los esclavos de 8 y 16 bits.

1. Maestro de 8 bits: debe manejar la línea de BHEN/ de manera que las líneas BHEN/ y DIR0\* sean iguales, es decir BHEN/ = DIR0\*.
2. Esclavo de 8 bits: debe ignorar la línea BHEN/.
3. Maestro de 16 bits y Esclavo de 16 bits: además de colocar los transceptores (transceivers) para comunicar la parte alta y baja de las líneas de datos del módulo usuario con la del sistema (DAT0\* a DAT7\* y DAT8\* a DAT15\* respectivamente), se necesita agregar un transceiver para conectar la parte alta de las líneas de datos del módulo con las líneas DAT0\* a DAT7\* del sistema.

### 1.2.12 Manejo de las líneas de interrupción.

La norma IBUS-III maneja el tipo de interrupciones no vectorizadas, es decir, que la rutina de servicio que el maestro deberá proporcionar, depende de la línea de interrupción activada por el esclavo solicitante. Para esto, existen 3 líneas de interrupción paralelas.

La estructura dada en IBUS-III permite que más de un esclavo se encuentre conectado a la misma línea de interrupción, lo que da la oportunidad de que varios esclavos puedan interrumpir al mismo maestro y además que el mismo esclavo pueda interrumpir a diferentes maestros. En este caso, el protocolo para interrumpir al maestro y que éste conteste a la petición, es el que sigue:

- El esclavo activa una de las líneas de interrupción, que se encuentra atendida por un maestro en específico.

- El maestro deberá buscar en qué espacio se encuentra el esclavo que solicita la rutina de servicio y una vez encontrado responderá al esclavo con un comando de lectura o escritura (que debe ir acompañado, por la naturaleza de la transferencia, de una dirección por PASP/ con las líneas correspondientes a la dirección geográfica del esclavo que solicita el servicio), con lo que éste deberá desactivar la línea de interrupción anteriormente utilizada. Con esto entra en acción la rutina de servicio.

## 2. Canal de interconexión SCSI

### 2.1 Introducción

Una de las interfases paralelas más complejas para ser desarrollada es la Small Computer System Interface (SCSI), que está basada en una interfaz para manejador de disco desarrollada por Shugart Associates, conocida como Shugart Associates System Interface (SASI). En la mayoría de los sistemas compuestos por un controlador de disco y una computadora, la tarjeta controladora de disco (llamada controlador puente) es la que establece la secuencia de la transferencia de datos sobre el canal SCSI, no la computadora.

#### Características Físicas

**Descripción física.** Los dispositivos SCSI están entrelazados mediante el sistema daisy-chain, usando un cable común. Todas las señales son comunes entre todos los dispositivos SCSI. Dos alternativas manejador(driver)/receptor están disponibles:

1) Drivers y receptores single-ended, que permiten una longitud de cable máxima de 6 m. (para conexión dentro de gabinete).

2) Drivers y receptores diferenciales, los cuales permiten una longitud de cable máxima de 25 m. (para conexiones fuera de gabinete).

En la descripción de las características del canal SCSI que se presenta a continuación, sólo se mencionarán aquellas que se refieran a la opción single-ended.

#### Requerimientos de cables

Se debe usar un tamaño mínimo de conductor de 28 AWG para minimizar los efectos de ruido y asegurar la distribución apropiada de la terminal power (opcional).

**Cable single-ended.** Se debe usar un cable plano de 50 hilos o un cable trenzado en pares de 25 señales. La longitud máxima del cable es de 6.0 m.

**Requerimientos de conector.** Se especifican conectores no blindados. Los conectores no blindados son típicamente usados para aplicaciones en gabinetes.

Los conectores no blindados del cable ( véase la figura 2.1) deben ser conectores de 50 conductores consistentes en 2 hileras de 25 terminales hembra con terminales adyacentes separados 2.54 mm. (0.1 in).

Los conectores no blindados de los dispositivos (véase la figura 2.2) deben ser conectores de 50 conductores, consistentes en 2 hileras de 25 terminales macho con terminales adyacentes separados 2.54 mm. (0.1 in). Se recomienda usar conectores marcados.

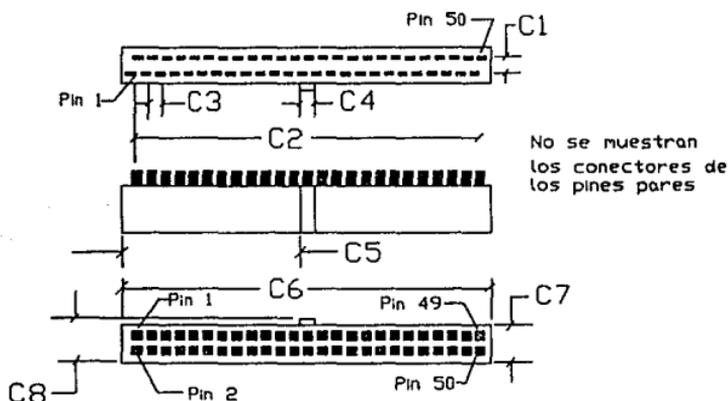


Figura 2.1 Conector no blindado hembra.

Dimensión	Milímetros	Pulgadas
C1	2.540	0.100
C2	60.960	2.400
C3	2.540	0.100
C4	3.302	0.130
C5	32.385	1.275
C6	68.072	2.680
C7	6.096	0.240
C8	7.620	0.300

Notas:

- 1) Cincuenta contactos a espacios de 1.27 mm (0.05 in) = 62.23 mm (2.450 in).
- 2) Las tolerancias son de  $\pm 0.127$  mm (0.005 in) no acumulativas.
- 3) Es opcional que el conector esté cubierto.

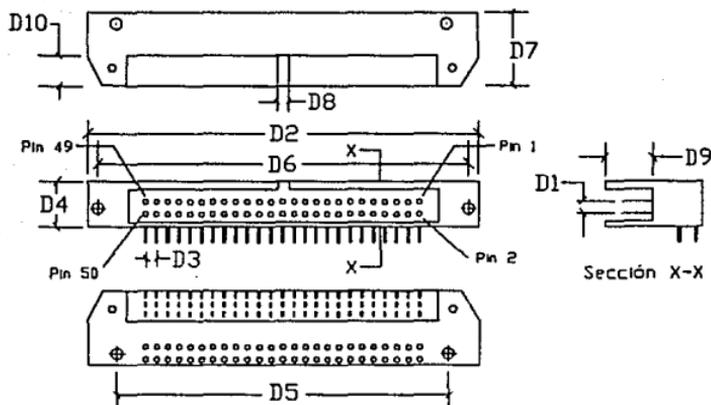


Figura 2.2 Conector no blindado macho

Dimensión	Milímetros	Pulgadas
D1	2.54	0.100
D2	82.80	3.260
D3	2.54	0.100
D4	8.89	0.350
D5	72.64	2.860
D6	78.74	3.100
D7	13.94	0.549
D8	$4.19 \pm 0.25$	$0.165 \pm 0.01$
D9	6.09	0.240
D10	6.60	0.260

Notas:

- 1) Dos hileras de 25 contactos espaciados 2.54 mm (0.100 in) = 60.96 mm (2.400 in).
- 2) Las tolerancias son de  $\pm 0.127$  mm (0.005 in) no acumulativas.

La asignación de las terminales del conector no blindado debe ser como muestra la tabla 2-1 para drivers single-ended.

Señal	número de terminal
-DB(0)	2
-DB(1)	4
-DB(2)	6
-DB(3)	8
-DB(4)	10
-DB(5)	12
-DB(6)	14
-DB(7)	16
-DB(P)	18
TIERRA	20
TIERRA	22
TIERRA	24
TERMPWR	26
TIERRA	28
TIERRA	30
-ATN	32
TIERRA	34
-BSY	36
-ACK	38
-RST	40
-MSG	42
-SEL	44
-C/D	46
-REQ	48
-I/O	50

TABLA 2-1  
Asignación de terminales single-ended

NOTAS:

- 1) Todas las terminales menos excepto la 25 deben estar conectados a tierra. La terminal 25 se debe dejar abierta. Algunos productos diseñados antes de la elaboración de este estándar conectan esta terminal a tierra.
- 2) El signo menos al lado de la señal indica activo bajo.

Alternativa de terminación sencilla (single ended). Todas las señales asignadas deben terminar con 220 ohms a +5 volts (nominal) y 330 ohms a tierra en cada extremo del cable (Tipo Thevenin, véase la figura 2.3). Todas las señales deben usar drivers colector abierto o de tres estados.

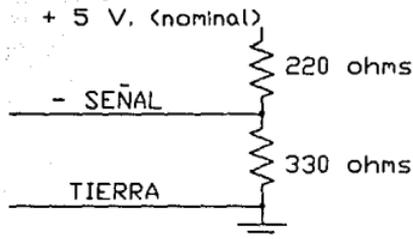


Figura 2.3 Arreglo Thevenin para todas las terminales.

**Características de salida.** Cada señal manejada por un dispositivo SCSI debe tener las siguientes características de salida, cuando se mide el conector del dispositivo SCSI:

- Activación de señal = 0.0 volts dc a 0.4 volts dc
- Capacidad mínima de salida del driver = 48 miliamp (sinking)  
a 0.5 volts dc
- Negación de señal = 2.5 volts dc a 5.25 volts dc

**Características de entrada.** Cada señal recibida por un dispositivo SCSI debe tener las siguientes características de entrada cuando se mide el conector del dispositivo SCSI:

- Señal verdadera = 0.0 volts dc a 0.8 volts dc
- Carga de entrada máxima total = -0.4 miliamps. a 0.4 volts dc
- Señal falsa = 2.0 volts dc a 5.25 volts dc
- Mínima histéresis de entrada = 0.2 volts dc

**Terminación Power (opcional).** Los dispositivos SCSI terminación sencilla que cuentan con la terminal de poder (power termination o TERMPWR) deben tener las siguientes características:

- $V_{term} = 4.0$  volts dc a 5.25 volts dc
- 800 miliamps. mínimos de capacidad drive fuente
- 1.0 miliamp. máxima capacidad sink (excepto para el propósito de proveer poder a una terminal interna) con limitador de corriente recomendado de 1.0 amp. (pe. un fusible).

**Nota:** El diseño de esta tarjeta no cuenta con la opción de terminal de poder.

Canal SCSI. La comunicación en el canal SCSI es permitida sólo entre dos dispositivos en cualquier tiempo dado, habiendo un máximo de 8 dispositivos. Cada dispositivo SCSI tiene un bit ID (identificador SCSI) asignado como muestra la figura 2-4.

Cuando dos dispositivos SCSI se comunican en el canal, uno actúa como iniciador y el otro como objetivo. El iniciador origina una operación y el objetivo la lleva a cabo. Un dispositivo SCSI, usualmente tiene un papel fijo ya sea como iniciador o como objetivo, pero algunos dispositivos pueden asumir uno u otro papel.

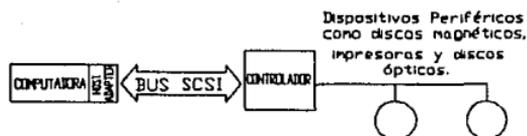
Un iniciador puede direccionar hasta 8 dispositivos periféricos que están conectados a un objetivo. Una opción, usando mensajes extendidos, permite hasta 2,048 dispositivos periféricos por objetivo. Tres ejemplos de configuración de sistema se muestran en la figura 2-5.

DB(7)	DB(6)	DB(5)	DB(4)	DB(3)	DB(2)	DB(1)	DB(0)	<-Canal de DATOS
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	SCSI ID = 0
:	:	:	:	:	:	:	:	SCSI ID = 1
:	:	:	:	:	:	:	:	SCSI ID = 2
:	:	:	:	:	:	:	:	SCSI ID = 3
:	:	:	:	:	:	:	:	SCSI ID = 4
:	:	:	:	:	:	:	:	SCSI ID = 5
:	:	:	:	:	:	:	:	SCSI ID = 6
:	:	:	:	:	:	:	:	SCSI ID = 7

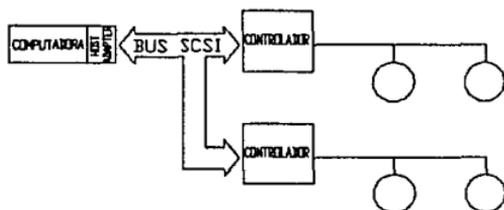
Figura 2.4 Bits SCSI ID.  
(Bits identificadores SCSI)

Hasta 8 dispositivos SCSI pueden ser soportados en el canal SCSI, pudiendo hacer cualquier combinación de iniciadores y objetivos.

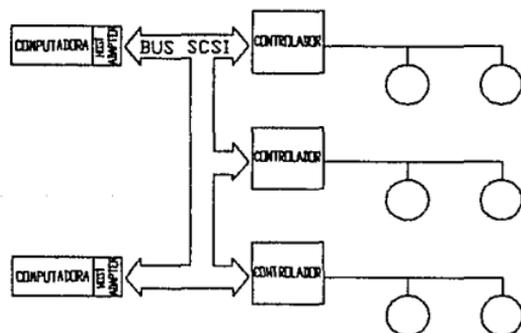
Ciertas funciones del canal SCSI están asignadas al iniciador y otras al objetivo. El iniciador puede arbitrar por el canal SCSI y seleccionar un objetivo en particular. El objetivo puede solicitar la transferencia de comandos, datos, estados (status) u otra información en el canal de DATOS; en algunos casos, puede arbitrar por el canal SCSI y reseleccionar un iniciador con el propósito de continuar una operación.



UN INICIADOR, UN OBJETIVO



UN INICIADOR, MULTIPLES OBJETIVOS



MULTIPLES INICIADORES, MULTIPLES OBJETIVOS

Figura 2.5 Ejemplos de configuración de sistemas.

La transferencia de información en el canal de DATOS es asíncrona y sigue un protocolo handshake REQ/ACK (saludo REQ/ACK) definido. Un byte de información es transferido con cada handshake. Se define una opción para transferencia síncrona de datos.

Señales del canal SCSI. Existen de 18 señales en total. 9 de ellas se usan para control y las 9 restantes se usan para datos. (Las señales de datos incluyen la opción de señal de paridad). Estas señales están descritas como sigue:

BSY (BUSY - Ocupado). Una señal OR-tied que indica que el canal está siendo usado.

SEL (SELECT - Selección). Una señal usada por el iniciador para seleccionar un objetivo o por el objetivo para reelegir un iniciador.

C/D (CONTROL/DATA - control/dato). Una señal manejada por el objetivo que indica si en el canal de DATOS hay datos o información de control. Verdadero indica control.

I/O (INPUT/OUTPUT - entrada/salida). Una señal manejada por el objetivo que controla la dirección del movimiento de datos en el canal de DATOS con respecto al iniciador. Verdadero indica entrada al iniciador. Esta señal también es usada para distinguir entre las fases de SELECCION y RESELECCION.

MSG (MESSAGE - mensaje). Una señal manejada por el objetivo durante la fase de MENSAJES.

REQ (REQUEST - solicitud). Una señal manejada por el objetivo para indicar una solicitud de un handshake REQ/ACK en la transferencia de un dato.

ACK (ACKNOWLEDGE - reconocimiento). Una señal manejada por un iniciador para indicar un reconocimiento en un handshake REQ/ACK en la transferencia de un dato.

ATN (ATTENTION - atención). Una señal manejada por un iniciador para indicar condición de ATENCION.

RST (RESET - reestablecimiento o inicilaización) Una señal OR-tied que indica la condición RESET.

DB(7-0,P) (DATA BUS - canal de datos). Ocho señales bit-dato más una señal bit-paridad que forman el canal de DATOS. DB(7) es el bit más significativo y tiene la mayor prioridad durante la fase de ARBITRAJE. El significado y la prioridad del número de bit decrecen hacia DB(0). Un bit dato, es definido como "1" cuando el valor de la señal es verdadero y es definido como "0" cuando el valor de la señal es falsa.

El Dato paridad DB(P) es impar. El uso de paridad es una opción del sistema (p. ej. un sistema está configurado para que todos los dispositivos SCSI en el canal generen paridad y tienen detección de paridad habilitada, o todos los dispositivos SCSI tienen deshabilitada o no utilizan la detección de paridad). La paridad no es válida durante la fase de ARBITRAJE.

Valores de las señales. Las señales pueden asumir valores de falso o verdadero. Hay dos métodos de manejar estas señales; en ambos casos, la señal debe ser activada (asserted). En el caso de drivers "OR-tied", el driver no conduce la señal al estado falso, es el circuito de polarización (bias) de la terminal del canal el cual mantiene la señal en falso siempre que es liberada por los drivers en todos los dispositivos SCSI. Si cualquier driver es activado, entonces la señal es verdadera. En el caso de drivers no-OR-tied, las señales pueden ser activamente llevadas a falso o negadas. En este estándar, donde quiera que el término negado sea usado, significa que la señal puede ser falsa o puede ser simplemente soltada (en cuyo caso los circuitos de polarización la mantiene en estado falso). La ventaja de manejar activamente la señal a falso, es que las transiciones de verdadero a falso ocurren más rápido y los márgenes de ruido se reducen; esto puede permitir transferencias de datos un poco más rápida.

Señales OR-tied. Las señales BSY y RST deben ser OR-Tied. En la operación ordinaria del canal, estas señales son manejadas a verdadero simultáneamente por varios drivers. Sólo las señales BSY, RST y DB(P) son simultáneamente manejadas por 2 o más drivers y cualquier otra señal excepto BSY y RST pueden emplear drivers OR-tied o no-OR-tied. DB(P) no debe ser llevado a falso durante la fase de ARBITRAJE. No hay problema de combinar drivers OR-tied y no-OR-tied en otras señales que BSY y RST.

**Fuentes de señales.** La tabla 2-2 indica que tipo de dispositivo SCSI puede generar cada señal. Todos los drivers de SCSI que no son fuentes activas deben estar en estado pasivo. Nótese que la señal RST puede ser generada por cualquier dispositivo SCSI en cualquier momento.

En la tabla 2-2 se muestra que dispositivo maneja las señales para cada fase.

Fase del Canal	señales				
	BSY	SEL	C/D, I/O MSG, REQ	ACK/ATN	DB(7-0, P)
Canal LIBRE	ninguno	ninguno	ninguno	ninguno	ninguno
ARBITRAJE	todos	ganador	ninguno	ninguno	SCSI ID
SELECCIÓN	I&O	inic.	ninguno	inic.	inic.
RESELECCIÓN	I&O	objet.	objet.	inic.	objet.
COMANDO	objet.	ninguno	objet.	inic.	inic.
DATOS ENTRADA	objet.	ninguno	objet.	inic.	objet.
DATOS SALIDA	objet.	ninguno	objet.	inic.	inic.
ESTADOS	objet.	ninguno	objet.	inic.	objet.
MENSAJE ENTRADA	objet.	ninguno	objet.	inic.	objet.
MENSAJE SALIDA	objet.	ninguno	objet.	inic.	inic.

TABLA 2-2

- todos:** Esta señal puede ser manejada por todos los dispositivos SCSI que están arbitrando activamente.
- SCSI ID:** Un bit dato único (el SCSI ID o identificador SCSI) debe ser manejado por cada dispositivo SCSI que está arbitrando activamente; los otros siete bits de datos deben ser soltados (no activándolos ni desactivándolos) por este dispositivo SCSI. El bit de paridad ( DB(P) ) puede ser manejado o no al estado verdadero, pero nunca debe ser llevado al estado falso durante esta fase.
- I&O:** La señal debe ser manejada por el iniciador, el objetivo o ambos, como se especifica en la fase de SELECCION y en la fase de RESELECCION.
- inic.:** Si esta señal es manejada, debe ser manejada sólo por el iniciador activo.

- ninguno: La señal debe ser soltada; esto es, no ser manejada por ningún dispositivo SCSI. La circuitería de polarización de las terminales del canal mantienen la señal en estado falso.
- ganador: La señal debe ser manejada por el dispositivo que ganó el arbitraje.
- objet.: Si la señal es manejada, debe ser manejada sólo por el objetivo activo.

### Características lógicas

Fases del canal SCSI. La arquitectura incluye ocho fases diferentes:

Fase BUS FREE (Canal libre)  
 Fase ARBITRATION (Arbitraje)  
 Fase SELECTION (Selección)  
 Fase RESELECTION (Reselección)  
 Fase COMMAND (Comandos) -----Estas fases son  
 Fase DATA (Datos) -----llamadas colectivamente  
 Fase STATUS (Estados) -----fases de transferencia  
 Fase MESSAGE (Mensajes)-----de datos.

El canal SCSI no puede estar en más de una fase a un mismo tiempo dado. A menos que se especifique en notas en las siguientes descripciones, las señales que no se mencionen no deberán ser activadas.

Fase Canal LIBRE. La fase canal LIBRE es usada para indicar que ningún dispositivo SCSI está activamente usando el canal SCSI y que está disponible para usuarios subsecuentes.

Fase de ARBITRAJE. La fase de ARBITRAJE permite a un dispositivo SCSI ganar el control del canal SCSI y poder asumir el papel de iniciador o de objetivo.

NOTA: La utilización de la fase de ARBITRAJE es una opción de sistema. Los sistemas que no utilizan esta opción, sólo podrán tener un iniciador. La fase de ARBITRAJE es requerida para sistemas que usan la fase RESELECCIÓN.

Fase de SELECCION. La fase de SELECCION permite al iniciador seleccionar un objetivo para el propósito de iniciar alguna función (por ejemplo, los comandos LEER o ESCRIBIR).

Fase de RESELECCION. La RESELECCION es una fase opcional que permite al objetivo reconectarse con el iniciador con el fin de continuar alguna operación que fue previamente comenzada por el iniciador pero que fue suspendida por el objetivo.

## Fases de transferencia de información

NOTA: Las fases COMMAND, DATA, STATUS y MESSAGE están agrupadas como fases de transferencia de información debido a que todas son usadas para transferencia de datos o de información de control a través del canal de DATOS.

Transferencia de información asíncrona. El objetivo debe controlar la dirección de la transferencia de información con significados de la señal I/O. Cuando I/O es verdadera, la información debe ser transferida del objetivo al iniciador. Cuando I/O es falso, la información debe ser transferida del iniciador al objetivo.

Transferencia de datos síncrona (opcional). Este modo de transferencia es opcional y podrá ser usado sólo en la fase de DATOS si previamente se ponen de acuerdo el iniciador y el objetivo, a través de un sistema de mensajes. El mensaje determina el uso de modo síncrono para los dos dispositivos y establece el desplazamiento (offset) y el período de transferencia REQ/ACK.

Fase COMMAND (fase de comandos). Esta fase permite al objetivo solicitar información de comandos del iniciador.

Fase de Datos. La fase de datos es un término que abarca las fases DATA IN y DATA OUT.

Fase DATA IN. (ENTRADA DE DATOS) Esta fase permite al objetivo solicitar que se envíen datos hacia el iniciador.

Fase DATA OUT. (SALIDA DE DATOS) Esta fase permite al objetivo solicitar que se envíen datos del iniciador hacia el objetivo.

Fase STATUS (fase de estados). Esta fase permite al objetivo solicitar la información de estados (status) que será enviada del objetivo al iniciador.

Fase MESSAGE (MENSAJES). La fase de mensajes es un término que hace referencia a las fases MESSAGE IN y MESSAGE OUT.

MESSAGE IN (ENTRADA DE MENSAJE): Permite al objetivo solicitar que los mensajes sean enviados del objetivo al iniciador.

MESSAGE OUT (SALIDA DE MENSAJE): Permite al objetivo solicitar que los mensajes sean enviados al objetivo desde el iniciador. El objetivo puede invocar esta fase a su conveniencia en respuesta a la condición ATTENTION (Atención) creada por el iniciador.

## Condiciones del canal

El canal SCSI tiene dos condiciones asíncronas, la condición ATENCION y la condición RESET. Estas condiciones causan que los dispositivos presenten ciertas acciones y puedan alterar la secuencia de fases.

### Condición ATENCION

Permite al iniciador informar al objetivo que tiene un mensaje listo. El objetivo puede obtener ese mensaje a su conveniencia cambiando a la fase de SALIDA DE MENSAJE.

### Condición RESET

Es usada para limpiar inmediatamente todos los dispositivos SCSI del canal. Esta condición debe tener prioridad sobre todas las otras fases y condiciones. Cualquier dispositivo puede crear la condición RESET activando RST por un mínimo de 25 us. Durante la condición de RESET el estado de las señales del canal es indefinido, excepto la de RST.

#### RESET por hardware

Al detectar el RESET deben:

- 1) Limpiar todos los comandos incompletos.
- 2) Liberar todas las reservaciones de dispositivos SCSI.
- 3) Regresar cualquier modo de operación del dispositivo a sus condiciones de default.

#### RESET por software.

Al detectar el RESET deben:

- 1) Intentar completar cualquier comando incompleto que esté plenamente identificado.
- 2) Mantener todas las reservaciones de los dispositivos SCSI.
- 3) Mantener cualquier modo de operación del dispositivo.

El RESET por software permite a un iniciador inicializar el canal sin disturbios a la operación de otros iniciadores en sistemas de múltiples iniciadores.

### Secuencia de fases del canal

El orden en que son usadas las fases en el canal SCSI sigue una secuencia preescrita.

En todos los sistemas, la condición RESET puede cancelar cualquier fase y siempre le sigue una fase canal LIBRE. Además, a cualquier otra fase le puede seguir a la fase canal LIBRE.

### Sistemas no arbitrados

La secuencia normal es de canal LIBRE a SELECCION y de SELECCION a una o más de las fases de transferencia de información.

