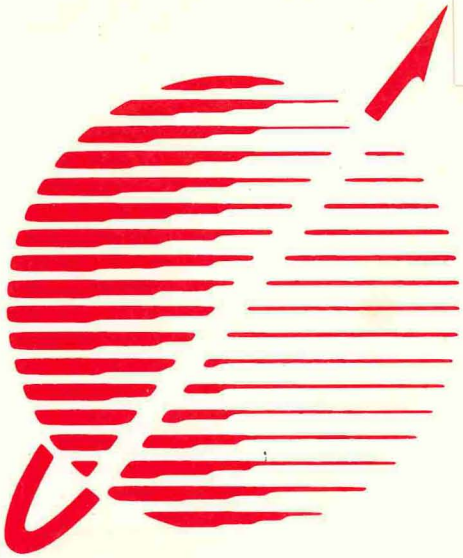


ICM

CPZ-186

SINGLE BOARD

CENTRAL PROCESSOR

4020 Leaverton Court • Anaheim, California 92807-1692 • (714) 630-0964 • Telex: 821375 SUPPORT UD

TABLE OF CONTENTS

L I S T O F T A B L E S

DISCLAIMER

Intercontinental Micro Systems Corp. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Intercontinental Micro Systems Corp. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Intercontinental Micro Systems Corp. to notify any person of such revision or changes.

1. INTRODUCTION

The **INTERCONTINENTAL MICRO SYSTEMS CORP.** (**ICM**) CPZ-186 single board central processor (SBCP) is a 80186 (tm) based computer board designed to meet or exceed the IEEE S-100 Bus specification. This fourth generation computer incorporates all the features necessary for a complete, stand alone CP/M (tm) or MSDOS (tm) system and is perfect for use in multi-processor or multi-user /multi-tasking architectures utilizing operating systems such as TurboDOS (tm), MP/M (tm), OASIS (tm), ZENIX (tm), UNIX (tm) and CP/NET (tm).

Features such as an independent interrupt structure, Direct Memory Access, a 4 Megabyte Memory Management Unit and a linearly addressable on-board 256Kbyte or 1 Megabyte memory coupled with I/O devices such as a floppy disk controller which controls 5 1/4" or 8" drives simultaneously, a 2-port serial controller and a 2 1/2-port parallel controller provides the user computing power on a single board heretofore unmatched in the S-100 Bus industry. Other features incorporated are listed as follows:

1.1. Technical Features

[ ]IEEE 696.1/D2 S-100 compliance. The CPZ-186 will interface with most IEEE S-100 Bus products on the market.

[ ] RS232 Communications and Floppy Controller Personality Boards included.

[ ] 8 MHz 80186 Operation

[ ] Floppy Disk Controller (FDC) with on-chip data separator. Single or double density. Single or double sided simultaneous 8" and 5-1/4" in any combination. The choice is yours, up to 4 drives.

[ ] Two synchronous (SCC) or asynchronous (ASCC) serial I/O channels. Both channels are programmable in interrupt or programmable I/O mode. One channel can be programmed in Direct Memory Access (DMA), interrupt, or programmable I/O mode.

[ ] Two parallel I/O channels (CIO). Both channels are programmable in interrupt or programmable I/O mode. One channel can be programmed in Direct Memory Access (DMA), interrupt, or programmable I/O mode.

[ ] Two channel on-chip DMA controller

[ ]256K Bytes of on-board RAM, expandable to 512KB or 1 Megabyte.

[ ] Memory Management Unit (MMU). Addresses up to 4 Megabytes of system memory.

[] Eighteen vectored priority interrupts.

[]Provisions for 2K, 4K or 8K of on-board EPROM. A boot up function in a 4K 2732 EPROM is supplied.

[] Software selectable baud rates. Eliminates costly, complicated hardware modifications to change baud rates. Up to 1 MegaBAUD in synchronous mode.

[] IBM Bisync, HDLC, SDLC and other protocols. All are handled through a Z8530 SCC chip. Permits communication with micro's, mini's or mainframes.

[] TurboDOStm operating system available.

[] DMA to extended memory.

1.2. Specifications

MICROPROCESSOR
Clock Rate ..... 8 MHz 80186

BUS INTERFACE...IEEE 696.1/D2 S100

DYNAMIC RAM MEMORY
  Capacity.....256K Bytes (64K x 1 DRAM's) or
               512KB or 1 MegaByte (256K x 1 DRAM's)
  Wait States...None

SERIAL I/O CHANNELS
  Synchronous Operation
    Baud Rate.....Up to 921.6K BAUD
    Data Transfer....DMA, interrupt or Programmed I/O
  Asynchronous Operation
    Baud Rate.....Up to 921.6K BAUD
    Clock Rate .....1, 16, 32, or 64 times Baud Rate
    Bits/Character....5, 6, 7 or 8
    Stop Bits.....1, 1-1/2 or 2
    Parity.....Odd, Even or None
    Data Transfer.....DMA, Interrupt or Programmed I/O
    I/O Interface.....Through Personality Boards

PARALLEL I/O CHANNELS
  DATA RATE.....Up to 500 KBytes/Sec
  Channel A Data Transfer...Interrupt, DMA or Programmed I/O
  Channel B Data Transfer...Interrupt or Programmed I/O
  Interface Signals.....16 Data Lines Plus 4 Handshaking Lines
  I/O Interface.....Through Personality Boards

FLOPPY DISK CONTROLLER
  Data Rate/8-inch Single-Density.....250,000 Bits/Sec
  Data Rate/8-inch Double-Density.....500,000 Bits/Sec
  Data Rate/5-1/4-inch Single-Density...125,000 Bits/Sec
  Data Rate/5-1/4-inch Double-Density...250,000 Bits/Sec
  Format.....IBM 3740 or 512 x 16 Sectors
  Data Transfer.....DMA, Interrupt or Programmed I/O
  I/O Interface.....Through Personality Boards

INTERRUPT CONTROL
  Number of Channels....18
  Priority.....Rotating or Fixed
  Interrupt Mode.....Master Cascade

REAL-TIME CLOCK
  Operation....Software Polled or Interrupt Driven
  Range.....37.5 Hz to 921.6 KHz

DIRECT MEMORY ACCESS CONTROLLER
  Channel 0.....Floppy Disk Controller
  Channel 1.....Channel A of ASCC Controller or Channel A
                of CIO Controller
  Memory-to-Memory utilizes both Channel 0 and 1

DIRECT EXTERNAL MEMORY TRANSFERS
  To/From ASCC (SCC), CIO, FDC or Memory

EPROM
  Type.....2716 2K, 2732 4K (standard), or 2764 8K
  Wait States...Three
  Function.....Boot up

POWER REQUIREMENTS
  Voltages...+8 VDC @ 3.0A (max)
            +16 VDC @ 0.2A (max)
            -16 VDC @ 0.2A (max)
  Power.....30.4W (max)

OPERATING ENVIRONMENT
  Temperature.....0 to 45 Degrees Celsius
  Relative Humidity...0 to 95%

CONSTRUCTION
  Circuit Board...Four Layer Glass Epoxy, Soldermask over copper
  All IC's in sockets
  Connectors.....Shrouded for Protection
  TESTING.....Completely tested and 24 hour burn-in
  WARRANTY.....One Year Warranty (Parts and Labor)

InterContinental Micro

Announcing!
Intercontinental Micro Systems'
CPZ-186
16-Bit Single Board Computer - Master Processor

TECHNICAL FEATURES

- ☐ IEEE 696.1/D2 S-100 compliance. The CPZ-186 will interface with most IEEE S-100 Bus products on the market.
- ☐ RS232 Communications and floppy controller Personality Boards included.
- ☐ 8MHz 80186 Operation
- ☐ Floppy Disk Controller (FDC) with on-chip data separator. Single or double density. Simultaneous 8" and 5¼" in any combination. The choice is yours, up to 4 drives.
- ☐ Two synchronous (SCC) or asynchronous (ASCC) serial I/O channels. One channel can be programmed in Direct Memory Access (DMA), interrupt, or programmable I/O mode.
- ☐ Two parallel I/O channels (CIO). Both channels are programmable in interrupt or programmable I/O mode.
- ☐ Two channel on-chip DMA controller.
- ☐ 256K Bytes of on-board RAM, expandable to 512KB or 1 Megabyte.
- ☐ Memory Management Unit (MMA). Addresses up to 4 megabytes of system memory.
- ☐ Eighteen vectored priority interrupts.
- ☐ Provisions for 2K, 4K or 8K of on-board EPROM. A boot up function in a 4K 2732 EPROM is supplied.
- ☐ Software selectable baud rates. Eliminates costly, complicated hardware modifications to change baud rates. Up to 1 MegaBAUD in synchronous mode.
- ☐ IBM Bisync, HDLC, SDLC and other protocols. All are handled through a Z8530 SCC chip. Permits communication with micro's, mini's or mainframes.
- ☐ TurboDOS™ operating system available.
- ☐ DMA to extended memory.

Specifications

MICROPROCESSOR
Clock Rate ..... 8 MHz 80186

BUS INTERFACE ..... IEEE 696.1/D2 S100

DYNAMIC RAM MEMORY
Capacity 256K Bytes (64K x 1 DRAM's) or 512KB or 1 MegaByte (256K x 1 DRAM's)
Wait States ..... None

SERIAL I/O CHANNELS
Synchronous Operation
Baud Rate ..... UP to 921.6K BAUD
Date Transfer ..... DMA, interrupt or Programmed I/O
Asynchronous Operation
Baud Rate ..... Up to 921.6K BAUD
Clock Rate ..... 1, 16, 32, or 64 times Baud Rate
Bits/Character ..... 5, 6, 7 or 8
Stop Bits ..... 1, 1½ or 2
Parity ..... Odd, Even or None
Date Transfer ..... DMA, Interrupt or Programmed I/O
I/O Interface ..... Through Personality Boards

PARALLEL I/O CHANNELS
DATA RATE ..... Up to 500 KBytes/Sec
Channel A Data Transfer ..... Interrupt or Programmed I/O
Channel B Data Transfer ..... Interrupt or Programmed I/O
Interface Signals ..... 16 DataLines Plus 4 Handshaking Lines
I/O Interface ..... Through Personality Boards

FLOPPY DISK CONTROLLER
Data Rate/8-inch Single-Density ..... 250,000 Bits/Sec
Data Rate/8-inch Double-Density ..... 500,000 Bits/Sec
Data Rate/5¼-inch Single Density ..... 125,000 Bits/Sec
Data Rate/5¼-inch Double-Density ..... 250,000 Bits/Sec
Format ..... IBM 3740 or 512 x 16 Sectors
Data Transfer ..... DMA, Interrupt or Programmed I/O
I/O Interface ..... Through Personality Boards

INTERRUPT CONTROL
Number of Channels 18
Priority ..... Rotating or Fixed
Interrupt Mode ..... Master Cascade

REAL-TIME CLOCK
Operation ..... Software Polled or Interrupt Driven
Range ..... 37.5 Hz to 921.6 KHz

DIRECT MEMORY ACCESS CONTROLLER
Channel 0 ..... Floppy Disk Controller
Channel 1 ..... Channel A of ASCC Controller or Channel A of CIO Controller
Memory-to-Memory utilizes both Channel 0 and 1

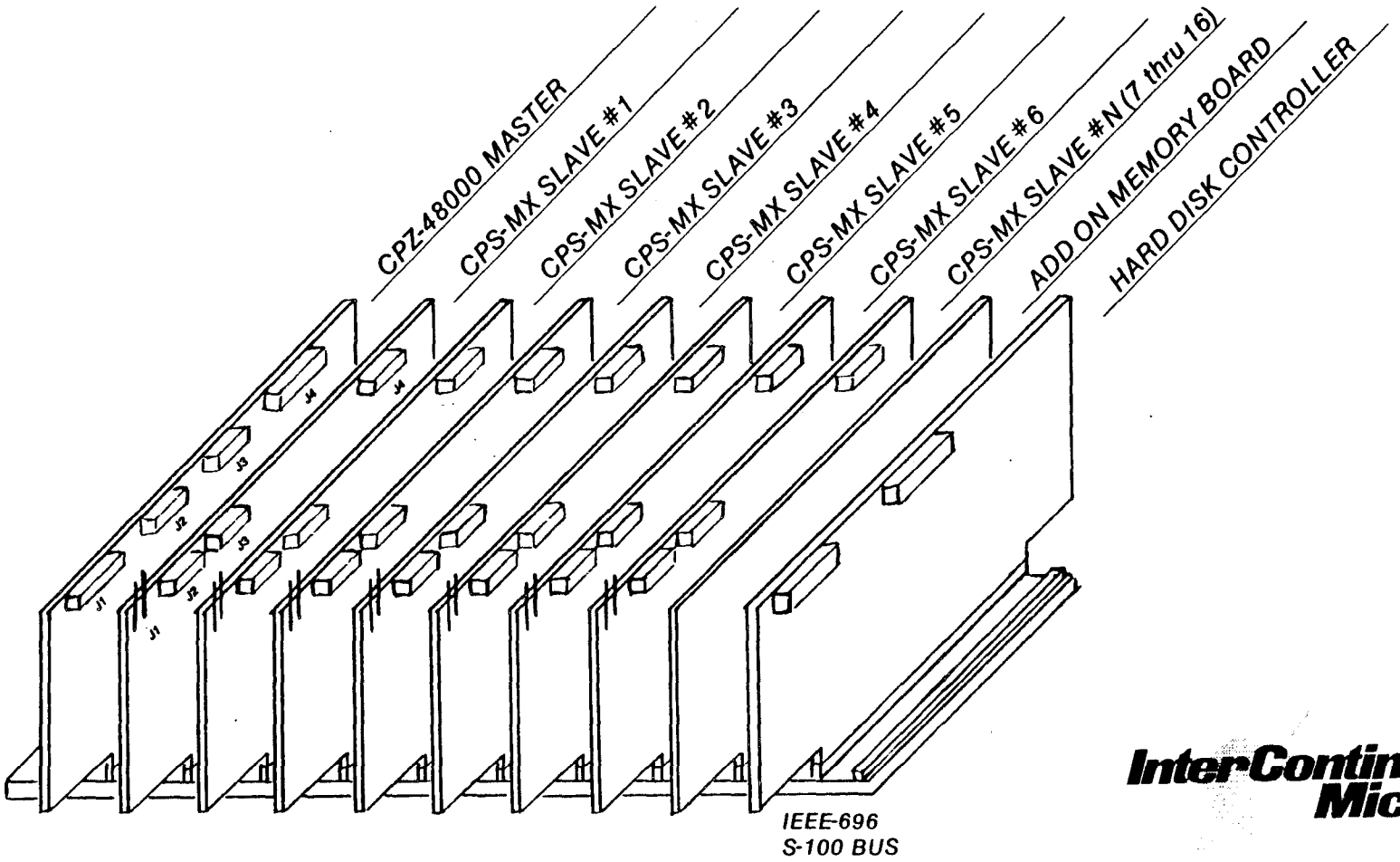
DIRECT EXTERNAL MEMORY TRANSFERS
To/From ASCC (SCC), CIO, FDC or Memory

EPROM
Type ..... 2716 2K, 2732 4K (standard), or 2764 8K
Wait States ..... Three
Function ..... Boot up

POWER REQUIREMENTS
Voltages ..... +8 VDC @ 3.0A (max)
+16 VDC @ 0.2A (max)
−16 VDC @ 0.2A (max)
Power ..... 30.4W (max)

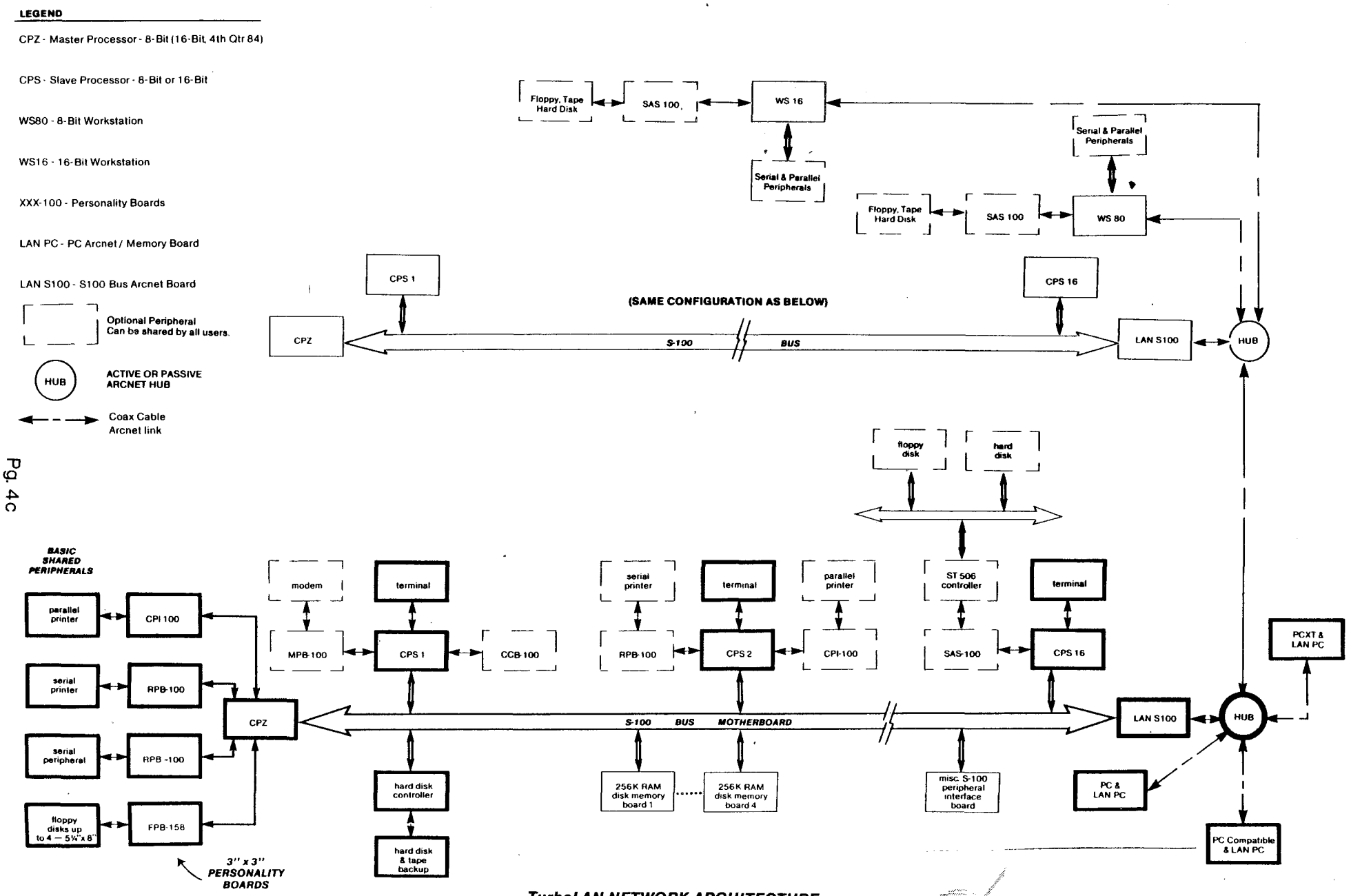
OPERATING ENVIRONMENT
Temperature ..... 0 to 45 Degrees Celsius
Relative Humidity ..... 0 to 95%

CONSTRUCTION
Circuit Board ..... Four Layer Glass Epoxy, Soldermask over copper
All IC's in sockets
Connectors ..... Shrouded for Protection
TESTING ..... Completely tested and 24 hour burn-in
WARRANTY ..... One Year Warranty (Parts and Labor)



InterContinental Micro

TYPICAL TURBODOS MASTER / SLAVE NETWORK



Pg. 4c

TurboLAN NETWORK ARCHITECTURE

InterContinental Micro

(see back for description)

INTERCONTINENTAL MICRO SYSTEMS
TurboLAN NETWORK ARCHITECTURE

ICM's revolutionary TurboLAN, multi-user architecture uses the sophisticated, cost effective TurboDOS Operating System. TurboLAN provides the flexibility of building multi-user systems with S-100 BUS Structured Networks and ARCnet protocol Local Area Networks. TurboLAN offers:

(A) Ability to network S-100 Bus Systems, IBM-PC's. PC Compatibles, ICM's WS80 & WS16 Workstations, and other computers with ARCnet capability
(B) Requires a Master Processor/File Server such as ICM's CPZ-4800X SBC
(C) Up to 400 users per network with 255 nodes per network segment
(D) Network self configures and efficiency increases as nodes are added
(E) Up to 40 miles between processors (Active Hub) or 2300 FT (Passive Hub)
(F) 2.5 MBIT/SEC data transfers across S-100 BUS, Coax cable, or twisted pair

TURBODOS OPERATING SYSTEM

TurboDOS is a true multi-user Operating System because it was designed from its beginnings to handle multiple computers running simultaneously on one or more networks. Each user is assigned an individual PC of a terminal attached to a Single Board Computer/Processor. This PC or individual processor is called a Slave Processor in the TurboDOS architecture and acts independently of all slaves on the network. A Master Processor, also known as a File Server, controls the network by downloading the operating system to each slave. The Master also downloads system files and orchestrates the use of all common peripherals. With its modular architecture, TurboDOS can increase the number of users or add peripherals on the network with a general software command that "links and patches". TurboDOS is the most sophisticated, yet cost effective Multi-user operating system available today. It offers FEATURES such as:

(1) Compatible with many OS's Application Software: CP/M, CP/M-86, PC-DOS, MS-DOS. MP/M, MP/M 86, CP.M PLUS, CONCURRENT CP/M 86
(2) Ability to mix Z-80, 8-bit; and 8086, 16-bit families of processors.
(3) Flexibility to build Bus Structured (Tightly Coupled) Networks and Local Area (Loosely Coupled) Networks using ICM's TurboLAN.
(4) Record and File Locking with File Sharing among multiple users
(5) Typically 300% faster than CP/M, MP/M, Oasis or similar multi-user, single processor, multi-tasking OS's
(6) 16 Logical Disk Drives per Master Processor/File Server
(7) Up to 1000 MB per drive and 134 MB per file
(8) 32 user areas (file libraries) on each disk
(9) 25% to 30% more floppy disk capacity
(10) Each user can independently STOP, RESUME of ABORT a program
(11) Sharing of costly peripherals and disk drives
(12) Read after Write verification of Floppy Disks and Hard Disks
(13) Logon/Logoff & Privileged/Non-privileged Security
(14) Background processing & Archival Back-up of files
(15) Queueing of multiple tasks-processing or printing
(16) Automatic Print Spooling
(17) Each individual TurboDOS OS has 4 Circuit Drivers with 255 nodes (slaves) per circuit driver = 1020 users per OS
(18) Multiple File Servers can be linked together with ICM's TurboLAN drivers — theoretically no limit to the number of users on a single network.

2. FUNCTIONAL DESCRIPTION

The CPZ-186 is functionally partitioned into the following major groups:

- INPUT/OUTPUT STRUCTURE
  - OFF-BOARD PERIPHERAL CONTROLLERS
    - SERIAL I/O PORT CONTROL
    - PARALLEL I/O PORT CONTROL
    - FLOPPY DISK CONTROL
  - ON-BOARD PERIPHERAL CONTROLLERS
    - DMA CONTROL
    - INTERRUPT CONTROL
    - MEMORY MANAGEMENT UNIT
- 256 KBYTE OR 1 MEGABYTE DYNAMIC RAM/LOGIC
- 2/4/8 KBYTE EPROM
- INPUT/OUTPUT CHIP SELECT LOGIC
- CPU CONTROL SIGNALS GENERATOR
- CLOCK GENERATOR
- POWER-ON CLEAR/RESET LOGIC
- S-100 BUS INTERFACE

Each group is described below to give the user a clear understanding of the hardware and software setup options and to give a full appreciation of the computing power available to the user. A block diagram is included in the following page.

2.1. Input/Output Structure

As a point of reference, an I/O device is defined as a device which, under program control of the 80186 CPU, controls a peripheral device or memory.

The I/O devices contained on the CPZ-186 consist of:

- 8531 ASCC (Serial Port Controller, SCC Optional)
- 8536 CIO (Parallel Port Controller, CIO)
- WD2793 (Floppy Disk Controller, FDC)
- 8259A (Universal Interrupt Controller, UIC)
- 74LS670 (Memory Management Unit, MMU)

Of these, the first three are used to communicate with off-board peripheral devices and will be referred to as the "OFF-BOARD" peripheral I/O controllers. The remaining are "ON-BOARD" I/O controllers.

Programmed I/O, Interrupt or Direct Memory Access (DMA) is possible to/from SCC port A, CIO port A and the FDC. No DMA is possible for SCC port B or CIO port B. The 80186 HOLD input line is assigned to the S-100 Bus DMA request line to allow temporary bus masters to capture the bus for DMA transfers to off-board memory. Fixed priority selection allows arbitration between internal DMA and external DMA requests from the S-100 Bus. Fixed priority gives the S-100 Bus the highest priority and the SCC or CIO ports, the lowest. The priority level of the Floppy port versus the SCC or CIO ports may be defined under software control. Thus,

DEVICE	PRIORITY
S-100 Bus	1 High
FDC	2 or 3 (under sftwr control)
SCC or CIO	2 or 3 (under sftwr control)

The CPZ-186 I/O PortAddress assignments are as follows:

0001 - FFFF	Available for External Use (except as noted below)
0002	Reserved for Internal Use
0004	Reserved for Internal Use
0006	Reserved for Internal Use
0008	Reserved for Internal Use
00F8 - 7FFF	Unavailable
8000 - FFFF	Available for External Use

T A B L E 2-1: I/O Port Address Assignments

Please refer to the SOFTWARE Section of this manual for further explanation.

2.1.1. Off-Board I/O Controllers

The Off-Board I/O Controllers consist of the Serial I/O Port Control, Parallel I/O Port Control and the Floppy Disk Control.

2.1.1.1. Serial I/O Port Control

The Serial I/O Port Control consists of the Serial I/O Controller and the Baud Rate Clock Generator.

2.1.1.2. Serial I/O Controller

The Serial I/O (SCC) Controller is a programmable dual channel device which provides formatting for serial data communications. The channels can handle either asynchronous or synchronous data transfers to/from serial peripheral devices. The SCC operates either under programmed I/O, Interrupt Control

or DMA control. DMA is provided for Port A only. All lines necessary to handle asynchronous, synchronous, synchronous bit oriented protocols and other serial protocols are available to the user at the interface connectors. In addition, +/- 16 volt DC and +5 volt DC power are available at these connectors. The SCC may be interfaced to peripheral devices requiring differing protocols. This interface is tailored to the exact device requirements by use of a Personality Module. The interface is implemented through two 16-pin Header connectors.

To program the ASCC or SCC the system software issues commands to initiate the mode of operation. Write registers exist for that purpose. In addition, read registers allow the programmer to read the status of each channel.

2.1.1.3. Baud Rate Clock Generator

The Baud Rate Clock Generator consists of a 1.8432 Mhz clock generator and an internal 80186 interval timer. The internal timer can, under software control, generate variable clock periods which are a multiple of the base input clock.

Two 80186 timer outputs are used to provide the SCC ports with a Baud Rate Clock. Channels 0 and 1 of the SCC are connected to these clock sources via jumper options PJA and PJB. These signals are also tied to the serial interface connectors. If clock signals are originated by the interfacing devices, the jumpers are cut appropriately. The jumpers provide for separate transmit and receive clock inputs from the interface connectors, J2 & J3, or may serve as baud rate generator outputs to the interface. This arrangement is intended to provide a clock to synchronous MODEM's via "external" clock in accordance with the EIA RS-232C standards. The modem can then return a transmit/receive clock to the serial controller. In summary, means are provided to implement serial interfaces accommodating asynchronous, synchronous, HDLC and a great number of currently defined communications protocols.

The Baud rates that can be derived from the 1.8432 Megahertz clock are listed as follows:

Baud Rate	Theoretical Frequency (16 x clock)
50	0.8 kiloHertz
75	1.2 kiloHertz
110	1.76 kiloHertz
134.5	2.152 kiloHertz
150	2.4 kiloHertz
300	4.8 kiloHertz
600	9.6 kiloHertz
1200	19.2 kiloHertz
1800	28.8 kiloHertz
2000	32.0 kiloHertz
2400	38.4 kiloHertz
3600	57.6 kiloHertz
4800	76.8 kiloHertz
7200	115.2 kiloHertz
9600	153.6 kiloHertz
19200	307.2 kiloHertz

T A B L E 2-2: Baud Rate Table

2.1.1.4. Parallel I/O Port Control Interface

The parallel I/O Port Control Interface consists of the Parallel I/O Controller (CIO). The Parallel I/O Controller is a programmable two-port LSI component, which interfaces peripheral devices to the 80186 microprocessor. The CIO provides data transfer to and from peripheral devices under programmed I/O, interrupt control or DMA control. Handshaking data transfer control lines are provided to the interface in addition to the two eight-bit data ports. The CPU reset line and the CPU clock are also connected to this interface. The CIO is flexible and may be connected to peripheral devices requiring differing protocols.

The interface is tailored to the exact device requirements by use of a "Personality Module". The Personality Module is a small external circuit board which connects to the CPZ-186 to provide the hardware drivers and receivers, logic and other circuitry as required. Refer to Personality Board Users Guide Section for a description of the parallel Personality Modules currently available.

An interrupt line is brought into the interface to give the user the capability of servicing interrupts. The interface is implemented through a 26-pin Header connector.

To program the CIO, the system software issues commands to initialize the mode of operation. Initialization is provided by loading the interrupt vector, mode, I/O and interrupt control registers.

2.1.1.5. Floppy Disk Controller (FDC)

The CPZ-186 uses the Western Digital WD2793 Floppy Disk Controller plus descrete support circuitry as the basis for the controller. A reliable phase-lock-loop circuit is implemented giving the user error free disk operation. Up to four 8-inch, four 5 1/4-inch or any combination of four 8 or 5 1/4-inch Floppy Disk drives may be connected. A mix of single- or double-sided drives and of single- or double-density drives may be interconnected. Any combination of single/double sided and single/double density drives may be connected.