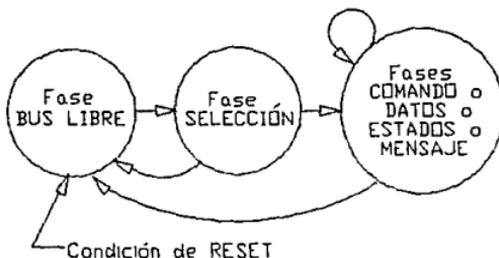


Figura 2.6 Fases del canal en sistemas no arbitrados.

## Sistemas arbitrados

La secuencia normal es de canal LIBRE a ARBITRAJE, de ARBITRAJE a SELECCION o RESELECCION y de SELECCION o RESELECCION a una o más fases de transferencia de información.

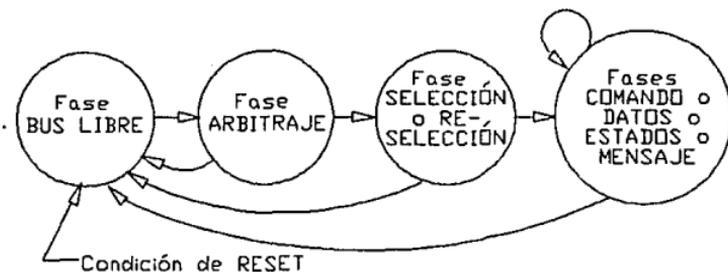


Figura 2.7 Fases del canal en sistemas arbitrados.

Todos los sistemas

No hay restricción en la secuencia entre fases de transferencia de información. Un tipo de fase puede ser seguida por el mismo tipo de fase.

## Apuntador SCSI

La arquitectura SCSI provee dos conjuntos (sets) de 3 apuntadores dentro de cada iniciador. Los apuntadores residen en la trayectoria de control del iniciador. El primer set de apuntadores es conocido como el de los apuntadores activos (o actuales). Estos apuntadores son usados para representar el estado de la interfaz y apuntar al siguiente comando, dato o byte de estado para ser transferido entre la memoria del iniciador y el objetivo. Sólo hay un set de apuntadores activos en cada iniciador. Estos son usados por el objetivo actualmente conectado con el iniciador.

El segundo set de apuntadores es conocido como apuntadores salvados. Hay un set de apuntadores salvados por cada comando que está actualmente activo (esté o no actualmente conectado). El apuntador del comando almacenado siempre apunta al inicio del bloque descriptor de comando (command descriptor block) para el comando activo. El apuntador del status almacenado siempre apunta al inicio del área status del comando activo. Al principio de cada comando, el apuntador de datos almacenado apunta al inicio del área de datos, permanece en este valor hasta que el objetivo envía un mensaje de SAVE DATA POINTER al iniciador. En respuesta a este mensaje, el iniciador guarda el valor del apuntador actual en el apuntador de datos almacenado (SAVED DATA POINTER). El objetivo puede recuperar el valor almacenado en cada apuntador enviando al iniciador un mensaje RESTORE POINTERS (reestablece apuntadores). El iniciador mueve los valores grabados de cada apuntador al correspondiente apuntador activo. Cuando un dispositivo SCSI se desconecta del canal, sólo los valores salvados del apuntador son retenidos. El valor del apuntador actual es recuperado de los valores almacenados hasta la siguiente reconexión.

#### Especificaciones del sistema de mensajes

El sistema de mensajes permite comunicación entre el iniciador y el objetivo para el manejo de trayectorias físicas.

#### Protocolo de Mensajes

Todos los dispositivos SCSI deben utilizar el mensaje COMANDO COMPLETO (COMMAND COMPLETE). Un dispositivo SCSI puede ser construido sin usar ningún otro mensaje, si el número de unidad lógica es especificado en el bloque descriptor de comando.

Activando la señal ATN, los dispositivos SCSI indican su habilidad de acomodar otros mensajes además del COMMAND COMPLETE. El iniciador indica esto en la fase de selección, activando ATN antes que en el canal se presente la condición de que SEL es verdadera y BSY es falsa. El objetivo indica su habilidad de acomodar más mensajes respondiendo a la condición ATENCION con la fase SALIDA DE MENSAJE, después de ir a través de la fase de SELECCION.

Para dispositivos SCSI que soportan otros mensajes además del COMMAND COMPLETE, el primer mensaje enviado por el iniciador después de la fase SELECCION debe ser el mensaje IDENTIFY (identificador). Esto permite el establecimiento de la trayectoria física para una unidad lógica particular especificada por el iniciador. Después de la fase RESELECCION, el primer mensaje del objetivo debe ser IDENTIFY. Esto permite a la trayectoria física ser reestablecida por el número de unidad lógica especificada por el objetivo. Bajo ciertas condiciones excepcionales, el iniciador puede enviar el mensaje de ABORT o el BUS DEVICE RESET en lugar del mensaje IDENTIFY como primer mensaje. Sólo el número de unidad lógica debe ser identificado por cualquier secuencia de selección, un segundo mensaje IDENTIFY con un nuevo número de unidad lógica no debe ser emitido antes de que el canal SCSI haya sido soltado (fase canal LIBRE).

Siempre que una trayectoria física se establezca en un iniciador que puede realizar desconexión y reconexión, el iniciador debe asegurarse de que los apuntadores activos de la trayectoria física sean iguales a los apuntadores grabados para esa unidad lógica en particular.

Los dispositivos SCSI que utilizan otros mensajes además de COMMAND COMPLETE, deben utilizar el mensaje MESSAGE REJECT.

## 3. PROPUESTA Y ANALISIS DE LA POSIBLE SOLUCION

El objetivo es realizar la interfaz entre el canal SCSI y el canal IBUS-III. Esto se reduce a un traductor para transferencias de información como lo muestra el diagrama a bloques de la figura 3.1.

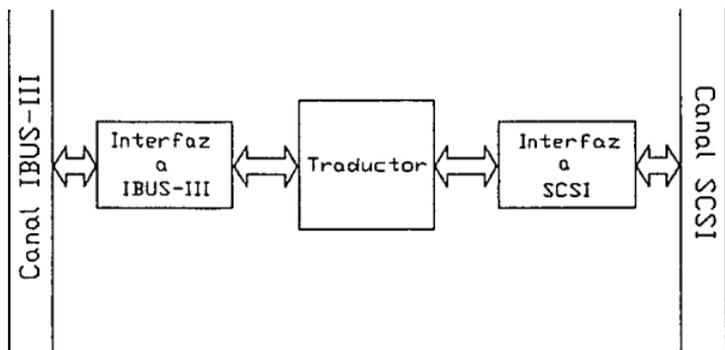


Figura 3.1 Diagrama de definición de objetivo

Dentro del sistema SAC, sólo se tienen dos tipos de tarjetas que son: tarjetas procesadoras o maestras, las cuales controlan y ordenan las funciones que se deberán realizar y en cada sistema sólo puede haber una procesadora (SAC 1887); tarjetas esclavas, las cuales realizan las operaciones que la tarjeta maestra indica, ya sea de salida o entrada, analógica o digital, o de comunicación. Otra tarjeta considerada como esclava es la expansora de memoria, la cual difiere de las otras en que esta no procesa información, sólo la almacena; además, esta tarjeta no se localiza por posición geográfica dentro del gabinete (*rack*), sino por páginas generales, es decir que para su acceso es necesario decodificar las 20 líneas de direcciones, sin importar en que posición se encuentre, mientras que para las otras esclavas se decodifican las 15 menos significativas y dependen de la posición donde sean colocadas dentro del gabinete (véase capítulo 2).

Las tarjetas de comunicación diseñadas anteriormente, son tarjetas esclavas localizadas por posición geográfica. Esto no era ningún inconveniente, considerando que la comunicación se realiza a través de canales de comunicación serial, por lo que era posible enviar un byte, leer los registros de banderas y hasta que estos lo indicaran, enviar el siguiente byte. Este sistema resulta

demasiado lento para las velocidades de transferencias que se pretenden lograr con el canal SCSI, por lo que el diseño se basa en una comunicación paralela, con lo que logramos una velocidad 8 veces más rápida que la comunicación serial.

Para lograr esto, se consideró utilizar una memoria de doble puerto, con lo que se obtendrá un doble beneficio: por un lado logramos que el movimiento de información sea en paralelo y por el otro lado, la tarjeta será una expansora de memoria, con lo que los accesos serán más rápidos y no se necesitarán registros de identificación ni palabras de estados (status word).

Actualmente la tarjeta maestra no cuenta con un controlador de accesos directos a memoria (DMA), el cual permitiría que las transferencias fueran aún más rápidas, pero los diseños actuales contemplan esta posibilidad, además de no utilizar 8 bits en el canal de datos, sino 16, lo que duplicaría la velocidad de transferencia.

Las características principales de la memoria de doble puerto son:

- Posee dos puertos independientes llamados puerto derecho y puerto izquierdo. Cada puerto consiste de un canal de datos de 8 bits bidireccionales, un canal de entrada de direcciones de 10 bits y las señales de control necesarias.
- Tiene un árbitro incluido para resolver contención entre los dos puertos.
- Cuando ocurre una contención entre los puertos, un puerto recibe la prioridad mientras que el otro recibe una señal de busy (ocupado).
- Tiene facilidades incluidas en el circuito para soportar semáforos.
- Las direcciones 3FEh y 3FFh sirven como generadores de interrupción.
- Cuenta con dos habilitadores de circuitos, cada uno para el puerto correspondiente.
- Todas las entradas y salidas son compatibles TTL y el dispositivo opera con una alimentación de +5 volts.
- Tiempos de acceso de 55 ns.

Con esto queda definido el uso de una memoria doble puerto como interfaz con el sistema SAC, ahora consideremos la interfaz SCSI.

Desde los inicios de este estándar en 1986, la compañía NCR ha desarrollado numerosos circuitos controladores para este canal. En la actualidad el dispositivo más avanzado es el 53C700, que cuenta con un microprocesador que logra transferencias de hasta 80 Mbytes/s, sirviendo además como procesador central en un microcomputadora. Tal vez esta sería la opción más adecuada desde el punto de vista técnico.

Para nuestros fines, bastará con el controlador SCSI NCR 53C80, el cual tiene las siguientes características:

#### Interfaz al canal SCSI

- Interfaz asíncrona a 1.5 Mbytes/s
- Soporta el papel de objetivo y/o iniciador
- Revisión de paridad opcional
- Control directo sobre las señales del canal
- Conexión directa al canal gracias a sus drivers de gran capacidad de corriente

#### Interfaz con el microprocesador

- Interfaz a través de puertos o memoria
- Transferencias por DMA o por entrada/salida (I/O)
- Transferencias de DMA normales o por bloques
- Interrupciones al microprocesador opcionales

Se escogió este circuito ya que, además de las características descritas anteriormente, se tiene facilidad de obtención, bajo costo, información disponible y antecedentes del circuito, pues este circuito ha sido utilizado por la compañía APPLE desde sus inicios para la transferencia de información.

Con esto, hemos definido los sistemas de interfaz entre los dos canales, lo que queda es definir que dispositivos servirán como traductores.

Resulta evidente la necesidad de un microprocesador para controlar el dispositivo NCR53C80, pero además es conveniente el uso de un controlador DMA ya que con este se lograrían velocidades mayores. Será necesario una memoria de sistema donde se tendrá el programa que controla al microprocesador (ROM) y una memoria para manejo de apuntadores (RAM). Con esto el traductor quedaría formado como muestra la figura 3.2

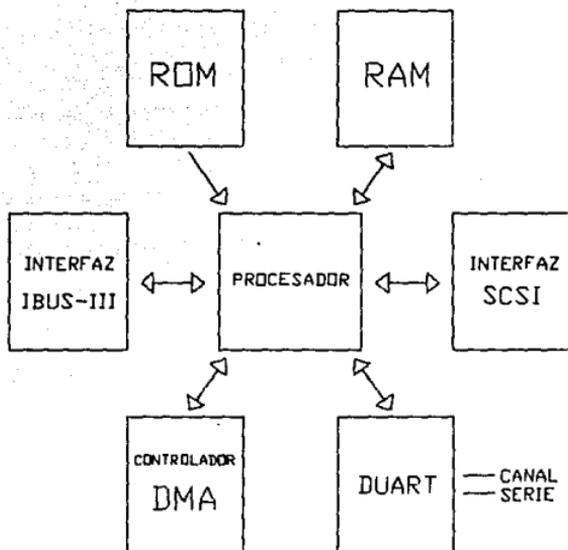


Figura 3.2 Diagrama del traductor de la tarjeta

Considerando que además de lo descrito se necesitarán circuitos extra para la decodificación de las direcciones y contemplando la necesidad de agregar circuitería que permita una comunicación serial para el monitoreo del desarrollo del sistema, se consideró como factor importante el espacio, ya que las dimensiones estándar de las tarjetas de la línea SAC son de 16.50 x 15.00 cm. La mejor opción tanto económica como de espacio fue el microprocesador 80C188, el cual además de su bajo costo y alta velocidad de procesamiento (reloj de 16 MHz), es un procesador de 8 bits externos y posee en el mismo chip dos canales de DMA, líneas especiales para la selección tanto de dispositivos periféricos como de memoria, posee un timer programable y controlador de interrupciones, todo en un circuito integrado de muy alta escala de integración (VLSI) de 68 terminales (pins).

Esto permitirá un mínimo de conexiones con otros dispositivos, logrando un considerable ahorro de espacio dentro de la tarjeta, además de que la programación será compatible con el software desarrollado para la tarjeta procesadora, ya que esta posee un microprocesador 80C88.

Las características principales del microprocesador 80C188 son:

- Microprocesador de 16 bits internos y 8 bits externos
- CHMOS de alta integración con modos de operación:
  - modo mejorado (enhanced) con refresco para DRAM y ahorro de energía (power-save)
  - modo compatible como reemplazo pin por pin del NMOS 80188 para aplicaciones no numéricas.
- CPU 80C86/C88 mejorado
- Generador de reloj
- 2 canales DMA independientes
- Controlador de interrupciones programable
- 3 contadores de 16-bits programables
- Unidad de control de refresco de RAM dinámica
- Lógica de selección de circuito de memoria y periféricos programable
- Generador de estados de espera programable
- Controlador de canal local
- Capacidad de direccionar directamente 1 Mbyte de memoria y 64 Kbytes de puertos.

A continuación se presenta una descripción más detallada de estos circuitos.

## 4. DESCRIPCION DE CIRCUITOS

### 4.1 PROCESADOR 80C188

#### 4.1.1 Características Generales

Este microprocesador de 16 bits, CHMOS de alta integración presenta los modos de operación: modo mejorado (enhanced) con refresco para DRAM y ahorro de energía (power-save) y modo compatible como reemplazo pin por pin del NMOS 80188 para aplicaciones no numéricas. Otras características son se presentaron en el capítulo 3.

#### 4.1.2 Interrupciones

Una Interrupción transfiere la ejecución a una nueva localidad de programa. La dirección vieja del programa (CS:IP) y los estados de la máquina (palabra de estados) son salvados en el stack para permitir la reanudación del programa interrumpido. Hay tres clases de interrupciones: iniciadas por hardware, instrucciones INT y excepciones de instrucciones. Las interrupciones iniciadas por hardware son respuesta a una entrada externa y se clasifican en mascarables y no mascarables.

Los programas pueden causar una interrupción con una instrucción de INT. Las excepciones de instrucciones ocurren cuando se detecta una condición unusual al tratar de ejecutar una instrucción. Si la excepción fue causada al intentar ejecutar la instrucción ESC, la instrucción de regreso (IRET) apuntará a la instrucción ESC o al segmento prefijo de anulación (segment prefix override), si está presente, que sigue a la instrucción ESC. En los demás casos, la dirección de regreso de una excepción apuntará a la instrucción siguiente de la instrucción que provocó la excepción.

Por cada interrupción, se debe suministrar un vector de 8 bits al 80C188, el cual identifica la entrada adecuada en la tabla. Las excepciones suministran el vector de interrupciones internamente. Además, los periféricos internos y las interrupciones externas que no están en cascada, generarán sus propios vectores a través del controlador de interrupciones interno. Las instrucciones INT implican el vector y permiten acceso a las 256 interrupciones. Las interrupciones mascarables iniciadas por hardware suministran el vector de 8 bits al CPU durante la secuencia de reconocimiento de la interrupción. Las interrupciones no mascarables de hardware usan un vector predefinido suministrado internamente.

Todas las fuentes de interrupción son atendidas por un llamado indirecto a través de un elemento de la tabla de vectores. Esta tabla de vectores está indexada usando el tipo de vector de interrupción multiplicada por 4 (véase tabla 4.1.1). Todas las interrupciones generadas por hardware son muestreadas a final de cada instrucción. Así, las interrupciones por software serán atendidas primero. Una vez que se ha entrado a una rutina de servicio y se habilitan las interrupciones, cualquier fuente de interrupción de suficiente prioridad puede interrumpir la rutina de servicio en progreso.

El 80C188 provee pines de solicitud de interrupción INTO-INT3. Además, el controlador de DMA y la unidad de reloj integrada del 80C188 pueden generar interrupciones mascarables. El tipo de vector de estas interrupciones se muestra en la tabla 4.1.1. El software habilita estas entradas activando la bandera IF en la palabra de control (instrucción STI).

Futuras interrupciones mascarables quedan deshabilitadas mientras se atiende una interrupción ya que el bit IF es desactivado como parte de la respuesta a la interrupción o a la excepción. La palabra de estados salvada reflejará el estado habilitado del procesador antes de la interrupción. La bandera de interrupción permanecerá en cero a menos que sea específicamente activada. La instrucción de regreso de interrupción (IRET) reestablece la palabra de estados, con ello reestablece el estado original del bit IF. Si el regreso de interrupción habilita las interrupciones y se tiene otra interrupción pendiente, el 80C188 atenderá inmediatamente la interrupción de mayor prioridad.

Se tiene una interrupción no-mascarable (NMI). Esta interrupción es atendida sin importar el estado del bit IF. El uso típico de esta interrupción es para activar la rutina de falla de poder. La activación de esta entrada causa una interrupción con un vector suministrado internamente de 2. No se realiza ninguna secuencia de reconocimiento de interrupción. El bit IF es desactivado al inicio de la interrupción NMI para evitar que otras interrupciones sean atendidas.

Las interrupciones por software son los tipos: 0, 1, 2, 3, 4, 5, 6 y 7 (véase tabla 4.1.1)

Las de hardware mascarable son: las entradas INTO-INT3, las generadas por el controlador de DMA y la unidad de contadores integrada. Estas interrupciones son atendidas si el bit IF de la palabra de estados está activo.

Nombre de Interrupción	Tipo de vector	Dirección de vector	Prioridad default
Divide error error de división	0	00h	1
Interrupción de paso sencillo single step	1	04h	1A
NMI	2	08h	1
interrupción breakpoint	3	0Ch	1
Sobreflujo Overflow	4	10h	1
array bounds	5	14h	1
Unused	6	18h	1
ESC	7	1Ch	1
Timer 0	8	20h	2A
Timer 1	18	48h	2B
Timer 2	19	4Ch	2C
Reservada	9	24h	3
DMA 0	10	28h	4
DMA 1	11	2Ch	5
INT 0	12	30h	6
INT 1	3	34h	7
INT 2	14	38h	8
INT 3	15	3Ch	9
reservadas	16,17	40h,44h	
reservadas	20-31	50h..7Ch	

Tabla 4.1.1 Tabla de vectores de interrupción.

#### 4.1.3 Condiciones iniciales del circuito

Las condiciones iniciales del circuito se realizan llevando a cero la entrada RES\, lo que provoca que se suspendan todas las actividades. Después de que RES\ se activa, el 80C188 ejecuta la instrucción localizada en la dirección física FFFF0h. Al poner en condiciones iniciales el microprocesador, algunos registros toman el valor predefinido que se muestra a continuación:

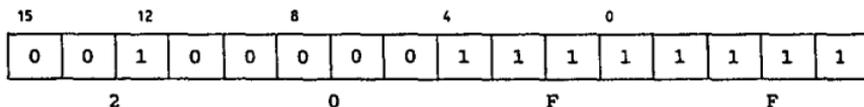
Status Word	F002h
Instruction Pointer	0000h
Code Segment	FFFFh
Data Segment	0000h
Extra Segment	0000h
Stack Segment	0000h
Relocation Register	20FFh
UMCS	FFFBh

Tabla 4.1.2 Condiciones iniciales.

#### 4.1.4 Interfaz con periféricos interna

Todos los periféricos integrados en el 80C188 son controlados por registros de 16 bits contenidos dentro de un bloque de control de 256 bytes. Este bloque de control puede ser mapeado dentro del espacio de memoria o de puertos. La lógica interna reconocerá la dirección del bloque y responderá a los ciclos del canal. Los registros de este bloque de control pueden ser leídos o se puede escribir en ellos en cualquier momento.

La dirección base del bloque de control está programada por un registro de 16 bits contenido en el bloque con el desplazamiento (offset) FEh de la dirección base.



Registro de relocalización (desplazamiento FEh)

- Bit 15      No importa
- Bit 14      0 = controlador de interrupciones como maestro  
              1 = controlador de interrupciones como esclavo
- Bit 13      No importa

Bit 12    0 = Bloque de control localizado en espacio de puertos  
          1 = Bloque de control localizado en espacio de memoria

Bit 11-8 Estos bits representan los bits 16 a 19 de la dirección base en donde se localizará el bloque de control. Si se localiza en espacio de puertos, estos 4 bits deberán ser ceros, debido a que en el espacio de puertos sólo se consideran 16 bits y no 20.

Bit 7-0 Estos bits representan los bits 8 a 15 de la dirección base en donde se localizará el bloque de control.

En la condición inicial, este registro contiene un 20FFh (el número presentado en la figura del registro de relocalización) lo que indica que:

- el controlador de interrupciones está en modo maestro
- el bloque de control está localizado en el espacio de puertos
- la dirección base del bloque de control es FF00h

El bloque de control se muestra a continuación:

	desplazamiento (offset)
Registro de relocalización	FEh
Descriptor de canal 1 DMA	DAh
	D0h
Descriptor de canal 0 DMA	CAh
	C0h
Registro de control de selección de <u>chip</u>	A8h
	A0h
Registro de control Timer 2	66h
	60h
Registro de control Timer 1	5Eh
	58h
Registro de control Timer 0	56h
	50h
Registro de controlador de interrupciones	3Eh
	20h

Figura 4.1.2 Mapa de registros internos.

El 80C188 contiene lógica que provee generación de selección de circuitos programable para memoria y para periféricos.

#### 4.1.5 Selección de circuitos de memoria

Se tienen 6 salidas de selección de circuitos para 3 áreas de direcciones: memoria superior (una salida), memoria inferior (una salida) y memorias de rango medio (cuatro salidas).

CS\ de Memoria superior:

Esta es la salida UCS\. Usualmente se usa la parte alta de la memoria como la memoria del sistema ya que después de poner las condiciones iniciales, se ejecuta la instrucción en la localidad FFFF0h.

El límite superior para esta señal es siempre FFFFh, mientras que el límite inferior es programable como se muestra en la tabla:

Dirección de inicio	Tamaño del bloque de memoria	Valor de UMCS
FFC00H	1K	FFF8H
FF800H	2K	FFB8H
FF000H	4K	FF38H
FE000H	8K	FE38H
FC000H	16K	FC38H
F8000H	32K	F838H
F0000H	64K	F038H
E0000H	128K	E038H
C0000H	256K	C038H

Tabla 4.1.3 Programación del registro UMCS.

UMCS es el registro de control para programar la selección de la memoria alta y se encuentra en el desplazamiento (offset) A0h. Los datos mostrados en la tabla son considerando R0 = R1 = R2 = 0.

15	12	8	4	0															
1	1	U	U	U	U	U	U	U	U	U	U	1	1	1	R2	R1	R0		

Registro UMCS (desplazamiento A0h)

Los valores permitidos para los bits 6-13 y su dirección base resultante se muestran en la tabla. Cualquier combinación no presentada resultará en una operación no definida. Después de las condiciones iniciales, este registro queda programado para acceder sólo 1K (UMCS = FFFBh). Los valores de R0-R2 se especifican más adelante (ver lógica de generación de READY).

CS\ de Memoria inferior:

Esta es la salida LCS\ . El límite inferior para esta señal es siempre 00000h, mientras que el límite superior es programable como se muestra en la tabla:

Dirección superior	Tamaño del bloque de memoria	Valor de LMCS
003FFH	1K	0038H
007FFH	2K	0078H
00FFFH	4K	00F8H
01FFFH	8K	01F8H
03FFFH	16K	03F8H
07FFFH	32K	07F8H
0FFFFH	64K	0FF8H
1FFFFH	128K	1FF8H
3FFFFH	256K	3FF8H

Tabla 4.1.4 Programación del registro LMCS.

LMCS es el registro de control para programar la selección de la memoria baja y se encuentra en el desplazamiento (offset) A2h. Los valores mostrados en la tabla son considerando R0 = R1 = R2 = 0.

15	12	8	4	0														
0	0	U	U	U	U	U	U	U	U	1	1	1	R2	R1	R0			

Registro LMCS (desplazamiento A2h)

Los valores permitidos para los bits 6-13 y su dirección base resultante se muestran en la tabla. Cualquier combinación no presentada resultará en una operación no definida. La línea LCS\ no se activará hasta que no se accese este registro. Los valores de R0-R2 se especifican más adelante (ver lógica de generación de READY).

CS\ de memorias de rango medio:

Se tienen cuatro líneas de selección de circuitos, las cuales son activadas dentro de un bloque de memoria localizado por el usuario. Este bloque se puede localizar en cualquier parte dentro del espacio de memoria de 1 Mbyte a excepción de los espacios definidos para UCS\ y LCS\ . Tanto como la dirección base, como la longitud de este bloque de memoria puede ser programado.

El tamaño del bloque de memoria está definido por los bits 8-14 del registro MPCS como se muestra a continuación:

Tamaño total del bloque	Tamaño individual de selección	MPCS bits 14 - 8
8K	2K	0000001B
16K	4K	0000010B
32K	8K	0000100B
64K	16K	0001000B
128K	32K	0010000B
256K	64K	0100000B
512K	128K	1000000B

Tabla 4.1.5 Programación del registro MPCS.

Como se ve, sólo uno de estos bits debe activarse cada vez, de otro modo la operación de las líneas MCS\ es impredecible. Cada una de las cuatro líneas de selección de circuitos se activa para cada uno de los cuatro divisiones iguales y contiguas del bloque de rango medio. Si el bloque es en total de 32K, cada selección de circuito se activa para 8K de memoria activando MCS0\ para el primer rango y MCS3\ para el último.

MPCS es el registro de control para programar la selección de la memoria de rango medio y se encuentra en el desplazamiento (offset) A8h. Los bits EX y MS se explicarán más adelante.

15	12	8	4	0											
1	M6	M5	M4	M3	M2	M1	M0	EX	MS	1	1	1	R2	R1	R0

Registro MPCS (desplazamiento A8h)

La dirección base del bloque de memoria de rango medio está definido por los bits 15-9 del registro MMCS. Estos bits corresponden a los bits A19-A13 de la dirección de memoria de 20 bits. Los bits A12-A0 de la dirección base son siempre ceros. Después de las condiciones iniciales, el contenido de estos dos registros es indefinido, sin embargo ninguna de las líneas MCS\ se activará hasta que se hayan accedido ambos registros (MMCS y MPCS).

15	12	8	4	0													
U	U	U	U	U	U	U	U	1	1	1	1	1	1	1	R2	R1	R0

A19

A13

Registro MMCS (desplazamiento A6h)



Bit	Descripción
MS	1 = Los periféricos se mapean en el espacio de puertos
	0 = Los periféricos se mapean en el espacio de memoria
EX	0 = 5 líneas PCS\ y A1, A2 disponibles
	1 = 7 líneas PCS\ y A1, A2 no disponibles

Tabla 4.1.6 Descripción de los bits MS y EX.

#### 4.1.7 Lógica de generación de READY

El 80C188 puede generar la señal de READY internamente para cada línea de CS\ ya sea de memoria o de puertos. El número de estados de espera a ser insertados es programado para cada uno y pueden ser de 0 a 3. Además, puede ser programado para aceptar o ignorar la señal externa de READY para cada CS\ individualmente.

El control de READY consiste de 3 bits para cada línea CS\. La combinación de esos bits y el resultado que se obtiene se muestra en la siguiente tabla:

R2	R1	R0	Número de ciclos de espera generados
0	0	0	0 ciclos de espera. Acepta Ready externo
0	0	1	1 ciclos de espera. Acepta Ready externo
0	1	0	2 ciclos de espera. Acepta Ready externo
0	1	1	3 ciclos de espera. Acepta Ready externo
1	0	0	0 ciclos de espera. No reconoce Ready externo
1	0	1	1 ciclos de espera. No reconoce Ready externo
1	1	0	2 ciclos de espera. No reconoce Ready externo
1	1	1	3 ciclos de espera. No reconoce Ready externo

Tabla 4.1.7 descripción de los bits R2-R0.

El generador de READY interno opera en paralelo con el READY externo, no en serie, si este se usa (R2 = 0). Por ejemplo: si el generador de READY interno está puesto para insertar dos ciclos de espera, pero la actividad en la línea READY externa va a insertar 4 ciclos de espera, el procesador insertará sólo cuatro ciclos y no seis, ya que los dos ciclos de espera internos se superponen con los cuatro externos.

Los bits R2-R0 de cada palabra de control especifican el modo de READY para el bloque correspondiente, con la excepción de que para los periféricos, PCS0\ - PCS3\ se controlan con R2-R0 de PACS mientras que PCS4\ - PCS6\ se controla con R2-R0 de MPCS.

## 4.1.8 Canales de DMA

El 80C188 provee dos canales de DMA independientes. Las transferencias de datos pueden ocurrir entre memoria y memoria, memoria y puertos y entre puertos y puertos. Cada canal mantiene ambos apuntadores de 20 bits, el del origen y el del destino, los cuales pueden ser incrementados, decrementados o permanecer sin cambio. Cada transferencia de datos toma 2 ciclos de canal (8 ciclos de reloj), uno para leer el dato y el otro para escribirlo.

## Operación:

Cada canal tiene 6 registros en el bloque de control los cuales definen la operación de cada canal. Los registros de control son: apuntador destino de 20 bits, apuntador origen de 20 bits, registro contador de transferencia y una palabra de control de 16 bits.

Nombre del registro	Dirección del registro	
	Canal 0	Canal 1
Palabra de control	CAh	DAh
Contador de transferencias	C8h	D8h
Apuntador destino (4 bits superiores)	C6h	D6h
Apuntador destino	C4h	D4h
Apuntador origen (4 bits superiores)	C2h	D2h
Apuntador origen	C0h	D0h

Figura 4.1.4 Mapa de registros de control de canales DMA.

El registro contador de transferencias especifica el número de transferencias DMA que se deben llevar a cabo. Hasta 64K byte o word se pueden transferir con terminación automática. La palabra de control define la operación del canal. Todos los registros pueden ser modificados o alterados durante cualquier actividad del DMA. Cualquier cambio a los registros se manifiesta inmediatamente en la operación del DMA.

## Registro palabra de control:

1	5	1	4	1	3	1	2	1	1	1	0	9	8	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Registro control de DMA

## Control de destino:

bit 15 M/IO\; 1 = el apuntador destino está en el espacio de memoria, 0 = en el espacio de puertos.

- bit 14 DEC: 1 = decreuenta el apuntador después de cada transferencia.
- bit 13 INC: 1 = incrementa el apuntador después de cada transferencia.

Si ambos (INC y DEC) están especificados (1) el apuntador no cambiará después de cada transferencia.

#### Control de origen:

- bit 12 M/IO\; 1 = el apuntador origen está en el espacio de memoria, 0 = en el espacio de puertos.
- bit 11 DEC: 1 = decreuenta el apuntador después de cada transferencia.
- bit 10 INC: 1 = incrementa el apuntador después de cada transferencia.

Si ambos (INC y DEC) están especificados (1) el apuntador no cambiará después de cada transferencia.

- bit 9 TC: Si está activo (1) el DMA terminará cuando el contenido del registro contador de transferencias llegue a cero. El bit ST/STOP\ también será puesto en condiciones iniciales al llegar a este punto. Si no se activa, el controlador de DMA decrementará el registro contador de transferencias después de cada ciclo DMA, pero las transferencias de DMA no se detendrán al llegar a cero.

- bit 8 INT Habilita interrupciones al CPU hasta la terminación de la cuenta de transferencias.

bit 7 SYN

- bit 6 SYN Estos dos bits determinan la sincronización.  
 00 = No sincronización (en este caso, el bit TC será ignorado y el bit ST/STOP\ será limpiado (0) cuando la cuenta del registro llegue a cero.  
 01 = El origen establece la sincronización.  
 10 = El destino establece la sincronización.  
 11 = No usado.

- bit 5 P Este bit establece la prioridad de uno de los canales sobre el otro.  
 0 = baja prioridad.  
 1 = alta prioridad.  
 Si ambos canales tienen la misma prioridad, los canales alternarán ciclos.
- bit 4 TDRQ Habilita/deshabilita (1/0) solicitud de DMA del contador 2.
- bit 3 No importa su contenido.
- bit 2 CHG/NOCHG Cambia/no cambia (1/0) el bit ST/STOP\.  
 Si este bit está activo cuando se escribe la palabra de control, el bit ST/STOP\ será programado por la escritura en la palabra de control. Si este bit no está activo cuando se escribe la palabra de control, el bit ST/STOP\ no será alterado; siempre será leído como 0.
- bit 1 ST/STOP\ Inicio/paro (1/0) de canal.
- bit 0 No importa su contenido.

#### Solicitud de DMA:

Las transferencias de los datos pueden ser sincronizadas ya sea por el destino o por la fuente. Además, las transferencias pueden ser no sincronizadas, esto es, que las transferencias se harán continuamente hasta que el contador de transferencias sea cero. Cuando se realizan transferencias no sincronizadas o sincronizadas por la fuente, el canal de DMA puede iniciar otra transferencia inmediatamente después del final de la transferencia previa. Esto permite que una transferencia completa se lleve 2 ciclos de canal (8 ciclos de reloj, asumiendo que no se tienen ciclos de espera). Cuando se realiza sincronización del destino, el dato no será enviado de la dirección fuente hasta que el dispositivo destino indique que está listo para recibirlo. También, el controlador de DMA solicitará el control del canal después de cada transferencia. Si no se ha iniciado ninguna otra actividad del canal, otra transferencia sincronizada por el destino se llevará a cabo después de 2 ciclos del procesador. Esto da tiempo al dispositivo destino a retirar su solicitud si no se desea otra transferencia. Debido a que el DMA solicitará el canal, el CPU puede iniciar un ciclo de canal. Como resultado, un ciclo completo de canal será insertado entre las transferencias sincronizadas por el destino. La tabla muestra los rangos máximos de transferencias de DMA.

Tipo de sincronización elegida	Velocidad de la transferencia
no sincronizada	2.0 Mbytes/s
sinc. fuente	2.0 Mbytes/s
sinc. destino	1.3 Mbytes/s

Tabla 4.1.8 Rangos máximos de transferencias DMA.

## Reconocimiento de DMA:

Este microprocesador no tiene un pulso explícito de DMA acknowledge. Dado que ambos apuntadores, el fuente y el destino, son mantenidos, una lectura de una fuente solicitante o una escritura a un destino solicitante debe ser usada como señal DMA acknowledge. Como las líneas de selección de circuitos pueden ser programadas para un determinado espacio de memoria o puertos y los apuntadores DMA pueden ser programados para apuntar al mismo bloque, una línea de CS puede ser usada para indicar el DMA acknowledge.

## Prioridades de DMA:

Los canales de DMA pueden ser programados para dar a uno prioridad sobre el otro, o se pueden programar para alternar ciclos cuando ambos tienen solicitudes de DMA pendientes. Los ciclos de DMA siempre tienen prioridad sobre los ciclos del CPU internos, a excepción de los accesos a memoria "locked"; también un canal hold externo tiene prioridad sobre los ciclos de DMA. Debido a que una solicitud de interrupción (INT request) no puede suspender la operación del DMA y el CPU no puede acceder memoria durante los ciclos de DMA, un se sufrirá retardo en la interrupción durante la secuencia de ciclos DMA continuos. Sin embargo, una solicitud de NMI (NMI request), causará un alto en todas las actividades internas del DMA. Esto permite al CPU responder rápido al NMI.

## Programación del DMA:

Los ciclos de DMA ocurrirán siempre que el bit ST/STOP\ del registro de control esté activo. Si se programan transferencias sincronizadas, se debe generar un DRQ. Los apuntadores de

transferencia, tanto fuente como destino y el registro de cuenta, si se usa, deben ser programados antes de activar el bit ST/STOP\.

Cada registro DMA puede ser modificado mientras el canal está operando. Si el bit CHG/NOCHG es limpiado mientras el registro de control es escrito, el bit ST/STOP\ del registro de control no será modificado con la escritura. Si se modifican múltiples registros del canal, se recomienda que se use una transferencia string "LOCKED" para prevenir que durante las modificaciones a los registros ocurran transferencias.

Canales DMA y condiciones iniciales:

Después de ponerlo en las condiciones iniciales, el estado de los canales de DMA será como sigue:

- \* El bit ST/STOP\ para cada canal está en STOP (0).
- \* Cualquier transferencia en proceso es suspendida
- \* Los valores de los registros de contador de transferencia, apuntador fuente y apuntador destino son indeterminados.

#### 4.1.9 Controlador de interrupciones

El 80C188 puede recibir interrupciones de varias fuentes tanto internas como externas. El controlador de interrupciones interno sirve para atender estas solicitudes en base a una prioridad, para ser servidas individualmente por el CPU.

Las fuentes internas de interrupción (timers y canales DMA) pueden ser deshabilitados por sus propios registros de control o por bits máscara dentro del controlador de interrupción. El controlador de interrupciones del 80C188 tiene su propio registro de control que establece el modo de operación del controlador.

El controlador de interrupciones resolverá la prioridad entre solicitudes pendientes simultáneamente. Nesting está provisto para que rutinas de interrupción de baja prioridad sean interrumpidas por interrupciones de mayor prioridad.

El 80C188 tiene un modo especial como esclavo, en el cual, el controlador de interrupciones interno actúa como un esclavo de un maestro externo. El controlador es programado por el bit 14 del registro de relocalización del bloque de control (desplazamiento FEh).

Operaciones en modo maestro:

Se proveen cinco pines para fuentes de interrupción externa. Uno de estos pines es NMI, interrupción no mascarable. NMI es generalmente utilizada para eventos poco usuales como interrupciones por fallas de poder. Los otros cuatro pines pueden ser configurados de alguna de las siguientes maneras:

\* Como 4 entradas de interrupción con generación de vectores de interrupción interna.

\* Como par de línea de interrupción y línea de acknowledge (modo cascada) con vectores de interrupción generados externamente, más 2 líneas de entrada de interrupción con generación de vectores interna.

\* Como dos pares de líneas de interrupción y acknowledge de interrupción (modo cascada) con vectores de interrupción generados externamente.

Las fuentes externas en modo cascada usan un vector de interrupción generado externamente. Cuando una interrupción es reconocida, dos ciclos de INTA\ son iniciados y es vector es leído en el 80C188 en el segundo ciclo. La capacidad de interfazar con un controlador de interrupción programable 82C59A está provista para cuando las entradas están configuradas en modo cascada.

#### Modos de operación del controlador de interrupciones:

Los modos básicos de operación del controlador de interrupciones en modo maestro, son similares al 82C59A. El controlador de interrupciones responde idénticamente igual para las interrupciones internas en los tres diferentes modos: la diferencia es sólo en la interpretación de función de los cuatro pines de interrupción externa. El controlador de interrupciones se deja en uno de estos tres modos programando los registros de control INTO e INT1 adecuadamente. Los modos de operación del controlador de interrupciones son:

##### Modo FULLY\_NESTED

En este modo, 4 pines son usados como solicitudes directas de interrupción. Los vectores para estas cuatro entradas son generados internamente. Se provee un bit de servicio para cada fuente de interrupción. Si un dispositivo de menor prioridad solicita una interrupción mientras el bit IS está activo, el controlador de interrupciones no generará la interrupción. Además, si otra solicitud de interrupción ocurre de la misma fuente de interrupción mientras el bit IS está activo, no se genera la interrupción. Esto permite a las rutinas de servicio a interrupciones operar con interrupciones habilitadas y ser suspendidas sólo por interrupciones de mayor prioridad que la del bit de servicio activo.

Cuando se completa una rutina de servicio, el bit IS correspondiente debe ser desactivado escribiendo el patrón adecuado en el registro EOI. Esto es necesario para permitir interrupciones subsecuentes de esa fuente y para permitir servir a interrupciones de menor prioridad. Un comando EOI es ejecutado al final de la rutina de servicio justo antes del regreso de interrupción (IRET).

### Modo cascada

Este modo establece dos pares de solicitud-reconocimiento (REQ/ACK) de interrupción, a diferencia del modo fully nested que establece cuatro entradas de solicitud de interrupción. Este esquema es utilizado cuando se conecta con otro controlador de interrupciones externo.

### Características del Modo maestro:

El usuario puede programar las fuentes de interrupción en ocho diferentes niveles de prioridad. La programación se realiza con tres bits de nivel de prioridad (0-7) en el registro de control de cada fuente de interrupción (la mayor prioridad es 0 y la menor prioridad es 7). Todas las interrupciones tienen un nivel de prioridad por default (ver tabla 4.1.1).

Si dos interrupciones con la misma prioridad están pendientes al mismo tiempo, el esquema de prioridades mostrado en la tabla 4.1.1 es usado. Si la rutina de servicio a una interrupción rehabilita las interrupciones, otras solicitudes de interrupción podrán ser atendidas.

### Comando EOI:

El comando EOI (fin de interrupción) es usado por el programador para desactivar el bit en servicio (IS) cuando se ha completado una rutina de servicio a interrupción. Este comando es enviado al escribir el patrón adecuado en el registro EOI. Se tienen dos tipos de comandos EOI, específico y no específico. El no específico no indica cual bit IS debe desactivarse; al ser emitido este comando, el controlador de interrupciones automáticamente desactiva el bit IS de mayor prioridad que tiene una rutina de servicio activa. El comando EOI específico requiere que el programador envíe el tipo de vector de la interrupción cuyo bit IS será desactivado.

### Modo de disparo:

Los cuatro pines de interrupción externas pueden ser programados para disparar por flanco o por nivel. La programación se realiza a través del bit LTM (modo de nivel de disparo). Todas las interrupciones son activo alto. En ambos modos de disparo, la interrupción debe permanecer en alto hasta que el CPU reconoce la interrupción. En el modo de sensado de flanco, si la interrupción permanece en alto después de ser reconocida, la entrada es deshabilitada y no se generaran futuras solicitudes. El nivel de entrada debe estar en bajo por lo menos un ciclo de reloj para restablecer la entrada.

## Vectores de interrupción:

El controlador de interrupciones del 80C188 generará vectores de interrupción para los canales de DMA integrados y para los timers integrados. Además, el controlador generará vectores de interrupción para las líneas de interrupción externas, si éstas están configuradas en modo directo (fully nested). Los vectores de interrupción generados son fijos y no pueden ser cambiados (tabla 4.1.1).

## Registros del controlador de interrupciones:

Registro de control INT3	3Eh	desplazamiento
Registro de control INT2	3Ch	
Registro de control INT1	3Ah	
Registro de control INT0	38h	
Registro de control DMA1	36h	
Registro de control DMA0	34h	
Registro de control TIMERS	32h	
Registro de estado de interrupciones	30h	
Registro de solicitud de interrupciones	2Eh	
Registro en servicio	2Ch	
Registro de máscara de prioridades	2Ah	
Registro de máscara	28h	
Registro de estados POLL	26h	
Registro POLL	24h	
Registro EOI	22h	

Figura 4.1.5 Mapa de registros del controlador de interrupciones.

Registro en servicio (in-service register):

Se puede escribir o leer en él. Contiene el bit de servicio para cada fuente de interrupción. El bit de servicio se activa para indicar que hay una rutina de servicio en progreso. Cuando hay un bit de servicio activo, el controlador no generará interrupciones al CPU cuando reciba una interrupción de menor prioridad. El bit TMR es el bit de servicio para los tres timers; D0 y D1 son los bits de servicio para los canales de DMA; I0-I3 son los bits de servicio para los pines de interrupción externa.

El bit IS se activa cuando el procesador reconoce una interrupción ya sea por un reconocimiento de interrupción o escribiendo en el registro POLL. El bit IS es desactivado al final de la rutina de servicio enviando el comando EOI.

15	12	8	4	0											
0	0	0	0	0	0	0	0	I3	I2	I1	I0	D1	D0	0	TMR

## Registro de solicitud de interrupción:

Las fuentes internas de interrupción tienen bits dentro del controlador de interrupciones. Una lectura en este registro reflejará el estado de estos bits. El bit TMR es el OR lógico de las solicitudes de los timers. D0 y D1 son los bits de solicitud de los canales de DMA.

También se indica el estado de los pines de entrada de interrupciones externas, los cuales no son una condición que se almacena en el controlador, por lo que no es posible escribir en los bits de interrupción externos.

Escribir en este registro sólo afecta los bits de solicitud de interrupción de los canales DMA. Activando cualquiera de ellos, se provoca la solicitud correspondiente, mientras que desactivando, se limpia la solicitud de interrupción. Los demás bits del registro sólo pueden ser leídos.

15	12	8	4	0											
0	0	0	0	0	0	0	0	I3	I2	I1	I0	D1	D0	0	TMR









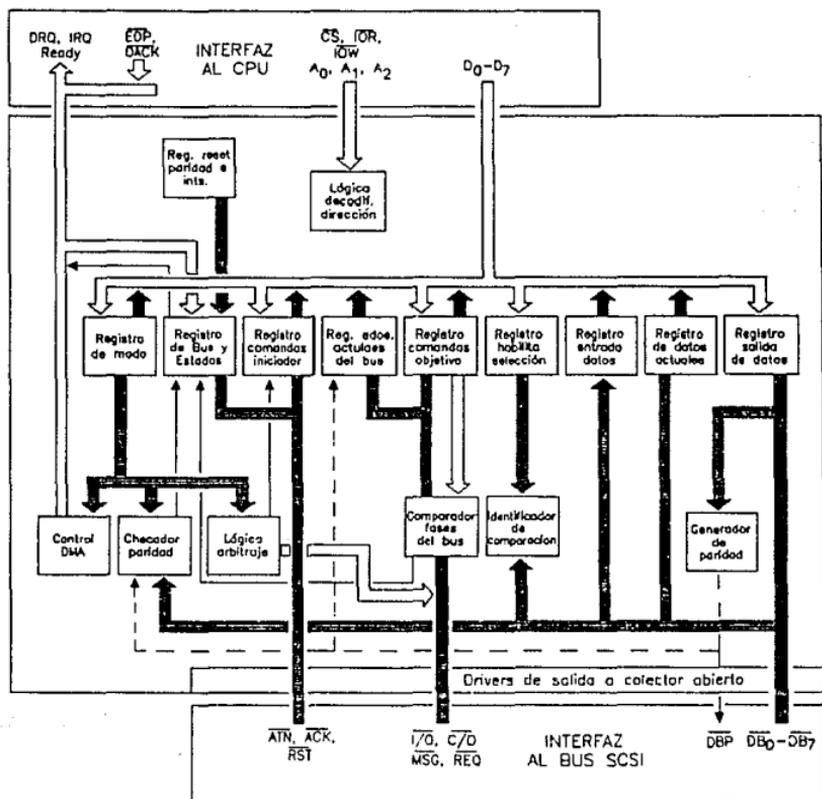


Figura 4.2.1 Diagrama a bloques del controlador SCSI NCR-53C80.

#### 4.2.2 Descripción de pines

Señales de alimentación:

VDD Fuente de poder del circuito +5 volts.

GND Tierra del circuito.

## Señales de interfaz al microprocesador:

A0-A2	I	Estas señales son usadas con CS\, IOR\ o IOW\ para direccionar todos los registros internos.
CS\<	I	Selección de circuito: habilita una lectura o escritura del registro interno seleccionado por A0-A2. Es activo bajo.
DACK\<	I	Reconocimiento de DMA: pone en condición inicial DRQ y selecciona el registro de datos para transferencias de entrada o salida de datos.
DRQ	O	Solicitud de DMA: indica que el registro de datos está listo para ser leído o escribir en él. DRQ ocurre sólo si el bit MODO DMA es verdadero (1) en el registro de comandos. Es limpiado por DACK\<.
D0-D7	I/O/HI	Canal de datos del microprocesador activo alto.
EOP\<	I	Señal de fin de proceso: es usada para terminar una transferencia de DMA. Si se activa durante un ciclo de DMA, el byte actual en el canal será transferido, pero no se solicitarán bytes adicionales.
IOR\<	I	Es usada para leer un registro interno seleccionado por CS\< y A0-A1. También selecciona el registro de entrada de datos cuando se usa con DACK\<. Es activo bajo.
IOW\<	I	Es usada para escribir en un registro interno seleccionado por CS\< y A0-A1. También selecciona el registro de salida de datos cuando se usa con DACK\<. Es activo bajo.
IRQ	O	Solicitud de interrupción: alerta al microprocesador de la existencia de alguna condición de error o de que se completó un evento.
READY	O	Puede ser usada para controlar la velocidad de transferencias en modo DMA a bloques. La señal se activa para indicar que el circuito está listo para enviar/recibir un dato y permanece falsa después de la transferencia hasta que el último byte es enviado o hasta que el bit MODO DMA es puesto en condición inicial.

RESET\ I Limpia todos los registros. No fuerza la señal RES\ del canal SCSI a ser activada. Es una señal activo bajo.

Señales de interfaz al canal SCSI: todas son bidireccionales, activo bajo y de colector abierto.

- ACK\ Manejada por un iniciador, indica un reconocimiento para el handshake REQ/ACK en la transferencia de datos. En el papel de objetivo, ACK\ es recibido como respuesta a la señal REQ\.
- ATN\ Manejada por un iniciador, indica una condición de ATENCION. Esta señal es recibida en el papel de objetivo.
- BSY\ Esta señal indica que el canal SCSI está siendo usado y puede ser controlada tanto por el iniciador como por el objetivo.
- CD\ Esta señal es manejada por el objetivo e indica si en el canal se tienen datos o información de control. Esta señal es recibida por el iniciador.
- IO\ Es una señal manejada por el dispositivo que controla la dirección de los movimientos de datos en el canal SCSI. Verdadero indica entrada al iniciador. Esta señal también es usada para distinguir entre las fases de selección y de reelección.
- MSG\ Es una señal manejada por el objetivo durante la fase de mensajes. Esta señal es recibida por el iniciador.
- REQ\ Manejada por el objetivo indica una solicitud para un handshake REQ/ACK de transferencia de datos. Esta señal es recibida por el iniciador.
- RST\ Indica una condición de poner en condición inicial el canal SCSI.
- DB0\-DB9\, DBP Los ocho bits de datos más el bit de paridad forman el bus de datos. DB7\ es el bit más significativo y tiene la mayor prioridad durante la fase de arbitraje. La paridad es non. La paridad siempre es generada y la revisión es opcional. La paridad no es válida durante el arbitraje.