The FDC is connected to the drives via Personality boards FPB100-XY or FPB158-XY and adaptor boards FPB100-XY or FPB158-XY The FPB158-30 accommodates both 5 1/4 and 8-inch drives by providing means to connect a 34 pin edge card connector for 5 1/4-inch drives and a 50 pin box connector for 8-inch drives. The FPB100-11 adapts 8-inch drives only and the FPB100-22 adapts 5 1/4-inch drives only. This technique greatly reduces the overall cost of interfacing to floppy drives. With a low cost personality board and even lower cost adaptor, the user may connect the drive configuration fitting their particular needs.

See the "PERSONALITY BOARD USERS GUIDE" section for clarification on the use of the Floppy Personality boards discussed above.

2.1.2. ON-BOARD I/O CONTROLLERS

The On-Board I/O controllers consist of the DMA Controller, Interrupt Control Logic and the Memory Management Unit.

2.1.2.1. DMA Controller

The DMA Controller is internal to the 80186 and consists of logic designed to allow external peripheral devices to transfer data directly to and from the on-board system memory. The use of this data transfer technique greatly enhances the system data throughput because the 80186 microprocessor CPU core does not have to deal directly with the transfers, and is free to perform other computing functions.

2.1.2.2. DMA Operations

The 80186 DMA Logic enables the programmer to free the CPU from the repetitive task of controlling data block transfers by providing hardware control over such operations. For example, the programmer may specify that a data block of "X" number of bytes contained in system memory starting at location "Y" is to be transferred. The programmer may further specify that at the end of said transfer an interrupt is to be generated (perhaps to initiate a subsequent transfer, or to determine the peripheral device status prior to initiating a subsequent transfer). Alternately, the programmer may wish to automatically re-initialize the data block transfer. Once the software command is transmitted to the DMA logic, it performs all of the indicated actions without further supervision from the 80186 CPU core. In all cases, the user of the CPZ-186 has full control over these parameters and events. Once the DMA transfer has begun (also enabled under software control), the CPU may then be used for other processing or for controlling other peripheral data transfers in a similar manner.

The DMA Control channels may be operated in either byte or word transfer mode. DMA transfers can occur between memory and I/O or between memory and memory. This means that in executing memory-to-memory transfers, DMA transfers can occur from on-board memory to off-board memory or from the I/O controllers to off-board memory. The transfer rate is 2 megabytes/sec. with DMA operating in burst mode.

2.1.2.3. DMA Channel Assignments

The CPZ-186 provides three channels of DMA. Channel 0 is dedicated to the S-100 Bus pHOLD line, channel 1 to the FDC Data Request Line and channel 2 to the SIO serial data channel A or channel A of the CIO parallel I/O port. The 80186 HOLD input line is assigned to the S-100 Bus DMA request line to allow temporary bus masters to capture the bus for DMA transfers to off-board memory. Fixed priority selection allows arbitration between internal DMA and external DMA requests from the S-100 Bus. Fixed priority gives the S-100 Bus the highest priority and the SCC or CIO ports, the lowest. The priority level of the Floppy port versus the SCC or CIO ports may be defined under software control. Thus,

DEVICE	PRIORITY
S-100 Bus	1 High
FDC	2 or 3 (under sftwr control)
SCC or CIO	2 or 3 (under sftwr control)

A memory-to-memory block transfer feature is provided which enables the user to transfer blocks of data from a source area of memory to a destination area of memory with an overall throughput increase of 5 times that available using Z80 (4 Mhz) block moves. Further, programming overhead is reduced in that the CPU need only initiate the DMA device and enable the DMA transfer. The CPU may then execute other code if so desired.

Combining DMA with the Memory Management Unit (MMU), a block of memory may be transferred from the on-board system memory to off-board system memory and vice-versa at DMA speeds. Additionally, FDC, SCC or CIO data may be transfered directly to off-board memory and vice-versa at DMA speeds. The MMU is loaded with appropriate address translation information. When the DMA transfers data to addresses translated by the MMU, the data is directed to the off-board memory. Memory-to-memory transfers within the on-board memory may also be made. While the 80186 executes block move transfers at 16 clock cycles per byte, the memory-to-memory function of the DMA controller will move a byte in 8 clock cycles, or 2 times faster.

The S-100 Bus channel (channel 0) is normally operated in "CASCADE" mode. The DMA Controller simply isolates the CPZ-186 from the S-100 Bus while the off-board DMA transfer occurs. The power of this technique is that any number of DMA type devices may reside on the S-100 Bus limited only by system data throughput considerations.

During power-up or reset, the DMA Controller is cleared to a state in which the Start/Stop bit for each channel will be set to STOP. Also, any transfer in progress is aborted.

Refer to the 80186 Data sheets for a detailed description of the DMA section.

2.1.2.4. Interrupt Control Logic

The interrupt control logic gives the CPZ-186 user the power to respond to the maskable interrupt (INT*) allowing the user a short form indirect call to any memory location within the 80186's address space.

2.1.2.5. Interrupt Controller/Select

The CPZ-186 interrupt controller consists of the 8259A Programmable Interrupt Controller. This is a LSI device which provides up to eight maskable interrupt request inputs. Upon receipt of an unmasked interrupt request, a byte of previously stored information is output to the data bus. This enables the CPU to process interrupt service routines by executing indirect jumps to those service routines. Expansion to the interrupt structure is provided by a priority technique in which enable in/enable out signals are connected in series ("daisy-chained").

The 8259A interrupt output is connected to Interrupt Channel 1 input of the 80186. The SCC and the CIO interrupt outputs are connected to channel 0 of the 80186 thus giving the SCC and CIO interrupts higher priority over the interrupts connected to the 8259A Controller. The higher priority interrupting device's enable input is set to logical ONE by permanently connecting it to a pull-up resistor. The SCC enable input line is pulled up to a logical ONE, its enable output line is tied to the enable input line of the CIO. The enable input line of the 8259A is pulled up. The eight interrupting channels are serviced on a fixed or rotating basis. Within the SCC, priority is fixed, Channel A is assigned a higher priority than Channel B. The receiver, transmitter, and external status are assigned priority in that order within each channel. Similarly, interrupt priority for the CIO is fixed, with Port A having higher priority than Port B.

During Interupt Acknowlege 0, the SCC or CIO will place its interrupt onto the data bus. During Interupt Acknowledge 1, the 8259A will place iits interrupt vectory onto the data bus.

In summary, the CPZ-186 interrupt priority daisy chain is as follows:

Priority	Device
1	SCC channel A receiver
2	SCC channel A transmitter
3	SCC channel A external status
4	SCC channel B receiver
5	SCC channel B transmitter
6	SCC channel B external status
7	CIO port A
8	CIO port B
9-16	8259A inputs (fixed or rotating)
17-nn	S-100 Bus interrupt device(s)

T A B L E 2-3: Interrupt Priority Daisy Chain

NOTE: Any I/O device in the S-100 Bus which uses the INT* line must use this priority chain scheme and must supply its own vector. The INT* signal must be connected to interupt 7 of the 8259A by jumpering B2 to C2 of block JB. The 8259A must be programmed so that the IR7 input has a slave. The I/O device must connect to the PCHAIN (Priority enable output Line, pin 21 of the S-100 Bus).

An additional feature of the CPZ-186 is that data transfers from the peripheral devices may be handled in a polled mode. This requires that the 8259A device be programmed for polled mode and the status register interrogated for the occurrence of the interrupt source signal. In polled mode no interrupts are generated, but the status signal indicating the occurrence of an event remains active. Having detected that occurrence, the remaining status is then interrogated to determine which of the eighteen events occurred.

Jumper options allow the user to choose among twelve S-100 Bus interrupt signals (VIO* to V17*, INT*, PWRFAIL*, NMI* and ERROR*), as well as two internally generated interrupt signals, FINT* and PINT* corresponding to the FDC interrupt and the parallel port interrupt. Four spare inputs are provided. The user selects eight of these signals to be inputs to the Interrupt Controller.

Signal	Source
VIO*-V17*	S-100 Bus
FINT*	FDC Interrupt
SERR*	S-100 BUS ERROR
PINT*	Parallel port interrupt

The S-100 Bus signal INT* is connected to the 8259A'S interrupt input 7 via jumper JB b2-to-c2.

The CPU's non-maskable interrupt line (INMI) may be selected to respond to signals on the S-100 Bus NMI* or PWRFAIL* line. Also, it may be left in its factory set default condition where it is permanently disabled. All of these options are implemented by use of jumper plugs.

2.1.2.6. MEMORY MANAGEMENT UNIT

The Memory Management Unit (MMU) consists of a 74LS670 4x4 Register File used as a memory mapping device plus associated logic to expand the 80186 20-bit address to 22 bits, increasing the addressing capability of the CPZ-186 from 1 Megabytes to 4 Megabytes. Two modes of operation are possible. These are the "PASS" and "MAP" modes. Bits A18 and A19 of the 80186 are input to the 74LS670. These bits address one of four 4-bit registers, the outputs of which are output on the address bus. In pass mode, the 80186's A18 and A19 address bits merely pass through the 74LS670 to the corresponding 74LS670 address outputs. The remaining 4 bits of extended address lines are forced to logic zero. In map mode, the contents of the addressed mapping register are output on the address bus. This technique proves to

be quite powerful since the extended address lines A20 and A21 appear on the bus dynamically. Address bits A22 and A23 are always at logic 0 level. This gives a total of 4 active bits of extended address constituting a "PAGE" address. The remaining 18 lower order address lines address the locations within each page. A "PAGE" consists of 256 Kbytes. There are sixteen 256 Kbyte pages to give a total of 4 megabytes of storage.

The Memory Management Unit allows the user to map any logical 256K block of memory to any physical 256K block within the 4 megabyte range. Thus, several programs or "TASKS" can share one main program by changing logical 256K block addresses.

2.2. 256 Kbyte/1 Megabyte Dynamic RAM/Controller

The 256 Kbyte/1 Megabyte Dynamic RAM consist of sixteen 64K-by-one-bit or 256K-by-one-bit Dynamic RAM's and the 4500A Dynamic RAM Controller.

Internal RAM Controller logic provides effective refresh techniques suitable for 80186 and S-100 Bus operations. The 4500A multiplexes 16-bit address lines (A1 through A16) to the RAM's. External logic multiplexes address lines A17 and A18 for the 9th multiplexed line required for 256 Kbyte RAM chip operation. A RAS/CAS/REFRESH circuit generates the required timing for the proper reading, writing and refresh operations of the RAM. RAM enable logic is provided to disable the on-board RAM when off-board RAM is addressed.

During cold-start boot-up, the first function is to open up a 4 or 8 Kbyte memory space for the boot-up EPROM. Next the EPROM contents is moved and executed. After boot-up, the EPROM is turned off and the full linear address of 1 megabyte becomes available.

2.3. 2K/4K/8K EPROM

The CPZ-186 may accommodate either a 2K (2716), 4K (2732) or 8K (2764) EPROM. A jumper (jumper PJJ) is made available to select either of the three EPROM types. The EPROM functions as both a boot-up and a monitor PROM. As a boot-up PROM, the EPROM contains the software routines necessary to manipulate the EPROM address and Deselect Circuitry and to load the required Disk Operating System contained on Floppy Disk Drive diskettes. The EPROM also contains monitor routines which are discussed in the SOFTWARE/PROM MONITOR sections.

2.4. I/O Chip Selects

The 80186 CPU generates the "chip select" signals for the SCC, CIO, Interrupt Controller, FDC, Memory Management Unit (MMU), Boot/Monitor Enable, Memory Deselect Logic, and the FDC Configuration Register.

2.5. POWER-ON CLEAR/RESET LOGIC

This logic provides reset signals to the CPU as well as to the S-100 Bus interface. The logic is activated under two conditions, when power is first applied to the board, and when the S-100 Bus signal RESET* is activated.

Signals asserted upon applying power are:

a. S-100 Bus signals POC*, RESET*, and SLAVE CLR*
b. Internal CPZ-186 reset

Signals asserted when RESET* is asserted are:

a. S-100 Bus signal SLAVE CLR*
b. Internal CPZ-186 reset

2.6. CLOCK GENERATOR

The 80186 CPU generates an 8 Mhz clock based on a 16 Mhz crystal connected to its inputs. The clock generator divides the 8 Megahertz clock signal to provide the internal CPZ-186 clock (8MHZ) and the S-100 Bus clocks (O and CLOCK). These clock signals are utilized to implement S-100 Bus signals in conformance with the IEEE standard for the S-100 Bus on a well-defined, clocked-logic basis.

2.7. S-100 BUS CONTROL SIGNALS GENERATOR

The S-100 Bus Control Signals Generator consists of the logic necessary to generate key S-100 Bus signals such as pSYNC, pSTVAL*, pWR*, pDBIN, pHLDA, sOUT, sM1, sINP, sMEMR, sHLTA, sINTA, sXTRQ and sWO.

The S100 Bus signal, SWO, may be output from the CPZ-186 so that the timing is in conformance with the write signal directly out of the 80186 CPU (pulsed mode) or with the timing in an NRZ form (latched mode). The CPZ-186 is factory configured for pulsed mode. Jumper PJL is provided to select the mode.

The S100 Bus signal, SMEMR, may be output from the CPZ-186 so that the timing is in conformance with the read signal directly out of the 80186 CPU (pulsed mode) or with the timing in an NRZ form (latched mode). The CPZ-186 is factory configured for pulsed mode. Jumper PJM is provided to select the mode.

2.8. S-100 Bus Interface

The S-100 Bus consists of 100 electrical signal lines. These are grouped into sets of lines used to transmit data and control among interconnected devices.

The groups are:

Group	No. of Lines
Address Bus	24
Input Data Bus	8
Output Data Bus	8
Status Bus	8
Control Input Bus	5
Control Output Bus	6
DMA Control Bus	8
Vectored Interrupt Bus	8
Utility Bus	8
System Power	9
Manufacturer specified lines	3
Reserved lines	5

Devices connected on the bus are classified as either bus masters or bus slaves and as either permanent or temporary masters. The CPZ-186 is a permanent bus master. Any other master connected to the S-100 bus may take control of the bus by making the appropriate DMA request provided no internal DMA by the SCC, FDC, or CIO is in progress. The S-100 Bus DMA request will be honored first if simultaneous DMA requests occur.

Each of the S-100 Bus signals utilized by the CPZ-186 are described on the following pages.

2.8.1. Address Bus

The address bus consists of 24 lines used to select a memory location or an input/output device during a bus cycle. All but A22 and A23 of the 24 address lines are active during a memory read, write or opcode fetch (M1) cycle unless the Memory Management Unit has been programmed for pass mode in which case the uppermost 4 bits (A20-A23) are forced to logic zero. The least significant byte of the address lines is active for input or output cycles. Address bus lines are enabled while ADSB* is inactive (no S-100 Bus DMA cycle in progress). The address bus lines are denoted as A0 through A23, with line A0 representing the least significant bit. Lines A0 through A7 compromise the least significant byte and lines A8 through A15 make up the "high" address byte with bits A16 through A23 constituting the extended address byte. Two octal-drivers and the Memory Management Unit are used to condition the lines in conformance with the characteristics required by the IEEE S-100 Bus standard.

2.8.2. Input Data Bus

There are eight input data lines (DI0-DI7) which are treated strictly as input when the CPZ-186 communicates with an 8 bit bus device. If a 16 bit bus device communicates with the CPZ-186, the Output Data Bus (DO0-DO7), which also consists of 8 bits, doubles up as an input bus to give a total of 16 data bits for input. The input data bus is enabled onto the CPZ-186 under the following conditions:

1. AN EXTERNAL I/O CYCLE IS INITIATED.
2. AN EXTERNAL MEMORY CYCLE IS INITIATED.
3. AN EXTERNAL DEVICE INTERRUPTS THE CPU AND PLACES A VECTOR ON THE DATA BUS.

2.8.3. Output Data Bus

There are eight data output lines (DO0-DO7) which are enabled by the signal DODSB*. A line driver conditions these lines to conform with the IEEE S-100 Bus standard. These 8 data lines are treated strictly as output when the CPZ-186 communicates with an 8 bit bus device. If a 16 bit bus device communicates with the CPZ-186, the Input Data Bus (DI0-DI7), which also consists of 8 bits, doubles up as an output bus to give a total of 16 data bits for output.

2.8.4. Status Bus

The status bus consists of seven output lines which define the current CPU bus cycle type. These lines are enabled while the enabling signal SDSB* is inactive. An eight line, sXTRQ, is output when the CPZ-186 is requesting 16 bit data transfers on the bus.

The eight lines of the Status Bus are:

Status	Function
sMEMR	Memory Read
sM1	Opcode Fetch
sINP	Input
sOUT	Output
sWO*	Write cycle
sINTA	Interrupt acknowledge
sHLTA	Halt acknowledge
sXTRQ	16-Bit Data Transfer Request

These are individually described below:

2.8.4.1. sMEMR (Memory Read)

sMEMR is a status signal indicating that a memory read cycle is in progress. This signal is valid during a normal memory read cycle (memory read or opcode fetch cycle).

sMEMR, may be output from the CPZ-186 so that the timing is in conformance with the read signal directly out of the 80186 CPU (pulsed mode) or with the timing in an NRZ form (latched mode). The CPZ-186 is factory configured for pulsed mode. Jumper PJM is provided to select the mode.

2.8.4.2. sM1 (Opcode Fetch)

sM1 is a status signal indicating that a memory read/opcode fetch cycle is in progress.

2.8.4.3. sINP (Input)

sINP is a status signal indicating that a peripheral device read cycle is in progress.

2.8.4.4. sOUT (Output)

sOUT is a status signal indicating that a peripheral device write cycle is in progress.

2.8.4.5. sWO* (Write Cycle)

sWO* is a status signal indicating that a write cycle is in progress, wherein data is transferred from an S-100 Bus master to a slave.

sWO*, may be output from the CPZ-186 so that the timing is in conformance with the write signal directly out of the 80186 CPU (pulsed mode) or with the timing in an NRZ form (latched mode). The CPZ-186 is factory configured for pulsed mode. Jumper PJL is provided to select the mode.

2.8.4.6. sINTA (Interrupt Acknowledge)

sINTA is a status signal indicating that an interrupt acknowledge cycle is in progress.

2.8.4.7. sHLTA (Halt Acknowledge)

sHLTA is a status signal indicating that the CPU is in a halt state.

2.8.4.8. sXTRQ* (16-Bit Data Transfer)

The status signal sXTRQ* (16-bit data transfer request) is used in the CPZ-186 to indicate to bus slaves that a 16 bit data transfer is requested by the CPZ-186. If the slave is able to respond to the masters request, it will cause the input signal, SIXTN* to go active which will then cause the CPZ-186 to transfer 16 bit data transfers using both 8 bit input and output data ports for bi-direction transfers. If the signal SIXTN* is inactive and if that transfer is output from the CPZ-186, the CPZ-186 will transfer the 16 bit data by sequencing that data out a byte at a time via the DO lines. If the signal SIXTN* is inactive and if that transfer is input to the CPZ-186, the CPZ-186 sequences in the 16 bits a byte at a time.

2.8.5. Control Input Bus

The Control Input Bus consists of six signals, five of which are used in the CPZ-186. These lines allow S-100 Bus slaves to synchronize the CPZ-186 with conditions internal to the bus slave, to request the relinquishment of the S-100 Bus (DMA request) and to disable the CPU from the S-100 Bus. The signals are conditioned by pull-up resistors and Schmitt-trigger input receivers.

The six lines of the Control Input Bus are:

Line	Function
RDY	Slave ready
XRDY	Special ready
INT*	Maskable interrupt request
NMI*	Non-maskable interrupt request
HOLD*	DMA request
SIXTN*	Sixteen acknowledge

These lines are described in the following paragraphs.

2.8.5.1. RDY (Slave Ready)

This control line is used by S-100 Bus slaves to suspend bus cycles by inserting wait states in a CPU cycle. Slaves may connect to this line by using an open-collector driver.

2.8.5.2. XRDY (Special Ready)

This control line is used as a special ready line to accommodate devices such as front panels. Only one slave device should connect into the XRDY line. This line also suspends bus cycles by introducing wait states to the CPU.

2.8.5.3. INT* (Maskable Int. Req.)

This control line is used to request service from the CPU on an interrupt basis. The INT* line is enabled (unmasked) or disabled (masked) under software control. When the INT* line is activated, the CPU responds with an acknowledge signal and subsequently gates the opcode or vector information asserted on the bus by the bus slave initiating the interrupt. Logic is provided to sense these conditions and to respond appropriately. INT* should be asserted as a continuous level and held active until a response is received.

2.8.5.4. NMI* (Non-maskable Int. Req.)

This control line is used to request service from the CPU on an interrupt basis. The NMI* is non-maskable, meaning it is always enabled. When an interrupt occurs on NMI*, a CPU acknowledge cycle is not generated.

Normally, only critical signals are connected to the NMI* line. The CPZ-186 provides the option to connect the S-100 Bus signal PWRFAIL* to the NMI* line via a jumper option. NMI* is sensed on a signal edge transition.

2.8.5.5. HOLD* (DMA Request)

This control line is used by S-100 temporary bus masters to request control of the S-100 Bus from the CPZ-186. A DMA cycle may be initiated by asserting this line. The CPZ-186 will respond with the signal pHLDA when the cycle is initiated, and will relinquish control to the temporary bus master.

2.8.5.6. SIXTN* (Sixteen Acknowledge)

The status signal sXTRQ* (16-bit data transfer request) is used in the CPZ-186 to indicate to bus slaves that a 16 bit data transfer is requested by the CPZ-186. If the slave is able to respond to the masters request, it will cause the input signal, SIXTN* to go active which will then cause the CPZ-186 to transfer 16 bit data transfers using both 8 bit input and output data ports for bi-direction transfers. If the signal SIXTN* is inactive and if that transfer is output from the CPZ-186, the CPZ-186 will transfer the 16 bit data by sequencing that data out a byte at a time via the DO lines. If the signal SIXTN* is inactive and if that transfer is input to the CPZ-186, the CPZ-186 sequences in the 16 bits a byte at a time.

2.8.6. Control Output Bus

The control output bus consists of five lines, one of which is optional. These lines are enabled when the enabling signal CDSB* is inactive. A line driver is used to condition these lines to conform with the characteristics required by the IEEE S-100 Bus standard.

The five lines of the Control Output Bus are:

Line	Function
pSYNC	Cycle start
pSTVAL*	Status valid
pDBIN	Read strobe
pWR*	Write strobe
pHLDA	Hold acknowledge

These lines are described in the following paragraphs.

2.8.6.1. pSYNC (Cycle Start)

pSYNC is a control signal which indicates the start of a new bus cycle. The signal becomes active when an I/O cycle, memory cycle, DMA read or DMA write cycle occurs. The signal remains active for approximately one bus clock in accordance with the IEEE S-100 Bus standard. pSYNC does not become active during a refresh cycle.

2.8.6.2. pSTVAL* (Status Valid)

pSTVAL* is a control signal which indicates that address, Data and Status signals have stabilized on the bus during the current bus cycle. It becomes active on the first CPU clock cycle after pSYNC becomes active, and goes inactive on the first CPU clock cycle after the bus cycle is complete. By using this signal as the latching signal, the address, data, and status signal timing will conform to the timing specified in the IEEE standard.

2.8.6.3. pDBIN (Read Strobe)

pDBIN is a control signal which gates data arriving on the CPU data bus from an external source. pDBIN goes active when the 80186 read signal goes active, thereby giving the user a maximum read access time window.

2.8.6.4. pWR* (Write Strobe)

pWR* is a control signal which performs the function of a write strobe to write data from the CPU data bus to an addressed peripheral or memory device. pWR* goes active at the specified time after pSTVAL goes active for I/O write cycles, DMA memory write cycles and CPU memory write cycles.

2.8.6.5. pHLDA (Hold Acknowledge)

pHLDA is a control signal which is active when the CPZ-186 relinquishes the address, data, control and status buses in response to a temporary master DMA request. This signal is generated by the 80186 Hold Acknowledge Output.

2.8.7. DMA Control Bus

The DMA Control Bus consists of eight input lines. Four of these are activated as required for the permanent bus master. The remaining four lines are utilized to isolate the CPU from the S-100 when the permanent bus master relinquishes control to the temporary bus master. The disable lines are connected to schmitt-trigger input receivers to provide noise immunity. The conditioned signals then disable the respective output line drivers. The DMA arbitration lines are used by the temporary masters to determine which temporary master has the use of the bus during a DMA cycle. The permanent bus master need not arbitrate. The eight DMA Control Bus lines are:

Line	Function
DMA0*	DMA arbitration line
DMA1*	DMA arbitration line
DMA2*	DMA arbitration line
DMA3*	DMA arbitration line
ADSB*	Address disable
DODSB*	Data out disable
SDSB*	Status disable
CDSB*	Control output disable

2.8.8. Vector Interrupt Bus

The Vector Interrupt Bus consists of eight lines, designated VI0* through V17*. VI0* is treated as the highest priority interrupt line. These lines should be asserted as levels, and should remain asserted until a response is received.

The Vectored Interrupt Bus lines are connected to interrupt option jumpers to connect the appropriate lines to the 8259A interrupt controller. This device then masks or unmasks the interrupts, prioritizes the requests, and asserts the INT* signal to the CPU.

2.8.9. Utility Bus

The Utility Bus consists of eight lines. Output lines are conditioned by drivers to conform with characteristics required by the IEEE S-100 Bus standard. The eight Utility Bus lines are:

Line	Function
0 (clock)	System clock (output)
CLOCK	Clock (output)
MWRITE	Memory write strobe (output)
POC*	Power-on clear (output)
SLAVE CLR*	Slave clear (output)
RESET*	Reset (input/output)

Each of these Utility Bus signals are described in the following paragraphs.

2.8.10. 0 (System Clock)

0 is the S-100 Bus system clock. 0 clock frequency is 8 Mhz.

2.8.11. CLOCK (Clock)

CLOCK is a 2 Megahertz Utility clock signal to be used by slave devices.

2.8.12. POC* (Power-on Clear)

The POC* line is active when initial power-up occurs on the S-100 Bus. When POC* is active, SLAVE CLR* and RESET* are asserted. POC* is guaranteed to stay active for at least 50 milliseconds.

2.8.13. SLAVE CLR* (Slave Clear)

SLAVE CLR* is the signal line which resets all slave devices on the S-100 bus. During power-on clear, this line is asserted by the CPZ-186 power-on clear logic. External devices may assert RESET* and, in doing so, assert SLAVE CLR* as well. RESET* is driven by an open-collector driver.

2.8.14. ERROR* (Error)

Error* is a signal generated by a slave device to indicate abnormal conditions such as parity error, CRC error, out of tape, etc. This line is connected to a jumper option where it may be selected as an interrupt source.

2.8.15. PWRFAIL* (Power Failure)

PWRFAIL* is a signal generated external to the CPZ-186 to indicate that a power failure has occurred. This signal remains active until power is restored and POC* is active. The signal is available to the user via a jumper so that it may be connected to the NMI* line of the CPU.

2.8.16. System Power

The system power lines consist of all lines supplying unregulated power to the CPZ-186 and other devices connected to the S-100 Bus. The nine System Power lines are:

Lines	Quantity	Pins
+8 VOLTS	2	1,51
+16 VOLTS	1	2
-16 VOLTS	1	52
GND	5	20,50,53,70,100

The +8 VOLT lines are connected to a +5 VDC regulator to supply +5 volt of regulated power to the CPZ-186. This line should not be greater than +10.0 VDC for best opperation.

The +16 VOLT connects to a +12 VDC regulator and the two serial port connectors. The -16 VDC line connects to the two serial port connectors. The 16 volt lines are utilized on the serial ports for supplying power to RS-232C driver circuitry.

All ground lines are connected to the ground plane to provide a low impedance path from the S-100 Bus ground to the CPZ-186 ground.

2.8.17. MANUFACTURER SPECIFIED LINES

The IEEE S-100 Bus standard reserves three of the 100 lines for special use by the manufacturer. The CPZ-186 utilizes these lines. One of these is required to implement the daisy chained priority interrupt expansion. The second supplies the 80186 refresh signal.

These lines may be connected through solder jumpers. See the section on Solder/Trace Cut Options.

The two Manufacturer specified lines are described in the following paragraphs.

2.8.17.1. PCHAIN (Interrupt Priority)

PCHAIN is an output signal which indicates the priority level of the interrupt in progress. If it is high, the interrupt response action is passed to the next interrupt device in the serial interrupt structure. PCHAIN utilizes pin 21 of the S-100 Bus.

2.8.17.2. RFSH* (Refresh)

RFSH* is the RAM Controller refresh signal buffered for use by external dynamic RAM memory devices connected to the S-100 Bus. RFSH* utilizes pin 66 of the S-100 Bus.

2.8.17.3. Reserved Lines

Five of the S-100 Bus lines are reserved for future use by the IEEE specification. The CPZ-186 makes no connection to these lines.

3. OPERATING INSTRUCTIONS

Instructions are given herein to configure the CPZ-186 from both the hardware and software standpoint. The user will be pleased to find that minimal setup procedures are required.