**SEL\** Es usada por el iniciador para seleccionar un objetivo, o por el objetivo para reelegir a un iniciador.

#### 4.2.3 Registros internos del 53C80

Este dispositivo aparece como un arreglo de 8 registros para el CPU controlador. Leyendo o escribiendo los registros adecuados, el CPU puede iniciar cualquier actividad del bus SCSI o puede muestrear o activar cualquier señal en el canal SCSI. Esto permite al usuario implementar todo a parte del protocolo SCSI en software. Estos registros son leídos (o escritos) activando CS\ con una dirección de A0-A2 y enviando el pulso IOR\ (IOW\). Esta sección describe la operación de los registros internos.

Los nombres de la señales son usados para describir el contenido de estos registros internos. AÚn cuando el canal es activo bajo, un "1" es usado para indicar activación de señal y un "0" es usado para indicar no-activado o estado inactivo.

A2	A1	A0	R/W	Nombre del registro.
0	0	0	R	Dato SCSI actual
0	0	0	W	Dato de salida
0	0	1	R/W	Comando de iniciador
0	1	0	R/W	Modo
0	1	1	R/W	Comando del objetivo
1	0	0	R	Estado del Bus SCSI actual
1	0	0	W	Habilita selección
1	0	1	R	Estado y Bus
1	0	1	W	Inicia envío DMA
1	1	0	R	Dato de entrada
1	1	0	W	Inicia Recepción DMA del objetivo
1	1	1	R	Cond. Inicial paridad/Interrupciones
1	1	1	W	Inicia Recepción DMA del iniciado

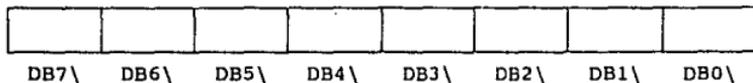
Tabla 4.2.1 Registros internos del controlador NCR-53C80.

#### Registros de datos:

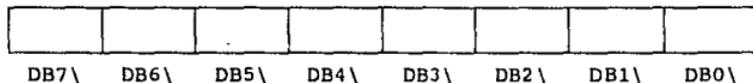
Los registros de datos son usados para transferir bytes de comandos, datos, estados y mensajes entre el canal de datos del microprocesador y el canal SCSI. Este microcontrolador no interpreta ninguna información que pasa a través de los registros de datos. Los registros de datos son: el registro de datos SCSI actual, el registro de datos de salida y el registro de datos de entrada.

**Registro de datos SCSI actual. Dirección 0 (sólo lectura)**

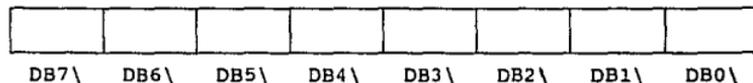
Este es un registro de sólo lectura que permite al microprocesador leer el canal de datos SCSI activo. Esto es acompañado por la activación de CS\ con la dirección A2-A0 de 000 y enviando un pulso de IOR\. Si la revisión de paridad está activa, la paridad del canal SCSI es revisada al principio del ciclo de lectura. Este registro se usa durante la lectura de datos programada como I/O ó durante el arbitraje, para revisar si hay dispositivos de mayor prioridad arbitrando. La paridad no es válida durante el arbitraje.

**Registro de datos de salida. Dirección 0 (sólo escritura)**

Este es un registro de sólo escritura que es usado para enviar datos al canal SCSI. El envío es realizado ya sea usando escritura normal del MPU o bajo el control de DMA, usando IOW\ o DACK\. Este registro también es usado para activar el bit identificador SCSI durante las fases de arbitraje y reelección.

**Registro de datos de Entrada. Dirección 6 (sólo lectura)**

Este registro es usado para leer los datos retenidos del canal SCSI. Los datos son retenidos durante la operación de recepción de DMA del objetivo cuando ACK\ (pin 14) cambia a activo o durante la recepción de DMA del iniciador cuando REQ\ (pin 20) se activa. El bit de Modo DMA (puerto 2, bit 1) debe estar activado antes de que los datos puedan ser retenidos en este registro. Este registro puede ser leído bajo el control de DMA usando IOR\ y DACK\. La paridad es opcionalmente revisada cuando el registro de dato de entrada es cargado.



**Registro de comandos del iniciador. Dirección 1  
(lectura/escritura)**

Este registro es usado para activar ciertas señales del canal SCSI, para monitorear estas señales y para monitorear el progreso del arbitraje. Muchos de estos bits son significativos sólo cuando se usa como iniciador, sin embargo, la mayoría pueden usarse en la operación como objetivo.

**REGISTRO DE LECTURA**

--	--	--	--	--	--	--	--

ACTIVA RST\	AIP	LA	ACTIVA ACK\	ACTIVA BSY\	ACTIVA SEL\	ACTIVA ATN\	ACTIVA BUS DE DATOS
----------------	-----	----	----------------	----------------	----------------	----------------	---------------------------

**REGISTRO DE ESCRITURA**

--	--	--	--	--	--	--	--

ACTIVA RST\	MODO PRUEBA	NO USADO	ACTIVA ACK\	ACTIVA BSY\	ACTIVA SEL\	ACTIVA ATN\	ACTIVA BUS DE DATOS
----------------	----------------	-------------	----------------	----------------	----------------	----------------	---------------------------

**Descripción de bits.**

**Bit 7. Activa RST\.**

Siempre que se escribe un "1" en este bit, la señal RST\ (pin 6) es activada en el canal SCSI. La señal RST\ permanece activa hasta que este bit es puesto en condición inicial o hasta que un RESET\ externo (pin 28) ocurre. Después de que este bit es activado, IRQ (pin 23) se activa y toda la lógica interna y los registros de control son puestos en condición inicial (excepto para el retenedor de interrupción y el bit ACTIVA RST\). Escribiendo un "0" a este bit se desactiva la señal RST\. Leyendo este registro simplemente refleja el estado del bit.

**Bit 6. Arbitraje en progreso (lectura) (AIP).**

Este bit es usado para determinar si el arbitraje está en progreso. Para que este bit se active, el bit de ARBITRAJE (puerto 2 bit 0) debe estar previamente activado. Indica que una condición de canal libre ha sido detectada y que el circuito ha activado BSY\ (pin 13) y ha puesto en el canal SCSI el contenido del registro de salida de datos (puerto 0). AIP permanecerá activo hasta que el bit ARBITRAJE sea puesto en condición inicial.

**Bit 6. Modo Prueba (escritura).**

Este bit puede ser escrito durante un ambiente de prueba para deshabilitar todos los manejadores de salida, removiendo, efectivamente, el NCR-53C80 del circuito. Poniendo en condición inicial este bit se regresa a operación normal.

**Bit 5. Pérdida de arbitraje (LA) (lectura/escritura).**

Este bit, activo, indica que el NCR-53C80 después de detectar una condición de canal libre, arbitrar para usar el canal activando BSY\ (pin 13) y poner su identificador en el canal de datos, ha perdido el arbitraje, debido a que SEL\ (pin 12) ha sido activado por otro dispositivo en el canal. Para que este bit se active, el bit de ARBITRAJE (puerto 2 bit 0) debe estar activo. Este bit no es usado para escritura.

**Bit 4. Activa ACK\.**

Este bit es usado por el iniciador del canal para activar ACK\ (pin 14) en el canal SCSI. Para activar ACK\ el bit MODO OBJETIVO (puerto 2 bit 6) debe ser falso. Escribiendo un "0" a este bit se pone en condición inicial ACK\ en el canal SCSI. Leer este registro simplemente refleja el estado de este bit.

**Bit 3. Activa BSY\.**

Escribiendo un "1" en esta posición se activa BSY\ (pin 13) en el canal SCSI. Un "0" pone en condición inicial la señal. La activación de BSY\ indica la exitosa selección o reelección y poniendo en condición inicial este bit se crea una condición de desconexión del canal. Leer este registro simplemente refleja el estado de este bit.

**Bit 2. Activa SEL\.**

Escribiendo un "1" en esta posición se activa SEL\ (pin 12) en el canal SCSI. SEL\ es normalmente activada después de que el arbitraje ha sido exitosamente completado. SEL\ puede ser desactivado poniendo en condición inicial este bit. Leer este registro simplemente refleja el estado de este bit.

**Bit 1. Activa ATN\.**

ATN\ (pin 15) es activado en el canal SCSI escribiendo en bit un "1", sólo si el bit MODO OBJETIVO (puerto 2 bit 6) es falso. ATN\ es normalmente activada por el iniciador para solicitar la fase de SALIDA DE MENSAJE (Message Out). Nótese que, dado que ACTIVA SEL\ y ACTIVA ATN\ están en el mismo registro, una selección con ATN\ puede ser implementada con un ciclo de escritura del MPU. ATN\ puede ser desactivado poniendo en condición inicial este bit. Leer este registro simplemente refleja el estado de este bit.

**Bit 0. ACTIVA Canal DE DATOS.**

Este bit, activado, permite que el contenido del registro de salida de datos sea habilitado como salidas del circuito en las señales DB0\ -DB7\. La paridad también es generada y activada en DBP\.

Cuando se conecta como iniciador, las salidas son habilitadas sólo si el bit MODO OBJETIVO (puerto 2 bit 6) es falso, la señal recibida I/O (pin 17) es falsa y las señales de fase (C/D, I/O, y MSG) coinciden con el contenido de ACTIVA C/D\, ACTIVA I/O\ Y ACTIVA MSG\ en el registro de comandos del objetivo.

Este bit debe estar activado también durante operaciones de envío con DMA.

**Registro de Modo. Dirección 2 (lectura/escritura)**

Este registro es usado para controlar la operación del circuito. Este registro determina: si el NCR 5380 opera como iniciador o como objetivo, si se están usando transferencias de DMA, si se está revisando paridad y si se están generando interrupciones por condiciones externas. Este registro puede ser leído para revisar el valor de estos bits internos de control. A continuación se describe la operación de cada bit de control.

MODO BLOQUE DMA	MODO OBJ.	HABIL. CHECA PARI.	HABIL. INT. PARI.	HABIL. INT. EOP.	MONIT. OCUPA- DO	MODO DMA	ARBI-TRAJE

**Bit 7. Modo de DMA a bloques.**

Este bit controla las características del handshake DRQ-DACK\ del DMA. Cuando el bit es puesto en condición inicial ("0") y el bit de MODO DMA está activo (1), se usa el handshake normal y el flanco de subida de DACK\ (pin 26) indica el final de cada transferencia de un byte. En el modo de operación a bloques, (el bit MODO DMA A BLOQUES en "1" y el bit MODO DMA en "1"), el final de IOR\ (pin 24) o IOW\ (pin 29) significa el final de cada transferencia de un byte y se le permite a DACK\ permanecer activa a través de las transferencias de DMA. READY (pin 25) puede ser usado para solicitar la siguiente transferencia.

**Bit 6. Modo Objetivo.**

Este bit permite al NCR-5380 operar ya sea como iniciador de canal SCSI (bit en "0") o como dispositivo objetivo del canal (bit en "1"). Para activar las señales ATN\ (pin 15) y ACK\ (pin 14) en el canal SCSI, el bit MODO OBJETIVO debe ser puesto en condición inicial ("0"). Para activar las señales C/D, I/O y MSG en el canal SCSI, el bit MODO OBJETIVO debe ser "1".

**Bit 5. Habilitador de chequeo de paridad.**

Determina si los errores de paridad se van a ignorar o a retener. En "0" la paridad será ignorada. Si es "1", los errores de paridad serán retenidos.

**Bit 4. Habilitador de interrupción de paridad.**

Cuando es "1", causará una interrupción (IRQ) sólo si se detecta un error de paridad. La interrupción de paridad se generará sólo si el bit HABILITA CHEQUEO DE PARIDAD (bit 5) está también habilitado ("1").

**Bit 3. Habilita la interrupción de fin de proceso (EOP).**

Cuando es "1", causa que al recibir la señal de fin de proceso (EOP pin 27) de la lógica del controlador de DMA, ocurra una interrupción.

**Bit 2. Monitor de BSY.**

Cuando es verdadero ("1"), causa que se genere una interrupción si hay una pérdida inesperada de la señal BSY\ (pin 13). Cuando la interrupción es generada por la pérdida de BSY\, los 6 bits menos significativos del registro de comandos del iniciador son puestos en sus condiciones iniciales ("0") y todas las señales son removidas del canal SCSI.

**Bit 1. Modo DMA.**

Es normalmente usado para habilitar transferencias DMA y debe ser activado ("1") antes de escribir a los puertos 5 a 7. Los puertos 5 a 7 son usados para iniciar transferencias de DMA. El bit MODO OBJETIVO (bit 6 puerto 2) debe ser consistente con las escrituras a los puertos 6 y 7 (p.e. activado ("1") para escribir al puerto 6 y en condición inicial ("0") para escribir al puerto 7). El bit de control Canal DE DATOS ACTIVO (puerto 1 bit 0) debe ser verdadero ("1") para todas las operaciones de envío por DMA. En modo de DMA, REQ\ (pin 20) y ACK\ (pin 14) son automáticamente controlados.

El bit MODO DMA es puesto en su condición inicial hasta la recepción de la señal EOP\. Cualquier transferencia de DMA puede ser detenida escribiendo un "0" a esta localidad, sin embargo, se debe tener cuidado de no causar un activación simultánea de CS\ y DACK\.

Nota: BSY\ debe estar activa para poder activar el bit de modo DMA.

**Bit 0. Arbitraje.**

Es activado para iniciar el proceso de arbitraje. Antes de activar este bit, el registro de salida de datos debe contener el valor del identificador (ID) de dispositivo SCSI. Sólo un bit de los datos debe estar activo para el arbitraje del canal. El NCR 5380 esperará por una condición de canal libre antes de entrar a la fase de arbitraje. El resultado de la fase de arbitraje puede ser determinado leyendo los bits de estados LA y AIP (puerto 1 bits 5 y 6 respectivamente).

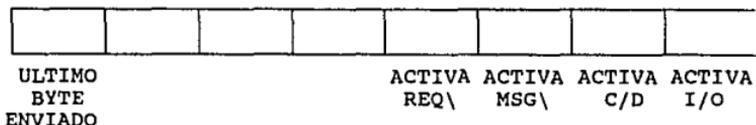
Registro de comandos del objetivo. Dirección 3 (lectura/escritura)

Cuando se conecta como dispositivo objetivo permite al MPU controlar las fases de transferencia de información del canal SCSI y/o activar REQ\ (pin 20) simplemente al escribir en este registro. El bit MODO OBJETIVO (puerto 2 bit 6) debe ser verdadero ("1") para que ocurra la activación del canal. Las fase del canal SCSI se describen en la siguiente tabla:

Fase del bus	ACTIVA I/O	ACTIVA C/D	ACTIVA MSG
Salida de datos	0	0	0
No especificada	0	0	1
Comandos	0	1	0
Salida de mensajes	0	1	1
Entrada de datos	1	0	0
No especificada	1	0	1
Estados	1	1	0
Entrada de mensajes	1	1	1

Tabla 4.2.2 Fases del canal SCSI.

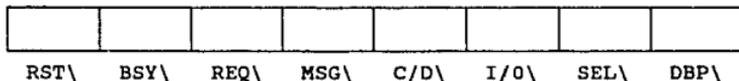
Cuando se conecta como iniciador con el bit MODO DMA verdadero, si las líneas (I/O, C/D MSG) no coinciden con los bits de fase del registro de comandos del objetivo, se generará una interrupción de no coincidencia de fases cuando REQ\ (pin 20) se active. Para poder enviar datos como iniciador, los bits ACTIVA I/O, ACTIVA C/D Y ACTIVA MSG deben coincidir con los correspondientes bits en el registro de estados del canal SCSI actual (puerto 4). El bit ACTIVA REQ\ (bit 3) no tiene significado cuando se opera como iniciador.



El NCR 53C80 usa el bit 7 de este registro para determinar cuando el último byte de transferencia DMA es enviado al canal SCSI. Esta bandera es necesaria ya que el bit fin de DMA en el registro Canal y Estados sólo refleja cuando el último byte fue recibido del DMA.

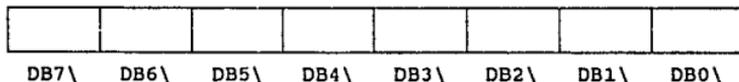
**Registro de estados del canal SCSI actual. Dirección 4. (sólo lectura)**

Este registro es usado para monitorear 7 señales de control del canal SCSI más el bit de paridad del canal. Por ejemplo, un dispositivo iniciador puede utilizar este registro para determinar la fase de canal actual y para poll REQ\ para transferencias de datos pendientes. Este registro también puede ser usado para determinar porqué ocurrió una interrupción en particular.



**Registro habilitador de selección. Dirección 4 (sólo escritura)**

Este registro es usado como máscara para monitorear un sólo identificador (ID) durante el intento de selección. La ocurrencia simultánea del bit ID correcto, BSY\ falso y SEL\ verdadero causará una interrupción. Esta interrupción puede ser deshabilitada poniendo en condiciones iniciales todos los bits en este registro. Si el bit de HABILITA CHEQUEO DE PARIDAD (puerto 2 bit 5) está activo, la paridad será revisada durante la selección.



**Registro de estados y canal. Dirección 5 (sólo lectura)**

Este registro puede ser usado para monitorear las señales restantes de control SCSI no encontradas en el registro Estados del canal SCSI actual (ATN\ y ACK\), así como otros seis bits de estados.

FIN DE DMA	SOLIC. DMA	ERROR PARI.	SOLIC. INT. ACTIVO	COINCI DE LA FASE	ERROR OCUPA DO	ATN\ DO	ACK\ DO

#### Bit 7. Fin de transferencia de DMA.

Este bit es activado si EOP\ (pin 27), DACK\ (pin 26) y, ya sea IOR\ (pin 24) o IOW\ (pin 29) están activos simultáneamente por un tiempo mínimo de 100 ns. Dado que EOP\ puede ocurrir durante el envío del último byte al registro de salida de datos (puerto 0), las señales REQ\ y ACK\ deben ser monitoreadas para asegurar que el último byte ha sido transferido. Este bit es puesto en su condición inicial cuando el bit MODO DMA es puesto en su condición inicial en el registro de modo (puerto 2).

#### Bit 6. Solicitud de DMA.

Este bit permite al MPU muestrear el pin de salida DRQ (Pin 22). DRQ puede ser limpiado activando DACK\ (pin 26) o poniendo en su condición inicial el bit MODO DMA (bit 1) en el registro de modo (puerto 2). La señal DRQ no se pone en condición inicial cuando ocurre una interrupción de error de fases.

#### Bit 5. Error de paridad.

Este bit es activado si ocurre un error de paridad durante la recepción de datos o en la selección de un dispositivo. Este bit puede ser activado sólo si el bit HABILITA CHEQUEO DE PARIDAD está activo (puerto 2 bit 5 en "1"). Este bit puede ser limpiado leyendo el registro de Establece (initialize) paridad/interrupción (puerto 7).

#### Bit 4. Activa solicitud de interrupción.

Este bit es activado si ocurre una condición de interrupción habilitada. Refleja el estado actual de la salida IRQ (pin 23) y puede ser limpiado leyendo el registro de Establece (Initialize) paridad/interrupción (puerto 7).

#### Bit 3. Coincidencia de fase.

Las señales SCSI MSG\, C/D\ e I/O\ (pines 19, 18 y 17) representan la fase actual de transferencia de información. Este bit indica si los la fase de canal SCSI actual coincide con los tres bits menos significativos del registro de comandos del objetivo. Este bit es continuamente actualizado y sólo es significativo cuando se opera como iniciador. Es necesario que coincidan las fases para que se realice la transferencia en el canal SCSI.

**Bit 2. Error en BSY.**

Este bit se activa si ha ocurrido una pérdida inesperada de la señal BSY\ (pin 13). Este retenedor sensible a nivel es activado siempre que el bit MONITOR DE BUSY (puerto 2 bit 2) es verdadero y BSY\ es falso. Una pérdida inesperada de BSY\ deshabilitará todas las salidas SCSI y pondrá en condiciones iniciales el bit MODO DMA (puerto 2 bit 1).

**Bit 1. Atención.**

Este bit refleja la condición de la señal de control ATN\ (pin 15). Esta señal normalmente es monitoreada por el objetivo.

**Bit 0. ACK (reconocimiento).**

Este bit refleja la condición de la señal de control ACK\ (pin 14). Esta señal normalmente es monitoreada por el objetivo.

**Registros de DMA**

Tres registros de sólo escritura son usados para iniciar todas las actividades de DMA. Estos son Inicia envío de DMA (puerto 5), Comienza el objetivo recepción de DMA (puerto 6) e Comienza el iniciador recepción de DMA (puerto 7). Simplemente escribiendo estos registros se inicia la transferencia de DMA. Los datos presentados al NCR 5380 en las señales D0-D7 durante la escritura del registro no tienen significado y no tiene efecto en la operación. Antes de escribir en estos registros el bit MODO DE DMA A BLOQUES (bit 7), el bit MODO DMA (bit 1) y el bit MODO OBJETIVO (bit 6) en el registro de modo (puerto 2) deben estar apropiadamente activados.

**Inicia envío DMA. Dirección 5 (sólo escritura)**

Al escribir en este registro se inicia el envío de DMA, desde el DMA al canal SCSI, tanto para papel de iniciador como para el de objetivo. El bit MODO DMA (puerto 2 bit 1) debe estar activado antes de escribir a este registro.

**Inicia recepción de DMA objetivo. Dirección 6 (sólo escritura)**

Al escribir en este registro se inicia la recepción de datos, desde el canal SCSI al DMA, sólo para el papel de objetivo. El bit MODO DMA (bit 1) y el bit MODO OBJETIVO (bit 6) en el registro de modo (puerto 2) debe estar ambos activados ("1") antes de escribir a este registro.

Inicia recepción de DMA iniciador. Dirección 7 (sólo escritura)

Al escribir en este registro se inicia la recepción de datos, desde el canal SCSI al DMA, sólo para el papel de iniciador. El bit MODO DMA (bit 1) debe ser verdadero ("1") y el bit MODO OBJETIVO (bit 6) debe ser falso ("0") en el registro de modo (puerto 2) antes de escribir a este registro.

Establece (Initialize) Paridad/Interrupción. Dirección 7 (sólo lectura)

leyendo este registro se pone en condición inicial el bit ERROR DE PARIDAD (bit 5), el bit SOLICITUD DE INTERRUPCIÓN (bit 4) y el bit ERROR DE BUSY (bit 2) en el reg. Canal y Estados (pto 5).

#### 4.2.4 Soporte de hardware en el circuito

El 53C80 es fácil de usar por su simple arquitectura. Este circuito permite control y monitoreo directo del canal SCSI ya que está provisto de un retenedor para cada señal. Sin embargo, parte de los tiempos definidos en el protocolo son muy rápidos para ser controlados por microprocesadores tradicionales. Por esto, se ha provisto de soporte de hardware para transferencias DMA, arbitraje, monitoreo de cambio de fase, desconexión, reset de canal, selección/reselección de dispositivo y generación y verificación de paridad.

El arbitraje se realiza usando un filtro de canal-libre para continuamente monitorear BSY/. Si BSY/ permanece inactivo por lo menos 400 ns, entonces el canal SCSI es considerado como libre y el arbitraje puede comenzar. El arbitraje comenzará sólo si el canal está libre, SEL/ está inactivo y el bit ARBITRAJE (puerto 2, bit 0) está activo. Una vez que el arbitraje ha comenzado (BSY/ activo), se presenta un retardo de arbitraje de 2.2 us antes de que el canal sea examinado para determinar si se ha ganado el arbitraje. Este retardo debe ser implementado en el software.

El 53C80 es un dispositivo sin reloj. Los retardos, como canal libre, estabilidad del canal y asentamiento del canal están implementados usando retardos puente. Estos retardos pueden diferir entre los dispositivos por variaciones inherentes del proceso, pero están dentro de la especificaciones propuestas por ANSI X3T9.2

#### 4.2.5 Interrupciones

El 53C80 provee una salida de interrupción (IRQ) para indicar que se ha completado una tarea a la ocurrencia de una condición anormal del canal. El uso de interrupciones es opcional y puede ser deshabilitado poniendo en condiciones iniciales los bits apropiados en el registro de MODO (puerto 2) o el registro habilitador de SELECCIÓN (puerto 4).

Cuando ocurre una interrupción, los registros Canal y Estados y Estados actuales del canal SCSI deben ser leídos para determinar que condición provocó la interrupción. IRQ (pin 23) puede ser inicializado simplemente leyendo el registro Initialize Parity/Interrupt (puerto 7) o por una condición externa del circuito (RESET/ activo por 200 ns).

Asumiendo que el 53C80 ha sido apropiadamente programado, una interrupción será generada si: el circuito es seleccionado o reseleccionado, si una señal de EOP/ ocurre durante transferencias DMA, si ocurre una condición inicial del canal SCSI, si ocurre un error de paridad durante una transferencia de datos, si no coinciden las fases o si ocurre una desconexión del canal SCSI.

#### Selección/Reselección

El NCR 53C80 puede generar una interrupción de selección si SEL\ (pin 12) es verdadero ("1"), su ID es verdadero ("1") y BSY\ (pin 13) es falso ("0") por lo menos 400 ns (bus settle delay). Si el pin I/O (terminal 17) está activo, se debe considerar una interrupción de reselección. El bit ID correcto es determinado si coincide con el registro habilitador de selección (puerto 4). Sólo un bit que coincida es necesario para generar la interrupción. Esta interrupción puede ser deshabilitada escribiendo ceros en todos los bits del registro habilitador de selección (puerto 4).

Si se tiene soportada la paridad, también ésta debe ser correcta durante la fase de selección. Sin embargo, si el bit HABILITA BIT DE PARIDAD (puerto 2 bit 5) está activo, el bit ERROR DE PARIDAD debe ser revisado para asegurar que la selección ha sido apropiada. El bit HABILITA INTERRUPCION DE PARIDAD necesita haber estado activo para que se genere esta interrupción.

La especificación SCSI propuesta también requiere que no más de 2 bits ID sean activos durante el proceso de selección. Para asegurar esto, se debe leer el registro de datos SCSI actuales (puerto 0).

Los valores adecuados para los registros Canal y estados (puerto 5) y Estado de canal SCSI actual (puerto 4) se muestran abajo.

Registro de canal y estados.

0	0	0	1	X	0	X	0
FIN DE DMA	SOLIC. DMA	ERROR PARI.	SOLIC. INT.	COINCI DE LA FASE	ERROR OCUPA DO	ATN\	ACK\

## Registro de estados del canal SCSI actual.

0	0	0	X	X	X	1	X
RST\	BSY\	REQ\	MSG\	C/D\	I/O\	SEL\	DBP\

## Interrupción de fin de proceso (EOP)

Esta señal (EOP\ terminal 27), que puede ocurrir durante transferencias DMA (bit MODO DMA verdadero), pondrá en "1" el bit de estado FIN DE DMA (puerto 5 bit 7) y opcionalmente generará una interrupción si el bit HABILITA INTERRUPCION DE FIN DE PROCESO (puerto 2 bit 3) es verdadero. El pulso de EOP\ no será reconocido (activación del bit FIN DE DMA) a menos que EOP\, DACK\ y, ya sea IOR\ o IOW\ sean concurrentemente activos por lo menos 100 ns. Las transferencias de DMA puede ocurrir aún si EOP\ no fue activada al tiempo correcto. Esta interrupción puede ser deshabilitada poniendo en condición inicial el bit HABILITA INTERRUPCION DE FIN DE PROCESO.

Los valores adecuados para los registros Canal y estados (puerto 5) y Estado de canal SCSI actual (puerto 4) para esta interrupción se muestran abajo.

## Registro de canal y estados.

1	0	0	1	0	0	0	X
FIN DE DMA	SOLIC. DMA	ERROR PARI.	SOLIC. INT. DE LA ACTIVO	COINCI. DE LA FASE	ERROR DE LA OCUPA DO	ATN\	ACK\

## Registro de estados del canal SCSI actual.

0	1	X	X	X	X	0	X
RST\	BSY\	REQ\	MSG\	C/D\	I/O\	SEL\	DBP\

El bit FIN DE DMA es usado para determinar cuando una transferencia de bloque se completó. Las operaciones de recepción se completan cuando no quedan más datos en el circuito y no ocurren handshakes adicionales. La única excepción a esto, es la recepción de datos como iniciador y el objetivo opta por enviar datos adicionales para la misma fase. En este caso, REQ\ se activa y el nuevo dato está presente en el registro de entrada de datos. Dado que no ocurrirá una interrupción por diferencia de fase, REQ\ y ACK\ deben ser muestreadas para determinar que el objetivo está intentando enviar más datos.

Para operaciones de envío, el bit FIN DE DMA es activado cuando el DMA termina su transferencia, pero la transferencia SCSI puede estar aún en progreso. Si está conectado como objetivo, REQ\ y ACK\ deben ser muestreadas hasta que ambas sean falsas. Si está conectado como iniciador, la interrupción de cambio de fase puede ser usada para señalar la conclusión de la fase previa. Es posible para el objetivo solicitar datos adicionales para la misma fase. En este caso, no ocurrirá un cambio de fase y REQ\ y ACK\ deben ser muestreados para determinar cuando fue transferido el último byte.

En el NCR 53C80, el bit ULTIMO BYTE ENVIADO (bit 7 del registro de comandos de objetivo) puede ser muestreado para determinar cuando el último byte ha sido transferido.

#### Condiciones iniciales del canal SCSI

El NCR 53C80 genera una interrupción cuando la señal RST\ (terminal 16) cambia a verdadero. El dispositivo suelta todas las señales dentro de un tiempo de 800 ns (bus clear delay) de esta transición. Esta interrupción también puede ocurrir después de poner el bit ACTIVA RST\ (puerto 1 bit 7). Esta interrupción no puede ser deshabilitada. (Nota: la señal RST\ no es retenida en el bit 7 del registro estados del canal SCSI actual y no puede estar activa cuando se lee este puerto. Para este caso, la interrupción Reset de Canal puede ser determinada por default).

Los valores adecuados para los registros Canal y estados (puerto 5) y Estado de canal SCSI actual (puerto 4) para esta interrupción se muestran abajo.

#### Registro de canal y estados.

0	0	0	1	X	0	0	0
FIN DE DMA	SOLIC. DMA	ERROR PARI.	SOLIC. INT. ACTIVO	COINCI DE LA FASE	ERROR OCUPA DO	ATN\	ACK\

#### Registro de estados del canal SCSI actual.

X	0	0	0	0	0	0	0
RST\	BSY\	REQ\	MSG\	C/D\	I/O\	SEL\	DBP\

### Error de paridad

Una interrupción es generada si se recibe un error de paridad y si el bit HABILITA CHEQUEO DE PARIDAD (bit 5) y el bit HABILITA INTERRUPTCIÓN DE PARIDAD (bit 4) son activados en el registro de modo (puerto 2). La paridad es revisada durante la lectura del registro de datos SCSI actual (puerto 4) y durante operaciones de recepción de DMA. Un error de paridad puede ser detectado sin generar una interrupción, deshabilitando el bit HABILITA INTERRUPTCIÓN DE PARIDAD y revisando la bandera de ERROR DE PARIDAD (puerto 5 bit 5).

Los valores adecuados para los registros Canal y estados (puerto 5) y Estado de canal SCSI actual (puerto 4) se muestran abajo.

#### Registro de canal y estados.

0	X	1	1	1	0	X	X
FIN DE DMA	SOLIC. DMA	ERROR PARI.	SOLIC. INT.	COINCI DE LA ACTIVO	ERROR DE LA OCUPA DO	ATN\	ACK\

#### Registro de estados del canal SCSI actual.

0	1	X	X	X	X	0	X
RST\	BSY\	REQ\	MSG\	C/D\	I/O\	SEL\	DBP\

### No coincidencia de fase de canal

Las líneas de fase del canal SCSI están comprendidas por las señales I/O\, C/D\ Y MSG\. Estas señales son comparadas con los bits correspondientes en el registro de comandos del objetivo: ACTIVA I/O\ (bit 0), ACTIVA C/D\ (bit 1) y ACTIVA MSG\ (bit 2). La comparación se hace continuamente y se refleja en el bit COINCIDE FASE (puerto 5 bit 3). Si el bit MODO DMA (puerto 2 bit 1) está activo y ocurre una diferencia de fase cuando REQ\ (terminal 20) cambia de falso a verdadero, se generará una interrupción (IRQ).

Una diferencia de fase previene el reconocimiento de REQ\ y remueve el circuito del canal durante operaciones de envío de un iniciador (las líneas DBO\-DB7\, DBP\ no serán manejados aún cuando el bit ACTIVA Canal DE DATOS (puerto 1 bit 0) está activo). Esta interrupción sólo tiene sentido cuando se conecta como iniciador y puede ser deshabilitada poniendo en condición inicial el bit MODO DMA (nota: es posible que esta interrupción ocurra cuando se conecta como objetivo, si otro dispositivo está llevando las líneas de fase a un estado diferente). Los valores adecuados para los registros Canal y estados (puerto 5) y Estado de canal SCSI actual (puerto 4) se muestran abajo.

## Registro de canal y estados.

0	0	0	1	0	0	X	0
FIN DE DMA	SOLIC. DMA	ERROR PARI.	SOLIC. INT.	COINCI. DE LA	ERROR OCUPA	ATN\ DO	ACK\ DO

## Registro de estados del canal SCSI actual.

0	1	1	X	X	X	0	X
RST\ DO	BSY\ DO	REQ\ DO	MSG\ DO	C/D\ DO	I/O\ DO	SEL\ DO	DBP\ DO

Pérdida de BSY\  
DO

Si el bit MONITOR DE BUSY (puerto 2 bit 2) está activo, se generará una interrupción si la señal BUSY (terminal 13) cambia a falso por lo menos durante 400 ns (bus settle delay). Esta interrupción puede ser deshabilitada poniendo en condición inicial el bit MONITOR DE BUSY.

Los valores adecuados para los registros Canal y estados (puerto 5) y Estado de canal SCSI actual (puerto 4) se muestran abajo.

## Registro de canal y estados.

0	0	0	1	X	1	0	0
FIN DE DMA	SOLIC. DMA	ERROR PARI.	SOLIC. INT.	COINCI. DE LA	ERROR OCUPA	ATN\ DO	ACK\ DO

## Registro de estados del canal SCSI actual.

0	0	0	0	0	0	0	0
RST\ DO	BSY\ DO	REQ\ DO	MSG\ DO	C/D\ DO	I/O\ DO	SEL\ DO	DBP\ DO

## Condiciones para establecer inicio

Existen tres posibles situaciones de poner en condiciones iniciales:

Condiciones iniciales por Hardware del circuito

Cuando la señal de RESET\  
DO (terminal 28) está activa por lo menos durante 200 ns, el dispositivo NCR 53C80 es reestablecido y todos los registros de control y lógica interna son limpiados. Es una puesta en condiciones iniciales del circuito y no crea una condición de inicio en el canal SCSI.

### Recepción de la señal RST\ del canal SCSI

Cuando se recibe la señal RST\ (terminal 16) una interrupción es generada y provoca una puesta en condiciones iniciales del circuito. Toda la lógica interna y los registros son limpiados, excepto el retenedor de la interrupción IRQ y el bit ACTIVA RST\ (puerto 1 bit 7). (Nota: la señal RST\ puede ser muestreada leyendo el puerto 4, sin embargo, esta señal no es retenida y puede no estar presente cuando se realice la lectura.

### Envío de la señal RST\ al canal SCSI

Si el CPU activa el bit ACTIVA RST\ (puerto 1 bit 7), la señal RST\ (terminal 16) se activa en el canal SCSI y una puesta en condiciones iniciales interna se lleva a cabo. De nuevo, toda la lógica interna y los registros son limpiados, excepto el retenedor de la interrupción IRQ y el bit ACTIVA RST\ (puerto 1 bit 7). La señal RST\ continuará activa hasta que el bit ACTIVA RST\ sea limpiado o hasta que ocurra una de RESET por hardware.

### 4.2.7 Transferencias de datos

Los datos pueden ser transferidos entre los dispositivos SCSI de cuatro maneras diferentes: I/O programado, DMA normal, DMA en modo bloques y pseudo DMA. (nota: para todas las operaciones de transferencia, DACK\ y CS\ nunca deben estar activas simultáneamente).

#### Transferencias programadas I/O

Esta es la forma más primitiva de transferencia de datos. Para el handshake REQ/ACK, las señales REQ\ (terminal 20) y ACK\ (terminal 14) son individualmente monitoreadas y activadas leyendo y re-escribiendo en los bits de los registros apropiados. Este tipo de transferencia es usada normalmente cuando se transfieren pequeños bloques de comandos o mensajes y bytes de estado.

Una operación de envío de un iniciador puede comenzar activando adecuadamente los bits de C/D\, I/O\ y MSG\ en el registro de comandos del objetivo el para que exista una coincidencia de fase. Además de la coincidencia de fase, es necesario que el bit ACTIVA Canal DE DATOS (puerto 1 bit 0) sea verdadero y la señal I/O\ sea falsa para que el 53C80 envíe datos.

Para cada transferencia, el dato es cargado en el registro salida de datos (puerto 0). El MPU espera entonces que el bit REQ (puerto 4 bit 5) se active. Una vez que REQ\ se activa, el bit COINCIDENCIA DE FASE (puerto 5 bit 3) es revisado y el bit ACTIVA ACK\ (puerto 1 bit 4) se prende. El bit REQ\ es muestreado hasta que es falso y entonces el MPU pone en condición inicial el bit ACK\ para completar la transferencia.

### Modo DMA Normal

Este modo es usado normalmente en transferencias de bloques largos. El circuito SCSI manda una solicitud de DMA (DRQ terminal 22) siempre que está listo para transferir un byte. La lógica externa DMA usa la señal DRQ para generar los pulsos DACK\ e IOR\ o IOW\ para el 5380. DRQ se desactiva cuando DACK\ se activa y DACK\ se desactiva poco tiempo después del ancho del pulso de lectura o escritura. Este proceso es repetido para cada byte. Para este modo, a DACK\ no se le debe permitir ciclar a menos que una transferencia se esté realizando.

### Modo DMA a bloques

El modo de bloques permite que un dispositivo DMA externo (Intel 8237) realice transferencias de DMA secuenciales sin solicitar el canal de datos al CPU. Manteniendo DACK\ activo se previene de que los CPUs de Intel ganen el acceso al canal del sistema. El handshake por sí mismo no incrementa la velocidad de transferencia. Evitando que el CPU comparta el canal del sistema se incrementa la transferencia de DMA, pero también detiene la operación del CPU.

Las transferencias en modo DMA a bloques es soportado para ambos papeles de operación, como iniciador o como objetivo. Cuando se usa este modo de operación, DRQ es activado para señalar el inicio de una transferencia de DMA. En respuesta a DRQ, DACK es activado y permanece así a través de toda la transferencia. READY se activa después de que el pulso IOW o IOR se inactiva, reemplazando efectivamente la señal DRQ.

Se debe tener cuidado con la operación de READY cuando se usa este modo. Si, por ejemplo, ocurre una interrupción por no coincidencia de fases, READY permanecerá en estado inactivo e INT se activará. Para esta condición, el circuito de DMA debe regresar el control del canal al CPU para que la interrupción sea atendida. READY tampoco regresa a estado activo cuando se recibe el pulso de EOP. Sin embargo, tal vez se quiera usar EOP para asegurar que el CPU obtuvo el control del canal después de que el último byte de DMA ha sido transferido. Igual que en el modo de DMA no a bloques, la señal EOP no desactiva ACK en el canal SCSI. Para completar exitosamente la operación de envío de DMA se debe, o escribir un byte de dato adicional al 5380 para permitir que ACK se inactivo, que el CPU inicialice el bit MODO DMA en el registro de modo.

Las transferencias de DMA no a bloques terminan cuando DACK cambia a falso, como las transferencias de bloques terminan cuando el IOR\ o IOW\ se vuelven inactivas después de cada byte. Las transferencias de DMA pueden iniciar antes en transferencias de modo de bloques.

Para obtener un desarrollo óptimo en operación de modo bloques, la lógica de DMA puede opcionalmente usar el interlocking handshake del modo normal de DMA. READY está aún disponible para regular la transferencia del DMA, pero DRQ es 30 a 40 ns más rápida que READY y puede ser usada para iniciar el ciclo antes.

#### Modo pseudo DMA

Para evitar el tedio de monitorear y activar la señales del handshake REQ/ACK para transferencias programadas I/O, el sistema puede ser diseñado para usar un modo pseudo DMA. Este modo es implementado programando al NCR 53C80 para operar en modo DMA, pero usando el MPU para emular el handshake del DMA. DRQ (terminal 22) puede ser detectado polling el bit SOLICITUD DE DMA (puerto 5 bit 6), muestreando la señal a través de un puerto externo o usándolo para generar una interrupción al MPU. Una vez detectado el DRQ, el MPU puede llevar a cabo una lectura o escritura de puerto y tener una transferencia DMA.

Frecuentemente se necesita lógica externa de decodificación para genera la señal CS\ del NCR 5380. Esta misma lógica puede ser usada para generar DACK\ sin costo extra del sistema y permite un desarrollo mejorado en transferencias programadas I/O.

#### Deteniendo operaciones de DMA

La señal EOP\ no es el único modo de detener transferencias de DMA. Una diferencia de fases o la puesta en condición inicial del bit MODO DMA (puerto 2 bit 1) también puede terminar un ciclo DMA para la fase actual del canal.

#### Usando la señal EOP\

Si se usa EOP\, esta se debe activar por lo menos 100 ns mientras DACK\ e IOR\ o IOW\ son simultáneamente activas. Notar, sin embargo, que si IOR\ o IOW\ no está activa, se generará una interrupción, pero la actividad del DMA continuará. La señal EOP\ no pone en condición inicial el bit MODO DMA. Dado que la señal EOP\ puede ocurrir durante el envío del último byte al registro de salida de datos (puerto 0), las señales REQ\ y ACK\ deben ser monitoreadas para asegurar que el último byte ha sido transferido.

#### Interrupción por no coincidencia de fases

Si se opera como iniciador, una interrupción por no coincidencia de fases puede ser usada para detener la transferencia. Usando este método, se libera al anfitrión de mantener el contador de longitud de transferencia y libera la lógica del DMA proveyendo la señal EOP\. Si se lleva a cabo una operación de envío como iniciador, el NCR 5380 requiere que DACK\ se cicle antes que ACK\ se active. Dado que no pueden ocurrir cambios de fase si DACK está activo, para recibir la interrupción

de no coincidencia de fases, se debe, ya sea que se cicle DACK\ después de que el último byte es enviado o que el bit MODO DMA sea puesto en condición inicial.

#### Poniendo en condición inicial el bit MODO DMA

Una operación de DMA puede ser detenida en cualquier momento poniendo en condición inicial el bit MODO DMA. Se recomienda poner así el bit MODO DMA después de recibir la señal EOP\ o la interrupción de no coincidencia de fase. El bit MODO DMA debe entonces ser activado antes de escribir cualquiera de los registros de inicio de DMA para las fases del canal subsecuentes.

En operaciones bajo el papel de objetivo, si se usa el residuo del bit MODO DMA en lugar de EOP\, se debe tener cuidado de poner en condición inicial el bit en el momento adecuado. Si se reciben datos como dispositivo objetivo, el bit MODO DMA se debe poner en condición inicial una vez que el último DRQ es recibido y antes de que DACK\ es activado, para prevenir que ocurran REQ\ adicionales. Poniendo en condición inicial este bit se causa que DRQ se inactíve. Sin embargo, el último byte recibido permanece en el registro de entrada de datos y puede ser obtenido ya sea realizando un ciclo de lectura del MPU en modo normal o ciclando DACK\ e IOR\. En la mayoría de los casos, es más fácil utilizar EOP\ cuando se opera como dispositivo objetivo.

### 4.3. MEMORIA RAM DOBLE PUERTO

#### 4.3.1 Descripción de la memoria estática de doble puerto CY-131

Este dispositivo tiene dos puertos independientes llamados puerto derecho y puerto izquierdo. Cada puerto consiste de un canal de datos de 8 bits bidireccionales, un canal de entrada de direcciones de 10 bits y las señales de control necesarias.

Tiene un árbitro incluido para resolver contención entre los dos puertos. Cuando ocurre una contención entre los puertos, un puerto recibe la prioridad mientras que el otro recibe una señal de busy (ocupado).

También tiene facilidades incluidas en el circuito para soportar semáforos. Las direcciones 3FEh y 3FFh sirven como generadores de interrupción. Si un dato se escribe en la dirección 3FFh del puerto izquierdo, una señal de interrupción se activa para el puerto derecho. La señal de interrupción es desactivada leyendo en el puerto derecho la misma dirección. La dirección 3FEh es usada de manera similar por el puerto derecho para activar la señal de interrupción para el puerto izquierdo.

Cuenta con dos habilitadores de circuito, cada uno para el puerto correspondiente. Antes de que alguna transacción en un puerto tenga lugar, su señal de habilitador de circuito correspondiente debe ser activada. Si no se activa la señal de habilitación, la circuitería correspondiente a ese lado automáticamente se apaga y queda en modo de espera (*standby*).

El circuito está en empaquetado 52 pin chip carrier. Todas las entradas y salidas son compatibles TTL y el dispositivo opera con una alimentación de +5 volts.

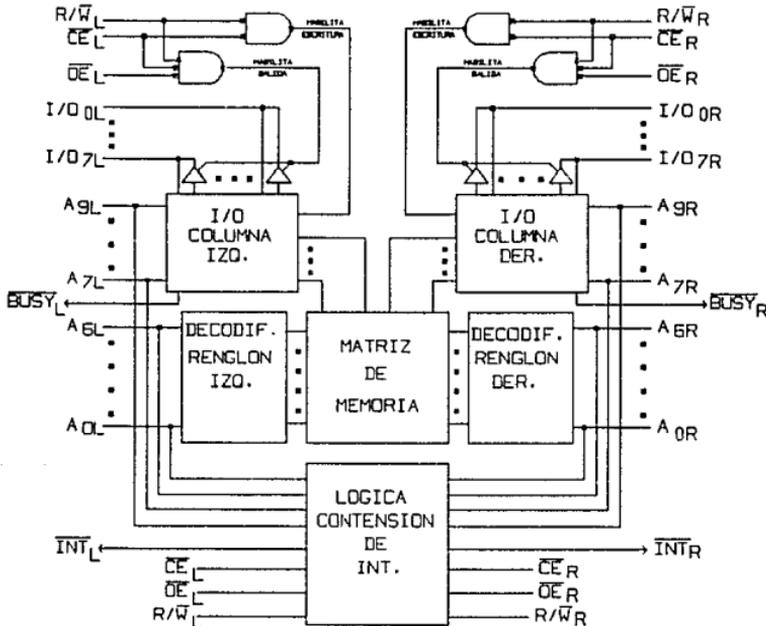


Figura 4.3.1 Diagrama a bloques de la RAM estática de doble puerto CY-131.

## 4.3.2 Descripción de terminales

A0L-A9L Puerto de direcciones izquierdo (entradas).

A0R-A9R Puerto de direcciones derecho (entradas).

BUSY\L (\R) Bandera de busy del puerto izquierdo (\derecho) (salida, colector abierto).

Esta salida de colector abierto requiere una resistencia de pull-up para la correcta operación. Un nivel bajo en esta salida indica que la lógica de arbitraje interna del circuito detectó contención entre los puertos y que al puerto derecho (\izquierdo) se le dio prioridad. Todas la señales del puerto izquierdo (\derecho) deben mantenerse estables hasta que esta salida indique un nivel alto.

La generación de esta señal es una función lógica de los puertos de direcciones derecho e izquierdo y las entradas CE\L y CE\R. El comportamiento del transitorio de la salida BUSY\L (\R) no se asegura mientras las entradas estén cambiando.

CE\L (\R) Habilitador del puerto izquierdo (\derecho) (entrada).

Esta entrada debe estar en nivel bajo antes que ninguna transacción en el puerto izquierdo (\derecho) y debe permanecer baja lo que dure la transacción. Cuando esta señal está en alto, los circuitos lógicos del puerto izquierdo (\derecho) entran a un modo de espera (standby) y permanecen en este modo mientras esta entrada permanece en alto. Debe notarse que desconectando el puerto izquierdo (\derecho) en modo standby no afecta las salidas INT\L e INT\R. Esta entrada al subir a nivel alto, también pone en condición inicial el retenedor interno de arbitraje. Se recomienda que CE\L (CE\R) pase a alto después de completar una transacción.

I/O 0-7L (0-7R) Canal izquierdo (\derecho) de entrada y salida de datos (entrada, salida y tercer estado).

INT\L Bandera de interrupción del puerto izquierdo (salida colector abierto).

Esta salida de colector abierto requiere una resistencia de pull-up para su correcta operación. Si el puerto derecho lleva cabo cualquier operación de escritura usando la dirección 3FEh, entonces esta salida cambia a nivel bajo. Permanecerá en este nivel hasta que el puerto izquierdo complete exitosamente una operación de lectura usando la dirección 3FEh. Se debe notar que el desconectar los puertos no tienen efecto sobre esta línea.

INT\R Bandera de interrupción del puerto derecho (salida colector abierto).

Esta salida de colector abierto requiere una resistencia de pull-up para su correcta operación. Si el puerto izquierdo lleva a cabo cualquier operación de escritura usando la dirección 3FFh, entonces, esta salida cambia a nivel bajo. Permanecerá en este nivel hasta que el puerto derecho complete exitosamente una operación de lectura usando la dirección 3FFh. Se debe notar que desconectar los puertos no tiene efecto sobre esta línea.

OE\L Habilitador de salida de datos del puerto izquierdo (entrada).

Esta señal no tiene efecto en las señales BUSY\L, BUSY\R, INT\L o INT\R. Sin embargo, los manejadores del puerto I/O izquierdo son deshabilitados cuando la entrada R/W\L cambia a nivel alto (operación de escritura). Se recomienda que la señal OE\L se mantenga en alto durante las operaciones de escritura al puerto izquierdo.

OE\R Habilitador de salida de datos del puerto derecho (entrada).

Esta señal no tiene efecto en las señales BUSY\L, BUSY\R, INT\L o INT\R. Sin embargo, los manejadores del puerto I/O derecho son deshabilitados cuando la entrada R/W\R cambia a nivel alto (operación de escritura). Se recomienda que la señal OE\R se mantenga en alto durante las operaciones de escritura al puerto derecho.

R/W\L Habilitador de lectura/escritura\ del puerto izquierdo (entrada).

Esta entrada es usada para especificar la función que el puerto izquierdo llevará a cabo. Nivel alto indica lectura y nivel bajo indica escritura.

Cuando CE\L es bajo, OE\L es bajo y R/W\L es alto, los datos de las localidades direccionadas por A0-A9L estarán disponibles en las líneas I/O0-I/O7L. Como ya se mencionó, leyendo del puerto izquierdo en la localidad 3FEh deshabilita la salida INT\L.

Cuando CE\L es bajo y R/W\L cambia a nivel bajo, los datos presentes en las líneas I/O0-I/O7L serán escritos en la localidad direccionada por A0-A9L. Se debe notar que la operación de escritura no es afectada por la entrada OE\L. Sin embargo, se recomienda que la entrada OE\L se mantenga en alto durante operaciones de escritura. Como se mencionó anteriormente, llevando a cabo una operación de escritura en el puerto izquierdo en la localidad 3FFh causa que la salida INT\R cambie a bajo.