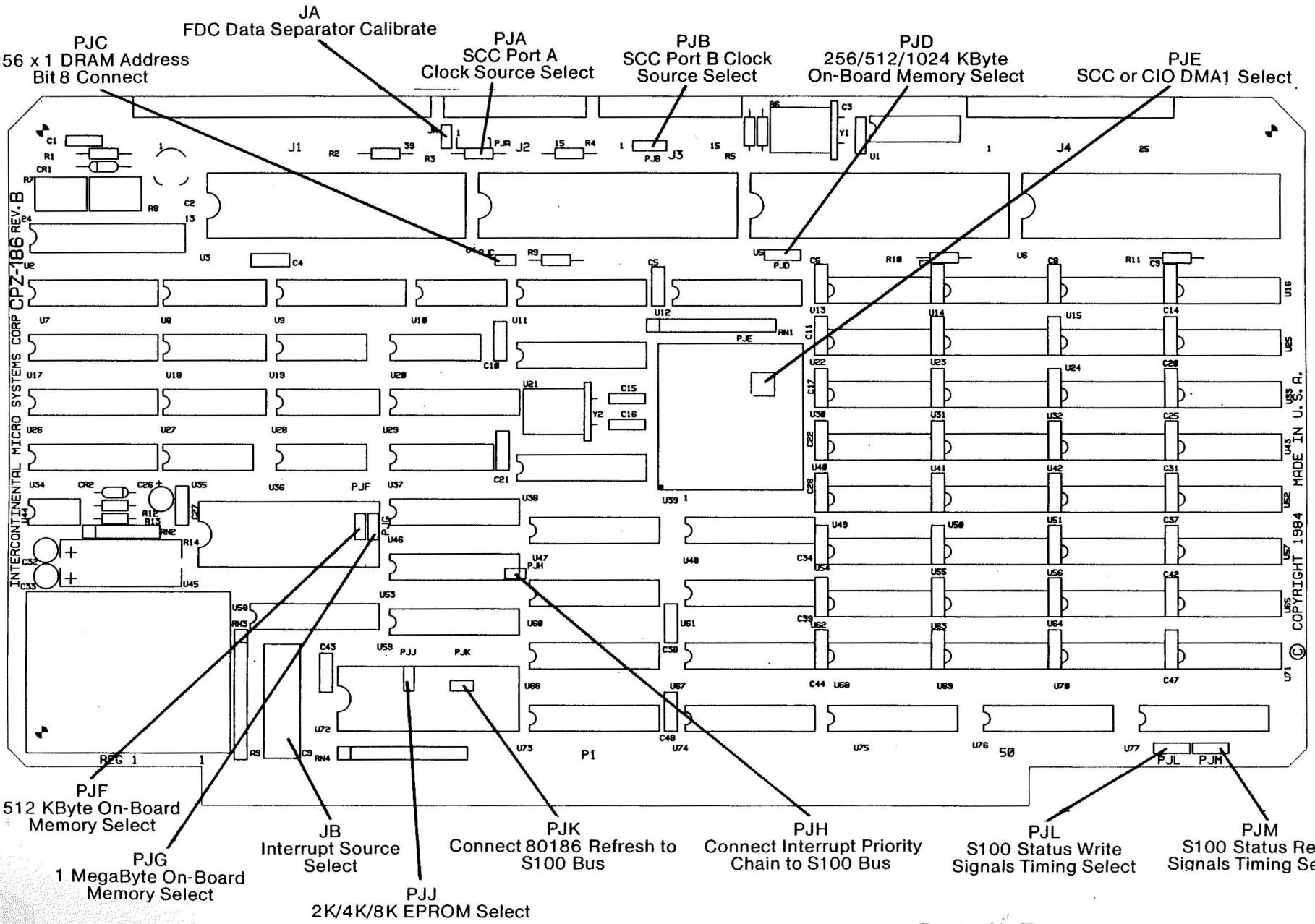
3.1. Hardware Setup Instructions

The hardware is configured via jumper options and solder/trace cut areas. The solder/trace cut areas are referred to as PJX, where X is the area designator. These jumpers are by nature rarely reconfigured. PJX options are located on the "solder" side of the board. The jumper options referred to as JX, where X is the jumper designator, gives the user flexibility in setting up the CPZ-186 for a multitude of applications. Jumper options are located on the "component" side of the board.

Instructions are also included on providing jumper option modifications for various popular floppy drives. These modifications must be executed prior to integrating the CPZ-186 to the floppy drives.

Refer to figure 1 to locate Jumper Options and Solder/Trace Cut areas.

A section is included on instructions for connecting personality boards to the CPZ-186.



- FIGURE 1 -
JUMPER - SOLDER/TRACE CUT OPTIONS

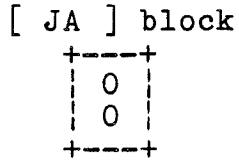
InterContinental Micro

3.1.1. JUMPER OPTIONS

Refer to figure 1 to locate JA and JB, the only jumper headers on the CPZ-186.

3.1.1.1. JA - FDC Data Separator Calibrate

This jumper option is for factory use only. The jumper is connected to enable the calibration of the FDC data separator circuitry through adjustments on potentiometers R7 & R8 and variable capacitor C2. These adjustments must not be modified in the field.

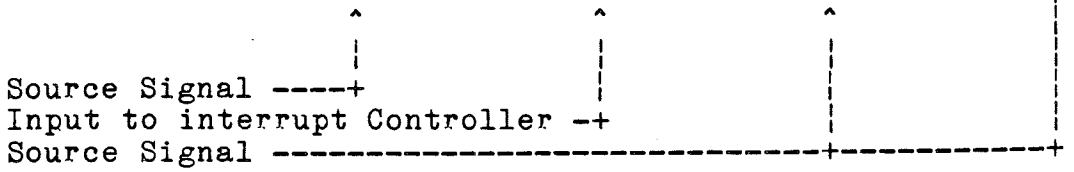


3.1.1.2. JB - Interrupt Source Select

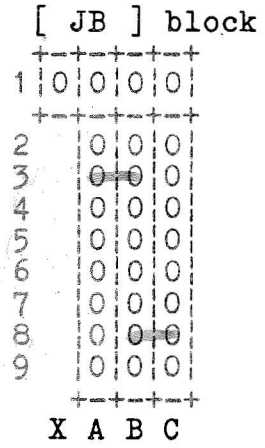
JB may be configured to select various signals as inputs to the interrupt controller. Jumpers are provided to select one of two signals available for each of 9 inputs. Wire-wrap or other means of interconnection may be used to select a signal in a different order from that assigned to the jumper block. This is clarified below:

Connection Table

Pin	C	B	A	X
1	NMI	INMI	GROUND	PWRFAIL
2	INT*	V7*	VI7*	
3	SERR*	V6*	VI6*	
4	PINT*	V5*	VI5*	
5	(SPARE)	V4*	VI4*	
6	(SPARE)	V3*	VI3*	
7	(SPARE)	V2*	VI2*	
8	FINT*	V1*	VI1*	
9	(SPARE)	VO*	VIO*	



** N O T E ** (B-1 is input to INMI of CPU)



EXAMPLES:

1) To connect S-100 Bus Vector line VI5* to the interrupt controller, install a jumper from position A4 to B4.
2) To connect Floppy interrupt signal FINT* to the interrupt controller, install a jumper from C8 to B8.
3) To connect the parallel port interrupt line to the highest priority interrupt input (V7*), install wire-wrap or any other adequate interconnection means from C4 to B9.

**** N O T E ****

a.) Highest priority input is V0 and the lowest is V7.
b.) NMI, GROUND and PWRFAIL are sources to the CPU non-maskable interrupt input, INMI. The CPZ-186 is normally delivered with ground connected to INMI. To connect either NMI or PWRFAIL to INMI, cut the trace from A1 to B1 and add the appropriate jumper. To connect PWRFAIL, solder a jumper from X1 to B1.

****
* WARNING *
****

ENSURE TRACE FROM A1 TO B1 IS CUT PRIOR
TO CONNECTING EITHER NMI OR PWRFAIL

c.) Signal source definition are as follows:

VIx = S-100 Bus vectored interrupt (X = 0 --> 7)
PWRFAIL = S-100 Bus power fail
FINT = Floppy interrupt
PINT = Parallel port interrupt
SERR = S-100 Bus error
NMI = S-100 Bus non-maskable interrupt
INT = S-100 Bus maskable interrupt

3.1.2. SOLDER/TRACE CUT OPTIONS

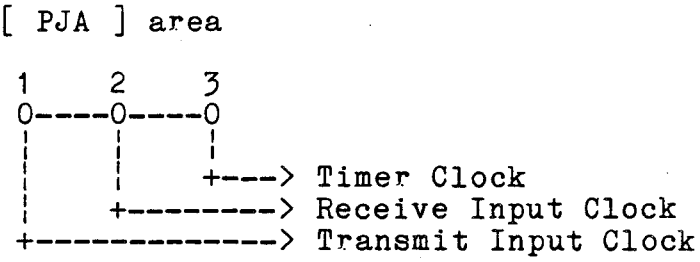
Refer to figure 1 to locate the PJX solder/trace options. The PJX options are listed as follows:

PJA - SCC Port A Clock Source Select
PJB - SCC Port B Clock Source Select
PJC - 256 x 1 DRAM Address Bit 8 Connect
PJD - 256/512/1024 KByte On-Board Memory Select
PJE - SCC or CIO DMA1 Select
PJF - 512 KByte On-Board Memory Select
PJG - 1 MegaByte On-Board Memory Select
PJH - Connect Interrupt Priority Chain to S100 Bus
PJJ - 2K/4K/8K EPROM Select
PJK - Connect 80186 Refresh to S100 Bus
PJL - S100 Status Write Signals Timing Select
PJM - S100 Status Read Signals Timing Select

3.1.2.1. PJA - SCC Port A Clock Source Select

The CPZ-186 comes configured so that the SCC ports receive their baud rate clocks from the 80186's on-chip programmable timers. The board could be reconfigured to source the clocks from the SCC serial port connectors. Such is the case when synchronous modems connect to the serial ports. The modem provides a clock to the SCC. Furthermore, the modem may receive the clock from the 80186's timer, condition the clock and return it to the input of the SCC. The transmit and receive clocks may be sourced separately on Port A. All combinations are possible through this jumper.

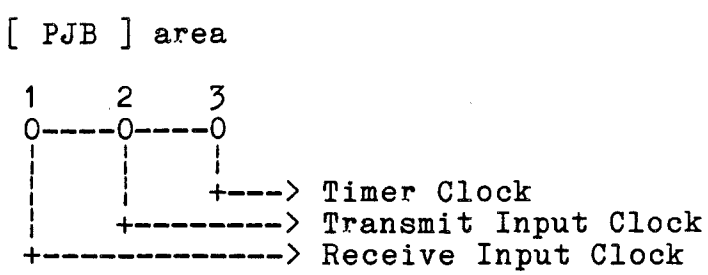
To source SCC PORT A inputs from the SCC connector only, cut the trace from PJA-2 to PJA-3. The source can now be connected through the personality board on either PIN P2-2 or P2-3. If the SCC PORT A inputs are to be sourced separately from the SCC connector, cut the trace from PJA-1 to PJA-2. The receive clock is now input on P2-3 and the transmit clock is input on P2-2.



3.1.2.2. PJB - SCC Port B Clock Source Select

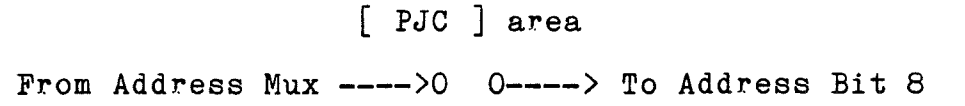
The CPZ-186 comes configured so that the SCC ports receive their baud rate clocks from the 80186's on-chip programmable timers. The board could be reconfigured to source the clocks from the SCC serial port connectors. Such is the case when synchronous modems connect to the serial ports. The modem provides a clock to the SCC. Furthermore, the modem may receive the clock from the 80186's timer, condition the clock and return it to the input of the SCC. The transmit and receive clocks may be sourced separately on Port B. All combinations are possible through this jumper.

To source SCC PORT B inputs from the SCC connector only, cut the trace from PJB-2 to PJB-1. The source can now be connected through the personality board on either PIN P3-2 or P3-3. If the SCC PORT B inputs are to be sourced separately from the SCC connector, cut the trace from PJB-1 to PJB-2. The receive clock is now input on P3-3 and the transmit clock is input on P3-2.



3.1.2.3. PJC - 256 x 1 DRAM Enable

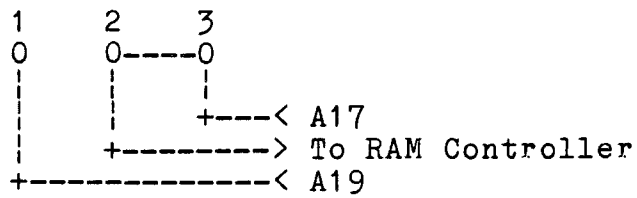
The CPZ-186 may be configured for 256 Kbytes of Dynamic RAM, 512 Kbytes or 1 Megabyte. Jumper area PJC is provided to input the multiplexed address lines A17 and A18 into the RAM array if the CPZ-186 is configured for 512 Kbytes or 1 Megabyte. Jumper area PJC is not connected if the CPZ-186 is configured for 256 Kbytes. It is connected if the CPZ-186 is configured for 512 Kbytes or 1 Megabyte.



3.1.2.4. PJD - 256/512/1024 K Select

The CPZ-186 may be configured for 256 Kbytes of Dynamic RAM, 512 Kbytes or 1 Megabytes. Jumper area PJD is provided to input the address lines A17, A19 or ground into the RAM Controller. If the CPZ-186 is configured for 256 Kbytes, A17 is input to the RAM Controller. If the CPZ-186 is configured for 512 Kbytes, ground is input to the RAM Controller and for 1 Megabyte, A19 is input to the RAM Controller. The CPZ-186 normally comes configured for 256 Kbytes.

[PJD] area



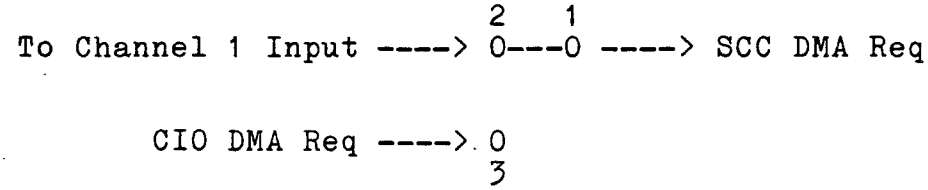
The CPZ-186 is normally configured for 256 Kbytes; therefore, PJD 2-to-3 is connected in order to input address line A17.

To configure the CPZ-186 for 512 Kbytes cut the trace from 2-to-3 and connect PJD-2 to U13 pin 16 (ground). For 1 Megabyte, cut the trace from 2-to-3 and connect PJD from 1-to-2 in order to input address line A19.

3.1.2.5. PJE - SCC or CIO DMA1 Select

Two channels of DMA are provided in the CPZ-186. Channel 0 is permanently assigned to the Floppy Controller signal "DRQ". Channel 1 may be optionally assigned to either the SCC Serial I/O controller or the CIO Parallel I/O Controller. The CPZ-186 is normally configured so that the SCC is connected to the DMA request channel. Note: PJE is located underneath U39.

[PJE] area (Solder Side View)



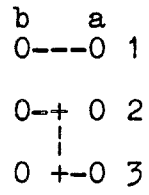
To configure the CPZ-186 for DMA operation on the CIO, cut the trace from 1-to-2 and solder a jumper from 2-to-3.

3.1.2.6. PJF - 512 KByte Memory Select

3.1.2.7. PJG - 1 Meg Memory Select

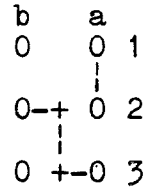
The CPZ-186 may be configured for 256 Kbytes of Dynamic RAM, 512 Kbytes or 1 Megabytes. Jumper areas PJF and PJG are provided to input the address lines A18 and/or A19 respectively, into logic which detects if either the on-board 256 Kbytes, 512 Kbytes or 1 Megabyte of RAM is being accessed or if external RAM is being accessed. These jumpers must be configured along with PJC & PJD when configuring the CPZ-186 for various RAM size options. The CPZ-186 is normally configured for 256 Kbytes of RAM and jumper PJF and PJG are set up as shown below. Note: Pads a1, a2 and b1 constitute jumper area PJF. Pads a3, b2 and b3 constitute jumper area PJG. Also PJF/PJG is located underneath U45.

[PJF/PJG] area (Solder Side View)



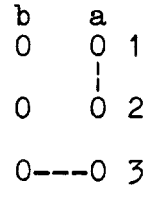
If the CPZ-186 is configured for 512 Kbytes, cut PJF a1-to-b1 and connect PJF a1-to-a2 as shown below.

[PJF/PJG] area (Solder Side View)



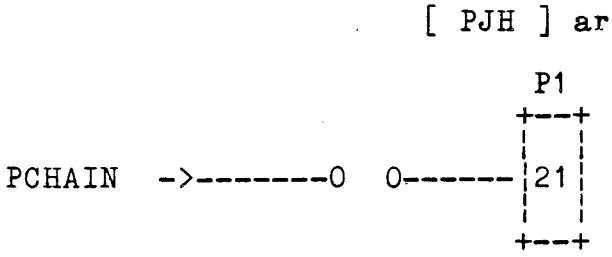
If the CPZ-186 is configured for 1 Megabyte, cut PJF a1-to-b1 and connect PJF a1-to-a2. Also cut PJG a3-to-b2 and connect PJG a3-to-b3 as shown below.

[PJF/PJG] area (Solder Side View)



3.1.2.8. PJH - Interrupt Priority Chain

The CPZ-186 may connect to off-board devices with priority interrupt structures which comply with the Intel Interrupt Controller 8259A method of resolving interrupt priority level. The method consists of serially chaining interrupt devices via a signal referred to as "PCHAIN". The CPZ-186 is factory configured so that this signal is not connected to the S-100 Bus. Solder a jumper in PJH if the interrupt structure is to be extended to other boards outside of the CPZ-186. Note PJH is located underneath U53.

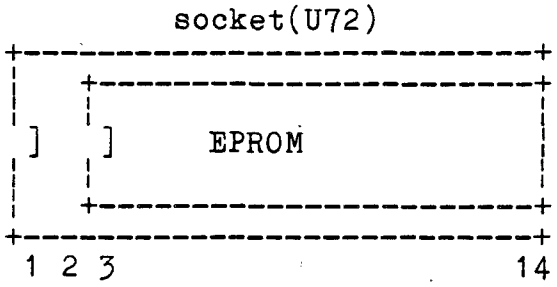


3.1.2.9. PJJ - 2K/4K/8K EPROM Select

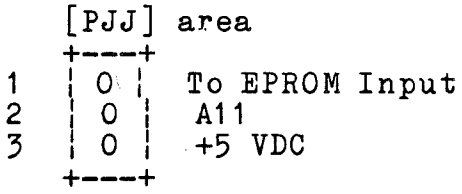
Either a 2716, 2732 or 2764 EPROM may be used with the CPZ-186. Instructions are given to incorporate any of these three EPROMS. Note: PJJ is located underneath U72.

2716/2732

A 28 pin socket is provided on the CPZ-186 in order to accommodate any of the three EPROM sizes. In the case of the 2716 and 2732, pin 1 of the EPROM must connect to pin three of the socket. This is illustrated below:



On jumper PJJ, for the 2716 connect 1-3 and for the 2732 connect 1-2.



2764

Install the EPROM in the normal fashion; i.e., pin 1 mates with pin 1. On jumper PJJ connect 1-2.

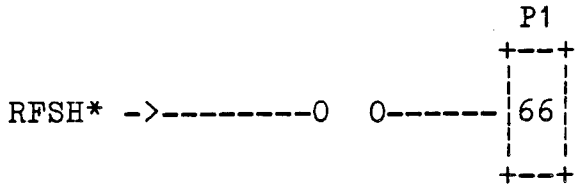
8K EPROM

4-5-87

3.1.2.10. PJK - Connect 80186 Refresh to Bus

If the S-100 Bus dynamic RAM memory boards require the 80186 refresh signal for proper operation, PJK may be connected to provide that signal.

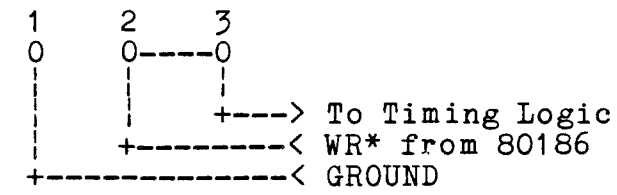
[PJK] area



3.1.2.11. PJL - Write Signal Timing

The S100 Bus signal, SW0, may be output from the CPZ-186 so that the timing is in conformance with the write signal directly out of the 80186 CPU (pulsed mode) or with the timing in an NRZ form (latched mode). The CPZ-186 is factory configured for pulsed mode.

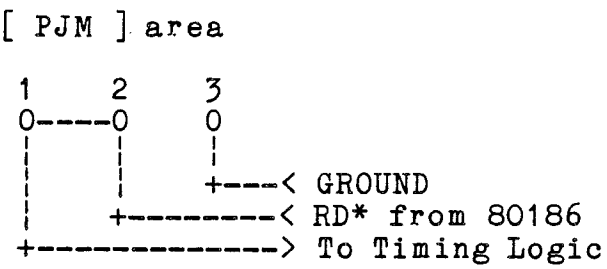
[PJL] area



To configure the CPZ-186 for Latched Write Mode, cut the trace from 2-to-3 and connect from 1-to-2.

3.1.2.12. PJM - Read Signal Timing

The S100 Bus signal, SMEMR, may be output from the CPZ-186 so that the timing is in conformance with the read signal directly out of the 80186 CPU (pulsed mode) or with the timing in an NRZ form (latched mode). The CPZ-186 is factory configured for pulsed mode.



To configure the CPZ-186 for Latched Read Mode, cut the trace from 1-to-2 and connect from 2-to-3.

3.1.3. FLOPPY DRIVE JUMPER OPTIONS

The CPZ-186 is compatible with all floppy drives which have Shugart standard interfaces. However various drives require particular jumper option settings to make them compatible with the CPZ-186 hardware. Jumper settings are given for the following Floppy drives:

SHUGART MODEL 800/801
SHUGART MODEL 850/851
QUME DATATRACK 8
TANDON TM848-1
TANDON TM848-2E
MITSUBISHI M2896-63
MITSUBISHI M2894-63
SEIMENS FDD100-8D
TANDON TM100-2

ICM technical support personnel shall provide the necessary assistance to customers wishing to integrate other drives compatible with the SHUGART standard interface.

3.1.3.1. SHUGART MODEL 800/801

(a) Do not modify etched trace options as delivered from the factory.

(b) Remove all jumpers on the disk drive and install the following:

A	Y
B	T1
C	T2
DS	800

(c) Install the following terminators in the last drive connected to the CPZ-186:

T3	T5
T4	T6

(d) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

3.1.3.2. SHUGART MODEL 850/851

(a) Do not modify etched trace options as delivered from the factory.

(b) Open the following shunts:

HL
X
S

(c) Remove all the jumpers on the disk drive and install the following:

A	FS
B	IW
C	RS
D	S2
I	2S
R	850
Z	

(d) Install the terminator pack in the last drive connected to the CPZ-186.

(e) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

3.1.3.3. QUME DATATRACK 8

(a) Close or open the following jumper options as indicated:

close	open
C	D
Y	DC
DS	DL
	HA
	T4O
	2S

(b) Close or open the following shunts as indicated:

close	open
A	X
B	Z
I	HL
R	

(c) Leave all other drive options as delivered from the factory.

(d) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

3.1.3.4. TANDON TM848-1

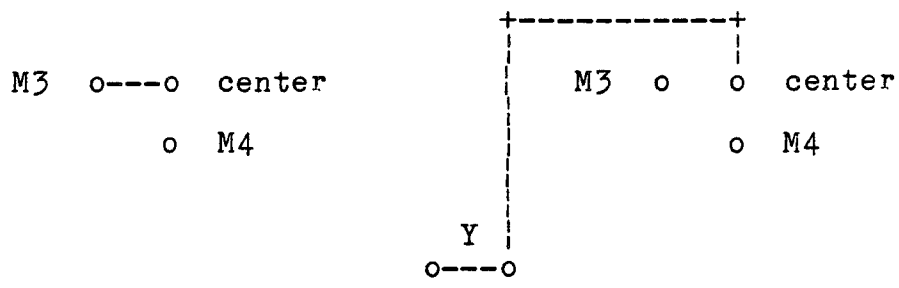
(a) Open the following shunts installed in U3:

1-to-16
2-to-15
5-to-12

(b) Open or close the following drive jumper options as indicated:

close	open
Y	M2
DS	
C	

(c) Remove connection from M3 to "center" and with a wire, connect "center" to Y as illustrated below:



(d) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

3.1.3.5. TANDON TM848-2E

(a) Install the following jumpers:

NL
M3
M4
IC
R

(b) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

(c) Install the terminator pack in the last drive in the chain.

3.1.3.6. **MITSUBISHI M2896-63 (HALF HEIGHT)**

(a) Open or close the following jumper options as indicated. All other jumper options are to remain as delivered from the factory except for the drive select jumpers and the terminator at location D2.

close	open
C	X
RM	RS

(b) Install the terminator at location D2 only on the last drive.

(c) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

3.1.3.7. MITSUBISHI 2894-63 (FULL HEIGHT)

(a) Open or close the following jumper options as indicated. All other jumper options are to remain as delivered from the factory except for the drive select jumpers and the terminator at location A5.

close	open
Z	Y
HUN	HUD

(b) Open the following shunts:

PJ4
PJ5
PJ8

(c) Install the terminator at location A5 only on the last drive.

(d) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

3.1.3.8. SEIMENS FDD100-8D

(a) Install jumper SS.

(b) Open trace from G to "center" and connect "center" to H as indicated below:

(c) Connect drive select jumpers as follows:

DS1= Drive A
DS2= Drive B
DS3= Drive C
DS4= Drive D

3.1.3.9. TANDON TM100-2 (5 1/4" drive)

Open or close the indicated shunts for drives A through D respectively:

	DRIVE A	DRIVE B	DRIVE C	DRIVE D
1-16	open	open	open	open
2-15	close	open	open	open
3-14	open	close	open	open
4-13	open	open	close	open
5-12	open	open	open	close
6-11	open	open	open	open
7-10	open	open	open	open
8-9	open	open	open	open

Personality Board Buyer's Guide
From Intercontinental Micro Systems

PERSONALITY BOARDS: PERIPHERAL INTERFACE

ICM'S PERSONALITY BOARDS offer a very versatile, cost effective technique for peripheral interface. The personality boards mount on the back of your system's mainframe or chassis. Whenever you decide to change peripherals or protocols, all you have to do is change the small, inexpensive personality board - not make expensive and complicated changes to your CPU board. ICM has existing PERSONALITY BOARDS and Software Drivers for most peripherals and protocols on the market today. New PERSONALITY BOARDS are constantly being developed as new peripherals enter the market.

PERSONALITY BOARDS are not only cost effective and inexpensive, they also protect the CPZ48000 SBC whenever new or different peripherals are placed on your system. First, since the buffers and drivers necessary to interface to the peripherals are on the personality board and not on the CPZ, there is no need to make costly and complicated cuts or jumpers on the CPZ. Second, any current surges on the communication lines between the CPZ and the peripheral will blow the surge suppressors on the Personality Board and not the CPZ. Thus you only have to replace a small, inexpensive board - not a large, expensive SBC board.

PERSONALITY BOARD - RS232/NO MODEM
Part Number - RPB100
Function

The RS232/NO MODEM Personality Board provides RS232 drivers and receivers, terminations and jumper options to interface any simple RS232 device such as CRT terminals, serial printers or any other serial device not requiring an extensive handshake protocol.

This module may be used with either the CPZ48000 SBCP or the CPS-MX SBSP.

PERSONALITY BOARD - RS232/FULL MODEM
Part Number - MPB100
Function

The RS232C/FULL MODEM Personality Board provides RS232 drivers/receivers and jumper options to interface asynchronous or synchronous modems with varying types of bit oriented protocols such as IBM Bi-Sync, HDLC or SDLC. Jumpers provided enable the user to configure the board for either asynchronous or synchronous operation.

This module may be used with either the CPZ48000 SBCP or the CPS-MX SBSP.

PERSONALITY BOARD - RS422 SERIAL COMMUNICATIONS
Part Number - FFT100
Function

The FFT100 personality board provides RS422 differential line drivers and receivers. These balanced drivers and receivers can provide serial communications for distances of up to 4000 feet at a communications rate of 100 Kbits/second. This assumes that 24 AWG twisted pair cable is used. Higher rates may be attained for shorter cable lengths. If the CPU's SIO controller is used in synchronous communications mode at its maximum rate of 800 Kbits/second, the maximum cable length recommended is 325 feet. Drivers and receivers are provided for all signals of the SIO to support full handshake protocols.

The FFT100 in combination with the Long Distance Serial Personality Board (LDS100), provides a means of connecting terminals, printers and other RS232 serial devices remotely located from the CPU mainframe. CPU-to-CPU communications may also be set-up over long distances by using the FFT100 at both CPUs. In this case, the interconnecting cable is cross-connected to tie receiver-to-transmitter and transmitter-to-receiver devices. No cross-connection is required between the FFT100 and the LDS100.

Jumper options are provided to minimize the number of cable lines required if no handshaking signals are required as in the case of simple RS232 Terminals where only transmit and receive signals are required. Ground is also provided but is not used in most cases.

This module may be used with either the CPZ48000 SBCP or the CPS-MX SBSP.

PERSONALITY BOARD - LONG DISTANCE SERIAL COMMUNICATIONS
Part Number - LDS100
Function

The LDS100 personality board provides RS422 differential line drivers and receivers. This SHORT HAUL MODEM can provide serial communications for distances of up to 4000 feet at a communications rate of 100 Kbits/second. This assumes that 24 AWG twisted pair cable is used. Drivers and receivers are provided to support full handshake protocols.

The LDS100 in combination with the RS422 Serial Communications Personality Board (FFT100), provides a means of connecting terminals, printers and other RS232 serial devices remotely located from the CPU mainframe. Jumper options are provided to minimize the number of cable lines connected if no handshaking signals are required as in the case of simple RS232 Terminals where only transmit and receive signals are used. Ground is also provided but is not used in most cases.

AC power must be provided to the board. The board may be strapped for either 115VAC/60HZ or 230VAC/50HZ operation.

PERSONALITY BOARD - FLOPPY DISK CONTROLLER
Part Number - FPB100-XY
Function

The FLOPPY DISK CONTROLLER personality board provides line drivers and receivers, terminators, logic and a jumper option to interface either an 8-inch of a 5¼-inch floppy disk drive with the CPZ48000 SBCP. A DB25 connector is available as the means to interface with the drive interface; however, if other types of commonly used connectors are required, adapters are available to tailor the interface appropriately.

This module is used only on the CPZ48000 SBCP.

PERSONALITY BOARD - CENTRONICS PRINTER
Part Number - CPI100
Function

The Centronics Printer Personality Board provides line drivers, receivers, terminators, jumper options and data strobe generator logic to interface to any printer compatible with the Centronics parallel interface.

This module may be used with either the CPZ48000 SBCP or the CPS-MX SBSP.

PERSONALITY BOARD - PRIAM INTELLIGENT HARD DISK
Part Number - PRI100
Function

PRIAM provides two intelligent hard disk interface controllers referred to as the "SMART" and the "SMART-E". These are preprogrammed microprocessor based controllers. They may be used for the entire line of PRIAM Winchester disc drives which range in capacity from 10 megabytes to 157 megabytes and come in eight or fourteen inch packaging. Up to four drives in any combination of drive sizes may be interconnected. The controllers support a variety of read sector, write sector and format commands. Data transfers may be either programmed I/O or DMA. The SMART-E has all the features that the SMART has in addition to error detection & correction, logical sector addressing, sector interleaving, parity generation & testing, direct data transfers and a 2 Kbyte data buffer (SMART has a 1 Kbyte buffer). The interface performs the entire function of detailed disc control while presenting to the host a basic and cost effective interface.

The PRI100 Personality Board connects the parallel port of the CPZ48000 SBCP or the CPS-MX SBSP to the SMART or SMART-E controllers. Thus, a very powerful disc subsystem may be directly connected to the ICM line of processors via the PRI100.

A jumper option is provided on the PRI100 to configure it for either the SMART or the SMART-E controller. The controllers mount along the drive sides alleviating the need for additional S-100 Bus slots. An adapter, PRI100-1, is provided allowing direct connection of the PRI100 to the smart controllers.

PERSONALITY BOARD - SHUGART ASSOCIATES SYSTEM INTERFACE
Part Number - SAS100
Function

The Shugart Associates System Interface (SASI) defines a Local I/O Bus which can be operated at data rates up to 1.5 megabytes per second. This bus provides I/O device independence so that disk drives, tape drives, printers and various other peripherals may be interfaced on the same I/O bus without modification to the host CPU's hardware or software. The interface protocol provides for connection of multiple initiators (devices capable of initiating an operation) and multiple targets (devices capable of responding to requests for operations). Arbitration logic is built in and a priority system awards control to the device that wins arbitration.

The SAS100 personality board converts the parallel port of either the CPZ48000 SBCP or the CPS-MX SBSP to a SASI I/O bus. Software is provided to emit bus timing in conformance with the SASI specification. The system integrator may interface SASI controllers such as the Data Technology Corporation, Zebec and Sysgen line of controllers. Each have powerful attributes such as connecting hard disks with floppies, hard disks with tape streamers and connecting to high performance SMD hard disk drives.

The SAS100 personality board is accompanied by an adapter board (SAS100-1). This adapter board converts the SAS100 DB25 connector interface to a 50 pin header connector interface with a pin assignment in exact conformance with the SASI Bus specification. The integrator may connect directly to the SAS100 with a DB25-to-SASI Interface cable or may connect via the SAS100-1 with a 50 pin flat ribbon cable.

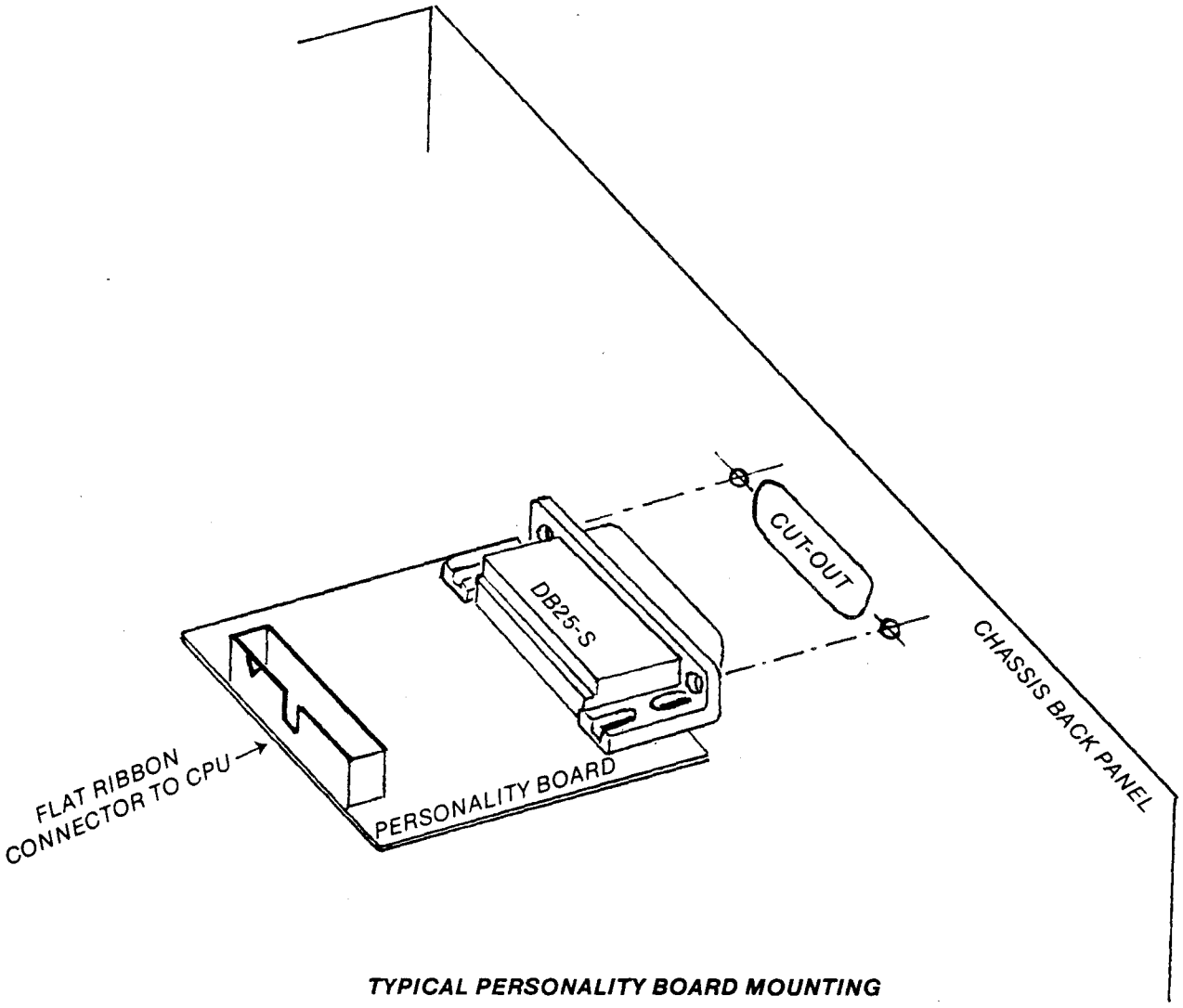
PERSONALITY BOARD - CLOCK/CALENDAR
Part Number - CCB100
Function

The CCB100 provides a highly accurate real time clock which may be set by the CPZ48000 SBCP or the CPS-MX SBSP under software control. The time of year, month, day, hour, minute and second is maintained and may be read back by the CPU. A Ni-Cad battery is used to provide backup power to the time control chip. In this manner the real time clock is continuously maintained even during extensive down time. This feature is quite useful for point-of-sale systems, inventory systems and other applications where continuous clock monitoring is required. This board is also very useful in operating systems which feature date and time stamping such as TurboDOS. In a TurboDOS based system, this board may be connected to the master (CPZ48000) parallel port or may be connected to any one slave (CPS-MX) parallel port.

Personality Board Index

		part number
1	RS232C/ NO MODEM	RPB100
2	RS232C/ FULL MODEM	MPB100
3	RS422 SERIAL COMMUNICATIONS	FFT100
4	LONG DISTANCE SERIAL COMMUNICATIONS	LDS100
5	FLOPPY DISK	FPB100
6	CENTRONICS PRINTER	CPI100
7	PRIAM INTELLIGENT HARD DISK	PRI100
8	SHUGART ASSOCIATES SYSTEMS INTERFACE	SAS100
9	CLOCK/CALENDAR (WITH BATTERY BACKUP)	CCB100

InterContinental Micro



TYPICAL PERSONALITY BOARD MOUNTING

4. PERSONALITY BOARD USERS GUIDE

PERSONALITY BOARD INDEX

4.1. INTRODUCTION

Since the introduction of the S100 Bus standard, compatability amongst S100 Bus products has been difficult to achieve. The CPZ-186 Single Board Central Processor(SBCP) solved these problems by effectively replacing four to five boards. By implementing the functions necessary to construct a system all on one board, interface and timing problems quickly disappeared. One problem, however, remained. The problem was to find an effective way of interfacing the SBCP with a great variety of peripheral devices without the necessity of modifying the SBCP for each devise. In the past, this entailed the necessity of modifying the PCB's with etch cuts and straps. This usually resulted in unattractive modifications to say nothing of the resultant inflexability for later integrating still other peripheral devices. This problem was effectively solved by customizing the peripheral interfaces through "personality" boards. Thus, the floppy, serial and parallel interfaces were brought out to connectors at the top of the board and those interfaces were tailored through small printed circuit boards connected to the main board by ribbon cables. In short, a personality board is a small circuit board containing line drivers/receivers, logic and other circuitry required to connect the CPZ-186 SBCP I/O controllers (Floppy Disk, Serial Controller and Parallel Controller) to a variety of peripheral devices.

ICM'S PERSONALITY BOARDS offer a very versatile, cost effective technique for peripheral interface. The personality boards mount on the back of your system's mainframe or chassis. Whenever you decide to change peripherals or protocols, all you have to do is change the small, inexpensive personality board - not make expensive and complicated changes to your CPU board. ICM has existing PERSONALITY BOARDS and Software Drivers for most peripherals and protocols on the market today. New PERSONALITY BOARDS are constantly being developed as new peripherals enter the market.

PERSONALITY BOARDS are not only cost effective and inexpensive, they also protect the CPZ-186 SBC whenever new or different peripherals are placed on your system. First, since the buffers and drivers necessary to interface to the peripherals are on the personality board and not on the CPZ, there is no need to make costly and complicated cuts or jumpers on the CPZ. Second, any current surges on the communication lines between the CPZ and the peripheral will blow the surge suppressors on the Personality Board and not the CPZ. Thus you only have to replace a small, inexpensive board not a large, expensive SBC board.

Typical S-100 Bus chassis provide DB25 connector cutouts at the chassis rear. The personality boards are designed to mount on DB25 connectors which in turn are mounted in the cutouts. In this manner, the personality boards do not require additional S-100 Bus slots and are conveniently mounted within the chassis. An additional connector is provided to connect the personality board to the SBCP. The connection is made with a simple point-to-point flat ribbon cable. See figure 1.

Intercontinental Micro Systems Corporation invites you, our valued customer, to submit your interface requirements if they are not covered by the line of personality boards available. Our engineering staff will evaluate those requirements and advice you of the feasibility of constructing your custom personality board.

4.2. Personality Board Interconnection Instructions

The CPZ-186 has four connectors at the top of the board numbered J1 through J4. These are listed below:

J1 - FDC Connector
J2 - DART/SIO Port A Connector
J3 - DART/SIO Port B Connector
J4 - PIO Connector

Tables A through D describe signal pin assignments for connectors J1 through J4 respectively.

At a minimum, the FDC and DART/SIO Port B personality boards must be installed. The instructions follow:

1.- Select a DB25 connector cutout at the rear of the chassis for the FDC personality board.
2.- Insert and hold the FDC personality board in the cutout. External to the chassis, plug in the desired connector adaptor and hold in place.
3.- Install #6 nuts, washers and bolts passing the bolts through the connector adapter and through the personality board's DB25 connector.
4.- Install the flat ribbon cable provided at the personality board and at the CPZ-186, connector J1.
5.- Follow the above procedure, except that an adapter is not used, for the DART/SIO Port B personality board.
6.- Install cables from the chassis connectors to the respective peripherals.

Table A

T A B L E 4-1: Connector J1 Pin Assignments

PIN NO.	SIGNAL NAME	DESCRIPTION
1	INT7*	INTERRUPT (LEVEL 7) TO CPU
2	DS1*	DRIVE SELECT #1 FROM CPU
3	GND	GROUND
4	DS2*	DRIVE SELECT #2 FROM CPU
5	GND	GROUND
6	DS3*	DRIVE SELECT #3 FROM CPU
7	GND	GROUND
8	DS4*	DRIVE SELECT #4 FROM CPU
9	GND	GROUND
10	DIRC	DIRECTION CONTROL FROM CPU
11	GND	GROUND
12	STEP	STEP CONTROL FROM CPU
13	GND	GROUND
14	WRITE DATA	WRITE DATA FROM CPU
15	GND	GROUND
16	WGATE	WRITE GATE FROM CPU
17	GND	GROUND
18	TRACK 0*	TRACK 0 STATUS TO CPU
19	GND	GROUND
20	WRITE PROT*	WRITE PROTECT TO CPU
21	GND	GROUND
22	READ DATA*	READ DATA TO CPU
23	GND	GROUND
24	SSO	SIDE SELECT OUTPUT FROM CPU
25	GND	GROUND
26	HLD	HEAD LOAD COMMAND FROM CPU
27	GND	GROUND
28	INDEX*	INDEX PULSE TO CPU
29	GND	GROUND
30	READY	READY STATUS TO CPU
31	GND	GROUND
32	MOTOR ON	MOTOR ON STATUS FROM CPU
33	GND	GROUND
34	TK43	TRACK 43 STATUS FROM CPU
35	GND	GROUND
36	+8VDC	+8VDC
37	GND	GROUND
38	HLTIMER	HEAD LOAD TIMER TO CPU
39	GND	GROUND
40	+5VDC	+5VDC

Table B

T A B L E 4-2: Connector J2 Pin Assignments

PIN NO.	SIGNAL NAME	DESCRIPTION
1	ADSR*	DATA SET READY TO CPU
2	ATXC*	TRANSMIT CLOCK TO/FROM CPU
3	ARXC*	RECEIVE CLOCK TO/FROM CPU
4	ATxD	TRANSMIT DATA FROM CPU
5	ARxD	RECEIVE DATA TO CPU
6	ARTS*	REQUEST TO SEND DATA FROM CPU
7	ACTS*	CLEAR TO SEND TO CPU
8	ADCD*	DATA CARRIER DETECT TO CPU
9	ADTR*	DATA TERMINAL READY FROM CPU
10	ARNG*	RINGING INDICATOR TO CPU
11	ABRCLK	BAUD RATE CLOCK FROM CPU
12	GND	GROUND
13	+16VDC	+16VDC
14	-16VDC	-16VDC
15	+5VDC	+5VDC
16	GND	GND

TABLE C

T A B L E 4-3: Connector J3 Pin Assignments

PIN NO.	SIGNAL NAME	DESCRIPTION
1	BDSR*	DATA SET READY TO CPU
2	BTXC*	TRANSMIT CLOCK TO/FROM CPU
3	BRXC*	RECEIVE CLOCK TO/FROM CPU
4	BTxD	TRANSMIT DATA FROM CPU
5	BRxD	RECEIVE DATA TO CPU
6	BRTS*	REQUEST TO SEND DATA FROM CPU
7	BCTS*	CLEAR TO SEND TO CPU
8	BDCD*	DATA CARRIER DETECT TO CPU
9	BDTR*	DATA TERMINAL READY FROM CPU
10	BRNG*	RINGING INDICATOR TO CPU
11	BBRCLK	BAUD RATE CLOCK FROM CPU
12	GND	GROUND
13	+16VDC	+16VDC
14	-16VDC	-16VDC
15	+5VDC	+5VDC
16	GND	GND

TABLE D

T A B L E 4-4: Connector J4 Pin Assignments

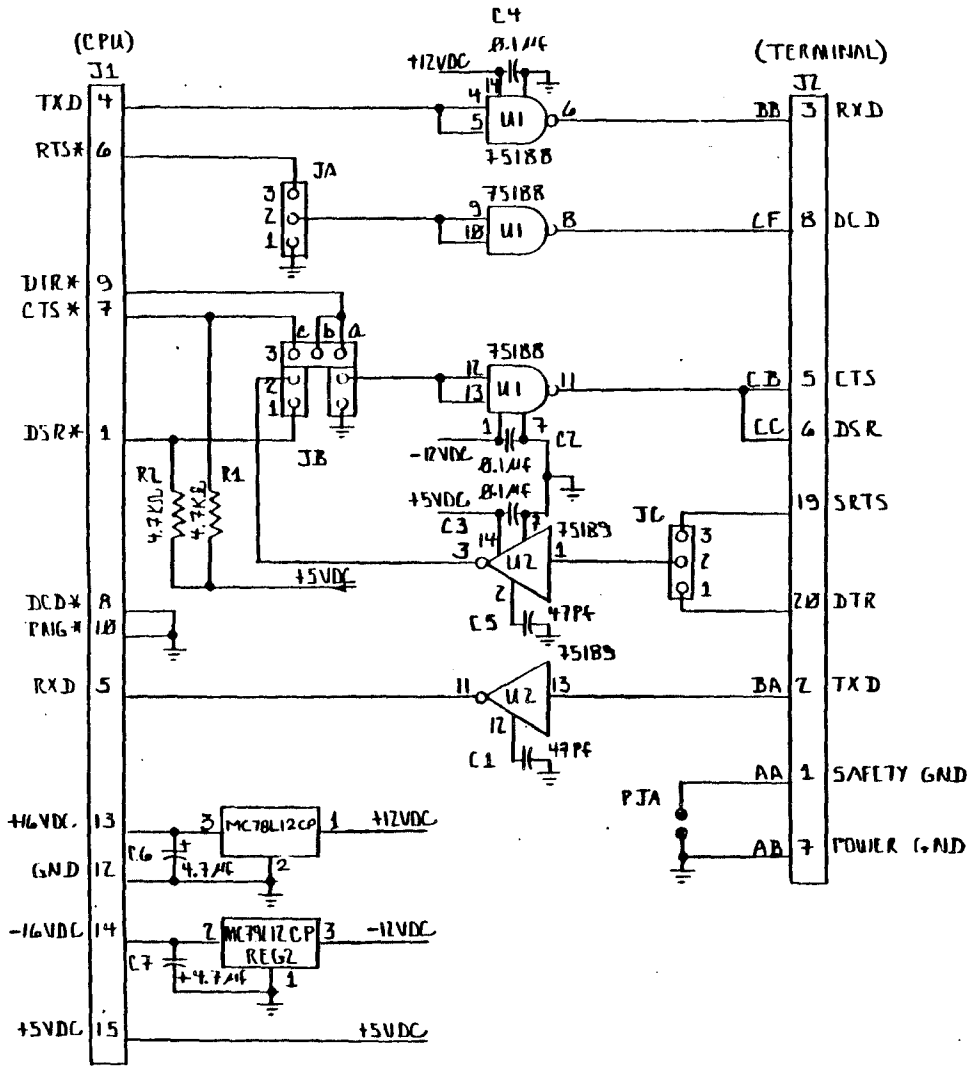
PIN NO.	SIGNAL NAME	DESCRIPTION
1	RDYA	PORT A READY TO PERIPHERAL
2	STBA*	PORT A STROBE TO CPU
3	RDYB	PORT B READY TO PERIPHERAL
4	STBB*	PORT B STROBE TO CPU
5	D0A	PORT A DATA BIT 0
6	D1A	PORT A DATA BIT 1
7	D2A	PORT A DATA BIT 2
8	D3A	PORT A DATA BIT 3
9	D4A	PORT A DATA BIT 4
10	D5A	PORT A DATA BIT 5
11	D6A	PORT A DATA BIT 6
12	D7A	PORT A DATA BIT 7
13	D0B	PORT B DATA BIT 0
14	D1B	PORT B DATA BIT 1
15	D2B	PORT B DATA BIT 2
16	D3B	PORT B DATA BIT 3
17	D4B	PORT B DATA BIT 4
18	D5B	PORT B DATA BIT 5
19	D6B	PORT B DATA BIT 6
20	D7B	PORT B DATA BIT 7
21	RESET*	SYSTEM RESET FROM CPU
22	GND	GROUND
23	PINT*	PARALLEL PORT INTERRUPT TO CPU
24	GND	GROUND
25	PCLK	PARALLEL PORT CLOCK FROM CPU
26	+5VDC	+5VDC

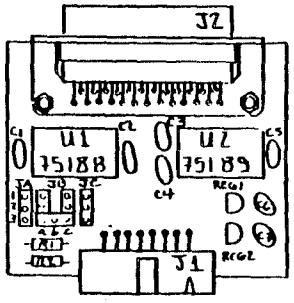
4.2.1. Description

Each personality board shall be described in the following sections. A brief functional description, interface requirements, mating connector requirements and set-up instructions, where applicable, are given for each board.

****
* WARNING *
****

Do not install or remove any personality board while the CPZ-186 SBCP or the CPS-MX SBSP power is on. This may result in damage to the personality board and/or the CPU board.





JUMPER OPTIONS

(1) 3 WIRE / NO HANDSHAKE :

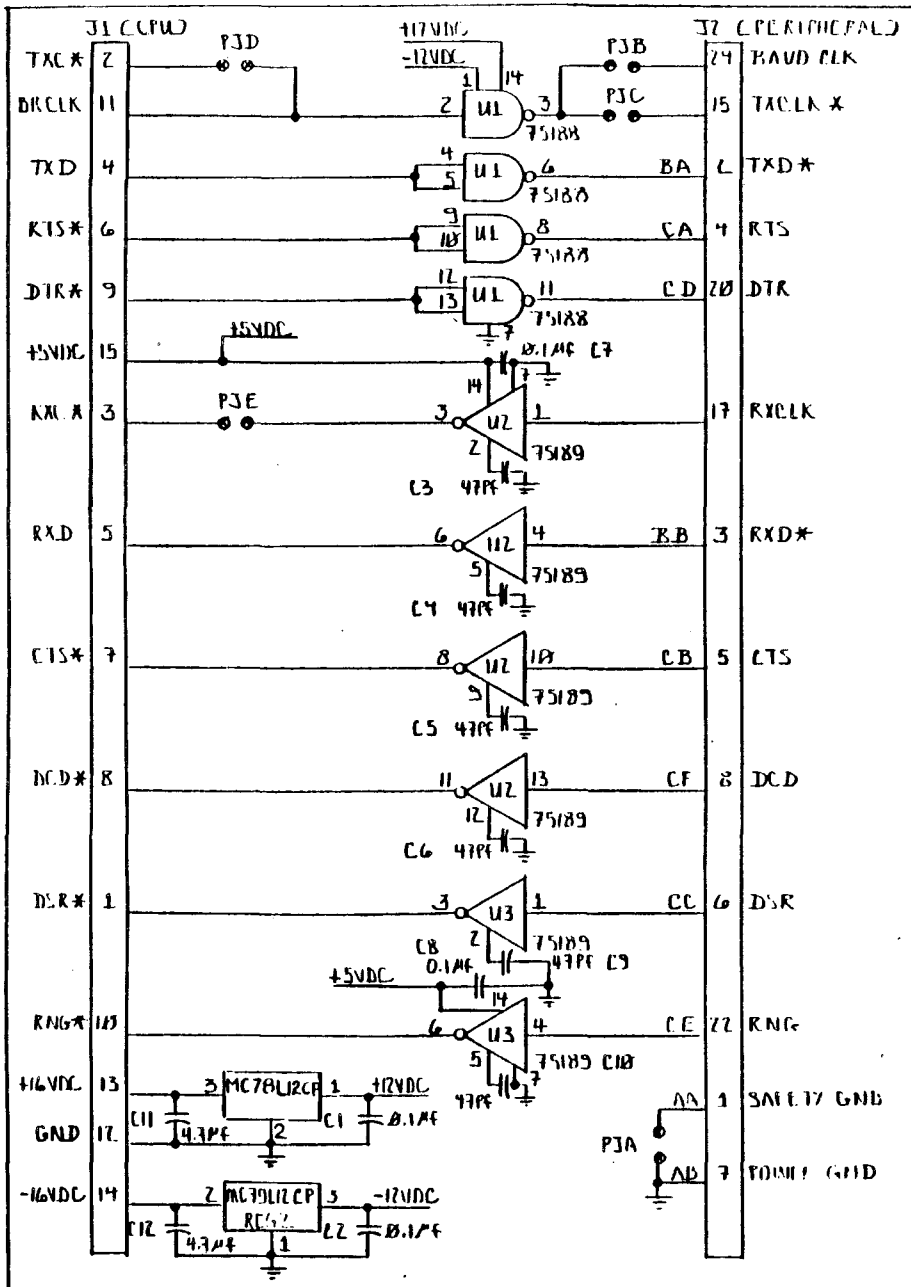
JA	JB	JC
1-2	A1-A2	1-2 = CRT
	C1-C2	2-3 = PRINTER (CHECK PRINTER MANUAL)
	B3-C3	

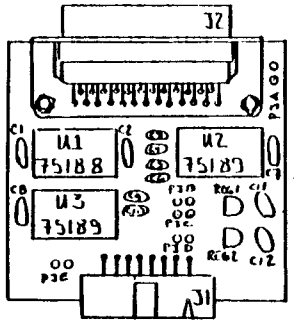
(2) FULL HANDSHAKE :

JA	JB	JC
2-3	A2-A3	1-2 = CRT
	C2-C3 OR	2-3 = PRINTER (CHECK PRINTER MANUAL)
	C1-C2	

InterContinental Micro

DESIGNED:	
REV: A	SCHEMATIC / LOGIC DIAGRAM FOR RS 232 / NO MODEM PERSONALITY BOARD
	PART NO. RPB 1188
	SHT 01 1





JUMPER OPTIONS

(1) ASYNCHRONOUS MODEM:

PJB, PJC, PJD, PJE ALL OPEN

(2) SYNCHRONOUS MODEM:

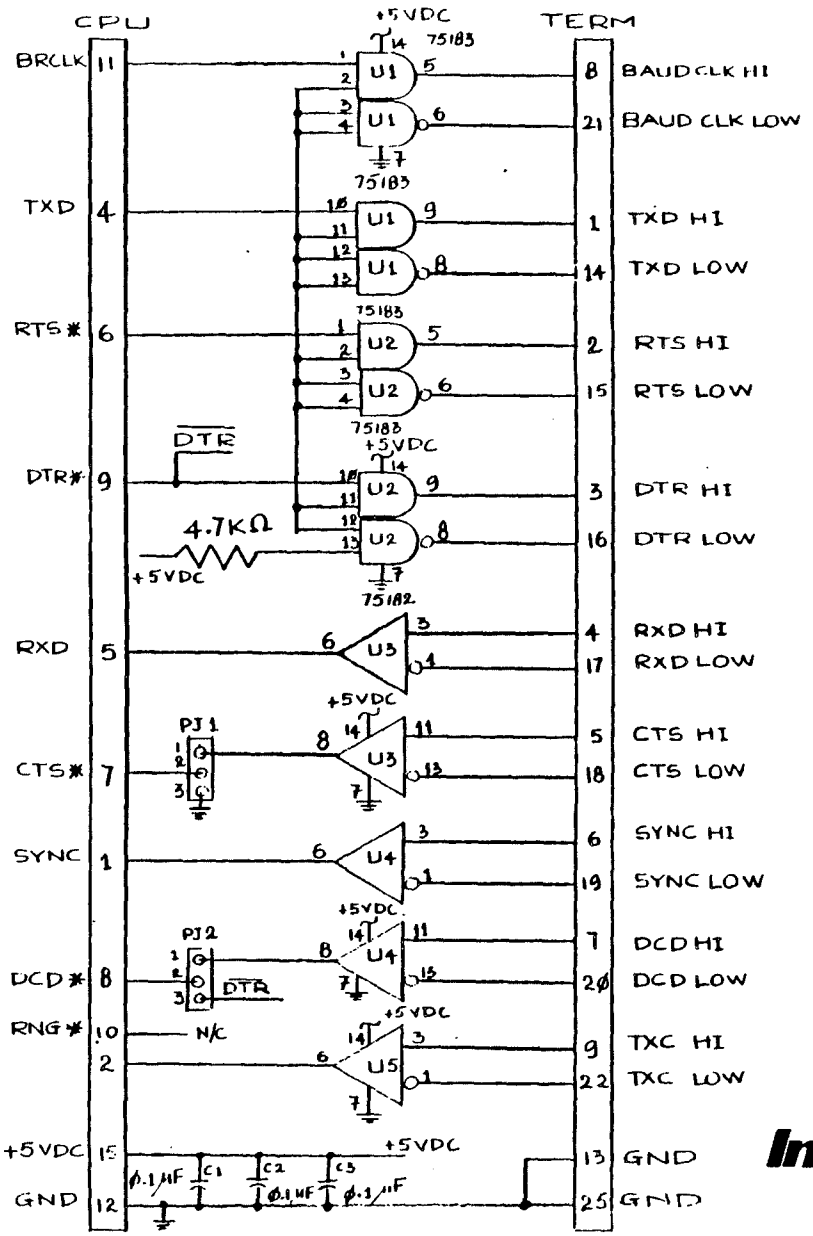
PJD & PJE CONNECTED; CUT PJB 1-2 & 2-3 ON CPZ-4800A IF PORT A

CUT PJA ON CPZ-4800A IF PORT B

PJB OR PJC CONNECTED (FUNCTION OF MODEM TYPE)