R/W\R Habilitador de lectura/escritura\ del puerto derecho (entrada).

Esta entrada es usada para especificar la función que el puerto derecho llevará a cabo. Nivel alto indica lectura y nivel bajo indica escritura.

Cuando CE\R es bajo, OE\R es bajo y R/W\R es alto, los datos de las localidades direccionadas por A0-A9R estarán disponibles en las líneas I/00-I/07R. Como ya se mencionó, leyendo del puerto derecho en la localidad 3FEh deshabilita la salida INT\R.

Cuando CE\R es bajo y R/W\R cambia a nivel bajo, los datos presentes en las líneas I/00-I/07R serán escritos en la localidad direccionada por A0-A9R. Se debe notar que la operación de escritura no es afectada por la entrada OE\R. Sin embargo, se recomienda que la entrada OE\R se mantenga en alto durante operaciones de escritura. Como se mencionó anteriormente, llevando a cabo una operación de escritura en el puerto derecho en la localidad 3FFh causa que la salida INT\L cambie a bajo.

Vcc +5 volts suministro de poder.

#### 4.3.3 Descripción funcional

Este dispositivo consiste de un arreglo de memoria con dos arreglos de decodificadores de direcciones y lógica asociada. Este arreglo permite el acceso de cada palabra en el arreglo de memoria desde dos fuentes independientes. Estas fuentes son llamadas lado (o puerto) izquierdo y lado (o puerto) derecho. Los datos accedidos por las entradas de direccionamiento del lado izquierdo aparecen en las líneas de datos del lado izquierdo del arreglo y están conectados a las terminales I/O del lado izquierdo a través de los buffers de tres estados asociados. La señal de control de habilitación para estos buffers es generada usando las entradas R/W\L, CE\L y OE\L. Si los buffers de I/O están deshabilitados en el circuito, las terminales I/O pueden ser usados como entradas. El dato a ser escrito desde el puerto izquierdo está presente en esta entrada. La escritura en el arreglo de memoria desde el lado izquierdo está controlado por la señal habilitadora de escritura del lado izquierdo, generada en el circuito usando las entradas R/W\L y CE\L. Un arreglo idéntico existe para el lado derecho. Además, hay una lógica de arbitraje en el circuito para dar prioridad a un puerto sobre el otro en caso de contención, una lógica de bandera de interrupción.

#### 4.3.4 Arbitraje de contención

La memoria de doble puerto provee la facilidad de dos accesos independientes para eliminar la interferencia física entre señales. Sin embargo, existen dos posibilidades significantes de interferencia "lógica" que no son tolerables: cuando un puerto está leyendo de una localidad mientras que el otro puerto está escribiendo a la misma localidad al mismo tiempo. En este caso, el dato recibido por el puerto que lee no puede no ser predecible. De manera similar considere la situación de que los dos puertos escriben información en la misma localidad simultáneamente. El dato resultante que queda finalmente en la memoria puede no ser válido. Estas dos situaciones son llamadas normalmente contención. El CY-131 tiene una lógica en circuito para detectar contención y dar prioridad a un puerto sobre el otro. En una verdadera memoria de doble puerto, lecturas simultáneas de ambos puertos a la misma dirección no corrompen los datos. Así, puede ser construido que no ocurra contención. Sin embargo, por simplicidad, y compatibilidad con los estándares de las prácticas industriales, el arbitraje del CY-131 está basado puramente en direcciones. Con esto, en el caso de lecturas simultáneas de ambos puertos a la misma dirección, la lógica de arbitraje sentirá contención y le dará prioridad a uno de los dos puertos. El otro puerto recibirá una indicación de busy.

La figura 4.3.2 es un diagrama lógico conceptual de lógica de arbitraje de contención. Consiste de dos comparadores de igualdad. El comparador izquierdo compara las entradas de direcciones del lado izquierdo contra la versión retardada de la dirección del puerto derecho. De manera similar, el comparador derecho compara las direcciones del puerto derecho contra la versión retardada de las direcciones del puerto izquierdo. La salida del comparador está conectada a un retenedor formado por dos compuertas NAND acopladas en forma cruzada. Las señales habilitadoras de circuito, CE\L y CE\R, también son entradas a este retenedor. Las salidas BUSY\L y BUSY\R son generadas puentando la salida del retenedor con la adecuada señal de habilitador de circuito. También hay que notar que las salidas del retenedor son usadas internamente como señales inhibitoras de escritura tanto derecha como izquierda. Por ejemplo, si la señal de inhibición de escritura del lado derecho en la figura 4.3.2 es baja, una escritura a la memoria no ocurrirá aún si la entrada R/W\R del CY-131 es baja.

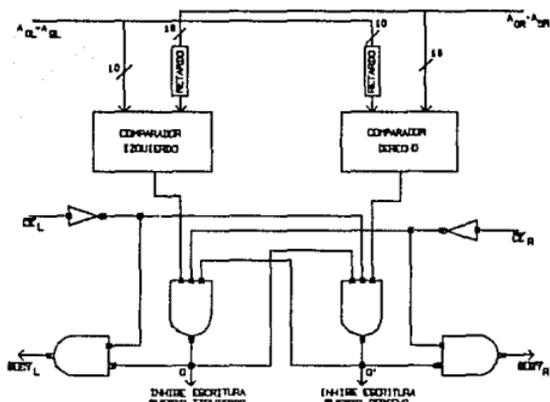


Figura 4.3.2 Diagrama de lógica conceptual de arbitraje.

Ahora, la operación del circuito de arbitraje puede ser explicada. Asumamos que la dirección del puerto izquierdo ha sido establecida y  $CE\setminus L$  es baja. Ambas salidas,  $Q$  y  $Q'$ , del retenedor serán altas debido a que ambas salidas de los comparadores son bajas (las direcciones son diferentes). Por lo tanto la salida  $BUSY\setminus$  en ambos lados es alta. Asumamos ahora que la dirección del puerto derecho cambia y es igual a la del lado izquierdo. La salida del comparador del lado derecho cambia a alto y la salida  $Q'$  del retenedor baja. Eventualmente la salida del comparador izquierdo también cambia a alto, pero debido al acoplamiento cruzado de  $Q'$  en la compuerta que genera la salida  $Q$ ,  $Q$  permanece alta. Tan pronto como  $CE\setminus R$  baja,  $BUSY\setminus R$  baja. Así el arbitraje da prioridad al puerto izquierdo enviando una señal de ocupado al puerto derecho. Así en este ejemplo, el puerto izquierdo es el ganador y el puerto derecho el perdedor en la contención por la memoria. Tarde o temprano el puerto izquierdo terminará su transacción a la localidad en contención y cambiará la dirección o el habilitador de circuito subirá. Con esto, cuando la contención termina la salida  $Q$  del retenedor cambiará a alto y  $BUSY\setminus r$  subirá. De manera similar se puede seguir el razonamiento para entender la operación del lado izquierdo. Debe ser claro que, en caso de contención, el árbitro decidirá que puerto es el ganador y el perdedor deberá esperar a que el ganador termine de usar la memoria. El puerto ganador debe indicar al árbitro que ha terminado su operación ya sea cambiando la dirección o poniendo la entrada habilitadora de circuito en estado alto. Sin una indicación, el árbitro no moverá la bandera de ocupado al puerto perdedor.

Las operaciones de lectura y escritura se realizan exactamente de la misma manera que una memoria RAM estándar.

## 5.1 Descripción general.

En la figura 5.1. se muestra el diagrama a bloques de la tarjeta interfaz canal SCSI <-> IBUS-III.

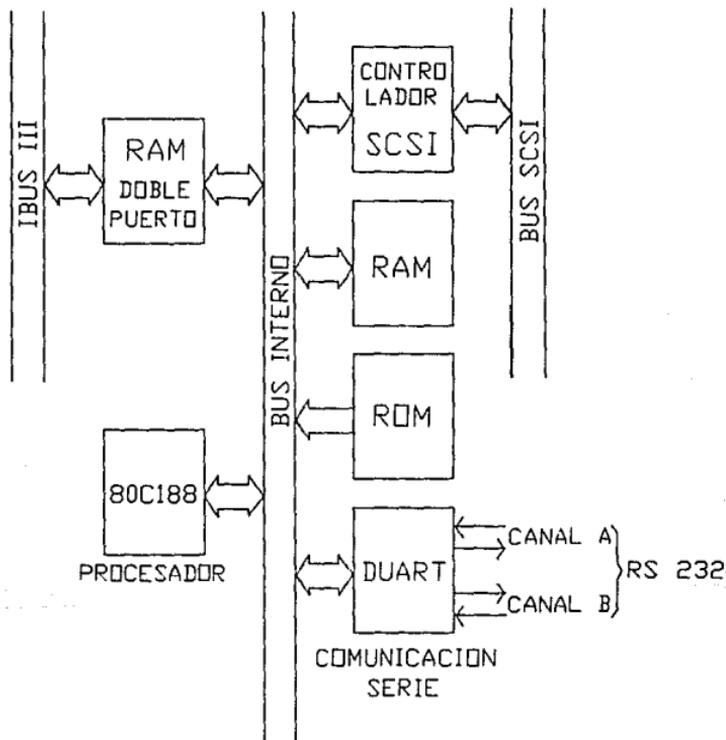


Figura 5.1 Diagrama a bloques de la interfaz SCSI IBUS-III

La tarjeta se compone de algunos circuitos especiales:

- **Procesador 80C188:** controla las operaciones de los dispositivos dentro de la tarjeta, ya que cuenta con lógica de selección de circuitos (chips), tanto para dispositivos periféricos a través de las líneas PCS, como para memoria a través de las líneas MCS; maneja dos canales de DMA, 3 contadores (timers) y posee un controlador de interrupciones, todo programable a través de sus registros internos.

- Memoria de doble puerto CY131: es la interfaz entre la tarjeta e IBUS-III; permite accesos simultáneos a diferente localidad y posee un árbitro que actúa cuando se intenta acceder a una misma localidad por los dos puertos. Su capacidad es de 1k x 8 bits y sus tiempos de acceso son de 55 ns, lo que facilita y acelera transferencias con DMA.
- Controlador SCSI NCR 53C80: soporta la norma ANSI SCSI X3T9.2; puede actuar como objetivo, como iniciador o ambos. Maneja un protocolo en el canal SCSI de 8 líneas de datos, 1 línea de paridad y 9 líneas de control. Se programa a través de 8 puertos, permitiendo transferencias de DMA a través de sus terminales DREQ y DACK (DMA request y DMA acknowledge).
- El prototipo posee dos canales serie compatibles con la norma RS-232, contenidos en el DUART (2681) para comunicación asincrónica. El circuito MAX233 de que dispone, permite convertir directamente entre los niveles eléctricos RS-232 y los HCMOS a partir de una sola fuente de alimentación de 5 volts.

Para describir el funcionamiento, consideremos que la tarjeta maestra (SAC 1887) debe enviar un bloque de información a un dispositivo en el canal SCSI. En forma breve, lo que debe hacer la tarjeta maestra es colocar en la memoria de doble puerto los datos a enviar y la información codificada de a que dispositivo se le van a enviar. Una vez que la información se encuentra en la RAM doble puerto, se genera una interrupción al procesador 80C188 y este analizará la información y establecerá la comunicación con el dispositivo deseado; una vez establecida la comunicación entre los dispositivos, el controlador de DMA del procesador 80C188 envía toda la información. De manera similar se realiza el proceso inverso, cuando la maestra requiere de la información de un dispositivo SCSI.

Una de las mayores ventajas que se tiene con este diseño es que la tarjeta maestra, una vez que ha dejado la información en la memoria de doble puerto, puede realizar otras operaciones; no queda bloqueada mientras se trasmite la información por el canal SCSI.

Los 2 canales serie RS-232 pueden ser usados para comunicación con un dispositivo en el canal SCSI, para comunicarse con la tarjeta maestra o simplemente para conocer los estados internos de la tarjeta.

Para la descripción detallada del diseño dividiremos la tarjeta en cuatro bloques principales: interfaz IBUS-III, que es la parte que comunica con IBUS-III, interfaz SCSI, que es la que comunica con el canal SCSI, la sección de procesamiento, que controla todas las operaciones para la comunicación y la sección comunicación serie.

## 5.2 Descripción de la interfaz a IBUS-III

Dentro de esta sección dividiremos en dos grupos, en uno veremos datos y direcciones y por el otro comandos y señales de control.

La figura 5.2 muestra el circuito electrónico de la interfaz para datos y direcciones. Como se puede notar, las señales de IBUS-III pasan a través de transceptores trapezoidales (U1, U2 y U3), los cuáles tienen características muy especiales por lo que la norma IBUS-III los especifica.

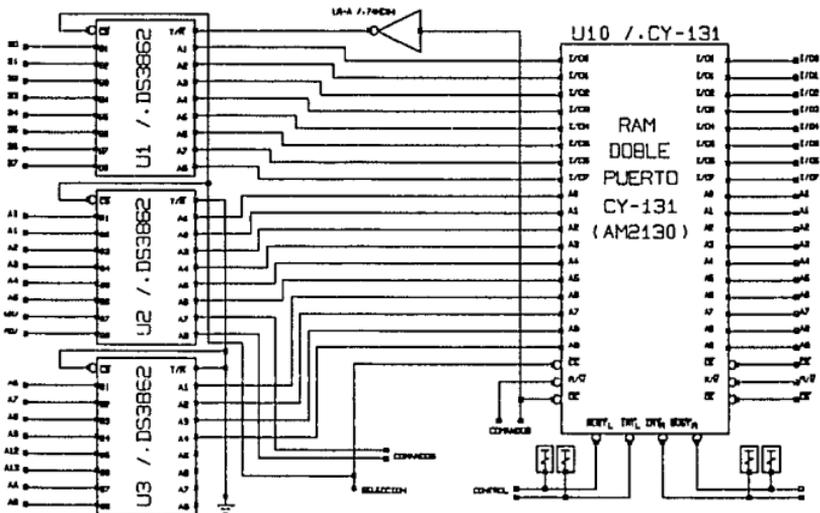


Figura 5.2 Interfaz a IBUS-III de datos y direcciones.

Las características de los transceptores trapezoidales son:

- Es un transceptor de canal para usarse con impedancias de terminación de línea de 120 ohms.
- Está especialmente diseñado para reducir el ruido en sistemas de distribución no balanceados.

- Sus drivers de colector abierto generan formas de onda trapezoidales precisas con tiempos de subida y bajada típicos de 9 ns, relativamente independiente de las condiciones de cargas capacitivas en la salida.
- Esto reduce el acoplamiento de ruido con líneas adyacentes sin ningún impacto apreciable en el rango máximo obtenible con transceptores de canal de alta velocidad.
- Niveles de umbral controlados estrechamente en el receptor que proveen igual rechazo a ambos pulsos de ruido en el canal, tanto positivos como negativos.
- La terminación externa de la línea es de 180 ohms a 5V y de 390 ohms a tierra.
- El canal puede estar terminado en uno o en ambos extremos.

La interfaz IBUS-III se basa en la memoria de doble puerto CY131. Existen otras memorias en el mercado como la Am2130, la cual difiere de esta sólo en la distribución de terminales, teniendo ambas una capacidad de 1K x 8 bits y un tiempo de acceso de 55 ns. Otras memorias de mayor capacidad pueden ser usadas como la IDT-7005 ó IDT-7006, las cuales tienen una capacidad de 8K x 8 y 16K x 8 bits, respectivamente.

La figura 5.3 muestra el circuito electrónico de la interfaz para comandos y señales de control. Como ya se mencionó, las señales de IBUS-III pasan a través de transceptores trapezoidales.

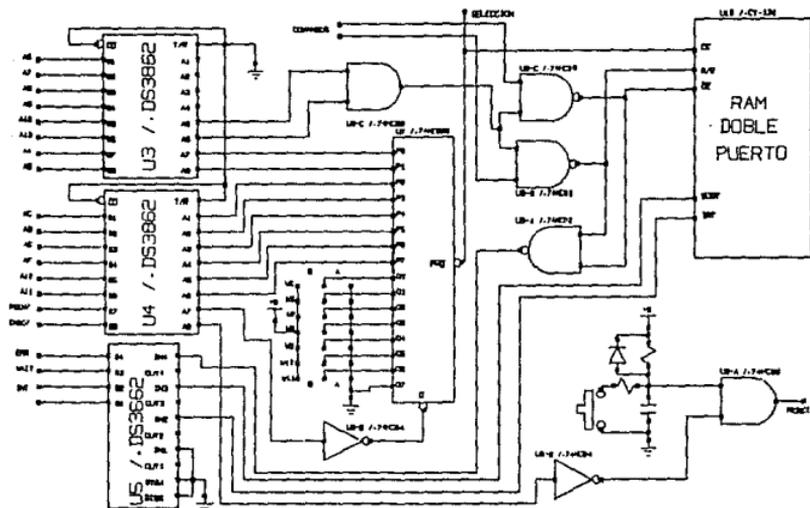


Figura 5.3 Interfaz a IBUS-III de comandos y señales de control.

#### Descripción funcional:

Para IBUS-III, esta es una tarjeta esclava mapeada en el espacio de memoria, por lo que para accederla es necesario decodificar las 20 líneas de direcciones.

La tarjeta maestra SAC-1887 ve a esta tarjeta como 1 Kbyte de memoria en páginas generales, localizable en el rango de direcciones C0000h - DFC00h. La dirección base es seleccionable a través de los jumpers W5-W11, como lo muestra la tabla 5.1

La señal PGEN y las direcciones AA, AB, AC, AD, AE, AF, A10, A11, A12 y A13 pasan a través de 2 transceptores trapezoidales (U3 y U4) que siempre están habilitados y en modo de recepción (CD y T/R están siempre en cero lógico), llegando a un comparador, por medio del cual se seleccionan los rangos de operación.

Una vez que se selecciona la tarjeta, la salida del comparador habilita los transeptores trapezoidales (U1 y U2), de modo que pueden ser recibidas el resto de las direcciones y los comandos; estos últimos son habilitados con la señal de selección por medio de compuertas NAND, generando la señal ERR que sale por el trapezoidal U5. La dirección del flujo de datos es controlada por la línea de comando RD, la cual está conectada al control T/R del trapezoidal de datos (U1); si ésta se activa, el trapezoidal estará en modo transmite, permitiendo la salida de los datos y si está desactivada, el trapezoidal estará en modo recibe, lo que permitirá que los datos entren a la memoria.

Las señales de control que genera la memoria de doble puerto (INT y BSY) salen directamente al canal pasando por el trapezoidal en U5 y llegando a IBUS-III como INT1 y WAIT respectivamente.

La señal de BSY (WAIT en IBUS-III), se genera cuando, tanto el canal IBUS-III como el microprocesador 80C188, quieren acceder la misma localidad al mismo tiempo, pero el arbitraje de la memoria doble puerto le cedió la prioridad al microprocesador.

La señal INT (INT1 en IBUS-II) se genera cuando el microprocesador escribe en la dirección base + 3FEh (la dirección base del procesador es la dirección que se programa para la activación de la línea MCS0, que en este diseño es 083FEh), permaneciendo activa hasta que se completa una lectura a la dirección base de IBUS-III + 3FEh (la dirección base de IBUS-III se obtiene de la tabla 5.1).

La señal INIC que también entra directamente a través del trapezoidal de selección (U4), llega a un circuito que pone a la tarjeta en condición inicial (Reset) como se muestra en la figura 5.3. La señal INIC después de pasar por un inversor llega a una compuerta NAND. La otra entrada de dicha compuerta la conforma un circuito RC. Este circuito RC es el reset por hardware. Al cerrar el interruptor, el capacitor se descarga, dando a la compuerta un valor de cero lógico, lo que producirá la puesta en condiciones iniciales al microprocesador. Los valores de R y de C se calcularon para que el capacitor alcanzara el valor de uno lógico después de 90 ms que es tiempo suficiente para que el microprocesador reconozca la señal de RESET. El diodo conectado en inversa sirve para que en el caso de que se presente una baja de tensión, el capacitor se descargue rápidamente a través de este, produciendo que la tarjeta se ponga en condiciones iniciales.

## 5.3 Direccionamiento en IBUS-III

La tarjeta maestra (SAC 1887) puede acceder esta memoria dependiendo de la posición de los "jumpers" W5 a W11 dispuestos en la tarjeta (ver figura 5.3).

W11	W10	W9	W8	W7	W6	W5	Rango de acceso
A	A	A	A	A	A	A	C0000h - C03FFh
A	A	A	A	A	A	B	C0400h - C07FFh
A	A	A	A	A	B	A	C0800h - C0BFFh
A	A	A	A	A	B	B	C0C00h - C0FFFh
A	A	A	A	B	A	A	C1000h - C13FFh
A	A	A	A	B	A	B	C1400h - C17FFh
A	A	A	A	B	B	A	C1800h - C1BFFh
A	A	A	A	B	B	B	C1C00h - C1FFFh
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
B	B	B	B	A	A	A	DE000h - DE3FFh
B	B	B	B	A	A	B	DE400h - DE7FFh
B	B	B	B	A	B	A	DE800h - DEBFFh
B	B	B	B	A	B	B	DEC00h - CFFFFh
B	B	B	B	B	A	A	DF000h - DF3FFh
B	B	B	B	B	A	B	DF400h - DF7FFh
B	B	B	B	B	B	A	DF800h - DFBFFh
B	B	B	B	B	B	B	DFC00h - DFFFFh

TABLA 5.1 Selección del rango de acceso en IBUS-III.

## 5.4 Descripción de la sección de procesamiento

Como ya se mencionó, la sección de procesamiento está basada en el microprocesador 80C188. En la tarjeta se tienen 3 bases para memorias, siendo la base U10 de 52 terminales para la RAM de doble puerto CY131; la base U11 para RAM estática configurable para memorias de 64 y 256 Kbits como se muestra en la tabla 5.2 y la base U12 para EPROM configurable para memorias de 64, 128 y 256 Kbits (tabla 5.3).

W1	W2	tipo de memoria	tamaño
A	cerrado	43256	32K x 8
B	abierto	HM6264	8K x 8

Tabla 5.2 Selección de tamaño de memoria RAM.

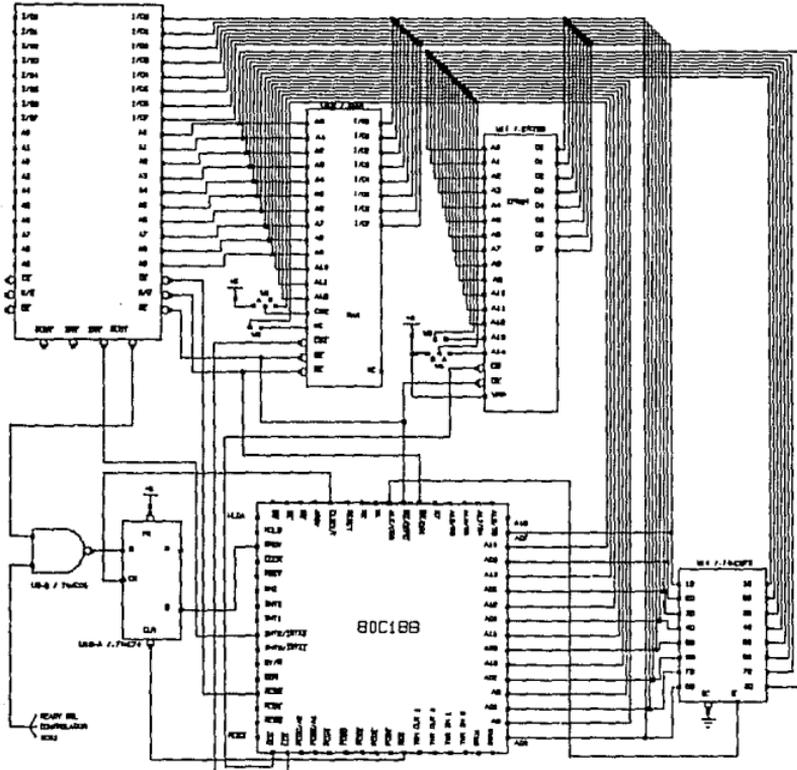


Figura 5.4 Diagrama de conexión de la sección de procesamiento.

W3	W4	tipo de memoria	tamaño
A	cerrado	27256	32K x 8
B	cerrado	27128	16K x 8
B	abierto	2764	8K x 8

Tabla 5.3 Selección de tamaño de memoria EPROM.

La memoria de doble puerto sólo podrá ser una CY131, ya que no hay otra que sea compatible terminal a terminal con esta.

Para la descripción del sistema prototipo, asumiremos que cuenta con una RAM 43256 y una EPROM 27256, por lo que los jumpers W1 a W4 quedarán

W1	A
W2	cerrado
W3	A
W4	cerrado

Asimismo, en donde aparezca RAM se refiere a la memoria 43256, y en donde aparezca RAM DP se refiere a la memoria de doble puerto CY131.

La memoria EPROM es habilitada por la línea UCS, programable para rangos de 1k a 256 kbytes; la RAM DP es habilitada por MCS0 programable para rangos de 2k a 128 kbytes y la RAM es habilitada por LCS, programable para rangos de 1k a 256 kbytes.

El procesador activará la línea de selección MCS0 para un rango de 2 kbytes como mínimo y la memoria que se usa sólo es de 1k. Consideremos que la señal MCS0 se activa para las direcciones 8000h a 87FFh; si dividimos a la mitad el rango de 2k, obtendremos 1 k definido de la 8000h a la 83FFh y el otro de la 8400h a la 87FFh. en el caso de que se desee enviar un dato a la dirección 8010h, las líneas de dirección A0 a A9 tendrán los siguientes valores:

0	0	0	0	0	1	0	0	0	0
A9									A0

si deseamos enviar un dato a la dirección 8410h, las líneas de dirección A0-A9 tendrán exactamente el mismo valor que si se envía a la 8010h, lo que provoca que el dato sea almacenado exactamente en la misma localidad de la memoria ya que esta sólo ve las líneas A0-A9. A este fenómeno se le conoce como "espejo". Para evitar este conflicto, se puede utilizar un circuito que habilite la memoria sólo cuando se está apuntando al bloque de 1kbyte inferior, como lo muestra la figura 5.5

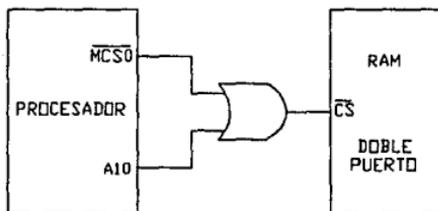


Figura 5.5 Diagrama para selección de memoria doble puerto

En la figura 5.5, la señal MCS0 está conectada a una compuerta OR, al igual que la señal A10; la salida de esta compuerta está conectada a la terminal habilitador de memoria con lo que logramos que la memoria sea seleccionada sólo cuando se cumple que MCS0 está activada (cero) y la línea de dirección A10 también es cero. Considerando que se necesitaría por lo menos otro circuito más en la tarjeta y que no es conflictivo tener este fenómeno debido a que no se utiliza más espacio de memoria, no se corrigió este fenómeno en el prototipo, por lo que el mapa de memoria del sistema con la configuración indicada se presenta a continuación

Línea  
activadora

UMCS	EPROM	FFFFh
MCS3 MCS2 MCS1	Disponible	F800h F7FFFh
MCS0	Espejo de RAM doble puerto	08800h 087FFh
	RAM doble puerto	08400h 083FFh
LMCS	RAM	08000h 07FFFh
		00000h

Figura 5.6 Mapa de memoria para el procesador 80C188.

En la memoria EPROM se tiene la rutina de para poner al procesador en condiciones iniciales, así como el programa de control del protocolo.

Al encender o poner en condiciones iniciales la tarjeta, el procesador comenzará a ejecutar las instrucciones a partir de la dirección FFFF0h. Después, el procesador activa UCS para las direcciones desde FFC00h hasta la FFFFh, es decir, queda programado para reconocer la memoria superior de 1k. Es por esto que en la localidad FFFF0h se tiene un salto a la localidad FFC00h y es aquí en donde se tiene la rutina para poner en condiciones iniciales al procesador.

Para obtener el mapa de memoria indicado en la figura 5.6, es necesario que la rutina de condiciones iniciales programe los registros de selección de circuito como se muestra a continuación:

Dir del Registro	Nombre del registro	Valor a escribir
FFFEh	Registro de relocalización	00FFh

Descripción: se programa al procesador como maestro, los registros internos estarán mapeados en el espacio de puertos, direccionados a partir de la FF00h

FFA0h	Registro de selección de memoria superior (UMCS)	F83Dh
-------	--	-------

Descripción: se programa para que la señal UCS se active en el rango de 32k (F8000h-FFFFh), con 1 ciclo de espera e ignorando la señal externa de READY.

FFA2h	Registro de selección de memoria inferior (LMCS)	07FCh
-------	--	-------

Descripción: se programa para que la señal LCS se active en el rango de 32k (00000h-07FFFh), con 0 ciclos de espera e ignorando la señal externa de READY.

FFA6h	Registro de selección de dirección base de la memoria de rango medio (MMCS)	09FCh
-------	---	-------

Descripción: se programa para que las señales MCS0-3 se activen a partir de la dirección 08000h, con 0 ciclos de espera e ignorando la señal externa de READY.

**FFA8h** Registro de selección del tamaño de la memoria de rango medio (MPCS) **81BCh**

Descripción: se programa para que las señales MCS0-3 se activen en rangos de 2K, comenzando por MCS0 en la dirección más baja (08000h-087FFh) y terminado con MCS3 en la más alta (09800h-09FFFh) con 0 ciclos de espera e ignorando la señal externa de READY. Además, los periféricos quedarán mapeados en el espacio de puerto, teniendo 7 líneas de selección de periféricos (PCSO-PCS7).

**FFA4h** Registro de selección de dirección base de dispositivos periféricos (PACS) **0038h**

Descripción: se programa para que los dispositivos periféricos queden mapeados a partir de la dirección 00000h hasta la 003FFh, ya que las líneas PCS0-7 se activan para cada bloque consecutivo de 128 bytes, quedando las señales activadas para los rangos mostrados en la tabla 5.4

Línea activadora	Rango de activación	Dispositivo seleccionado
PCS0	000h - 07Fh	Controlador SCSI
PCS1	080h - 0FFh	DUART
PCS2	100h - 17Fh	Lectura de SCSI ID
PCS3	180h - 1FFh	Disponible
PCS4	200h - 27Fh	Disponible
PCS5	280h - 2FFh	Disponible
PCS6	300h - 37Fh	Disponible
PCS7	380h - 3FFh	Disponible

Tabla 5.4 Mapeo de dispositivos periféricos.

Estarán programados con 0 ciclos de espera, pero reconocerán la señal externa de READY, insertando ciclos de espera mientras esta esté activada.

Una vez que se ha escrito el registro de control del UCS, se puede acceder el código desde la dirección F8000h, que es en donde localizaremos el programa controlador de las transferencias (programa principal).

La RAM será activada en el rango de la direcciones 00000h-7FFFFh. Aquí se tienen las variables del sistema, los apuntadores de las transferencias y los vectores de interrupción.

La memoria RAM DP (CY131) será activada en el rango de direcciones 08000h a 083FFh y se tendrá un espejo a partir de la 08400h. Esta memoria es el buffer de las transferencias, el área de comandos y banderas. A continuación se presenta el mapeo de esta memoria para el programa de pruebas del procesador 80188. Esta configuración será modificada al desarrollar el sistema final para comunicación con dispositivos comerciales.

Dirección	Descripción
8000h	comando
8001h-8003h	dirección de inicio de la transferencia
8004h-8005h	cantidad de bytes a enviar
8006h-800Fh	disponibles para banderas y mensajes
8010h-83FFh	datos a ser transferidos
83FEh	deshabilita la interrupción generada por IBUS-III
83FFh	activador de la señal de interrupción para IBUS-III.

Tabla 5.5 Mapa de memoria CY-131 para el procesador 80C188.

La señal de BSY del circuito CY-131 se conecta directamente a la señal ARDY del procesador, ya que éste tiene la capacidad de sincronizar internamente la señal. Otra opción de conectarlo es como se ve en la figura 5.4, en donde la señal de BSY entra a un circuito de sincronización basado en un flip-flop, llegando al procesador a través de la terminal SRDY. El flip-flop queda sincronizado por medio de la señal CLKOUT del procesador.

Una opción sencilla para lograr un mayor buffer de transferencia es utilizando los circuitos IDT7005 o IDT7006, los cuales tienen capacidad de 8K x 8 y 16K x 8 bits respectivamente, o bien conectando varias memorias de doble puerto, formando un banco de memoria.

#### Descripción funcional:

El procesador está dedicado únicamente a controlar la transferencia al canal SCSI; este no realiza ninguna operación ni ordena tareas a otros dispositivos. Lo único que debe hacer es interpretar las instrucciones que se le mandan a través de la memoria doble puerto, por lo que la mayor parte del tiempo el



Este circuito es accesado en el espacio de puertos a partir de la dirección 00000h, con lo que sus ocho registros son accesados con instrucciones in y out a los puertos 0 a 7.

Dirección	Lectura	Escritura
00h	Dato SCSI actual	Reg. de salida de datos
01h	Reg. de comandos del iniciador	Reg. de comandos del iniciador
02h	Registro de Modo	Registro de Modo
03h	Reg. de comandos del objetivo	Reg. de comandos del objetivo
04h	Estados actuales del canal SCSI	Reg. habilitador de selección
05h	Reg. del canal y estados	Inicia envío DMA
06h	Reg. de entrada de datos	Inicia objetivo recepción en DMA
07h	Inicializa paridad e interrupción	Inicia iniciador recepción en DMA

Tabla 5.6 Mapa de registros internos del controlador NEC53C80.

Para acceder al controlador SCSI es necesario haber programado los registros MMCS y MPCS como se indica en la sección de procesamiento.

Como se puede ver en la figura 5.7, ninguna línea de selección de circuito (PCSx) se encuentra directamente conectada al controlador (CS). Esto es debido a que existen dos maneras de accederlo: para transferencias I/O, se debe activar la señal CS y para transferencias DMA se debe activar la señal DACK. Estas líneas nunca deben ser activadas simultáneamente. Por esta razón, la selección del circuito se realiza a través de un decodificador 2-4, teniendo como entradas PCS0 y A19/S6 y como salidas CS y DACK.

La tabla de valores del decodificador se presenta a continuación:

G	B	A	Y0	Y1	Y2	Y3
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	x	x	1	1	1	1

Tabla 5.7 Tabla de verdad de decodificador 2 a 4.

Como se ve en la tabla 5.7, manteniendo la compuerta G habilitada (conectada a 0) y conectando las entradas B con PCS0 y A con A19/S6 y las salidas Y0 con CS y Y1 con DACK, tendremos la siguiente codificación:

PCS0	A19/S6	CS	DACK
0	0	0	1
0	1	1	0
1	X	1	1

Tabla 5.8 Tabla de verdad para activar controlador SCSI.

Para la decodificación se utilizan la señales PCS0, por ser la selectora de periféricos y A19/S6, que tiene la característica de indicar durante los tiempos del procesador T2, T3, Tw y T4 si se está en un ciclo del procesador al activarse (1) o si se está en un ciclo DMA al desactivarse (0). Los ciclos de DMA son solicitados por el controlador de canal SCSI a través de su señal DRQ, la cual se activa un poco después de que se desactiva el comando IOR o IOW. Las transferencias DMA se inician al escribir en el registro correspondiente 5, 6 ó 7, lo que generará la activación de DRQ. Al concluir las transferencias es necesario poner en condición inicial el bit de MODO DMA para que la señal DRQ se desactive.

En la figura 5.7 se puede ver un DIP switch, el cual sirve para seleccionar el SCSI ID (identificador SCSI) de la tarjeta. Esto se realiza activando uno y sólo uno de los switches, con lo que tendremos activo un bit en el canal. Para poder leer este valor, es necesario conectar un buffer, para evitar que exista contención en el canal interno. El buffer permite el paso del dato sólo al activarse ambas compuertas G0 y G1 del circuito U15. Para activar estas señales se utilizó un circuito sencillo por medio de una compuerta, el cual habilita la salida de las señales al activarse simultáneamente la señal PCS2 y el comando RD. Esto se logra a través de lecturas a cualquiera de los puertos en las direcciones 100h - 17Fh.

El controlador SCSI se pone en condiciones iniciales, al poner en esta condición a toda la tarjeta, por medio de la salida RESET del procesador, la cual permanece activa sólo un ciclo de reloj menos de lo que permanece activada la señal de entrada al procesador RST.

El controlador cuenta con una señal de solicitud de interrupción (IRQ), la cual se conecta directamente a la entrada del procesador INT1. IRQ se activa cuando ocurre una selección o una condición especial en el canal SCSI, lo que permitirá que sea detectada por el procesador a través de la rutina de servicio de la interrupción (INT1).

**Descripción funcional:**

Este circuito tiene la característica de no interpretar ningún dato que pasa a través de él. Para controlarlo es necesario escribir en uno o varios de sus registros para lograr la comunicación. Lo primero es pasar por la fase de arbitraje, la cual no es necesaria cuando sólo se tiene un iniciador y un objetivo; en esta fase se logra que en el canal sólo se encuentre un iniciador, ya que en este sistema no se permite que dos dispositivos actúen como iniciadores a la vez. Ya logrado el control del canal SCSI se procede a la selección del dispositivo, en la cual se indica a qué dispositivo en el canal se desea enviar o recibir información; ya seleccionado el dispositivo se procede a una fase de comandos, en la que el iniciador le indica al objetivo que tipo de transferencia se va a realizar, cuantos bytes se van a enviar y si el iniciador va a recibir o a enviar. Una vez que los dos dispositivos saben que operaciones se realizarán comienza la fase de transferencia de datos, en la cual cada byte que se escribe al puerto de salida del dispositivo que envía es leído en el puerto de entrada del dispositivo que recibe; este proceso es controlado a través de las líneas REQ y ACK del canal SCSI. En el capítulo 6 se presentan los programas de prueba, en donde se puede ver como se deben programar los registros para cada fase. Es importante que tanto el iniciador como el objetivo se encuentren en la misma fase, de otro modo, la operación no se realizará; por esto, si alguno de los dos cambia de fase, el otro debe entrar a la misma fase. Esto debe ser controlado por el iniciador.

**5.6 Descripción de canales de comunicación serie**

La tarjeta se diseñó con dos canales de comunicación serie; esto en parte por la facilidad que presenta el canal serie para detectar y corregir errores al momento de ejecutar programas y por otro lado, para poder cargar estos programas a RAM a través de uno de los dos canales serie, usando la adaptación del programa monitor desarrollado para la tarjeta maestra SAC-1887. El dispositivo utilizado es un DUART (Dual Asynchronous Receiver/Transmitter) MC2781, el cual tiene las siguientes características:

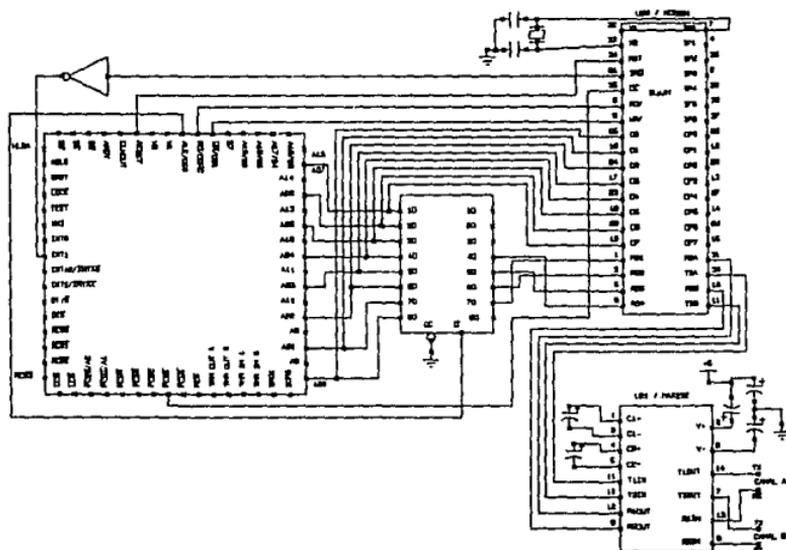
- Dos canales independientes Full-Duplex.
- Baud-rate programable independiente para cada canal en 18 rangos fijos.
- Formato de datos programable:
  - de 5 a 7 bits de datos más paridad
  - paridad par, impar, no paridad o paridad forzada
  - bits de paro programable en uno, uno y medio o dos

- Modos de canal programable:
  - normal (Full-Duplex)
  - eco automático
  - loopback local
  - loopback remoto

- Tiene una interfaz de propósito general que puede ser usada con microprocesadores síncronos y asíncronos.

- Tiene un puerto de entrada de 7 bits y uno de salida de 8 bits, ambos de propósitos múltiples.

La figura 5.8 muestra el diagrama de conexión



En la figura 5.8 se ve como el procesador se conecta directamente con el DUART, por medio de las 8 líneas de datos, 4 líneas de direcciones (A4-A1), para acceder los registros internos, los comandos de lectura y escritura, una línea de selección del dispositivo y una línea de interrupción. En este sistema, los canales paralelos de entrada y salida no se utilizaron. Las líneas de datos de canal serie (Tx y Rx) se conectan al circuito MAX232, el cual convierte las señales de niveles TTL (+5V) a señales de niveles RS232 ( $\pm 12V$ ).

Los registros internos del DUART quedan mapeados como se muestra en la tabla 5.9

Dir	Lectura	Escritura
80h	Reg. de modo A	Reg. de modo A
82h	Reg. de estados A	Reg. selección de reloj A
84h	No acceder	Reg. de comandos A
86h	Buffer recepción A	Buffer transmisión A
88h	Reg. cambio de puerto de entrada	Reg. de control auxiliar
8Ah	Reg. estados de interrupción	Reg. de máscara de interrupción
8Ch	Modo de contador MSB actual de cuenta	Registro de timer/contador superior
8Eh	Modo de contador LSB actual de cuenta	Registro de timer/contador inferior
90h	Reg. de modo B	Reg. de modo B
92h	Reg. de estados B	Reg. selección de reloj B
94h	No acceder	Reg. de comandos B
96h	Buffer recepción B	Buffer transmisión B
98h	Registro de vector de interrupción	Registro de vector de interrupción
9Ah	Puerto de entrada (no retenido)	Reg. configuración de puerto de salida
9Ch	Comando de iniciar conteo	Reg. puerto de salida activa comando
9Eh	Comando de iniciar conteo	Reg. puerto de salida inicializa comando

Tabla 5.9 Mapa de registros internos de DUART.

## 6. PROGRAMACION.

### 6.1 Rutina de inicialización del microprocesador 80C188.

A continuación se presenta una descripción del programa que se desarrolló para inicializar los registros internos del microprocesador.

Se programa al procesador como maestro, los registros internos estarán mapeados en el espacio de puertos, direccionados a partir de la FF00h

La señal UCS se activa en el rango de 32k (F8000h-FFFFFh), con 1 ciclo de espera e ignorando la señal externa de READY.

La señal LCS se activa en el rango de 32k (00000h-07FFFh), con 0 ciclos de espera e ignorando la señal externa de READY.

Las señales MCS0-3 se activan a partir de la dirección 08000h, con 0 ciclos de espera e ignorando la señal externa de READY.

Las señales MCS0-3 se activan en rangos de 2K, comenzando por MCS0 en la dirección más baja (08000h-087FFh) y terminado con MCS3 en la más alta (09800h-09FFFh) con 0 ciclos de espera e ignorando la señal externa de READY. Además, los periféricos quedarán mapeados en el espacio de puerto, teniendo 7 líneas de selección de periféricos (PCS0-PCS7).

Los dispositivos periféricos queden mapeados a partir de la dirección 00000h hasta la 003FFh, ya que las líneas PCS0-7 se activan para cada bloque consecutivo de 128 bytes, quedando las señales activadas para los rangos mostrados en la tabla 5.4. Estarán programados con 0 ciclos de espera, pero reconocerán la señal externa de READY, insertando ciclos de espera mientras esta esté activada.

Este programa se compiló, ligó y localizó junto con las rutinas del monitor MSB-87, la cuales fueron modificadas anteriormente para hacerlas completamente compatibles con el microprocesador 80C188.

Los dos programas que se presentan a continuación sirven únicamente para comunicar las dos tarjetas que se desarrollaron, ya que el objetivo era probar la comunicación a través del canal SCSI y lograr determinar la máxima velocidad de transferencia que se puede lograr con el sistema.

## 6.2 Rutina para tarjeta Interfaz IBUS-III <-> SCSI.

Este programa sólo sirve para comunicarse con la segunda tarjeta que se armó, ya que la configuración está totalmente especificada y no contempla la posibilidad de conectarse con equipos comerciales ya que este no cumple con la norma SCSI.

El objetivo del programa es transferir información de la memoria doble puerto, la cual actúa como buffer de transferencias, a la tarjeta que simula un dispositivo periférico. La dirección 08000h es de donde o a donde se envía la información, que es la dirección de inicio de la memoria doble puerto.

Este programa presenta un menú de 4 opciones:

- 0 = envío por I/O
- 1 = envío por DMA
- 2 = recibe por I/O
- 3 = recibe por DMA

Una vez seleccionada la opción, se pregunta la dirección de inicio; esta dirección de 5 nibbles es la dirección del dispositivo periférico en donde se almacenarán o de donde se obtendrán los datos de la tarjeta no. 2 (simuladora de dispositivo SCSI). En este caso, sólo se puede acceder de la dirección 00000h a la 08000h (memoria RAM de la tarjeta no. 2) y de la F8000h a la FFFFFh (memoria ROM de la tarjeta no. 2), debido a que esta tarjeta sólo cuenta con 32 Kbytes de RAM y 32 Kbytes de EPROM.

### Dirección de inicio (00000 - FFFFF):

Se debe tener cuidado de no acceder las direcciones en donde se encuentran localizados los programas y los vectores de interrupción, ya que de modificarlos, se tendrá que cargar de nuevo el programa a la tarjeta.

Una vez obtenida la dirección, se pregunta la longitud de transferencia, es decir, la cantidad de bytes que serán transferidos.

### Longitud de transferencia (001 - 3E0):

Esta longitud queda limitada, ya que sólo estamos considerando la capacidad del buffer de transferencias, que en este caso será la capacidad de la memoria doble puerto, menos algunos bytes que servirán como control.

Una vez que se tienen estos datos, se realiza la transferencia de información entre las dos tarjetas, comenzando con la fase de arbitraje y selección que se describe a continuación:

**Fase de arbitraje:**

Escribe el ID del dispositivo al puerto 00h

Activa el bit de arbitraje (pto. 2 bit 0)

Revisa que el arbitraje está en progreso (pto. 1 bit 6). Si no lo está, vuelve a leer el puerto hasta que el bit 6 este activo

Espera 2.2 us

Revisa si se perdió el arbitraje (pto. 1 bit 5). Si se perdió, vuelve a comenzar

Lee del puerto 0 si hay alguien más arbitrando. Si lo hay, revisa que su ID sea mayor. Si es mayor, pierde el arbitraje y vuelve a comenzar

Si no hay otro arbitrando o si no es mayor su ID, revisa si se perdió el progreso del arbitraje (pto. 1 bit 5). Si se perdió, vuelve a comenzar

**Fase de selección:**

Activa el bit SEL (pto. 1 bit 2)

Espera 1.2 us

Escribe su ID y el del dispositivo objetivo en el puerto 0

Activa BSY y el canal de datos (pto. 1 bits 3,0)

Desactiva el bit de arbitraje (pto. 2 bit 0)

Limpia el puerto 4 para evitar que se active una interrupción

Desactiva BSY (pto. 1 bit 3)

Retardo de 400 ns

Espera que el objetivo active BSY dentro de 250 ms. Si no se activó BSY, se envía un mensaje de error en la selección y se termina proceso

Si se activó BSY, desactiva el bit SEL (pto. 1 bit 2)

Una vez que se logró la selección, se inicia la fase de comandos:

**Fase de comandos:**

Activa canal de datos (pto. 1 bit 0)

Compara que en el pto. 4 coincida la fase de comandos (MSG = 0, C/D = 1, I/O = 0, SEL = 0 y BSY = 1). Si no coinciden hay error y termina proceso

Se activan los bits correspondientes a la fase de comandos en el puerto 3 (MSG = 0, C/D = 1, I/O = 0), para que quede habilitado el canal de datos (pto. 0)

Se programa un apuntador de datos de comando

Se programa un contador para enviar 6 bytes

Espera que se active REQ (pto. 4 bit 5)

Escribe un byte de comando en puerto 0

Activa ACK (pto. 1 bit 4)

Espera que se desactive REQ (pto. 4 bit 5)

Desactiva ACK (pto. 1 bit 4)

Si el contador está en cero, termina fase de comandos

Incrementa el apuntador de datos de comando

Decrementa contador y repite los 7 pasos anteriores

Una vez terminada la fase de comandos, se procede la fase de transferencia de datos, en la cual se tienen cuatro posibilidades:

**Fase DATA IN con transferencias vía I/O:** en esta fase los datos son transferidos del objetivo al iniciador.

Programa contador de datos a recibir

Programa apuntador de dirección en memoria para recepción de datos (dirección 08000h)

Revisa que BSY esté activo (pto. 6 bit 4). Si no está, envía señal de error y termina proceso.

Espera que se active REQ (pto. 4 bit 5)

Lee el byte de del puerto 0

Almacena ese dato en la localidad indicada por el apuntador de memoria

Activa ACK (pto. 1 bit 4)

Espera que se desactive REQ (pto. 4 bit 5)

Desactiva ACK (pto. 1 bit 4)

Si el contador está en cero, termina fase de datos

Incrementa apuntador de localidad de memoria

Decrementa contador y repite los 8 pasos anteriores

Fase DATA OUT con transferencias vía I/O: en esta fase los datos son transferidos del iniciador al objetivo.

Programa contador de datos a enviar

Programa apuntador de dirección en memoria para envío de datos (dirección 08000h)

Activa el canal de datos (pto. 1 bit 0)

Se activan los bits correspondientes a la fase de salida de datos en el puerto 3 (MSG = 0, C/D = 0, I/O = 0), para que quede habilitado el canal de datos (pto. 0)

Revisa que BSY esté activo (pto. 6 bit 4). Si no está, envía señal de error y termina proceso.

Espera que se active REQ (pto. 4 bit 5)

El dato en la localidad indicada por el apuntador de memoria lo transfiere al puerto 0

Activa ACK (pto. 1 bit 4)

Espera que se desactive REQ (pto. 4 bit 5)

Desactiva ACK (pto. 1 bit 4)

Si el contador está en cero, termina fase de datos

Incrementa apuntador de localidad de memoria

Decrementa contador y repite los 7 pasos anteriores

Fase DATA IN con transferencias vía DMA: en esta fase los datos son transferidos del objetivo al iniciador.

\* Escribe en los registros del controlador de DMA del microprocesador:

La dirección fuente (la dirección de recepción vía DMA del controlador SCSI es el puerto 6)

La dirección destino (será la dirección de inicio de la memoria doble puerto, que en este caso será la 08000h)

Contador de transferencias DMA, cantidad de bytes a recibir.