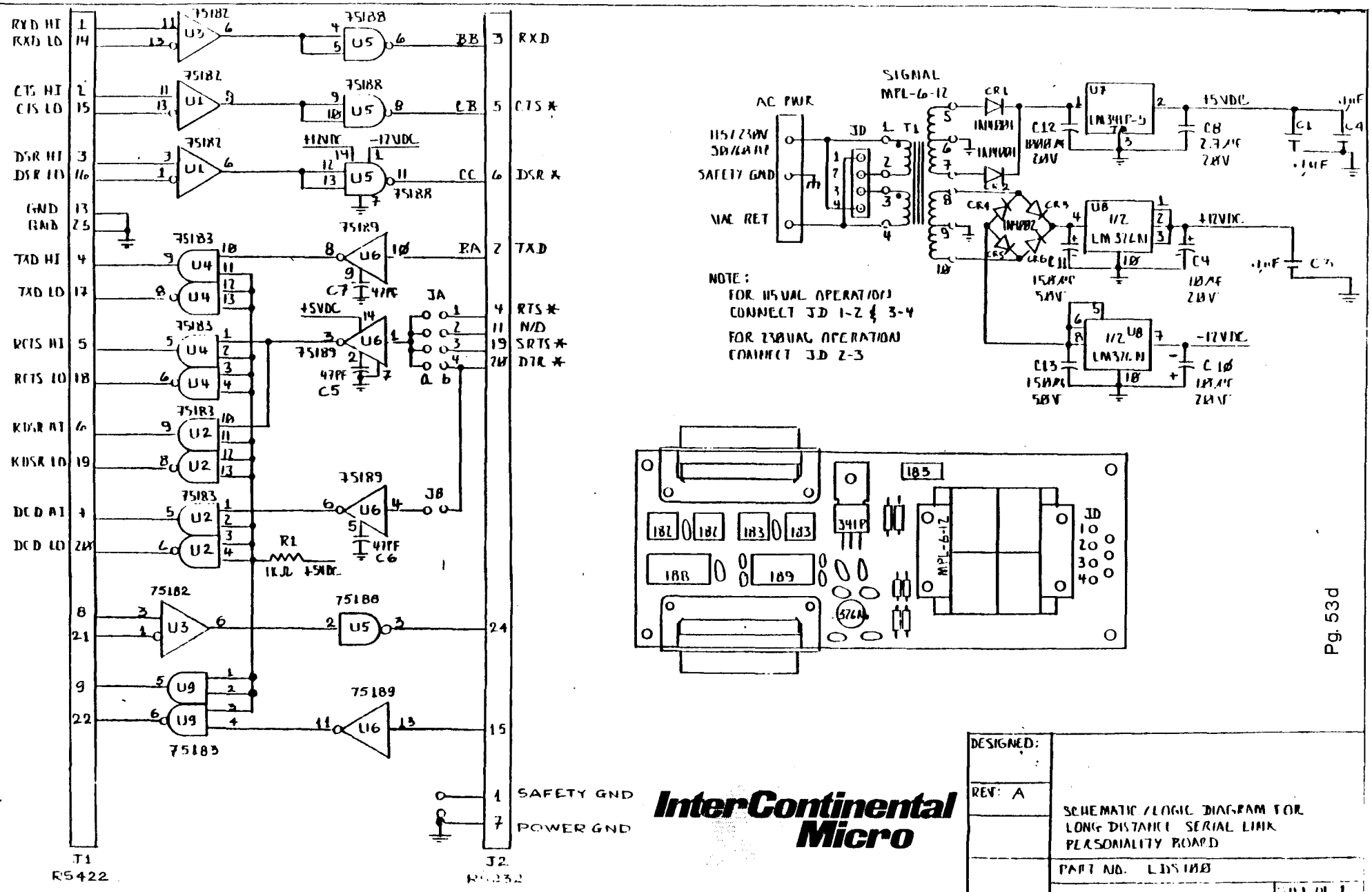
InterContinental Micro

DESIGNED:	
REV: A	SCHEMATIC / LOGIC DIAGRAM FOR RS232 C FULL MODEM PERSONALITY BOARD
	PART NO. CMPB1180
	SHT 01 OF 1



InterContinental Micro

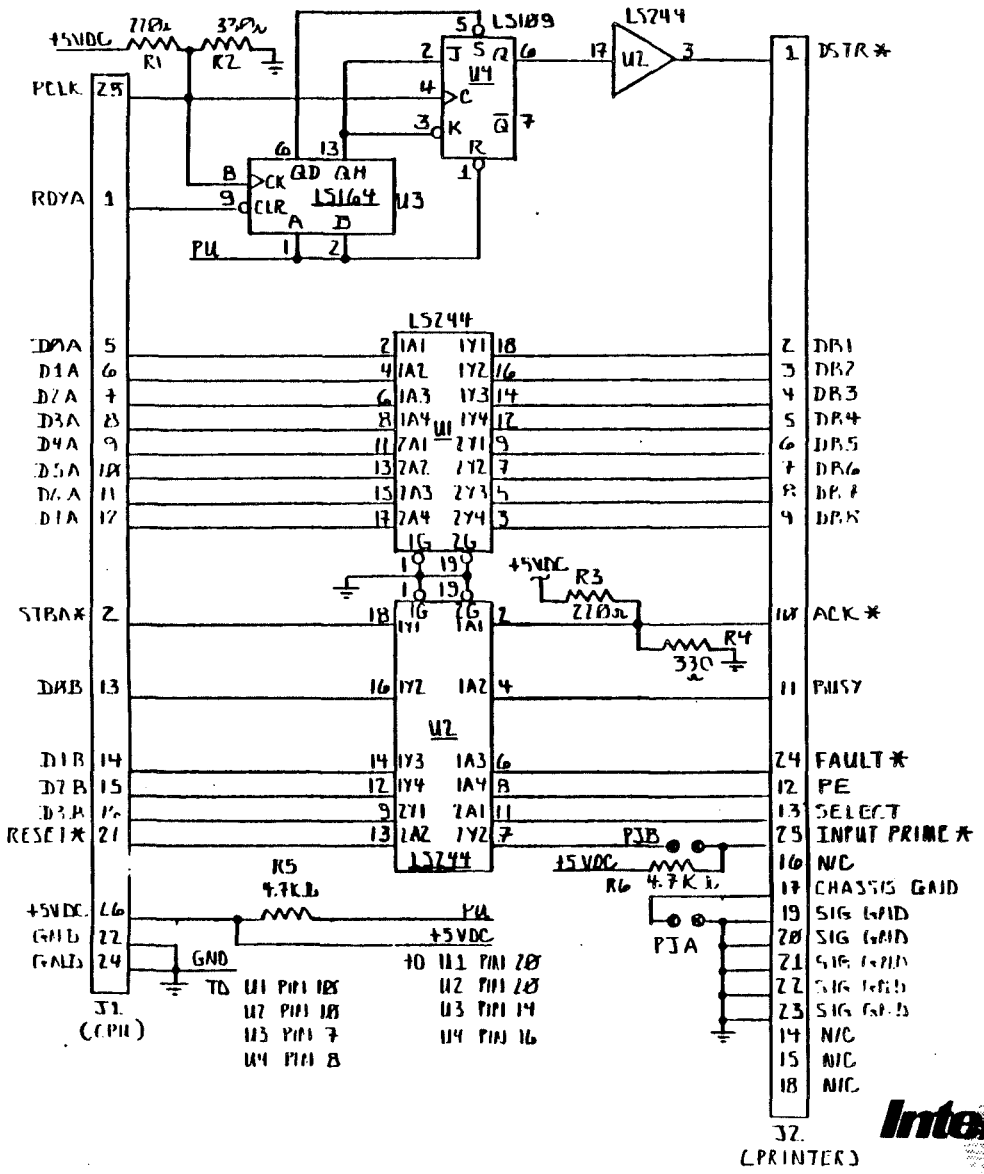
DESIGNED:	
REV: A	SCHEMATIC/LOGIC DIAGRAM FOR RS422 SERIAL COMMUNICATION PERSONALITY BOARD
	PART NO. FTTIØØ-1
	SH1 OF 1

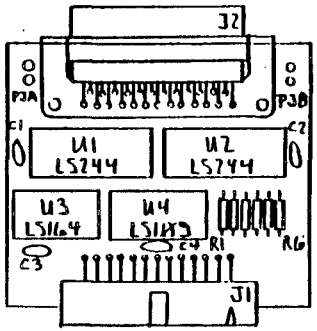


Pg. 53d

DESIGNED:	SCHEMATIC / LOGIC DIAGRAM FOR LONG DISTANCE SERIAL LINK PERSONALITY BOARD
REV: A	PART NO. LDS100
	SHT NO. 1

InterContinental Micro





NOTE: C1 - C4 = DECOUPLING CAPS 0.1 μf

DESIGNED:

REV: -

SCHEMATIC/LOGIC DIAGRAM FOR CENTRONICS PRINTER PARALLEL INTERFACE PERSONALITY BOARD

PART NO. CPI 188

SHT 01 OF 2

InterContinental Micro

CENTRONICS		P1	
1	DATA STROBE	1	DSTR *
2	DATA 1	2	DB1
3	DATA 2	3	DB2
4	DATA 3	4	DB3
5	DATA 4	5	DB4
6	DATA 5	6	DB5
7	DATA 6	7	DB6
8	DATA 7	8	DB7
9	DATA 8	9	DB8
10	ACKNLG*	10	ACK*
11	BUSY	11	BUSY
12	PE	12	PE
13	SLCT	13	SELECT
14	+/- 5V		N/C
15	CSCX		N/C
16	+/-5V OR EMPTY	16	EMPTY *
17	CHASSIS GND	17	CHASSIS GND
18	+5V		N/C
19	GND	19	GND
20	GND	20	GND
21	GND	21	GND
22	GND	22	GND
23	GND	23	GND
24	GND		GND
25	GND		GND
26	GND		GND
27	GND		GND
28	GND		GND
29	GND		GND
30	GND		GND
31	INPUT PRIME *	25	INPUT PRIME *
32	FAULT *	24	FAULT
33	N/C		N/C

CENTRONICS		P1
34	LINE COUNT PULSE	N/C
35	GND	GND
36	N/C	N/C

Pg. 53f

InterContinental Micro

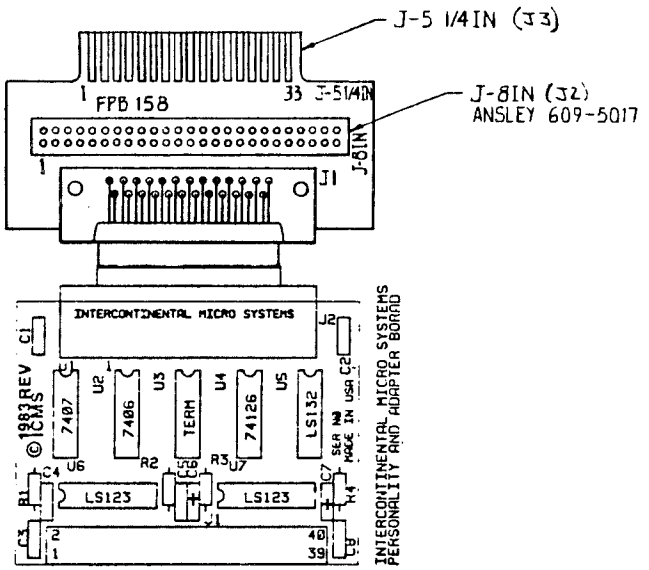
CONNECTION TABLE

CPI INIT

SH 2 OF 2

FUNCTION	J2	J-8IN[1]	J-5¼IN[1]
DS1*	1	26	10
DS2*	2	28	12
DS3*	3	30	14
DS4*	4	32	6
DIRC*	5	34	18
STEP*	6	36	20
WRITE DATA*	7	38	22
WRITE GATE*	8	40	24
TRACK 0*	9	42	26
WRITE PROT*	10	44	28
READ DATA*	11	46	30
SSO*	12	14	32
HEAD LOAD*	13	18	--
INDEX*	14	20	8
READY*	15	22	--
MOTOR ON*	16	24	16
TK 43*	17	2	--
	18	--	--
	19	--	--
GND	20	--	--
GND	21	--	--
GND	22	--	--
GND	23	--	--
GND	24	--	--
GND	25	--	--

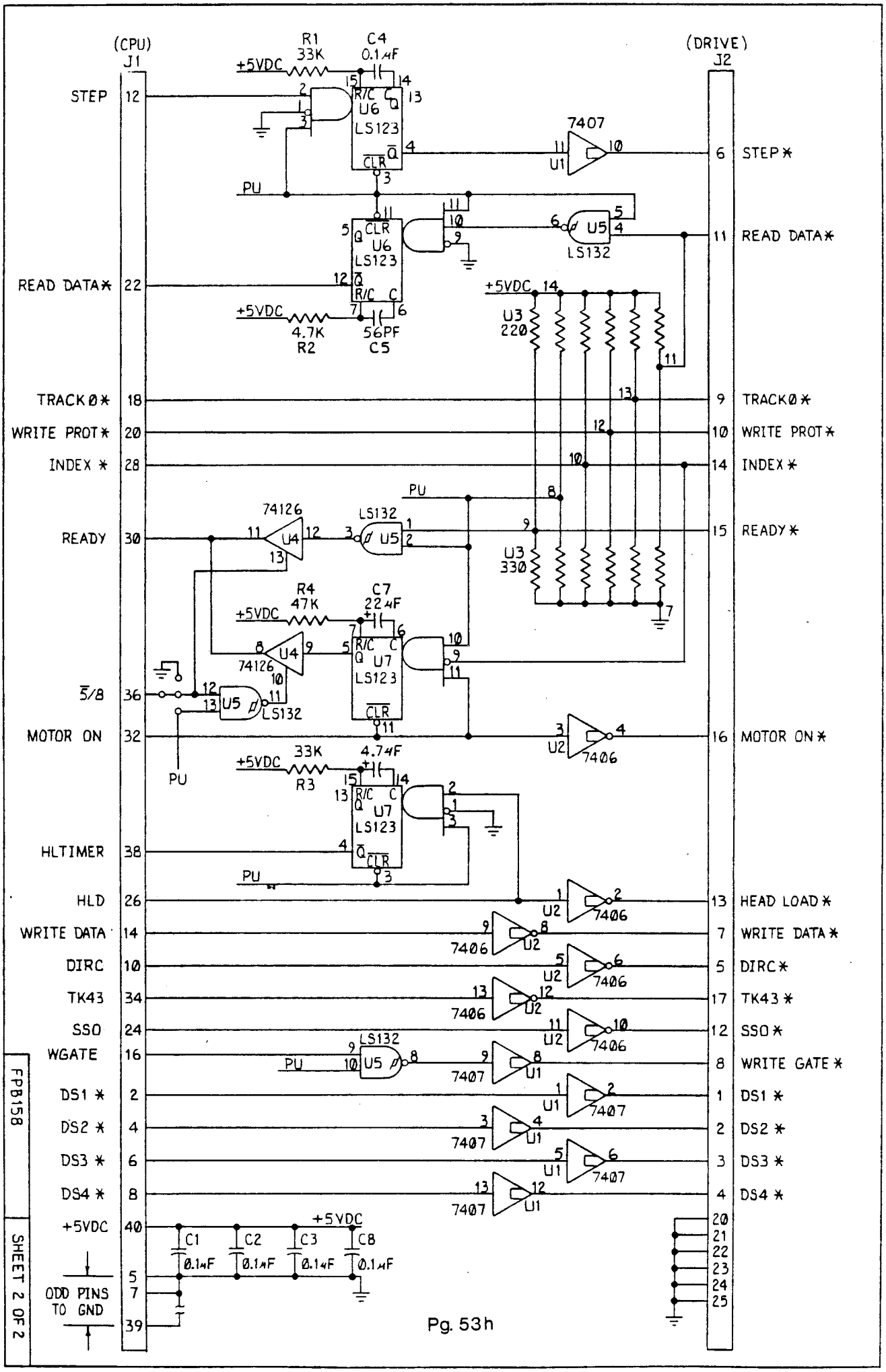
NOTE 1: On J-8IN & J-5¼IN, all odd pins are grounded; all even pins not listed are open.



SCHEMATIC/LOGIC DIAGRAM
8-5¼ INCH
FLOPPY DISK CONTROLLER
PERSONALITY BOARD

PART NO. FPB158-XY

OCT. 25, 1983 SHEET 1 OF 2


(CPU)
J1
(DRIVE)
J2
R1
33K
C4
0.1uF
+5VDC
STEP
12
R/C
Q
U6
LS123
CLR
7407
U1
STEP*
6
PU
READ DATA*
22
U5
LS132
READ DATA*
11
+5VDC
4.7K
R2
56PF
C5
U3
220
TRACK0*
18
TRACK0*
9
WRITE PROT*
20
WRITE PROT*
10
INDEX*
28
INDEX*
14
74126
READY
30
READY*
15
U3
330
R4
47K
C7
22uF
5/8
36
U7
MOTOR ON
32
MOTOR ON*
16
7406
U2
33K
R3
4.7uF
HLTIMER
38
HLD
26
HEAD LOAD*
13
WRITE DATA
14
WRITE DATA*
7
DIRC
10
DIRC*
5
TK43
34
TK43*
17
SSO
24
SSO*
12
WGATE
16
WRITE GATE*
8
DS1*
2
DS1*
1
DS2*
4
DS2*
2
DS3*
6
DS3*
3
DS4*
8
DS4*
4
+5VDC
40
C1
C2
C3
C8
0.1uF
ODD PINS
TO GND
5
7
39
20
21
22
23
24
25
FPB158
SHEET 2 OF 2

Pg. 53h

4.2.1.1. RS232/NO MODEM

PART NUMBER - RPB100

FUNCTION

The RS232/NO MODEM Personality Board provides RS232 drivers and receivers, terminations and jumper options to interface any simple RS232 device such as CRT terminals, serial printers or any other serial device not requiring an extensive handshake protocol.

This module may be used with either the CPZ-186 SBCP or the CPS-MX SBSP.

INTERFACE REQUIREMENTS

Connects to J2 or J3 of the CPZ-186 or the CPS-MX.

J1(CPU)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DSR*	DATA SET READY TO CPU
2	n/c	n/c
3	n/c	n/c
4	TxD	TRANSMIT DATA FROM CPU
5	RxD	RECEIVE DATA TO CPU
6	RTS*	REQUEST TO SEND DATA FROM CPU
7	CTS*	CLEAR TO SEND TO CPU
8	DCD*	DATA CARRIER DETECT TO CPU
9	DTR*	DATA TERMINAL READY FROM CPU
10	RNG*	RINGING INDICATOR TO CPU
11	n/c	n/c
12	GND	GROUND
13	+16VDC	+16VDC
14	-16VDC	-16VDC
15	+5VDC	+5VDC
16	n/c	n/c

J2(PERIPHERAL)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	SAFETY GND	SAFETY GROUND
2	TXD	TRANSMIT DATA TO PERIPHERAL
3	RXD	RECEIVE DATA FROM PERIPHERAL
4	n/c	n/c
5	CTS*	CLEAR TO SEND TO PERIPHERAL
6	DSR*	DATA SET READY TO PERIPHERAL
7	POWER GND	POWER GROUND
8	DCD*	DATA CARRIER DETECT TO PERIPHERAL
9	n/c	n/c
10	n/c	n/c
11	n/c	n/c
12	n/c	n/c
13	n/c	n/c
14	n/c	n/c
15	n/c	n/c
16	n/c	n/c
17	n/c	n/c
18	n/c	n/c
19	SRTS*	SECONDARY REQUEST TO SEND FROM PERIPHERAL
20	DTR*	DATA TERMINAL READY FROM PERIPHERAL
21	n/c	n/c
22	n/c	n/c
23	n/c	n/c
24	n/c	n/c
25	n/c	n/c

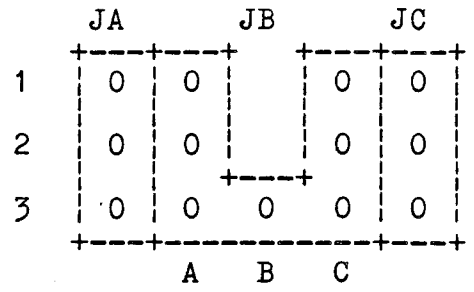
CONNECTOR REQUIREMENTS

PERSONALITY BOARD CONNECTORS	MATING CONNECTORS
J1 - ANSLEY 609-1617	ANSLEY 609-1630 (ICM SUPPLIED)
J2 - CANNON DB25P-731	CANNON DB 25S-731 (CUSTOMER SUPPLIED)

SET UP INSTRUCTIONS

Three Jumper Areas are provided: JA, JB and JC. Refer to the figure below for the following set-up instructions:

[JA/JB/JC] area



JA

The CPU may be required to provide handshaking with the peripheral through the signal "DCD". If that handshaking signal is required, connect pin 2 to pin 3 with the jumper provided. If no handshaking signal is required , connect pin 2 to pin 1.

JB

The CPU may required to provide handshaking with the peripheral through the signall "CTS". Furthermore, it may accept the signals "DTR" or "SRTS" through the input "CTS". The following options are available:

JB	Configuration
1A-2A	no handshaking provided to peripheral at "CTS"
2A-3A	peripheral's "CTS" activated by CPU's "DTR"
3A-3B	not used
3B-3C	no handshaking provided to CPU's "CTS" by peripheral's "DTR" or "SRTS"
2C-3C	peripheral's "DTR" or "SRTS" activates CPU's "CTS"
1C-2C	peripheral's "DTR" or "SRTS" activates CPU's "DSR"

JC

The peripheral may provide either of two handshaking signals "SRTS" or "DTR". This jumper may select either signal as the source to the CPU's "CTS" or "DSR" inputs.

To connect "DTR" handshaking which is on pin 20 of the RS232/C interface, connect JC-1 to JC-2.

To connect "SRTS" handshaking which is on pin 19 of the RS232/C interface, connect JC-2 to JC-3.

EXAMPLES

1) Configure JA, JB and JC as follows for a simple terminal interface:

    JA = 1-2
    JB = 1A-2A / 3B-3C
    JC = none required

2) Configure JA, JB and JC as follows for an Anadex Serial Printer, model DP-9501

    JA = 2-3
    JB = 1A-2A / 2C-3C
    JC = 2-3

4.2.1.2. RS232/FULL MODEM

PART NUMBER - MPB100

FUNCTION

The RS232C/FULL MODEM PERSONALITY BOARD provides RS232 drivers/receivers and jumper options to interface asynchronous or synchronous modems with varying types of bit oriented protocols such as IBM Bi-Sync, HDLC or SDLC. Jumpers provided enable the user to configure the board for either asynchronous or synchronous operation.

This module may be used with either the CPZ-186 SBCP or the CPS-MX SBSP.

INTERFACE REQUIREMENTS

Connects to J1 or J2 of the CPZ-186 or the CPS-MX.

CPU (J1)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DSR*	DATA SET READY TO CPU
2	n/c	n/c
3	n/c	n/c
4	TXD	TRANSMIT DATA FROM CPU
5	RXD	RECEIVE DATA TO CPU
6	RTS*	REQUEST-TO-SEND FROM CPU
7	CTS*	CLEAR-TO-SEND TO CPU
8	DCD*	DATA CARRIER DETECT TO CPU
9	DTR*	DATA TERMINAL READY FROM CPU
10	RNG*	RINGING INDICATOR TO CPU
11	n/c	n/c
12	GND	GROUND
13	+16VDC	+16VDC
14	-16VDC	-16VDC
15	+5VDC	+5VDC
16	n/c	n/c

J2(PERIPHERAL)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	SAFETY GND	SAFETY GROUND
2	TXD	TRANSMIT DATA TO PERIPHERAL
3	RXD	RECEIVE DATA FROM PERIPHERAL
4	RTS*	REQUEST-TO-SEND TO PERIPHERAL
5	CTS*	CLEAR-TO-SEND FROM PERIPHERAL
6	DSR*	DATA SET READY FROM PERIPHERAL
7	POWER GND	POWER GROUND
8	DCD*	DATA CARRIER DETECT FROM PERIPHERAL
9	n/c	n/c
10	n/c	n/c
11	n/c	n/c
12	n/c	n/c
13	n/c	n/c
14	n/c	n/c
15	TXCLK*	TRANSMIT CLOCK TO PERIPHERAL
16	n/c	n/c
17	RXCLK*	RECEIVE CLOCK FROM PERIPHERAL
18	n/c	n/c
19	n/c	n/c
20	DTR*	DATA TERMINAL READY TO PERIPHERAL
21	n/c	n/c
22	RNG*	RINGING INDICATOR FROM PERIPHERAL
23	n/c	n/c
24	BAUD CLK	BAUD CLOCK TO PERIPHERAL
25	n/c	n/c

CONNECTOR REQUIREMENTS

PERSONALITY BOARD CONNECTORS	MATING CONNECTORS
J1 - ANSLEY 609-1617	ANSLEY 609-1630 (ICM SUPPLIED)
J2 - CANNON DB25P-731	CANNON DB 25S-731 (CUSTOMER SUPPLIED)

SET UP INSTRUCTIONS

The board may be configured for either asynchronous or synchronous modem requirements.

a) Asynchronous Modems

PJB, PJC, PJD, PJE and PJF are all open.

b) Synchronous Modems

1) MODEM SUPPLIES TRANSMIT AND RECEIVE CLOCK

Connect PJE and PJF only.

2) CPZ-186 OR CPS-MX SUPPLY TRANSMIT CLOCK

Connect PJC, PJD and PJE of MPB100 only.

Note: If using CPZ-186 Port A of the SIO, cut PJB A-B and B-C.

If using CPZ-186 Port B of the SIO, cut PJC.

If using CPS-MX Port A of the SIO, cut PJA A-B and B-C.

If using CPS-MX Port B of the SIO, cut PJB.

3) CPZ-186 OR CPS-MX SUPPLY BAUD RATE CLOCK

Same as (2) above except that on the MPB100, PJB is connected instead of PJC and PJF is disconnected.

c) If safety ground of the modem is to be tied to logic ground. connect PJA on the MPB100.

4.2.1.3. RS422 SERIAL COMMUNICATIONS

PART NUMBER - FTT100

FUNCTION

The FTT100 personality board provides RS422 differential line drivers and receivers. These balanced drivers and receivers can provide serial communications for distances of up to 4000 feet at a communications rate of 100 kbits/second. This assumes that 24 AWG twisted pair cable is used. Higher rates may be attained for shorter cable lengths. If the CPU's DART/SIO controller is used in synchronous communications mode at its maximum rate of 800 kbits/second, the maximum cable length recommended is 325 feet. Drivers and receivers are provided for all signals of the SIO to support full handshake protocols.

The FTT100 in combination with the Long Distance Serial Personality Board (LDS100), provides a means of connecting terminals, printers and other RS232 serial devices remotely located from the CPU mainframe. CPU-TO-CPU communications may also be set-up over long distances by using the FTT100 at both CPUs. In this case, the interconnecting cable is cross-connected to tie receiver-to-transmiter and tansmiter-to-receiver devices. No cross-connection is required between the FTT100 and the LDS100.

Jumper options are provided to minimize the number of cable lines required if no handshaking signals are required as in the case of simple RS232 Terminals where only transmit and receive signals are required. Ground is also provided but is not used in most cases.

This module may be used with either the CPZ-186 SBCP or the CPS-MX SBSP.

INTERFACE REQUIREMENTS

Connects to J1 or J2 of the CPZ-186 or the CPS-MX.

J1(CPU)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DSR*	DATA SET READY TO CPU
2	TXC*	TRANSMIT CLOCK FROM CPU
3	n/c	n/c
4	TXD	TRANSMIT DATA FROM CPU
5	RXD	RECEIVE DATA TO CPU
6	RTS*	REQUEST-TO-SEND FROM CPU
7	CTS*	CLEAR-TO-SEND TO CPU
8	DCD*	DATA CARRIER DETECT TO CPU
9	DTR*	DATA TERMINAL READY FROM CPU
10	n/c	n/c
11	BRCLK	BAUD RATE CLOCK FROM CPU
12	GND	GROUND
13	+16VDC	+16VDC
14	-16VDC	-16VDC
15	+5VDC	+5VDC
16	n/c	n/c

J2(TERMINAL)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	TXD HI	TRANSMIT DATA HIGH to terminal
2	RTS HI	REQUEST-TO-SEND HIGH to terminal
3	DTR HI	DATA TERMINAL READY to terminal
4	RXD HI	RECEIVE DATA HIGH from terminal
5	CTS HI	CLEAR-TO-SEND HIGH from terminal
6	SYNC HI	SYNC HIGH from terminal
7	DCD HI	DATA CARRIER DETECT HIGH from terminal
8	BAUDCLK HI	BAUD RATE CLOCK HIGH to terminal
9	TXC HI	TRANSMIT CLOCK HIGH from terminal
10	n/c	n/c
11	n/c	n/c
12	n/c	n/c
13	GND	GROUND
14	TXD LO	TRANSMIT DATA LOW to terminal
15	RTS LO	REQUEST-TO-SEND LOW to terminal
16	DTR LO	DATA TERMINAL READY LOW to terminal
17	RXD LO	RECEIVE DATA LOW from terminal
18	CTS LO	CLEAR-TO-SEND LOW from terminal
19	SYNC LO	SYNC LOW from terminal
20	DCD LO	DATA CARRIER DETECT LOW from terminal
21	BAUDCLK LO	BAUD RATE CLOCK LOW to terminal
22	TXC LO	TRANSMIT CLOCK LOW from terminal
23	n/c	n/c
24	n/c	n/c
25	GND	GROUND

CONNECTOR REQUIREMENTS

PERSONALITY BOARD CONNECTORS	MATING CONNECTORS
J1 - ANSLEY 609-1617	ANSLEY 609-1630 (ICM SUPPLIED)
J2 - CANNON DB25P-731	CANNON DB 25S-731 (CUSTOMER SUPPLIED)

SET UP INSTRUCTIONS

SIMPLE TERMINAL

To provide less interconnecting lines for terminal not requiring full handshake protocol, jumper PJ1 2-to-3 and PJ2 2-to-3. Provide twisted pair lines for TXD and RXD only.

FULL PROTOCOL

To provide for full handshaking, jumper PJ1 1-to-2 and PJ2 1-to-2. Provide twisted pair lines for the signals required.

SYNCHRONOUS TERMINAL

To use CPZ-186 Port A of the SIO, cut PJB B-C on the CPZ-186.