Palabra de control

\* En los registros del controlador SCSI se debe:

Revisar que el bit BSY esté activo (pto. 4 bit 6)

Activar el bit de Modo DMA (pto. 2 bit 1)

Limpiar el puerto 1

Se activan los bits correspondientes a la fase de entrada de datos en el puerto 3 (MSG = 0, C/D = 0, I/O = 1, bits 2, 1 y 0 respectivamente)

Realiza un ciclo de escritura en el puerto 7 para iniciar la recepción vía DMA

Termina fase de transferencia de datos

Fase DATA OUT con transferencias vía DMA: en esta fase los datos son transferidos del iniciador al objetivo.

\* Escribe en los registros del controlador de DMA del microprocesador:

La dirección fuente (será la dirección de inicio de la memoria doble puerto, que en este caso será la 08000h)

La dirección destino (la dirección de envío vía DMA del controlador SCSI es el puerto 0)

Contador de transferencias DMA, cantidad de bytes a enviar

Palabra de control

\* En los registros del controlador SCSI se debe:

Activar el canal de datos (pto. 1 bit 0)

Revisar que el bit BSY esté activo (pto. 4 bit 6)

Activar el bit de Modo DMA (pto. 2 bit 1)

Se activan los bits correspondientes a la fase de entrada de datos en el puerto 3 (MSG = 0, C/D = 0, I/O = 0, bits 2, 1 y 0 respectivamente)

Realiza un ciclo de escritura en el puerto 5 para iniciar la transmisión vía DMA

Termina fase de transferencia de datos

### 6.3 Rutina para la tarjeta que simula un dispositivo SCSI.

El programa que se muestra a continuación se realizó para la tarjeta no 2, la cual actúa como un dispositivo objetivo en el canal SCSI.

En esta rutina, la tarjeta queda en un estado durmiente, en espera de que se active una interrupción a través del controlador SCSI.

Para que esta interrupción se active es necesario escribir en el puerto 4 el valor de su ID. Con esto, al presentarse la condición de que en el canal SCSI, SEL esta activa, el bit ID coincide con el dato del canal SCSI y BSY esta desactivada durante 400 ns.

Al activarse la interrupción, se le indica que se encuentra en fase de selección, en la cual debe:

Activar la señal de BSY (pto. 1 bit 3)

Activar el modo de objetivo (pto. 2 bit 6)

Inmediatamente después, la tarjeta entrará a la fase de comandos y recibirá 6 bytes, en los cuales se le indica:

- Tipo de transferencia (entrada o salida, I/O ó DMA)
- Dirección de inicio para recepción o para envío (según sea el caso)
- Cantidad de bytes a ser transferidos

El proceso para la fase de comandos es:

Programar apuntador para la recepción de comandos

Programar contador para recepción de 6 bytes

Activa C/D (pto. 3 bit 1)

Activa REQ (pto. 3 bit 3)

Espera a que se active ACK (pto. 5 bit 0)

Lee el byte de comando y lo almacena

Desactiva REQ (pto. 3 bit 3)

Espera a que se desactive ACK

Si el contador está en cero termina fase de comandos

Incrementa apuntador de datos de comando

Decrementa contador y repite los 6 pasos anteriores

Una vez que se tienen estos datos, se procede, según el tipo de transferencia seleccionado, de alguna de las 4 maneras que se muestran a continuación:

Fase DATA IN con transferencias vía I/O: en esta fase los datos son transferidos del objetivo al iniciador.

Programa contador de datos a enviar

Programa apuntador de dirección en memoria para envío de datos (la dirección la obtuvo en la fase de comandos)

Activa bit de modo objetivo (pto. 2 bit 6)

Activa BSY y canal de datos (pto. 1 bits 3,0)

Activa I/O (pto. 3 bit 0) para indicar la fase de entrada de datos

Manda el dato de la localidad apuntada al puerto 0

Activa REQ (pto. 3 bit 3)

Espera que se active ACK (pto. 5 bit 0)

Desactiva REQ (pto. 3 bit 3)

Espera que se desactive ACK (pto. 5 bit 0)

Si el contador de datos es cero, termina fase de transferencia de datos

Incrementa apuntador

Decrementa contador y repite los 7 pasos anteriores

Fase DATA OUT con transferencias vía I/O: en esta fase los datos son transferidos del iniciador al objetivo.

Programa contador de datos a recibir

Programa apuntador de dirección en memoria para recepción de datos (la dirección la obtuvo en la fase de comandos)

Activa bit de modo objetivo (pto. 2 bit 6)

Activa BSY y canal de datos (pto. 1 bits 3,0)

Desactiva I/O, MSG y C/D (pto. 3) para indicar la fase de salida de datos

Activa REQ (pto. 3 bit 3)

Espera que se active ACK (pto. 5 bit 0)

Lee el dato del puerto 0 y lo almacena

Desactiva REQ (pto. 3 bit 3)

Espera que se desactive ACK (pto. 5 bit 0)

Si el contador de datos es cero, termina fase de transferencia de datos

Incrementa apuntador

Decrementa contador y repite los 7 pasos anteriores

Fase DATA IN con transferencias vía DMA: en esta fase los datos son transferidos del objetivo al iniciador.

\* Escribe en los registros del controlador de DMA del microprocesador:

La dirección fuente (será la dirección de inicio que se obtuvo en la fase de comandos)

La dirección destino (la dirección de envío vía DMA del controlador SCSI es el puerto 0)

Contador de transferencias DMA, cantidad de bytes a enviar

Palabra de control

\* En los registros del controlador SCSI se debe:

Activar BSY y canal de datos (pto. 1 bits 3,0)

Activar el Modo DMA y el modo objetivo (pto. 2 bits 6,1)

Activar I/O (pto. 3 bit 1)

Realizar un ciclo de escritura al puerto 5 para iniciar el envío vía DMA

Termina fase de transferencia de datos

Fase DATA OUT con transferencias vía DMA: en esta fase los datos son transferidos del iniciador al objetivo.

\* Escribe en los registros del controlador de DMA del microprocesador:

La dirección fuente (la dirección de recepción vía DMA del controlador SCSI es el puerto 6)

La dirección destino (será la dirección de inicio que se obtuvo de la fase de comandos)

Contador de transferencias DMA, cantidad de bytes a recibir

Palabra de control

\* En los registros del controlador SCSI se debe:

Activar BSY (pto. 1 bit 3)

Activar el Modo DMA y el modo objetivo (pto. 2 bit 6,1)

Se activan los bits correspondientes a la fase de entrada de datos en el puerto 3 (MSG = 0, C/D = 0, I/O = 0 , bits 2, 1 y 0 respectivamente)

Realiza un ciclo de escritura en el puerto 6 para iniciar la transmisión vía DMA

Termina fase de transferencia de datos

### Resultados y conclusiones.

Los resultados que se presentan a continuación se obtuvieron del analizador de estados lógicos, al hacer transferencias de información tanto de la tarjeta interfaz IBUS-III <-> SCSI a la tarjeta simuladora de un dispositivo periférico, como de la simuladora a la interfaz IBUS-III <-> SCSI.

Considerando que lo que nos interesa es obtener la máxima velocidad de transferencia posible, sólo analizaremos los resultados obtenidos al realizar transferencias a través de accesos directos a memoria (DMA), ya que este será el método utilizado para transferir, una vez que se tenga desarrollado el software para transferencias con dispositivos comerciales. A continuación se presenta el diagrama de tiempos que se obtuvo al transferir información de la tarjeta interfaz IBUS-III a la tarjeta que simula un dispositivo periférico en el canal SCSI.

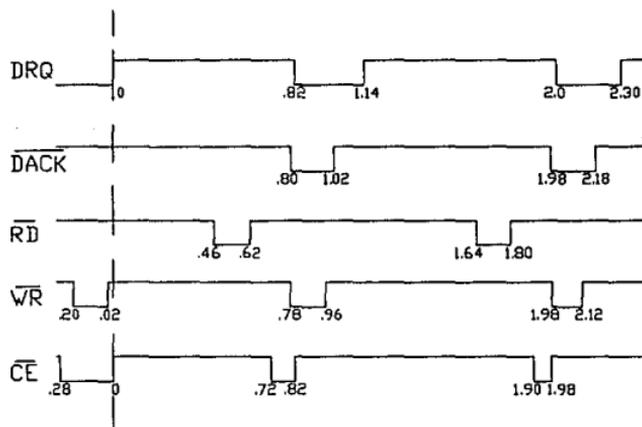


Figura 1 Diagrama de tiempos  
ciclos de envío de información.

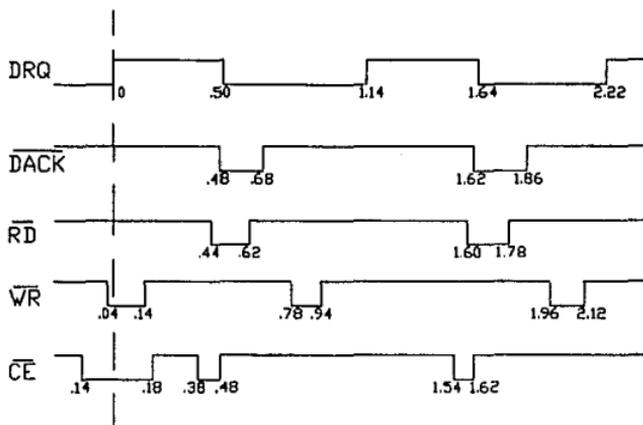


Figura 2 Diagrama de tiempos  
ciclos de recepción de información.

Del analizador de estados lógicos, se obtuvieron los tiempos en que se activa la señal DREQ, tanto para envío, como para recepción de información, con una precisión de 2 us. A continuación se muestra la tabla de estos resultados:

envío		recepción	
t(us)	diferencia	t(us)	diferencia
0	0	0	0
1.14	1.14	1.14	1.14
2.30	1.16	2.22	1.08
3.46	1.16	2.30	1.08
4.62	1.16	4.38	1.08

Tabla 8.1 Tiempos de envío y recepción de información.

Como se puede ver, el ciclo de envío de cada byte se realiza en 1.16 us lo que nos da una velocidad de transferencia de 862 Kbytes/s que es aproximadamente el 80% de lo que se esperaba obtener, mientras que para la recepción, cada ciclo se realiza en 1.08 us lo que es igual a 926 Kbytes/s. equivalente a un 87% de lo esperado.

Estas velocidades de envío y recepción se lograron con el microprocesador 80C188 corriendo a una velocidad de 12 MHz. Si aumentamos su velocidad a 16 Mhz, la cual es soportada por este dispositivo, aumentamos los procesos en un 33.3 % con lo que las velocidades de transferencia a través de accesos directos a memoria (DMA) serán de 1.15 Mbytes/s en envíos y de 1.23 Mbytes/s en recepción, obteniendo así las velocidades de transferencia esperadas.

Para el caso de las transferencias de IBUS-III a la tarjeta interfaz, la velocidad de envío es tan rápida como la puede realizar la tarjeta procesadora SAC-1887, ya que esta la ve sólo como una expansión de memoria, realizando accesos en 700 ns, lo que teóricamente nos da una velocidad de transferencia de 1.4 Mbytes/s.

Este factor no es tan importante, ya que, en el caso de envío de información, esta puede ser elaborada en el mismo espacio de memoria que sirve como buffer de transferencias, evitando así un tiempo extra en el manejo de esa información para transferirla. En el caso de recepción de archivos, estos ya quedan en un espacio de memoria en donde la tarjeta maestra puede trabajar con la información, lo que evita otro tiempo de envío de la tarjeta interfaz a la tarjeta de memoria del sistema.

En resumen, el uso de la memoria doble puerto permite que los tiempos de transferencia de información del sistema a la tarjeta de interfaz IBUS-III <-> SCSI sean mínimos y en algunos casos nulos, ya que la información, una vez que se tiene en la memoria doble puerto (tarjeta interfaz), esta puede ser accesada por cualquiera de los dos dispositivos. En el caso de este prototipo, el espacio de memoria compartido es inferior a 1 Kbyte, considerando que algunos bytes son usados para control, por ejemplo, para indicar a que dispositivo se envía la información, el tamaño del archivo, bytes de estados, etc. pero es posible utilizar memorias doble puerto de mayor capacidad, o conectar varias, formando un banco de memoria, según la aplicación que se le de al sistema.

Una mejora que se le puede hacer al sistema es manejar toda la lógica de accesos y selecciones a través de PLD's (Programable Logic Device), con lo que se reduciría espacio y se aceleraría la activación de líneas de selección y comandos.

Las posibles mejoras son:

- un banco de memoria más grande
- utilizar adecuadamente los recursos del DUART
- uso de PLD's
- utilizar todas las ventajas que ofrece el microprocesador 80C188 (alta velocidad, controladores de DMA, timers, etc.)

Conclusiones:

Se logró un sistema muy adecuado para transferencias a disco, considerando que en la actualidad el proceso de almacenamiento de archivos es un tanto lento; el hardware que se diseñó es suficiente para establecer una comunicación con dispositivos comerciales, pero es necesario un desarrollo de software más específico para lograrlo. Se logró un equipo de bajo costo, ya que este se estima de 250 dólares.

Los objetivos no se lograron plenamente, ya que, aunque se tiene un dispositivo para la línea SAC de comunicación al canal SCSI, con la característica de terminación single-ended, para aplicaciones en un gabinete y bajo costo, no se logró una velocidad de transferencia superior a 1.0 Mbytes/s, sin embargo, ya se tiene una infraestructura para una posible aplicación con el canal SCSI, además de un antecedente en algunos circuitos que anteriormente no se habían utilizado.

Otro motivo por el cual no se lograron plenamente los objetivos, fue el hecho de que este trabajo era de interés particular del Ing. Carlos Masallach, antiguo jefe de departamento de electrónica en el IIE. Con su salida se perdió un gran apoyo para este proyecto. Además de esta situación, meses después se presentó la renuncia del asesor de tesis Ing. Ernesto Duhart, con lo que el proyecto quedó completamente sin apoyo. Sin embargo, agradezco el apoyo desinteresado del Ing. Pedro Molina F. ya que con su ayuda se logró la conclusión del proyecto, aunque esta no fue del todo satisfactoria. Lo lamentable de esta situación es que no sólo a este proyecto, sino a muchos otros, se les da un apoyo al inicio de los mismos, pero a la conclusión de estos, los proyectos son archivados y olvidados, con lo que tanto recursos humanos como económicos, son desperdiciados.

## BIBLIOGRAFIA

NORMA ANSI X3.131-1986  
Small Computer System Interface (SCSI)

REFERENCIA TECNICA IBUS III  
IIE DEPARTAMENTO DE ELECTRONICA  
2a. VERSION  
MOLINA, PEDRO Y ZARATE, ALEJANDRO  
1990

REFERENCIA TECNICA DE LA SAC-1887  
IIE DEPARTAMENTO DE ELECTRONICA  
JUNIO 1989

MSB-87 MANUAL DE USUARIOS  
IIE DEPARTAMENTO DE ELECTRONICA  
1987

MICROPROCESSOR AND PERIPHERAL HANDBOOK  
VOLUMEN I  
INTEL 1989

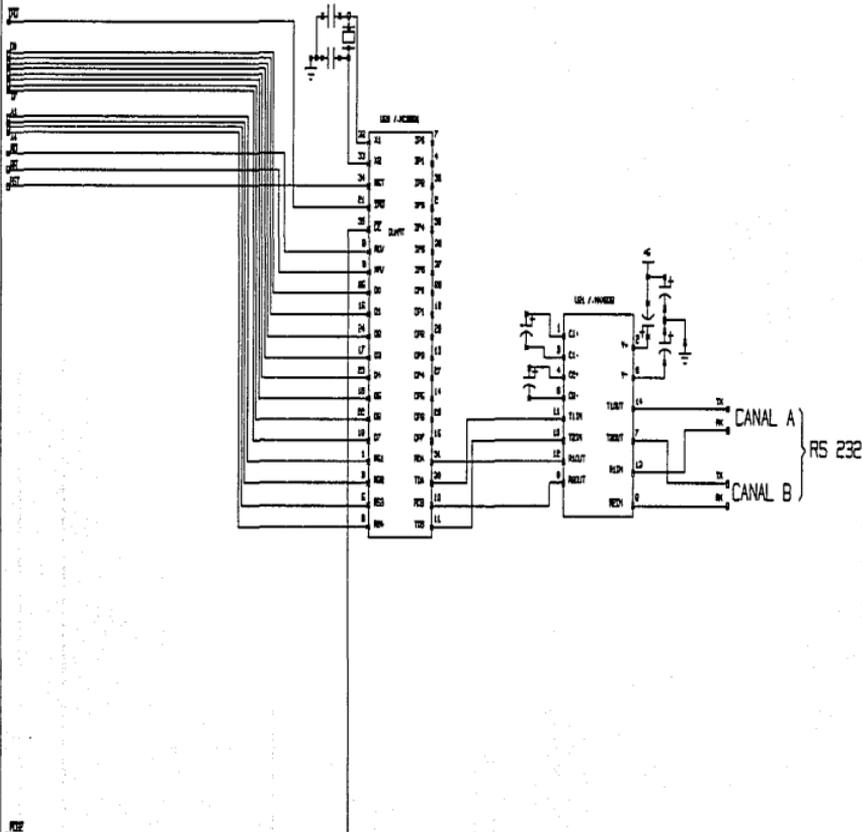
MICROCONTROLLER HANDBOOK  
VOLUMEN I  
INTEL 1989

STANDAR PRODUCTS DATA BOOK  
NCR 1988

SPECIALTY MEMORY PRODUCTS DATA BOOK  
AMD 1988

**APENDICE**

A continuación se presentan los diagramas eléctricos de la tarjeta interfaz SCSI <-> IBUS-III.



PROJETO Nº	SISTEMA DE		
	INDICAÇÕES ELETRONICAS		
	FORNIDA EM CARTAS DE		
	ELECTRONICA		
APROV	FECHA	TITULO	INTERNA Nº III - 100
DESENH		ESCALA	
REVISÃO			
APROV			

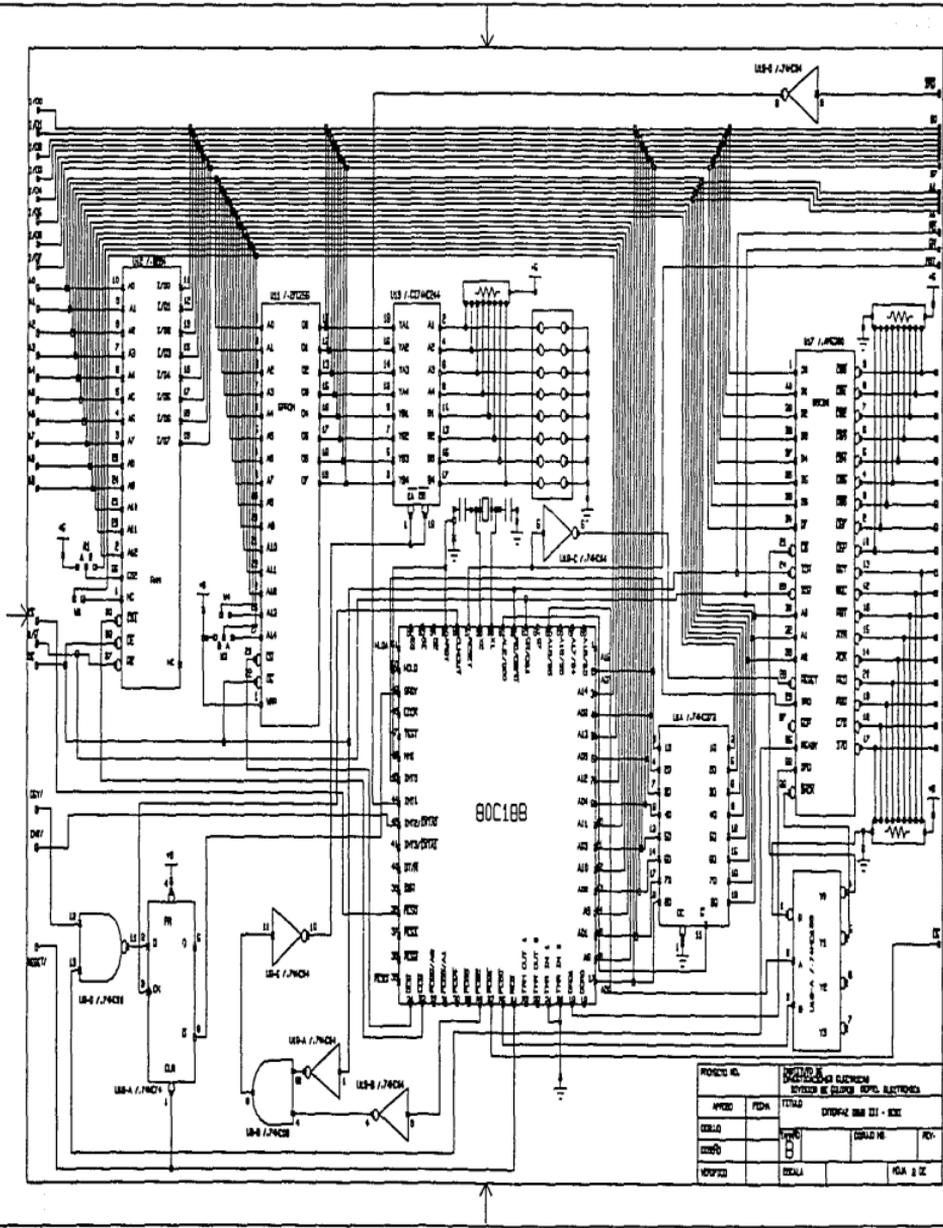


FIGURE NO.	74LS198-199-188 DECODER		
REVISED	DATE	BY	DESCRIPTION
1			74LS198-199-188 DECODER
2			74LS198-199-188 DECODER
3			74LS198-199-188 DECODER
4			74LS198-199-188 DECODER
5			74LS198-199-188 DECODER
6			74LS198-199-188 DECODER
7			74LS198-199-188 DECODER
8			74LS198-199-188 DECODER
9			74LS198-199-188 DECODER
10			74LS198-199-188 DECODER
11			74LS198-199-188 DECODER
12			74LS198-199-188 DECODER
13			74LS198-199-188 DECODER
14			74LS198-199-188 DECODER
15			74LS198-199-188 DECODER
16			74LS198-199-188 DECODER
17			74LS198-199-188 DECODER
18			74LS198-199-188 DECODER
19			74LS198-199-188 DECODER
20			74LS198-199-188 DECODER
21			74LS198-199-188 DECODER
22			74LS198-199-188 DECODER
23			74LS198-199-188 DECODER
24			74LS198-199-188 DECODER
25			74LS198-199-188 DECODER
26			74LS198-199-188 DECODER
27			74LS198-199-188 DECODER
28			74LS198-199-188 DECODER
29			74LS198-199-188 DECODER
30			74LS198-199-188 DECODER
31			74LS198-199-188 DECODER
32			74LS198-199-188 DECODER
33			74LS198-199-188 DECODER
34			74LS198-199-188 DECODER
35			74LS198-199-188 DECODER
36			74LS198-199-188 DECODER
37			74LS198-199-188 DECODER
38			74LS198-199-188 DECODER
39			74LS198-199-188 DECODER
40			74LS198-199-188 DECODER
41			74LS198-199-188 DECODER
42			74LS198-199-188 DECODER
43			74LS198-199-188 DECODER
44			74LS198-199-188 DECODER
45			74LS198-199-188 DECODER
46			74LS198-199-188 DECODER
47			74LS198-199-188 DECODER
48			74LS198-199-188 DECODER
49			74LS198-199-188 DECODER
50			74LS198-199-188 DECODER
51			74LS198-199-188 DECODER
52			74LS198-199-188 DECODER
53			74LS198-199-188 DECODER
54			74LS198-199-188 DECODER
55			74LS198-199-188 DECODER
56			74LS198-199-188 DECODER
57			74LS198-199-188 DECODER
58			74LS198-199-188 DECODER
59			74LS198-199-188 DECODER
60			74LS198-199-188 DECODER
61			74LS198-199-188 DECODER
62			74LS198-199-188 DECODER
63			74LS198-199-188 DECODER
64			74LS198-199-188 DECODER
65			74LS198-199-188 DECODER
66			74LS198-199-188 DECODER
67			74LS198-199-188 DECODER
68			74LS198-199-188 DECODER
69			74LS198-199-188 DECODER
70			74LS198-199-188 DECODER
71			74LS198-199-188 DECODER
72			74LS198-199-188 DECODER
73			74LS198-199-188 DECODER
74			74LS198-199-188 DECODER
75			74LS198-199-188 DECODER
76			74LS198-199-188 DECODER
77			74LS198-199-188 DECODER
78			74LS198-199-188 DECODER
79			74LS198-199-188 DECODER
80			74LS198-199-188 DECODER
81			74LS198-199-188 DECODER
82			74LS198-199-188 DECODER
83			74LS198-199-188 DECODER
84			74LS198-199-188 DECODER
85			74LS198-199-188 DECODER
86			74LS198-199-188 DECODER
87			74LS198-199-188 DECODER
88			74LS198-199-188 DECODER
89			74LS198-199-188 DECODER
90			74LS198-199-188 DECODER
91			74LS198-199-188 DECODER
92			74LS198-199-188 DECODER
93			74LS198-199-188 DECODER
94			74LS198-199-188 DECODER
95			74LS198-199-188 DECODER
96			74LS198-199-188 DECODER
97			74LS198-199-188 DECODER
98			74LS198-199-188 DECODER
99			74LS198-199-188 DECODER
100			74LS198-199-188 DECODER