To use CPS-MX Port A of the SIO, cut PJA B-C on the CPS-MX.

4.2.1.4. LONG DISTANCE SERIAL COMMUNICATIONS

PART NUMBER - LDS100

FUNCTION

The LDS100 personality board provides RS422 differential line drivers and receivers. These balanced drivers and receivers can provide serial communications for distances of up to 4000 feet at a communications rate of 100 kbits/second. This assumes that 24 AWG twisted pair cable is used. Drivers and receivers are provided to support full handshake protocols.

The LDS100 in combination with the RS422 Serial Communications Personality Board (FTT100), provides a means of connecting terminals, printers and other RS232 serial devices remotely located from the CPU mainframe. Jumper options are provided to minimize the number of cable lines required if no handshaking signals are required as in the case of simple RS232 Terminals where only transmit and receive signals are required. Ground is also provided but is not used in most cases.

AC power must be provided to the board. The board may be strapped for either 115VAC/60HZ or 230 VAC/50HZ operation.

INTERFACE REQUIREMENTS

J1 connects to RS422 Personality Board via long distance cable.
J2 connects to RS232 serial device with standard RS232 cable.

J1(RS422 INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	TXD HI	TRANSMIT DATA HIGH to terminal
2	RTS HI	REQUEST-TO-SEND HIGH to terminal
3	DTR HI	DATA TERMINAL READY to terminal
4	RXD HI	RECEIVE DATA HIGH from terminal
5	CTS HI	CLEAR-TO-SEND HIGH from terminal
6	SYNC HI	SYNC HIGH from terminal
7	DCD HI	DATA CARRIER DETECT HIGH from terminal
8	BAUDCLK HI	BAUD RATE CLOCK HIGH to terminal
9	TXC HI	TRANSMIT CLOCK HIGH from terminal
10	n/c	n/c
11	n/c	n/c
12	n/c	n/c
13	GND	GROUND
14	TXD LO	TRANSMIT DATA LOW to terminal
15	RTS LO	REQUEST-TO-SEND LOW to terminal
16	DTR LO	DATA TERMINAL READY LOW to terminal
17	RXD LO	RECEIVE DATA LOW from terminal
18	CTS LO	CLEAR-TO-SEND LOW from terminal
19	SYNC LO	SYNC LOW from terminal
20	DCD LO	DATA CARRIER DETECT LOW from terminal
21	BAUDCLK LO	BAUD RATE CLOCK LOW to terminal
22	TXC LO	TRANSMIT CLOCK LOW from terminal
23	n/c	n/c
24	n/c	n/c
25	GND	GROUND

J2(PERIPHERAL)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	SAFETY GND	SAFETY GROUND
2	TXD	TRANSMIT DATA TO PERIPHERAL
3	RXD	RECEIVE DATA FROM PERIPHERAL
4	RTS*	REQUEST-TO-SEND TO PERIPHERAL
5	CTS*	CLEAR-TO-SEND FROM PERIPHERAL
6	DSR*	DATA SET READY FROM PERIPHERAL
7	POWER GND	POWER GROUND
8	n/c	n/c
9	n/c	n/c
10	n/c	n/c
11	RESERVED	RESERVED FOR SPECIAL USE HANDSHAKE
12	n/c	n/c
13	n/c	n/c
14	n/c	n/c
15	TXCLK*	TRANSMIT CLOCK FROM PERIPHERAL
16	n/c	n/c
17	n/c	n/c
18	n/c	n/c
19	SRTS*	SECONDARY REQUEST-TO-SEND FROM PERIPHERAL
20	DTR*	DATA TERMINAL READY TO PERIPHERAL
21	n/c	n/c
22	n/c	n/c
23	n/c	n/c
24	BAUDCLK	BAUD CLOCK TO PERIPHERAL
25	n/c	n/c

CONNECTOR REQUIREMENTS

PERSONALITY BOARD CONNECTORS	MATING CONNECTORS
J1 - CANNON DB25S-731	CANNON DB25P-731 (CUSTOMER SUPPLIED)
J2 - CANNON DB25S-731	CANNON DB25P-731 (CUSTOMER SUPPLIED)

SET UP INSTRUCTIONS

(1)AC POWER SET-UP

The LDS100 may be configured to operate with 115VAC/60HZ or 230VAC/50HZ through jumper options. Use 18 or 16 AWG wire for jumpers in this setting.

To configure the LDS100 for 115VAC, solder two jumpers. One is soldered at JD 1-to-2 and the other is soldered at JD 3-to-4.

To configure the LDS100 for 230VAC, solder one jumper at JD 2-to-3.

AC power may now be installed. Connect AC HI and AC LO in the indicated solder pads. Connect SAFETY GROUND to the pad marked "CH".

(2)SAFETY GROUND CONNECTION

Solder a strap in jumper area JC if Safety Ground should be connected to Power Ground.

(3)SIMPLE TERMINAL(NO HANDSHAKING) SET-UP

Most terminals do not require handshaking for RS232/C communication. In this case, no jumpers are required in jumper areas JA and JB. Connect receive and transmit data lines only between the FTT100 and the LDS100 boards.

(4)TERMINAL/PRINTER(FULL HANDSHAKING)

Connect JA in accordance with the type of handshaking signal required to be transmited to the CLEAR-TO-SEND input of the CPU. The options are as follows:

JA	handshake signal
a1-to-b1	request-to-send(RTS)
a2-to-b2	manufacture defined
a3-to-b3	secondary request-to-send(SRTS)
a4-to-b4	data terminal ready(DTR)

Connect JB if DATA TERMINAL READY(DTR) is required to be connected to the DATA CARRIER DETECT(DCD) signal of the CPU. Connect all corresponding signal lines from FTT100 to the LDS100.

4.2.1.5. FLOPPY DISK CONTROLLER

PART NUMBER - FPB158-XY

FUNCTION

The FLOPPY DISK CONTROLLER personality board provides line drivers and receivers, terminators and logic to interface either an 8-inch or a 5 1/4-inch or both 8-inch and 5 1/4-inch floppy disk drives with the CPZ-186 SBCP. A DB25 connector is available as the means to interface with the drive interface; however, if other types of commonly used connectors are required, adapters are available to tailor the interface appropriately.

This module is used ONLY on the CPZ-186 SBCP. It is not to be used with the CPZ48000 SBCP.

```
**********
*        *
*  WARNING  *
*        *
**********
```

THE FPB158-XY BASE BOARD MUST NOT BE CONNECTED TO THE MODEL CPZ48000 SBCP AS DAMAGE MAY RESULT TO THE SBCP. NOTE HOWEVER THAT THE FPB100-11 OR THE FPB100-22 ADAPTERS MAY BE USED WITH THE FPB158-XY BASE BOARD. THE FPB158-30, AN ADAPTER WHICH ACCOMMODATES BOTH 5 1/4- INCH EDGE CARD AND 8-INCH HEADER CONNECTORS, IS TO BE USED ONLY WITH THE FPB158-XY BASE BOARD.

INTERFACE REQUIREMENTS

FPB158-XY

J1(CPU)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	n/c	n/c
2	DS1*	DRIVE SELECT #1 FROM CPU
3	n/c	n/c
4	DS2*	DRIVE SELECT #2 FROM CPU
5	GND	GROUND
6	DS3*	DRIVE SELECT #3 FROM CPU
7	GND	GROUND
8	DS4*	DRIVE SELECT #4 FROM CPU
9	GND	GROUND
10	DIRC	DIRECTION CONTROL FROM CPU
11	GND	GROUND
12	STEP	STEP CONTROL FROM CPU
13	GND	GROUND
14	WRITE DATA	WRITE DATA FROM CPU
15	GND	GROUND
16	WGATE	WRITE GATE FROM CPU
17	GND	GROUND
18	TRACK 0*	TRACK 0 STATUS TO CPU
19	GND	GROUND
20	WRITE PROT*	WRITE PROTECT TO CPU
21	GND	GROUND
22	READ DATA*	READ DATA TO CPU
23	GND	GROUND
24	SS0	SIDE SELECT OUTPUT FROM CPU
25	GND	GROUND
26	HLD	HEAD LOAD COMMAND FROM CPU
27	GND	GROUND
28	INDEX*	INDEX PULSE TO CPU
29	GND	GROUND
30	READY	READY STATUS TO CPU
31	GND	GROUND
32	MOTOR ON	MOTOR ON STATUS FROM CPU
33	GND	GROUND
34	TK43	TRACK 43 STATUS FROM CPU
35	GND	GROUND
36	5*/8	DRIVE SIZE SELECT
37	GND	GROUND
38	HLTIMER	HEAD LOAD TIMER
39	GND	GROUND
40	+5VDC	+5VDC

J2(MODIFIED DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DS1*	DRIVE SELECT #1 to DRIVE interface
2	DS2*	DRIVE SELECT #2 to DRIVE interface
3	DS3*	DRIVE SELECT #3 to DRIVE interface
4	DS4*	DRIVE SELECT #4 to DRIVE interface
5	DIRC*	DIRECTION CONTROL to DRIVE interface
6	STEP*	STEP CONTROL to DRIVE interface
7	WRITE DATA*	WRITE DATA to DRIVE interface
8	WRITE GATE*	WRITE GATE to DRIVE interface
9	TRACK 0 *	TRACK 0 STATUS from DRIVE interface
10	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
11	READ DATA*	READ DATA to DRIVE inerface
12	SS0*	SIDE SELECT OUTPUT to DRIVE interface
13	HEAD LOAD*	HEAD LOAD COMMAND to DRIVE interface
14	INDEX*	INDEX PULSES from DRIVE interface
15	READY*	READY STATUS from DRIVE interface
16	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
17	TK43*	TRACK 43 STATUS to DRIVE interface
18	n/c	n/c
19	n/c	n/c
20	GND	GROUND
21	GND	GROUND
22	GND	GROUND
23	GND	GROUND
24	GND	GROUND
25	GND	GROUND

ADAPTER FPB100-11

Connects to FPB100 or FPB158 Personality Board.

J1(MODIFIED DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DS1*	DRIVE SELECT #1 to DRIVE interface
2	DS2*	DRIVE SELECT #2 to DRIVE interface
3	DS3*	DRIVE SELECT #3 to DRIVE interface
4	DS4*	DRIVE SELECT #4 to DRIVE interface
5	DIRC*	DIRECTION CONTROL to DRIVE interface
6	STEP*	STEP CONTROL to DRIVE interface
7	WRITE DATA*	WRITE DATA to DRIVE interface
8	WRITE GATE*	WRITE GATE to DRIVE interface
9	TRACK 0 *	TRACK 0 STATUS from DRIVE interface
10	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
11	READ DATA*	READ DATA to DRIVE inerface
12	SS0*	SIDE SELECT OUTPUT to DRIVE interface
13	HEAD LOAD*	HEAD LOAD COMMAND to DRIVE interface
14	INDEX*	INDEX PULSES from DRIVE interface
15	READY*	READY STATUS from DRIVE interface
16	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
17	TK43*	TRACK 43 STATUS to DRIVE interface
18	GND	GROUND
19	GND	GROUND
20	GND	GROUND
21	GND	GROUND
22	GND	GROUND
23	GND	GROUND
24	GND	GROUND
25	GND	GROUND

ADAPTER FPB100-11

J2(8-INCH DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	TK43*	TRACK 43 STATUS to DRIVE interface
3	GND	GROUND
4	n/c	n/c
5	GND	GROUND
6	n/c	n/c
7	GND	GND
8	n/c	n/c
9	GND	GROUND
10	n/c	n/c
11	GND	GROUND
12	n/c	n/c
13	GND	GROUND
14	SSO	SIDE SELECT OUTPUT to DRIVE interface
15	GND	GROUND
16	n/c	n/c
17	GND	GROUND
18	HEAD LOAD*	HEAD LOAD COMMAND to DRIVE interface
19	GND	GROUND
20	INDEX*	INDEX PULSES from DRIVE interface
21	GND	GROUND
22	READY*	READY STATUS from DRIVE interface
23	GND	GROUND
24	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
25	GND	GROUND
26	DS1*	DRIVE SELECT #1 to DRIVE interface
27	GND	GROUND
28	DS2*	DRIVE SELECT #2 to DRIVE interface
29	GND	GROUND
30	DS3*	DRIVE SELECT #3 to DRIVE interface
31	GND	GROUND
32	DS4*	DRIVE SELECT #4 to DRIVE interface
33	GND	GROUND
34	DIRC*	DIRECTION CONTROL to DRIVE interface
35	GND	GROUND
36	STEP*	STEP COMMAND to DRIVE interface
37	GND	GROUND
38	WRITE DATA*	WRITE DATA to DRIVE interface
39	GND	GROUND
40	WRITE GATE*	WRITE GATE to DRIVE interface

41	GND	GROUND
42	TRACK 0 *	TRACK ZERO STATUS from DRIVE interface
43	GND	GROUND
44	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
45	GND	GROUND
46	READ DATA*	READ DATA to DRIVE interface
47	GND	GROUND
48	n/c	n/c
49	GND	GROUND
50	n/c	n/c

ADAPTER FPB100-22

Connects to FPB100 or FPB158 Personality Board.

J1(MODIFIED DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DS1*	DRIVE SELECT #1 to DRIVE interface
2	DS2*	DRIVE SELECT #2 to DRIVE interface
3	DS3*	DRIVE SELECT #3 to DRIVE interface
4	DS4*	DRIVE SELECT #4 to DRIVE interface
5	DIRC*	DIRECTION CONTROL to DRIVE interface
6	STEP*	STEP CONTROL to DRIVE interface
7	WRITE DATA*	WRITE DATA to DRIVE interface
8	WRITE GATE*	WRITE GATE to DRIVE interface
9	TRACK 0 *	TRACK 0 STATUS from DRIVE interface
10	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
11	READ DATA*	READ DATA to DRIVE inerface
12	SSO*	SIDE SELECT OUTPUT to DRIVE interface
13	HEAD LOAD*	HEAD LOAD COMMAND to DRIVE interface
14	INDEX*	INDEX PULSES from DRIVE interface
15	READY*	READY STATUS from DRIVE interface
16	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
17	TK43*	TRACK 43 STATUS to DRIVE interface
18	GND	GROUND
19	GND	GROUND
20	GND	GROUND
21	GND	GROUND
22	GND	GROUND
23	GND	GROUND
24	GND	GROUND
25	GND	GROUND

ADAPTER FPB100-22

J2(5 1/4-INCH DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	n/c	n/c
3	GND	GROUND
4	n/c	n/c
5	GND	GROUND
6	DS4*	DRIVE SELECT #4 to DRIVE interface
7	GND	GROUND
8	INDEX*	INDEX* PULSE STATUS from DRIVE interface
9	GND	GROUND
10	DS1*	DRIVE SELECT #1 to DRIVE interface
11	GND	GROUND
12	DS2*	DRIVE SELECT #2 to DRIVE interface
13	GND	GROUND
14	DS3*	DRIVE SELECT #3 to DRIVE interface
15	GND	GROUND
16	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
17	GND	GROUND
18	DIRC*	DIRECTION CONTROL to DRIVE interface
19	GND	GROUND
20	STEP*	STEP COMMAND to DRIVE interface
21	GND	GROUND
22	WRITE DATA*	WRITE DATA to DRIVE interface
23	GND	GROUND
24	WRITE GATE*	WRITE GATE to DRIVE interface
25	GND	GROUND
26	TRACK 0 *	TRACK ZERO STATUS from DRIVE interface
27	GND	GROUND
28	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
29	GND	GROUND
30	READ DATA*	READ DATA to DRIVE interface
31	GND	GROUND
32	SS0*	SIDE SELECT OUTPUT to DRIVE interface
33	GND	GROUND
34	n/c	n/c

ADAPTER FPB158-30

Connects to FPB158 Personality Board.

J1(MODIFIED DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DS1*	DRIVE SELECT #1 to DRIVE interface
2	DS2*	DRIVE SELECT #2 to DRIVE interface
3	DS3*	DRIVE SELECT #3 to DRIVE interface
4	DS4*	DRIVE SELECT #4 to DRIVE interface
5	DIRC*	DIRECTION CONTROL to DRIVE interface
6	STEP*	STEP CONTROL to DRIVE interface
7	WRITE DATA*	WRITE DATA to DRIVE interface
8	WRITE GATE*	WRITE GATE to DRIVE interface
9	TRACK 0 *	TRACK 0 STATUS from DRIVE interface
10	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
11	READ DATA*	READ DATA to DRIVE inerface
12	SSO*	SIDE SELECT OUTPUT to DRIVE interface
13	HEAD LOAD*	HEAD LOAD COMMAND to DRIVE interface
14	INDEX*	INDEX PULSES from DRIVE interface
15	READY*	READY STATUS from DRIVE interface
16	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
17	TK43*	TRACK 43 STATUS to DRIVE interface
18	n/c	n/c
19	n/c	n/c
20	GND	GROUND
21	GND	GROUND
22	GND	GROUND
23	GND	GROUND
24	GND	GROUND
25	GND	GROUND

ADAPTER FPB158-30

J2(8-INCH DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	TK43*	TRACK 43 STATUS to DRIVE interface
3	GND	GROUND
4	n/c	n/c
5	GND	GROUND
6	n/c	n/c
7	GND	GND
8	n/c	n/c
9	GND	GROUND
10	n/c	n/c
11	GND	GROUND
12	n/c	n/c
13	GND	GROUND
14	SSO	SIDE SELECT OUTPUT to DRIVE interface
15	GND	GROUND
16	n/c	n/c
17	GND	GROUND
18	HEAD LOAD*	HEAD LOAD COMMAND to DRIVE interface
19	GND	GROUND
20	INDEX*	INDEX PULSES from DRIVE interface
21	GND	GROUND
22	READY*	READY STATUS from DRIVE interface
23	GND	GROUND
24	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
25	GND	GROUND
26	DS1*	DRIVE SELECT #1 to DRIVE interface
27	GND	GROUND
28	DS2*	DRIVE SELECT #2 to DRIVE interface
29	GND	GROUND
30	DS3*	DRIVE SELECT #3 to DRIVE interface
31	GND	GROUND
32	DS4*	DRIVE SELECT #4 to DRIVE interface
33	GND	GROUND
34	DIRC*	DIRECTION CONTROL to DRIVE interface
35	GND	GROUND
36	STEP*	STEP COMMAND to DRIVE interface
37	GND	GROUND
38	WRITE DATA*	WRITE DATA to DRIVE interface
39	GND	GROUND
40	WRITE GATE*	WRITE GATE to DRIVE interface

41	GND	GROUND
42	TRACK 0 *	TRACK ZERO STATUS from DRIVE interface
43	GND	GROUND
44	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
45	GND	GROUND
46	READ DATA*	READ DATA to DRIVE interface
47	GND	GROUND
48	n/c	n/c
49	GND	GROUND
50	n/c	n/c

ADAPTER FPB158-30

J3(5 1/4-INCH DRIVE INTERFACE)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	n/c	n/c
3	GND	GROUND
4	n/c	n/c
5	GND	GROUND
6	DS4*	DRIVE SELECT #4 to DRIVE interface
7	GND	GROUND
8	INDEX*	INDEX* PULSE STATUS from DRIVE interface
9	GND	GROUND
10	DS1*	DRIVE SELECT #1 to DRIVE interface
11	GND	GROUND
12	DS2*	DRIVE SELECT #2 to DRIVE interface
13	GND	GROUND
14	DS3*	DRIVE SELECT #3 to DRIVE interface
15	GND	GROUND
16	MOTOR ON*	MOTOR ON COMMAND to DRIVE interface
17	GND	GROUND
18	DIRC*	DIRECTION CONTROL to DRIVE interface
19	GND	GROUND
20	STEP*	STEP COMMAND to DRIVE interface
21	GND	GROUND
22	WRITE DATA*	WRITE DATA to DRIVE interface
23	GND	GROUND
24	WRITE GATE*	WRITE GATE to DRIVE interface
25	GND	GROUND
26	TRACK 0 *	TRACK ZERO STATUS from DRIVE interface
27	GND	GROUND
28	WRITE PROT*	WRITE PROTECT STATUS from DRIVE interface
29	GND	GROUND
30	READ DATA*	READ DATA to DRIVE interface
31	GND	GROUND
32	SS0*	SIDE SELECT OUTPUT to DRIVE interface
33	GND	GROUND
34	n/c	n/c

CONNECTOR REQUIREMENTS

Use the following table to determine the type of mating connector to use:

CONFIGURATION	PART NUMBER	CONNECTOR TYPE
8-INCH OR 5 1/4-INCH W/O ADAPTER	FPB158	CANNON DB23S-731
8-INCH/HEADER PLUG ADAPTER	FPB100-11	ANSLEY 609-5017
5 1/4-INCH/EDGE CONNECTOR ADAPTER	FPB100-22	AMP 840-225F-A34-1
8 OR 5 1/4-INCH ADAPTER	FPB158-30	ANSLEY 609-5017 & AMP 840-225F-A34-1

FPB158	MATING CONNECTORS
J1 - ANSLEY 609-4017	ANSLEY 609-4030 (ICM SUPPLIED)
J2 - CANNON DB25S-731	CANNON DB25P-731 (*see note below)

FPB100-11	MATING CONNECTORS
J1 - CANNON DB25P-731	CANNON DB25S-731 (ICM SUPPLIED)
J2 - ANSLEY 609-5017	ANSLEY 609-5030 (CUSTOMER SUPPLIED)

FPB100-22	MATING CONNECTORS
J1 - CANNON DB25P-731	CANNON DB25S-731 (ICM SUPPLIED)
J2 - (34 PIN EDGE)	AMP 840-225F-A34-1(CUSTOMER SUPPLIED)

FPB158-30	MATING CONNECTORS
J1 - CANNON DB25P-731	CANNON DB25S-731 (ICM SUPPLIED)
J2 - (34 PIN EDGE)	AMP 840-225F-A34-1(CUSTOMER SUPPLIED)
J3 - ANSLEY 609-5017	ANSLEY 609-5030 (CUSTOMER SUPPLIED)

* Customer supplied if connecting directly to FPB158. ICM supplied if using FPB100-XY or FPB158-XY Adapters.

SET UP INSTRUCTIONS

None Required.

4.2.1.6. CENTRONICS PRINTER

PART NUMBER - CPI100

FUNCTION

The Centronics Printer Personality Board provides line drivers, receivers, terminators, jumper options and data strobe generator logic to interface to any printer compatible with the Centronics parallel interface.

This module may be used with either the CPZ-186 SBCP or the CPS-MX SBSP.

INTERFACE REQUIREMENTS

Connects to J4 of CPZ-186 SBCP or CPS-MX SBSP.

J1(CPU)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	RDYA	READY handshake from CPU, Channel A
2	STBA	STROBE handshake to CPU, Channel A
3	n/c	n/c
4	n/c	n/c
5	D0A	DATA BIT 0, Channel A
6	D1A	DATA BIT 1, Channel A
7	D2A	DATA BIT 2, Channel A
8	D3A	DATA BIT 3, Channel A
9	D4A	DATA BIT 4, Channel A
10	D5A	DATA BIT 5, Channel A
11	D6A	DATA BIT 6, Channel A
12	D7A	DATA BIT 7, Channel A
13	D0B	DATA BIT 0, Channel B
14	D1B	DATA BIT 1, Channel B
15	D2B	DATA BIT 2, Channel B
16	D3B	DATA BIT 3, Channel B
17	D4B	DATA BIT 4, Channel B
18	D5B	DATA BIT 5, Channel B
19	D6B	DATA BIT 6, Channel B
20	D7B	DATA BIT 7, Channel B
21	RESET*	RESET from CPU (active low)
22	GND	GROUND
23	n/c	n/c
24	GND	GROUND
25	PCLK	4 MHZ Auxilliary Clock from CPU
26	+5VDC	+5VDC

J2(PRINTER)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DSTR*	DATA STROBE to the Printer
2	DB1	DATA BIT 1 to the Printer
3	DB2	DATA BIT 2 to the Printer
4	DB3	DATA BIT 3 to the Printer
5	DB4	DATA BIT 4 to the Printer
6	DB5	DATA BIT 5 to the Printer
7	DB6	DATA BIT 6 to the Printer
8	DB7	DATA BIT 7 to the Printer
9	DB8	DATA BIT 8 to the Printer
10	ACK*	ACKNOWLEDGE from the Printer
11	BUSY	BUSY Status from the Printer
12	PE	PAPER EMPTY Status from the Printer
13	SELECT	SELECT Status from the Printer
14	n/c	n/c
15	n/c	n/c
16	n/c	n/c
17	CHASSIS GND	Printer Chassis Ground
18	n/c	n/c
19	SIG GND	SIGNAL GROUND
20	SIG GND	SIGNAL GROUND
21	SIG GND	SIGNAL GROUND
23	SIG GND	SIGNAL GROUND
24	FAULT*	FAULT Status from the Printer
25	INPUT PRIME*	RESET to the Printer

CONNECTOR REQUIREMENTS

CPI100	MATING CONNECTORS
J1 - ANSLEY 609-2617	ANSLEY 609-2630 (ICM SUPPLIED)
J2 - CANNON DB25S-731	CANNON DB25P-731 (* see note below)

*NOTE: Customer is to supply cabling from the CPI100 to the Centronics compatible printer. If a flat ribbon cable is desired, one can be provided by using a flat ribbon type DB25 connector at one end and a Centronics type connector (AMP 57-10360 or equivalent) at the other end. In this case, all pins are to be connected at the DB25 end except for pins 24 and 25. The software normally does not use the FAULT status and most printers have a power-up reset circuit and do not need subsequent reset operations; therefore, pins 24 and 25 are not required and a flat ribbon cable will be usable. If all signals are required, the customer must use a descrete wire harness to connect all signals.

SET UP INSTRUCTIONS

If signal ground is to be connected to chassis ground, solder a jumper in jumper area PJA.

If the CPU is to provide reset signals to the printer, solder a jumper in jumper area PJB.

4.2.1.7. PRIAM INTELLIGENT HARD DISK

PART NUMBER - PRI100

FUNCTION

PRIAM provides two intelligent hard disk interface controllers referred to as the "SMART" and the "SMART-E". These are preprogrammed microprocessor based controllers. They may be used for the entire line of PRIAM Winchester disc drives which range in capacity from 10 megabytes to 157 megabytes and come in eight or fourteen inch packaging. Up to four drives in any combination of drive sizes may be interconnected. The controllers support a variety of read sector, write sector and format commands. Data transfers may be either programmed I/O or DMA. The SMART-E has all the features that the SMART has in addition to error detection & correction, logical sector addressing, sector interleaving, parity generation & testing, direct data transfers and a 2 Kbyte data buffer (SMART has a 1 Kbyte buffer). The interface performs the entire function of detailed disc control while presenting to the host a basic and cost effective interface.

The PRI100 Personality Board connects the parallel port of the CPZ-186 SBCP or the CPS-MX SBSP to the SMART or SMART-E controllers. Thus, a very powerful disc subsystem may be directly connected to the ICM line of processors via the PRI100.

A jumper option is provided on the PRI100 to configure it for either the SMART or the SMART-E controller. The controllers mount along the drive sides alleviating the need for additional S-100 Bus slots. An adapter, PRI100-1, is provided allowing direct connection of the PRI100 to the smart controllers.

INTERFACE REQUIREMENTS

Connects to J4 of either the CPZ-186 SBCP or the CPS-MX SBSP.

J1(CPU)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	RDYA	READY handshake from CPU, Channel A
2	STBA*	STROBE handshake to CPU, Channel A
3	n/c	n/c
4	n/c	n/c
5	D0A	DATA BIT 0, Channel A
6	D1A	DATA BIT 1, Channel A
7	D2A	DATA BIT 2, Channel A
8	D3A	DATA BIT 3, Channel A
9	D4A	DATA BIT 4, Channel A
10	D5A	DATA BIT 5, Channel A
11	D6A	DATA BIT 6, Channel A
12	D7A	DATA BIT 7, Channel A
13	D0B	DATA BIT 0, Channel B
14	D1B	DATA BIT 1, Channel B
15	D2B	DATA BIT 2, Channel B
16	D3B	DATA BIT 3, Channel B
17	D4B	DATA BIT 4, Channel B
18	D5B	DATA BIT 5, Channel B
19	D6B	DATA BIT 6, Channel B
20	D7B	DATA BIT 7, Channel B
21	RESET*	RESET from CPU (active low)
22	GND	GROUND
23	PINT*	PORT INTERRUPT (active low)
24	GND	GROUND
25	n/c	n/c
26	+5VDC	+5VDC

J2(MODIFIED PRIAM)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	HCBUS0	HOST DATA BUS 0
3	HCBUS1	HOST DATA BUS 1
4	HCBUS2	HOST DATA BUS 2
5	HCBUS3	HOST DATA BUS 3
6	HCBUS4	HOST DATA BUS 4
7	HCBUS5	HOST DATA BUS 5
8	HCBUS6	HOST DATA BUS 6
9	HCBUS7	HOST DATA BUS 7
10	GND	GROUND
11	HRD*	ENABLE REGISTER TO HOST-BUS
12	GND	GROUND
13	HWR*	ENABLE HOST-BUS TO REGISTER
14	GND	GROUND
15	HAD2	HOST ADDRESS BUS 2
16	HAD1	HOST ADDRESS BUS 1
17	HAD0	HOST ADDRESS BUS 0
18	GND	GROUND
19	RESET*	RESET TO CONTROLLER
20	GND	GROUND
21	HIR*	HOST INTERRUPT
22	DTREQ*	DATA TRANSFER REQUEST TO HOST
23	HREAD*	DATA DIRECTION CONTROL TO CONTROLLER
24	DBUSENA*	CONTROLLER-READY TO HOST
25	BUSREQ*	DATA TRANSFER REQUEST TO HOST (SMART-E ONLY)

ADAPTER PRI100-1

J1(MODIFIED PRIAM)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	HCBUS0	HOST DATA BUS 0
3	HCBUS1	HOST DATA BUS 1
4	HCBUS2	HOST DATA BUS 2
5	HCBUS3	HOST DATA BUS 3
6	HCBUS4	HOST DATA BUS 4
7	HCBUS5	HOST DATA BUS 5
8	HCBUS6	HOST DATA BUS 6
9	HCBUS7	HOST DATA BUS 7
10	GND	GROUND
11	HRD*	ENABLE REGISTER TO HOST-BUS
12	GND	GROUND
13	HWR*	ENABLE HOST-BUS TO REGISTER
14	GND	GROUND
15	HAD2	HOST ADDRESS BUS 2
16	HAD1	HOST ADDRESS BUS 1
17	HAD0	HOST ADDRESS BUS 0
18	GND	GROUND
19	RESET*	RESET TO CONTROLLER
20	GND	GROUND
21	HIR*	HOST INTERRUPT
22	DTREQ*	DATA TRANSFER REQUEST TO HOST
23	HREAD*	DATA DIRECTION CONTROL TO CONTROLLER
24	DBUSENA*	CONTROLLER-READY TO HOST
25	BUSREQ*	DATA TRANSFER REQUEST TO HOST (SMART-E ONLY)

ADAPTER PRI100-1

J2(PRIAM)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	HCBUS0	HOST DATA BUS 0
3	HCBUS1	HOST DATA BUS 1
4	HCBUS2	HOST DATA BUS 2
5	HCBUS3	HOST DATA BUS 3
6	HCBUS4	HOST DATA BUS 4
7	HCBUS5	HOST DATA BUS 5
8	HCBUS6	HOST DATA BUS 6
9	HCBUS7	HOST DATA BUS 7
10	GND	GROUND
11	HRD*	ENABLE REGISTER TO HOST-BUS
12	GND	GROUND
13	HWR*	ENABLE HOST-BUS TO REGISTER
14	GND	GROUND
15	HAD2	HOST ADDRESS BUS 2
16	HAD1	HOST ADDRESS BUS 1
17	HAD0	HOST ADDRESS BUS 0
18	GND	GROUND
19	RESET*	RESET TO CONTROLLER
20	GND	GROUND
21	HIR*	HOST INTERRUPT
22	GND	GROUND
23	HREAD	DATA DIRECTION CONTROL TO CONTROLLER
24	DBUSENA*	CONTROLLER-READY TO HOST
25	GND	GROUND
26	DTREQ*	DATA TRANSFER REQUEST TO HOST
27	GND	GROUND
28	BUSREQ*	DATA TRANSFER REQUEST TO HOST (SMART-E ONLY)
29	GND	GROUND
30	HCBUS8	HOST DATA BUS PARITY (SMART-E ONLY)
31	GND	GROUND
32	RES	RESERVED
33	RES	RESERVED
34	RES	RESERVED
35	RES	RESERVED
36	RES	RESERVED
37	RES	RESERVED
38	RES	RESERVED
39	RES	RESERVED
40	RES	RESERVED

CONNECTOR REQUIREMENTS

PRI100	MATING CONNECTORS
J1 - ANSLEY 609-2617	ANSLEY 609-2630 (ICM SUPPLIED)
J2 - CANNON DB25S-731	CANNON DB25P-731 (* see note below)

PRI100-1	MATING CONNECTORS
J1 - CANNON DB25P-731	CANNON DB25S-731 (ICM SUPPLIED)
J2 - ANSLEY 609-4017	ANSLEY 609-4030 (CUSTOMER SUPPLIED)

* Customer supplied if connecting directly to PRI100. ICM supplied if using PRI100-1 Adapter.

SET-UP INSTRUCTIONS

To configure the PRI100 for the SMART controller, solder a jumper on JA from B-to-C.

To configure the PRI100 for the SMART-E controller, solder a jumper on JA from A-to-B.

4.2.1.8. SHUGART ASSOCIATES SYSTEM INTERFACE

PART NUMBER - SAS100

FUNCTION

The Shugart Associates System Interface(SASI) defines a Local I/O Bus which can be operated at data rates up to 1.5 megabytes per second. This bus provides I/O device independence so that disk drives, tape drives, printers and various other peripherals may be interfaced on the same I/O bus without modification to the host CPU's hardware or software. The interface protocol provides for connection of multiple initiators (devices capable of initiating an operation) and multiple targets(devices capable of responding to requests for operations). Arbitration logic is built in and a priority system awards control to the device that wins arbitration.

The SAS100 personality board converts the parallel port of either the CPZ-186 SBCP or the CPS-MX SBSP to a SASI I/O bus. Software is provided to emit bus timing in conformance with the SASI specification. The system integrator may interface SASI controllers such as the Data Technology Corporation's, Zebec and Sysgen line of controllers. Each have powerful attributes such as connecting hard disks with floppies, hard disks with tape streamers and connecting to high performance SMD type hard disks.

The SAS100 personality board is accompanied by an adapter board (SAS100-1). This adapter board converts the SAS100 DB25 connector interface to a 50 pin header connector interface with a pin assignment in exact conformance with the SASI Bus specification. The integrator may connect directly to the SAS100 with a DB25-to-SASI Interface cable or may connect via the SAS100-1 with a 50 pin flat ribbon cable.

INTERFACE REQUIREMENTS

Connects to J4 of either the CPZ-186 SBCP or the CPS-MX SBSP.

J1(CPU)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	RDYA	READY handshake from CPU, Channel A
2	STBA*	STROBE handshake to CPU, Channel A
3	RDYB	READY handshake from CPU, Channel B
4	STBB*	STROBE handshake to CPU, Channel B
5	DOA	DATA BIT 0, Channel A
6	D1A	DATA BIT 1, Channel A
7	D2A	DATA BIT 2, Channel A
8	D3A	DATA BIT 3, Channel A
9	D4A	DATA BIT 4, Channel A
10	D5A	DATA BIT 5, Channel A
11	D6A	DATA BIT 6, Channel A
12	D7A	DATA BIT 7, Channel A
13	DOB	DATA BIT 0, Channel B
14	D1B	DATA BIT 1, Channel B
15	D2B	DATA BIT 2, Channel B
16	D3B	DATA BIT 3, Channel B
17	D4B	DATA BIT 4, Channel B
18	D5B	DATA BIT 5, Channel B
19	DB6	DATA BIT 6, Channel B
20	D7B	DATA BIT 7, Channel B
21	n/c	n/c
22	GND	GROUND
23	PINT*	PORT INTERRUPT (active low)
24	GND	GROUND
25	n/c	n/c
26	+5VDC	+5VDC

J2(MODIFIED SASI)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	DO	DATA BIT 0
2	D2	DATA BIT 2
3	D4	DATA BIT 4
4	D6	DATA BIT 6
5	GND	GROUND
6	BSY*	BUSY
7	ACK*	ACKNOWLEDGE
8	RST*	RESET
9	MSG*	MESSAGE
10	SEL*	SELECT
11	C/D*	CONTROL/DATA
12	REQ*	REQUEST
13	I/O*	INPUT/OUTPUT
14	D1	DATA BIT 1
15	D3	DATA BIT 3
16	D4	DATA BIT 5
17	D7	DATA BIT 7
18	GND	GROUND
19	GND	GROUND
20	GND	GROUND
21	GND	GROUND
22	GND	GROUND
23	GND	GROUND
24	GND	GROUND
25	GND	GROUND

APAPTER SAS100-1

Connects to J2 of the SAS100 Personality Board.

J1(MODIFIED SASI)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	D0	DATA BIT 0
2	D2	DATA BIT 2
3	D4	DATA BIT 4
4	D6	DATA BIT 6
5	GND	GROUND
6	BSY*	BUSY
7	ACK*	ACKNOWLEDGE
8	RST*	RESET
9	MSG*	MESSAGE
10	SEL*	SELECT
11	C/D*	CONTROL/DATA
12	REQ*	REQUEST
13	I/O*	INPUT/OUTPUT
14	D1	DATA BIT 1
15	D3	DATA BIT 3
16	D4	DATA BIT 5
17	D7	DATA BIT 7
18	GND	GROUND
19	GND	GROUND
20	GND	GROUND
21	GND	GROUND
22	GND	GROUND
23	GND	GROUND
24	GND	GROUND
25	GND	GROUND

ADAPTER SAS100-1

J2(SASI)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	GND	GROUND
2	DB0	DATA BIT 0
3	GND	GROUND
4	DB1	DATA BIT 1
5	GND	GROUND
6	DB2	DATA BIT 2
7	GND	GROUND
8	DB3	DATA BIT 3
9	GND	GROUND
10	DB4	DATA BIT 4
11	GND	GROUND
12	DB5	DATA BIT 5
13	GND	GROUND
14	DB6	DATA BIT 6
15	GND	GROUND
16	DB7	DATA BIT 7
17	GND	GROUND
18	n/u	n/u
19	GND	GROUND
20	n/u	n/u
21	GND	GROUND
22	n/u	n/u
23	GND	GROUND
24	n/u	n/u
25	GND	GROUND
26	n/u	n/u
27	GND	GROUND
28	n/u	n/u
29	GND	GROUND
30	n/u	n/u
31	GND	GROUND
32	n/u	n/u
33	GND	GROUND
34	n/u	n/u
35	GND	GROUND
36	BSY*	BUSY
37	GND	GROUND
38	ACK*	ACKNOWLEDGE
39	GND	GROUND

40	RST*	RESET
41	GND	GROUND
42	MSG*	MESSAGE
43	GND	GROUND
44	SEL*	SELECT
45	GND	GROUND
46	C/D*	CONTROL/DATA
47	GND	GROUND
48	REQ*	REQUEST
49	GND	GROUND
50	I/O*	INPUT/OUTPUT

CONNECTOR REQUIREMENTS

SAS100	MATING CONNECTORS
J1 - ANSLEY 609-2617	ANSLEY 609-2630 (ICM SUPPLIED)
J2 - CANNON DB25S-731	CANNON DB25P-731 (*see note below)

SAS100-1	MATING CONNECTORS
J1 - CANNON DB25P-731	CANNON DB25S-731 (ICM SUPPLIED)
J2 - ANSLEY 609-5017	ANSLEY 609-5030 (CUSTOMER SUPPLIED)

* Customer supplied if connecting directly to SAS100. ICM supplied if using SAS100-1 Adapter.

SET-UP INSTRUCTIONS

(none required)

4.2.1.9. PERSONALITY BOARD - CLOCK/CALENDAR

PART NUMBER - CCB100

FUNCTION

The CCB100 provides a highly accurate real time clock which may be set by the CPZ-186 SBCP or the CPS-MX SBSP under software control. The time of year, month, day, hour, minute and second is maintained and may be read back by the CPU. A Ni-Cad battery is used to provide backup power to the time control chip. In this manner the real time clock is continously maintained even during extensive down time. This feature is quite useful for point-of-sale systems, inventory systems and other applications where continous clock monitoring is required. This board is also very useful in operating systems which feature date and time stamping such as TurboDOS. In a TurboDOS based system, this board may be connected to the master (CPZ-186) parallel port or may be connected to any one slave (CPS-MX) parallel port.

INTERFACE REQUIREMENTS

Connects to J4 of the CPZ-186 SBCP or the CPS-MX SBSP. No other interface cable is required.

CPU (J1)

PIN NO.	SIGNAL NAME	DESCRIPTION
1	n/c	n/c
2	GND	GROUND
3	n/c	n/c
4	n/c	n/c
5	D0A	DATA BIT 0, Channel A
6	D1A	DATA BIT 1, Channel A
7	D2A	DATA BIT 2, Channel A
8	D3A	DATA BIT 3, Channel A
9	GND	GROUND
10	GND	GROUND
11	GND	GROUND
12	GND	GROUND
13	D0B	DATA BIT 0, Channel B
14	D1B	DATA BIT 1, Channel B
15	D2B	DATA BIT 2, Channel B
16	D3B	DATA BIT 3, Channel B
17	D4B	DATA BIT 4, Channel B
18	D5B	DATA BIT 5, Channel B
19	D6B	DATA BIT 6, Channel B
20	D7B	DATA BIT 7, Channel B
21	n/c	n/c
22	GND	GROUND
23	n/c	n/c
24	GND	GROUND
25	n/c	n/c
26	+5VDC	+5VDC

J1(CPU) - Ansley 609-2617 or equivalent

SETUP INSTRUCTIONS

No hardware setup instructions are required, software instructions follow.

The CCB-100 can be used under CP/M by attaching the personality board to the parallel port of the CPZ-186 and using the CLKSETM program to set the time and DSPCLKM to display the time.

Under the TurboDOS operating system the CCB-100 clock module can be placed on the CPZ-186 master or any CPS-MX slave processor.

If the CCB is on the CPZ-186 the user can set the time by using the program CLKSETM. The time can be displayed by executing the program DSPCLKM. The CCB can automatically be read as system date and time when the module MSTRCLK is included in the sys file that is loaded into the CPZ-186.

If the CCB is on the CPS-MX slave processor the user can set the time by using the program CLKSETS. The time can be displayed by executing the program DSPCLKS. The CCB can automatically be read as system date and time when the program SLVCLK.AUT is executed as a TurboDOS cold start program. There is no problem executing the program if the card is not attached, since it will simply return to the operating system.

5. CRT TERMINAL SET-UP INSTRUCTIONS

Firmware in the CPZ-186 is structured to communicate with RS232 terminals and with the terminals set-up in a particular fashion. The terminals must be set-up as described below otherwise booting-up the CPZ-186 will not be possible. The description given is for CP/M configurations only. Refer to the " TurboDOS Users Guide" manual for instructions on setting-up terminals for TurboDOS based systems:

number of stop bits = 2
number of data bits = 8
parity bit = not used

When the CPZ-186 is shipped configured for CP/M, a PROM monitor is installed which stores firmware to examine the baud rate of the terminal. The user must strike the "return" key function until the CPZ-186 adjusts itself to the rate set-up on the terminal. The baud rates which may be set-up on the terminal are listed below:

Baud Rate
300
600
1200
2400
4800
9600
19200
38400
76800

6. HARD DISK COMPATABILITY GUIDE

Two general methods exist for integrating hard disk drives with the CPZ-186. The system integrator may install hard disk drives through the parallel port using personality boards which interface to various intelligent hard disk controllers. The other option is to install an IEEE hard disk controller in the S100 Bus. Each are discussed below:

6.1. PARALLEL PORT INTERFACE

The following personality boards are available:

-PRIAM INTELLIGENT HARD DISK PERSONALITY BOARD [PRI100]

Compatible with "SMART" or "SMART-E" Priam Intelligent Hard Disk controller. These controllers interface any PRIAM drive ranging in capacity from 10 to 157 megabytes and in 8 or 14 inch packaging.

-SHUGART ASSOCIATES SYSTEMS INTERFACE (SASI) PERSONALITY BOARD [SAS100]

Compatible with boards from the following manufactures:

-XEBEC Systems, Inc.
-Data Technology, Corp.
-Sysgen, Inc.

Any SASI compatible controller should interface with the SAS100. Some boards provide features others don't. For example, a XEBEC board allows SMD removable hard disk drives whereas the Sysgen controller interfaces ST506 drives on the same bus astape streaming drives.

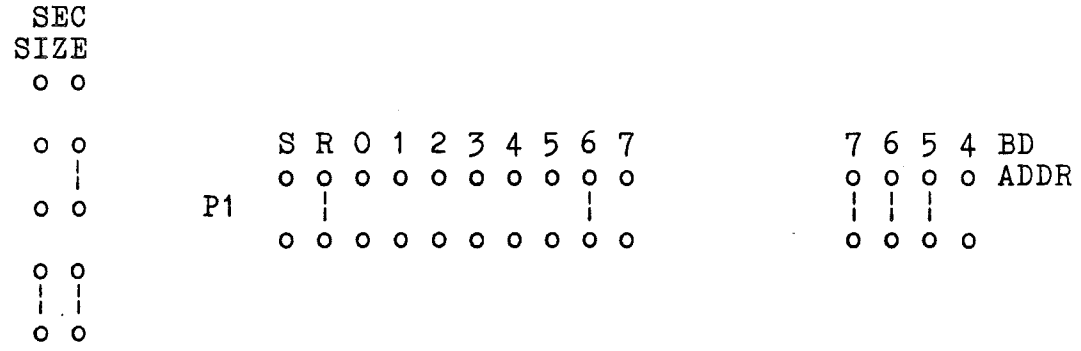
6.2. S100 BUS

Any hard disk controller which is IEEE 696.D2 compatible may be integrated with the CPZ-186. CP/M and TurboDOS drivers are available for the MONITOR DYNAMICS, INC. hard disk controller. TurboDOS drivers are also available for the ADES GYPSY hard disk/tape backup controller and the KONAN SMC200 Removable Cartridge hard disk drive controller. In any case, the only condition imposed is that the controller be compatible with the IEEE specification.

Much experience has been gained in the field with the MONITOR DYNAMICS hard disk controller and has received wide acceptance. Benchmarks have proven this controller to be the fastest in its catagory. To assist in the integration of this controller, the following instructions are given:

The CPZ-186 communicates with the Monitor Dymnamics Modules 1010, 1012 , 1016 or 1013 using I/O port 10 hex and interrupt vector 6 on the S-100 bus. The CPZ-186 must be configured to detect interrupt vector 6 (see JUMPER OPTIONS-JF section). To enable the Monitor Dynamics controller to communicate and issue interrupts, place jumpers as follows:

6.2.1. MD1010 & MD1013 JUMPER OPTIONS



6.2.2. MD1013/1016 Drive List

The Monitor Dynamics 1013 controller uses the program TESTMD1.COM to format, verify and set the disk paramater block for the hard disk drive being used. The model 1013 is being directly sold and supported by I.C.M. and will control the following 5 1/4" drives:

Manafacture	Model
Atasi	3020/3033/3046
Rodime	R201/202/203/204
Tandon	TM602/TM603/TM502/TM503
Quantum	Q2020/30/40/80 (8 inch)
Ampex Corp.	PYXIS 7/13/20/27
Seagate Technology	ST506/406/412/418
CDC	9415-5 (WREN)
Micropolis	1303/1303/1304
Evotek	ET-5510/ET-5520/ET-5530/EY-5540
Vertex	V130/V150/V170
Shugart Associates	SA1002/1004 (8 Inch)
Shugart Associates	SA602/604/606
Computer Memories	CM5206/5412/5619
Rotating Memories	RM504/509/513/518
Memorex/Fujitsu	MRX101/102 (8 inch)
Memorex/Fujitsu	MRX306/310/512/513/514
Olivetti	HD561/1,/2,/3
IMI	5006H/5012H/5018H
Miniscribe Corp.	II (2006), III (3006,3012) IV (2012)

For other controller models or Hard disk drives please feel free to contact Richard Turner or Gary Clinard at Monitor Dynamics Inc., 1121 West 9th Street, Upland CA (714)985-7214.

7. SOFTWARE SECTION

This section of the manual describes the Software Interface for the CPZ-186.

The 16-bit master uses a powerful 80186 processor to orchestrate the S-100 network. It operates at 8-Mhz with no onboard wait-states required. It comes standard with 256K of memory, which is expandable to 1 megabyte with the addition of 256*1K dynamic RAM chips and a prom. Its advantage as a network server over an 8-bit master is not only the increased word size and speed, but also its abundance of onboard memory. It can now handle any size operating system without sacraficing TPA space. The onboard caching buffers can also be increased to an amount which is more suitable for system disk buffering. With the 1 megabyte option, 700k can be alotted for buffers. As with the rest of ICM's products, the 16-bit master can be used with the rest of our existing products, including 8-bit slaves.

Before bringing up the system, go over the Bus Voltage Check discussed at the beginning of the manual. The board will boot up using the configuration 'I' 80186 boot diskette. If using a Monitor Dynamics Hard Disk controller, there is a 16-bit version of the format program available on the ICM-UTILITIES diskette named TESTMD.CMD. The configuration 'I' diskette also comes with a 16-bit assembler(TASM.CMD) and a linker program which replaces the 8-bit GEN command(TLINK.CMD).

T A B L E 7-1: File extensions used in 16-bit TurboDOS

??????.CMD	-------->	16-bit executable commands
??????.O	-------->	16-bit relocatable files
??????.A	-------->	16-bit source files

Both 8 and 16-bit files can exist on the same drive since both operating systems will default to their own extensions. Only the 16-bit operating system will look for .CMD files when executing a command. For good housekeeping, it is a good idea to keep the .REL and .O files on seperate users.

8. CPZ-186 MASTER MULTI-USER SYSTEM CHANGES

8.1. Software Modifications

The GEN files for the CPZ-186 based system are very similar to the 8-bit versions presented earlier. The PAR files are very similar as well. The major difference between the 8-bit and the 16-bit versions is that to denote a HEX (base 16) number has to be preceded with a 0x, and word (16 bit wide) number or address are in [ ] instead of ().

Listed below is an example of the port assignment table for the slaves:

```
8-bit       PATCSA = 7F,7E,7C

16-bit      PATCSA = 0x7F,0x7E,0x7C
```

The 0x tells the linker TLINK to take the next number as a hex value. If the 0x is not supplied, the number is assumed to be decimal.

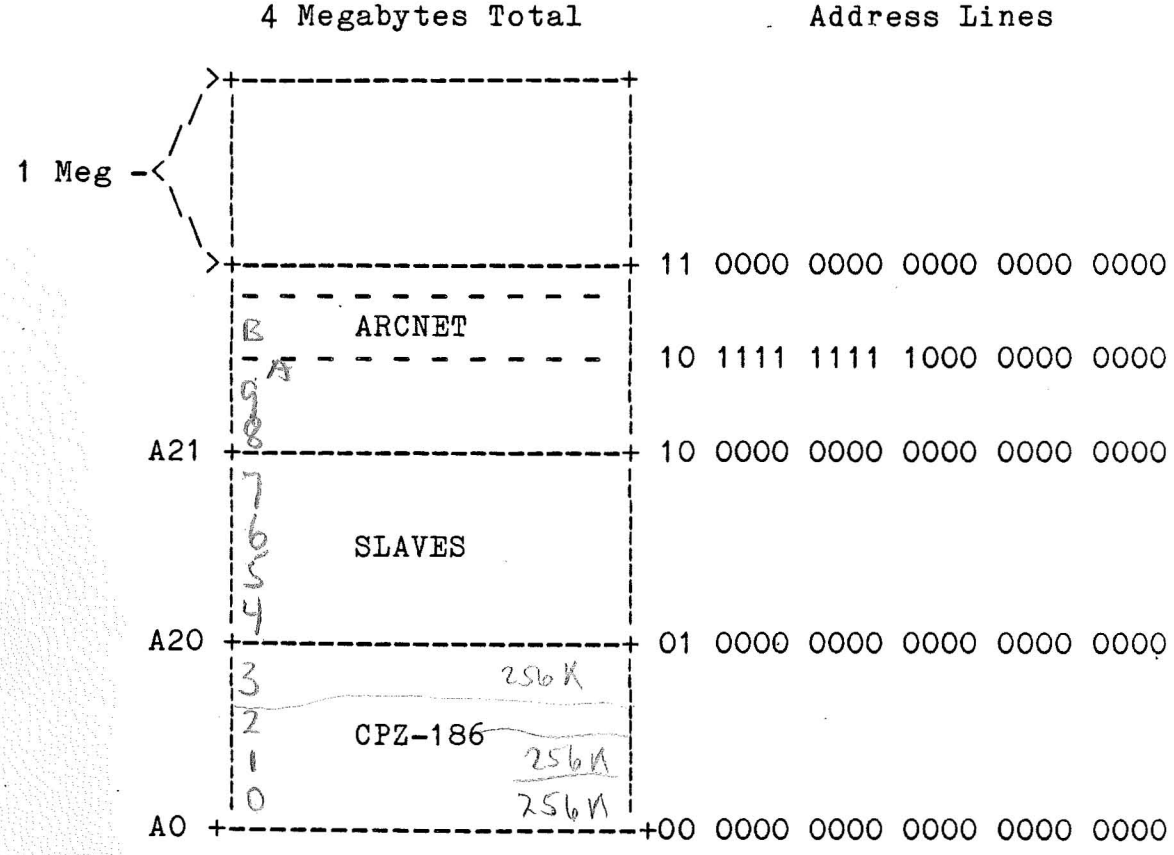
Below is an example of the CPZ-186 as a multi-user master supporting 8-bit and 16-bit slaves:

```
;CPZMST16.GEN    CPZ-186 NETWORK MASTER
STDMASTR         ; STANDARD 16-Bit Multi-User OS
;NETREQ          ; Support module for Despooling/Networking
;MSGFMT          ;    "       "       "       "          "
;NETLOD          ;    "       "       "       "          "
;NETFWD          ;    "       "       "       "          "
PATCH            ; SYSTEM PATCH AREA
RTCCPZ           ; RTC DRIVER MODULE
CPMSUP           ; CP/M-86 SUPPORT MODULE
MSTCPZ           ; MEMORY DESCRIPTOR TABLE
NITCPZ           ; HARDWARE INITIALIZATION
CONREM           ; Use remote console for slaves
;CON192          ; NULL 19.2K BAUD CONSOLE DRIVER
;CON96           ; NULL 9600 BAUD CONSOLE DRIVER
SPDCPZ           ; SERIAL/PARALLEL I/O DRIVER
LSTCTS           ; LIST CLEAR-TO-SEND DRIVER
;LSTPAR          ; LIST PARALLEL DRIVER
DSKCPZ           ; CPZ-16 Floppy disk driver module
DSKFMT8          ; ICM 8" TurboDOS disk format tables
MD131DRV         ; Monitor Dynamics MD-1013 with 1 drive
;MD132DRV        ; Monitor Dynamics MD-1013 with 2 drives
MCDCPS           ; 16 Bit Master/Slave circuit driver module
;LANCPZ          ; TurboLAN Network Driver Module
```

```
;CPZMST16.PAR   CPZ-186 NETWORK MASTER
SRHDRV = 0xFF  ; Search Default System Drive
COMPAT = 0xF8  ; Compatibility Flags
AUTUSR = 0x80  ; Auto Logon to User 0 Privileged
;
CONAST = 0x01,CONDRA  ; Assign Console to channel 1 serial
;
;
PATCSA = 0x7F,0x7E,0x7D,0x7C     ; Status port table for CPS-MX)
         0x7B,0x7A,0x79,0x78     ; (default values)
         0x3F,0x3E,0x3D,0x3C     ; Status port table for CPS-16
         0x3B,0x3A,0x39,0x38     ; (default values)
;
SSTCSA = " B       "      ; 0/S suffix table for CPS-MX
         "ZZZZZZZZ"       ; 0/S suffix table for CPS-16
PTRAST = 0x00,LSTDRA      ; assign printer to port 00
QUEAST = 0x00,(0x0000)
;
LDCOLD = 0x00             ; Cold start autoload flag
;COLDFN = 0,"TDTIMEM CMD"; Init TurboDOS Date/Time function
LDWARM = 0x00             ; Warm start autoload flag
;
OSMLEN = (1024)                   ; allow 16K for Dyn. expansion
;
NMBUFS = 128                      ; Number of disk buffers
BUFBAS = (0x2000)                 ; 16K memory segment reserved
BUFLEN = (8192)                   ; paragraphs to reserve
MEMTBL+3 = (0x1FFF-0x0050)        ; define memory we have for 0/S
;
BFLDLY = (0x64)                   ; Flush buffers every 2 seconds
;
FLSRT = 0x01                      ; 6ms step rate
DSKAST = 0x01,DSKDRB,0x00,DSKDRA,0X01,DSKDRA,0X02,DSKDRA
         0XFF,(0X0000),0XFF,(0X0000),0XFF,(0X0000),0XFF,(0X0000),
         0XFF,(0X0000),0XFF,(0X0000),0XFF,(0X0000),0XFF,(0X0000),
         0XFF,(0X0000),0XFF,(0X0000),0XFF,(0X0000),0XFF,(0X0000)
```

8.2. Hardware Modifications

The CPZ-186 TurboDOS software is set up to occupy the first 16 64K pages of memory, which is equal to 1 megabyte in length. In a system that had been using the 8-bit CPZ-48000, the slaves will have to be re-addressed since their old settings will conflict with the new master's address space usage. The CPZ-186 dedicates the first MEG of ram to internal use even if the larger RAM chips are not installed. Commencing at the 1 MEG boundary, a CPS-MX 8-bit will occupy the next 64K, and the CPS-16 will start at the same base address, but extend over the next 4 64K pages. Having all the slaves at the same address is possible since only one slave is active at a time. The activation to triggered via their unique I/O port number.

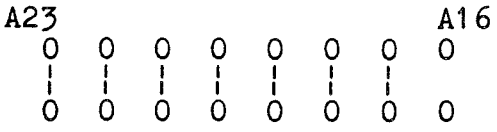


T A B L E 8-1: CPZ-186 Memory Usage Map

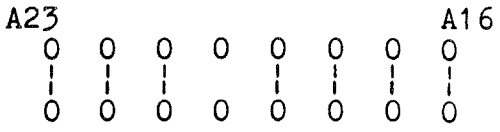
You will need a de-wire wrapping tool or a pair of tweezers and some jumpers to make the Extended Addressing Select changes:

8-BIT SLAVES
CPS-MX or CPS-BMX

Using CPZ4800x:

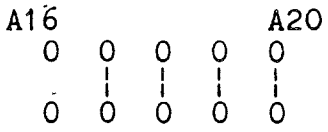


Using CPZ-186:

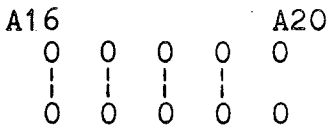


16-Bit SLAVES
CPS-16

Using CPZ4800x:

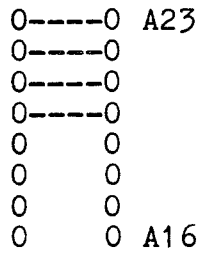


Using CPZ-186:

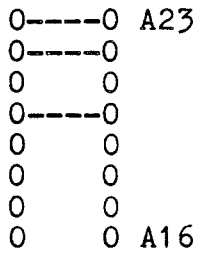


LANS100
S-100 ARCnet controller in memory mapped mode

Using CPZ4800x:



Using CPS-16:



8.3. 16 Bit Slaves

The 16-bit CPZ-186 works without modification with our 8-bit slave boards but have a slight modification to any CPS-16A REV A boards you may have must be incorporated in order to run reliably with the CPZ-186. Cut the trace between pins 4 and 5 of U18 on the back of the board. Also cut the trace leading to pin 2 of U37on the back of the board. Add a jumper wire from U37 pin 4 to U18 pin 4 and also a jumper from U18 to pin 5 to U67 (74LS240) pin 8. Also jumper PJM should be changed to alloOw for 16-bit transfers by cutting the horizontal trace on the back and adding a jumper to the two vertical pads.

Configuring a system with 16-bit slaves is very similar to 8-bit slaves. The main difference is the Relocatable modules are called *.O files instead of *.REL files. Another difference is that all values in the parameter files have to be preceded with a Ox to distinguish it as a HEX value; the default base is decimal. All 16-bit executable commands have a tail defaulting at *.CMD. There is a Z80 emulator on the MUTD-G disk called TZ80.CMD. This will enable you to run most CP/M-80 programs from a 16-bit slave. Because each command needs to run through the emulator, you can expect a substantial decrease in speed especially with screen intensive programs. To exit the emulator, press the "break" and "^C" keys.

The 16-bit slave uses the "CPSSLV16" Gen and Par files available on the MUTD-G diskette. This can be modified on either an 8-bit editor or a 16-bit editor. Since this is a 16-bit operating system, it can only be linked on a 16-bit processor. It is a good rule of thumb to link this file before modifying the master Gen and Par files. TLINK.CMD will replace the GEN.COM on the 16-bit slave:

```
OA}TLINK CPSSLV16 OSSLAVEZ.SYS
```

The diskette already has an "OSSLAVEZ.SYS" file on it which only assumes a 9600 baud terminal with 8 data bits, 2 stop bits, and no parity. The "CPZMASTR.PAR" already has this suffix included in its slave suffix table at port address 3F,3E,3D,3C,3B,3A,39 and 38:

```
PATCSA = 7F,7E,7D,7C        ; Status port table for CPS-MX
         7B,7A,79,78
         3F,3E,3D,3C        ; Status port table for CPS-16
         3B,3A,39,38
;
SSTCSA = "         "        ; O/S suffix table for CPS-MX
         "ZZZZZZZZ"         ; O/S suffix table for CPS-16
```

Below is an example CPSSLV16 Gen and Par File with a Direct Printing Local Serial Printer:

CPSSLV16.GEN

```
STDSLAVE        ; Standard 16 Bit Slave O/S Module
PATCH           ; Include system patch area
SOMCPS          ; I.C.M. Slave Sign-on Module
RTCNUL          ; Null RTC driver module
CPMSUP          ; CP/M-86 Support Module
MSTCPS          ; 16 Bit Memory Descriptor Table
NITCPS          ; 16 Bit Slave hardware init module
;CON192          ; Include Null 19.2K baud Terminal Driver
CON96           ; Include Null 9600 baud Terminal driver
SPDCPS          ; Include Serial I/O driver
LSTCTS          ; Include Serial List Clear to Send driver
;LSTPAR          ; Include Parallel List Null Driver
RESCPS          ; 16 Bit Slave Keyboard Reset Module
SCDCPS          ; 16 Bit Slave Circuit Driver Module
```

CPSSLV16.PAR

```
USRSOM = OXOD,OXOA,"ICM CPS-16 SLAVE"
SRHDRV = OxFF  ; Search Default System Drive
COMPAT = OxF8  ; Compatibility Flags
AUTUSR = Ox80  ; Auto Logon to User O Privileged
;
ATNCHR = OxOO  ; Define Attention Char as BREAK Key
RESKEY = OXOO  ; Define the reset detection key
CONAST = Ox01,CONDRA  ; Assign Console to channel 1 serial
FFCHR  = Ox1A  ; Console form feed character
;
PTRAST = OxOO,LSTDRA ; Local printer as A
PRTMOD = O             ; Print Mode = Direct
;
LDCOLD = OxOO  ; Cold start autoload flag (O=disabled,FF=enabled)
LDWARM = OxOO  ; Warm start autoload flag (O=disabled,FF=enabled)
;
;TurboDOS V1.4x Patches
;Patch # 1.41-2
;
CINTRY+OX41 = OXE9,[PATCH]
PATCH = OX31,OXCO,OX89,OXC3,OXC3
;
;Patch # 1.41-3
;
PLFCN+OX39 = [PATCH+OXO5]
PATCH+OXO5 = OXC7,OX06,BASPAG-OX128,OXOO,OXOO,OXE9,[INCMOB]
```

8.3.1. One Megabyte 16 Bit Slaves

With the availability of the 256k*1 dynamic RAM chips, the CPS-16 can hold up to 1 Megabyte, with the addition of a PAL #(310-04) in IC location U66. For this example, we'll use the name OSSLAVE1.SYS for the 1 Megabyte Slave.

As described in Appendix B, the Processor Reset Memory segment table needs to be modified in the CPZMASTR.PAR so that it knows where the last 64k is on the slave:

CPZMASTR.PAR

```
AUTUSR = 80                ; Auto Log-on to user one, privileged
SRHDRV = OFF               ; Search system disk for command files
;
;------- If Using CB-80 V1.3 Only ---------------+
;       Must be the same in Slave files          |
;COMPAT = OB8    ; If using CB-80 V1.3            |
;CPMVER = 22     ; Inhibit CB-80 Record Locking   |
;-------- ELSE using CB-80 V1.4 -----------------+
;       Must be the same in Slave files          |
COMPAT = OF8     ; File/Record Locking Flags      |
;CPMVER = 30     ; Allow CB-80 Record Locking     |
;-----------------------------------------------+
;
NMBUFS = OA                ; Default number of Disk Buffers (hex)
BUFSIZ = 03                ; Default disk buffer size
MEMRES = (0400)            ; Dynamic Memory Expansion of TurboDOS
;
PATCSA = 7F,7E,7D,7C       ; Status port table for CPS-MX
         7B,7A,79,78
         3F,3E,3D,3C       ; Status port table for CPS-16
         3B,3A,39,38
;
SSTCSA = "         "       ; O/S suffix table for CPS-MX
         "1Z1ZZZZZ"        ; O/S suffix table for CPS-16
;
PRM86A = 3,3,3,3,3,3,3,3   ; Processor Reset Memory Table
         OE,3,OE,3,3,3,3,3
;
NMBCKT = 2
;
CONAST = 01,CONDRA         ; Console on port 1 of CPZ-48000
STOPBB = 44
;
PTRAST = 00,LSTDRA         ; List assignment table
STOPBA = 44
;
(Rest of File omitted)
```

As shown in Appendix B, the Memory Table Variable has to be changed in the CPSSLV16.PAR file:

CPSSLV16.PAR

```
USRSOM = OXOD,OXOA,"ICM CPS-16 SLAVE"
SRHDRV = OxFF  ; Search Default System Drive
COMPAT = OxF8  ; Compatibility Flags
AUTUSR = Ox80  ; Auto Logon to User 0 Privileged
;
ATNCHR = OxOO  ; Define Attention Char as BREAK Key
RESKEY = OXOO  ; Define the reset detection key
CONAST = Ox01,CONDRA  ; Assign Console to channel 1 serial
FFCHR  = Ox1A  ; Console form feed character
MEMTBL+3 = [0x0000-0x50] ;Memory table for 1MByte slave
;
PTRAST = OxOO,LSTDRA
PRTMOD = O
;
LDCOLD = OxOO  ; Cold start autoload flag (O=disabled,FF=enabled)
LDWARM = OxOO  ; Warm start autoload flag (O=disabled,FF=enabled)
;
;TurboDOS V1.4x Patches
;Patch # 1.41-2
;
CINTRY+OX41 = OXE9,[PATCH]
PATCH = OX31,OXCO,OX89,OXC3,OXC3
;
;Patch # 1.41-3
;
PLFCN+OX39 = [PATCH+OXO5]
PATCH+OXO5 = OXC7,OXO6,BASPAG-OX128,OXOO,OXOO,OXE9,[INCMOB]
```

This example shows the 1-Megabyte,16-bit slaves at I/O port 3F and 3D. Type in TPA to verify.

8.3.2. Mixing 8 And 16 Bit Slaves

The suffix table supplied with the CPZMASTR.PAR file has already made provisions for mixing 8 and 16-bit slaves. The Master Circuit Driver is set up to address the first eight 8-bit slaves and the first eight 16-bit slaves:

```
PATCSA = 7F,7E,7D,7C     ; Status port table for CPS-MX
         7B,7A,79,78
         3F,3E,3D,3C     ; Status port table for CPS-16
         3B,3A,39,38
;
SSTCSA = "        "      ; O/S suffix table for CPS-MX
         "ZZZZZZZZ"      ; O/S suffix table for CPS-16
```

The Circuit Initialization routine will go to the bus and scan for the number of slaves present in a system, so the number of slaves does not have to be modified in the CPZMASTR.PAR as in the older versions of Turbo-Dos. The 8-bit slaves are defaulted as OSSLAVE.SYS (no suffix) and the 16-bit slaves are defaulted as OSSLAVEZ.SYS (Z suffix). This enables you to bring up a basic Turbo-Dos operating system with no modifications.

If you need to include more than eight 8-bit slaves or more than eight 16-bit slaves in your system, you will need to include the Slave Master Circuit Driver twice. When you gen the CPZMASTR with two drivers, it will give you a duplicate symbol error, but it will still link that module in corectly. A second Port Assignment Table and Suffix Table will also have to be patched. Refer to the example below: .ii more than 16 slaves

```
CPZMASTR.GEN
STDMASTR          ; Standard networking master
PATCH             ; Include PATCH Module
FASLOD            ; Use fast disk loader module
;NETREQ           ; Network request module
;MSGFMT           ; Message format module
CPMSUP            ; CP/M function support module
CONREM            ; Use remote console module
;CON192           ; Null 19.2 Kbaud Console Driver
;CON96            ; Null 9600 Baud Console driver
NITCPZ            ; CPZ-48000 hardware initialization
SPDCPZ            ; CPZ-48000 Serial and Parallel I/O
LSTCTS            ; TI810 CTS Driver (LSTDR@ assigned to LSTDRA)
;LSTPAR           ; Parallel Printer driver (CPI-100 centronics board)
RTCCPZ            ; CPZ-48000 real time clock driver
;MSTRCLK           ; Include ICM ccb board drvr to set TurboDOS date/time
DSKCPZ            ; CPZ-48000 floppy disk driver
DSKFMT8           ; Disk specification tables for 8-INCH diskettes
;MD131DRV         ; Monitor Dynamics Model #1013 with 1 drive installed
;MD132DRV         ; Monitor Dynamics Model #1013 with 2 drive installed
MCDCPS            ; 1st CPZ-48000 / CPS-MX Master Circuit Driver
MCDCPS            ; 2nd CPZ-48000 / CPS-MX Master Circuit Driver
```

CPZMASTR.PAR

```
AUTUSR = 80              ; Auto Log-on to user one, privileged
SRHDRV = OFF             ; Search system disk for command files
;
;------- If Using CB-80 V1.3 Only ---------------+\
;      Must be the same in Slave files            \  "Don't forget
;COMPAT = 0B8    ; If using CB-80 V1.3              + the patches for
;CPMVER = 22     ; Inhibit CB-80 Record Locking    /  CB-80 V1.3"
;-------- ELSE using CB-80 V1.4 -----------------+/
;      Must be the same in Slave files           |
COMPAT = 0F8     ; File/Record Locking Flags     |
;CPMVER = 30     ; Allow CB-80 Record Locking    |
;-----------------------------------------------+
;
NMBUFS = 0A              ; Default number of Disk Buffers (hex)
BUFSIZ = 03              ; Default disk buffer size (log2(size/128))
MEMRES = (0400)          ; Allow for Dynamic Memory Expansion of TurboDOS
;
PATCSA = 7F,7E,7D,7C     ; Status port table for CPS-MX (std default value
         7B,7A,79,78
         3F,3E,3D,3C     ; Status port table for CPS-16 (std default value
         3B,3A,39,38
PATCSB = 77,76,75,74     ; 2nd Status port table for CPS-MX
         73,72,71,70
         37,36,35,34     ; 2nd Status port table for CPS-16
         33,32,31,30
;
SSTCSA = "        "      ; 0/S suffix table for CPS-MX (std default values
         "1Z1ZZZZZ"      ; 0/S suffix table for CPS-16 (std default values
SSTCSB = "        "      ; 2nd 0/S suffix table for CPS-MX
         "ZZZZZZZZ"      ; 2nd 0/S suffix table for CPS-16
;
PRM86A = 3,3,3,3,3,3,3,3,0E,3,0E,3,3,3,3,3
;
NMBCKT = 2
;
CONAST = 01,CONDRA       ; Console on port 1 of CPZ-48000
STOPBB = 44
;
PTRAST = 00,LSTDRA       ; List assignment table
STOPBA = 44
;
FLSRT  = 01              ; 6ms Floppy step rate (0=3ms,1=6ms,2=10ms,3=15ms
ATNCHR = "^@"            ; New attention character (Break Key)
PRTMOD = 0               ; Print mode (0 = direct, 1 = Spooled)
;
;MPAGE = 02              ; Use memory above slave address for Turbo-disk
;
;following defines floppy disk drives
;
(rest of file omitted)
```

8.4. Automatic Logon

Automatic Logon provides a method of having each slave execute the LOGON command after it boots. This command will not let you into the system unless you can enter the correct ID and password. Each slave booting in this method will need its slaves PAR file modified:

```
;LOGON SECURITY PASSWORD INCLUDED
;AUTUSR = 80     ; DEFAULT = User 0, Privileged
                  ;(delete for LOGON function)
SRHDRV = OFF     ; Search System Disk for .COM Files
;
;------- If Using CB-80 V1.3 Only ---------------+
;    Must be the same in CPZMASTR.PAR file       |
;COMPAT = 0B8     ; If using CB-80 V1.3           |
;CPMVER = 22      ; Inhibit CB-80 Record Locking  |
;-------- ELSE using CB-80 V1.4 -----------------+
;    Must be the same in CPZMASTR.PAR file       |
COMPAT = 0F8     ; File/Record Locking Flags     |
;CPMVER = 30      ; Allow CB-80 Record Locking    |
;-----------------------------------------------+
;
ATNCHR = "^@"    ; Use "BREAK" Key for Attention
RESKEY = "^\"    ; Define slave reset key
;
CONAST = 01,CONDRA       ; Console on port 1 of CPS-80
STOPBB = 44              ; Define 1 stop bit
;
LDCOLD = 000     ; Disable Cold start autoload
                 ; (change to OFF to enable)
LDWARM = OFF     ; Enable Warm start autoload
                 ; (change to 000 to disable)
;
;LDCOLD must be enabled if you want to auto-init TurboDOS time
;and date function using the ICM CCB-100 Clock/Calendar board.
;
;COLDFN = 0,"SLVCLK  ","AUT"     ;init system clock
WARMFN = 0,"WARMSTRT","AUT"      ;init system logon
;
;patch for TurboDOS V1.4x
;Patch # 1.41-1
;
CINTRY+22 = PATCH,44
PATCH = 21,00,00,7D,0C9
```

Here is an example 16-bit slave PAR file with LOGON enabled:

```
USRSOM = 0x0D,0x0A,"Intercontinental Micro Systems, Corp."
         0x0D,0x0A,"8086 Slave."
;
SRHDRV  = 0xFF           ;Search Default System Drive
COMPAT  = 0xF8           ;Compatability Flags
;AUTUSR = 0x80           ;Auto Logon to User 0 Privileged
;
ATNCHR = 0x00            ;Define Attention Char as "BREAK" Key
CONAST = 0x01,CONDRA     ;Assign Console to Port B Serial
;
MAXMBS = 0x03
MAXRPS = 0x03
;
MEMTBL+3 = (0x3FFF-0x0050)       ; 256 Kbyte Slave memory specs
;MEMTBL+3 = (0x7FFF-0x0050)      ; 512 Kbyte Slave memory specs
;MEMTBL+3 = (0xFFFF-0x0050)      ; 1 Mbyte Slave memory specs
;
LDCOLD = 0x00              ;Disable Cold Load function
;COLDFN = 0,"TDTIMES CMD";Init TurboDOS Data/Time using CCB-100
LDWARM = 0xFF              ;Enable Warm Load function
WARMFN = 0,"16STRT  ","AUT";Execute Logon Security
```

"AUTUSR = 80" was disabled so that the slave would come up on user 31. "LDWARM = OFF" was enabled so that the slave could automatically execute a command on boot. The command which it will execute will be WARMSTRT.AUT which is a default parameter in the CPSSLAVE.PAR file (WARMFN = 0,"WARMSTRT","AUT"). The command that needs to be executed is LOGON.COM which is in user 0, so copy the LOGON.COM command to user 31 and rename it WARMSTRT.AUT. If you are using both 16-bit and 8-bit slaves, you will have to make WARMFN patch on the 16-bit slave equal to a different name such as 0,"16STRT","AUT". Then copy LOGON.CMD to user 31, renaming it 16STRT.AUT.

```
OA}COPY OA:LOGON.COM 31A:WARMSTRT.AUT
OA}COPY OA:LOGON.CMD 31A:16STRT.AUT
```

Now the LOGON command is set up to look for a file called USERID.SYS. USERID.SYS is a text file created by the integrator to contain the user passwords with a word processor such as WordStar. The format is as follows:

[USER ID],[PASSWORD],[USER # (and P for Privileged)],[DRIVE],

The commas are mandatory !

Example:

```
OA}TYPE USERID.SYS

CHUCK,ROAST,OP,A
JOHN,SMITH,1P,B
BOB,RUIZ,26,A

OA}
```

In this USERID file, Chuck Roast will be logged onto user 0, and the drive on which it booted, in this case drive A. John Smith was logged onto 1B, and Bob Ruiz was logged onto 26A. Chuck Roast and John Smith are privileged users and can change User numbers any time they wish, but Bob Ruiz will be logged on User 26 and stuck there. Because a lot of users will be stuck on higher user levels, and the commands are located on user 0, all of the commands you want available to others must be set global:

NOTE: The backslash '\', allows mutiple commands to be put on a single line.

```
OA}SET *.COM;G\SET *.CMD;G
```

8.5. Batch Processing

Batch processing is a way to set aside a slave whose only purpose is to receive commands from other slaves and execute them on a First-In First-Out basis. This will enable you to send time consuming commands such as linking, compiling, or long copy routines out to that slave and go on to other jobs. The slave set up for batch cannot have a console attached to it and cannot receive commands which require console input in order to execute.

To set up a slave for batch processing, set up the slave suffix table in the CPZMASTR.PAR to know of only one batch slave. In this example, the batch is known as 'B' at 7F:

```
SSTCSA = "B       "     ; O/S suffix table for CPS-MX
         "ZZZZZZZZ"     ; O/S suffix table for CPS-MX
```

Make changes in the CPSSLAVE.PAR so that the slave will execute the command "BATCH.AUT" when it boots:

```
LDWARM = OFF     ; Enable Warm start autoload (change to OFF to enable)
;
WARMFN = 0,"BATCH    ","AUT"
```

You will now want to create a file called BATCH.AUT. The file that needs to be executed in this case will be a DO file. Fortunately Turbo-Dos provides a command which can accomplish this called AUTOLOAD. The file which we want to create into an autoload file will be DO BATCH.DO:

```
OA}AUTOLOAD DO BATCH.DO

Auto load file created

OA}RENAME AUTOLOAD.AUT BATCH.AUT

OA}SET BATCH.AUT;G
```

Put BATCH.AUT in user 0 and set it global since the batch slave will have to be set up as a privileged user. The file you set up as BATCH.DO will now have to be set up as a FIFO. We will set it up to suspend processing in case the FIFO is empty so that it won't waste the masters processing time by executing BATCH.AUT over and over again:

```
0A}FIFO BATCH.DO

FIFO file not found, creating new file
Enter FIFO type (Ram/Disk): D
Suspend processing on full/empty conditions? (Yes/No): Y
Enter maximum number of records (1-65535): 128
FIFO file created

0A}SET BATCH.DO;G
```

To test out, let the batch processor copy a file up to different drive using the BATCH command:

```
5A}BATCH COPY A: B:;N

5A}DIR B:<CR>
```

NOTICE : The COPY command that was sent to the Batch slavehad the option ";n" to insure that it wouldn't expect any console input from the local console.

To set up a 16-bit slave for Batch Processing, do the same as above except use the CPSSLV16.PAR and CMD commands.

8.5.1. Master Batching

In extreme cases, where small application programs have to access to bus I/O ports directly; batch processing may be implemented on the master processor. WARNING! If the application program requires direct console input, like the FORMAT program, the batching will hang. This can be compensated for by putting a terminal on the master. This may be accomplished by enabling the Warm Boot Function to execute BATCH.AUT on the master. Set up the BATCH.AUT and BATCH.DO in the same manner as it was for Batch on the slave. The main difference to be made is to enable CON96 on the CPZMASTR.PAR file instead of CONREM. This will ensure that no one will try to attach to the master and provides a facility to monitor the batch processing if needed during integration. Below is an example CPZMASTR GEN and PAR file for master batching:

```
; CPZMASTR.GEN
STDMASTR        ; Standard networking master
PATCH           ; Include PATCH Module
FASLOD          ; Use fast disk loader module
NETREQ          ; Network request module
MSGFMT          ; Message format module
CPMSUP          ; CP/M function support module
;CONREM         ; Use remote console module
;CON192         ; Null 19.2 Kbaud Console Driver
CON96           ; Null 9600 Baud Console driver
NITCPZ          ; CPZ-48000 hardware initialization
SPDCPZ          ; CPZ-48000 Serial and Parallel I/O
LSTCTS          ; TI810 CTS Driver (LSTDR@ assigned to LSTDRA)
;LSTPAR         ; Parallel Printer driver (CPI-100 centronics board)
RTCCPZ          ; CPZ-48000 real time clock driver
;MSTRCLK        ; Include ICM ccb board drvr to set TurboDOS date/time
DSKCPZ          ; CPZ-48000 floppy disk driver
DSKFMT8         ; Disk specification tables for 8-INCH diskettes
;MD131DRV       ; Monitor Dynamics Model #1013 with 1 drive installed
;MD132DRV       ; Monitor Dynamics Model #1013 with 2 drive installed
;TURBO          ; Turbo-Disk driver module
;TURDSK         ; Turbo-Disk definition module
MCDCPS          ; CPZ-48000 / CPS-MX Master Circuit Driver
```

```
;CPZMASTR.PAR
AUTUSR = 80              ; Auto Log-on to user one, privileged
SRHDRV = 01              ; Search A: for command files
;
NMBUFS = 0A              ; Default number of Disk Buffers (hex)
BUFSIZ = 03              ; Default disk buffer size (log2(size/128))
MEMRES = (0400)          ; Allow for Dynamic Memory Expansion of TurboDOS
;
PATCSA = 7F,7E,7D,7C     ; Status port table for CPS-MX (std default value
         7B,7A,79,78
         3F,3E,3D,3C     ; Status port table for CPS-16 (std default value
         3B,3A,39,38
;
SSTCSA = "        "      ; O/S suffix table for CPS-MX (std default values
         "ZZZZZZZZ"      ; O/S suffix table for CPS-16 (std default values
;
CONAST = 01,CONDRA       ; Console on port 1 of CPZ-48000
STOPBB = 44
;
PTRAST = 00,LSTDRA       ; List assignment table
STOPBA = 44
;
FLSRT  = 1               ; 6ms Floppy step rate (0=3ms,1=6ms,2=10ms,3=15ms
ATNCHR = "^S"            ; New attention character (Break Key)
PRTMOD = 0               ; Print mode (0 = direct, 1 = Spooled)
;
;LDWARM = OFF
;WARMFN = 0,"BATCH    ","AUT"
;
;MPAGE = 02              ; Use memory above slave address for Turbo-disk
;
;following defines disk drives
;
DSKAST = 000,DSKDRA,001,DSKDRA,002,DSKDRA,003,DSKDRA ; floppies
;
;the following defines floppies + Monitor Dynamics Hard Disk
;with bootup from 1st Monitor Dynamics hard disk.
;
;DSKAST = 001,DSKDRB,000,DSKDRA,001,DSKDRA,002,DSKDRA    ;MD + Floppies
;         000,DSKDRB             ;deselect for hard disk before power down
;
LDWARM = OFF ; Enable Warm Start Autoload(000 to disable)
;
WARMFN = 0,"BATCH    ","AUT"
```

9. I/O Port Address Assignments

The CPZ-186 uses certain internal I/O ports addresses. Below is a breakdown of these I/O ports by port function and their corresponding addresses in hex.

9.1. Serial Port A and B Assignments

SCC Port A Data Reg.	1FE Hex
SCC Port A Control Reg.	1FC Hex
SCC Port B Data Reg.	1FA Hex
SCC Port B Control Reg.	1F8 Hex

9.2. Floppy Disk Controller Assignment

FDC Command/Status Reg.	100 Hex
FDC Track Reg.	102 Hex
FDC Sector Reg.	104 Hex
FDC Data Reg.	106 Hex

9.3. Parallel Port A and B Assignment

CIO Port C	200 Hex
CIO Port B	202 Hex
CIO Port A	204 Hex
CIO Control Reg.	206 Hex

9.4. Interrupt Controller Assignments

Interrupt Data Reg.	2FA Hex
Interrupt Command Reg.	2F8 Hex

9.5. Control Registers

On-board Functions	308 Hex {write only}
FDC Wait Reg. (program data xfer use)	308 Hex {read only}

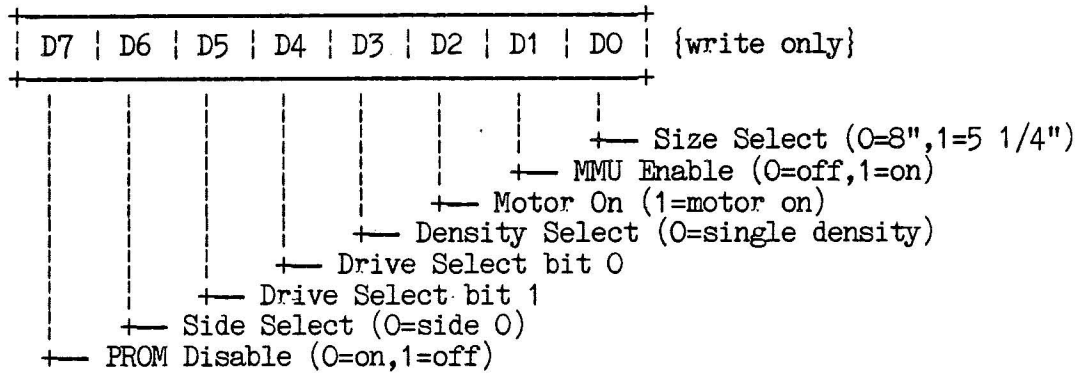
9.6. Memory Management Registers

MMU Address Reg 1	300 Hex
MMU Address Reg 2	302 Hex
MMU Address Reg 3	304 Hex
MMU Address Reg 4	306 Hex

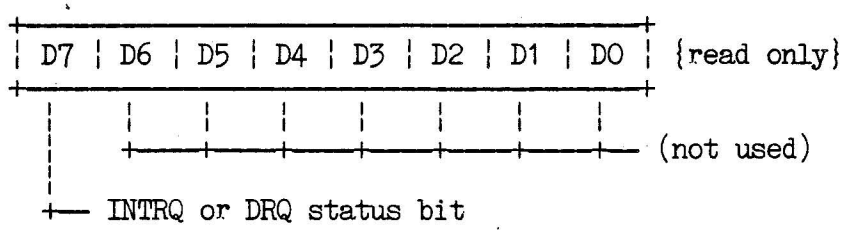
9.7. Control Register Bit Assignments

This is a description of the Control Registers and the corresponding bit assignments used on the CPZ-186.

9.7.1. On-board Functions Register (Port 308 Hex)



9.7.2. FDC Wait Register (Port 308 Hex)



9.7.3. S-100 Bus I/O Address Space

I/O addresses 0000-00F7 are allocated for slaves on the S-100 bus with the exception of addresses 0 - 8 which are reserved for on-board I/O devices.

10. WARRANTY

All products sold hereunder are under warranty on a return to factory basis against defects in workmanship and material for a period of one (1) year from the date of delivery.

Conditions of this warranty are as follows: Purchaser must 1) obtain a return material authorization (RMA) number and shipping instructions, 2) product must be shipped prepaid, 3) written description of the failure must be included with the defective product. All transportation charges inside the continental U.S. will be paid by Intercontinental Micro Systems (ICM) Corp. For products returned from all other locations, transportation must be prepaid. Should ICM determine that the products are not defective, the purchaser must pay all return transportation charges. All repairs will be provided at repair rates being charged at the time by ICM. Under the above product warranty, ICM may, at its option, either repair or replace any component which fails during the warranty period providing the purchaser has reported same in a prompt manner. All replaced products or parts shall become property of ICM.

All above warranties are contingent upon proper use of the product. These warranties will not apply 1) if any repair, parts replacement, or adjustments are necessary due to accident , unusual physical, electrical or electromagnetic stress, neglect, misuse, failure of electric power, air conditioning, humidity control, transportation, failure of rotating media not furnished by ICM, operation with media not meeting or not maintained in accordance with ICM specifications or causes other than ordinary use, 2) if the product has been modified by purchaser, 3) where ICM's serial numbers or warranty date decals have been removed or altered, 4) if the product has been dismantled by purchaser without the supervision of or prior written approval of ICM.

EXCEPT FOR THE EXPRESS WARRANTIES CONTAINED HEREIN, ICM DISCLAIMS ALL WARRANTIES ON THE PRODUCTS FURNISHED HEREUNDER, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS; and the stated express warranties are in lieu of all obligations or liabilities on the part of ICM arising out of or in connection with the performance of the products. ICM is not liable for any indirect or consequential damages.

After the warranty period, the products will be repaired for a service charge plus parts, provided that it is returned prepaid to ICM after retaining a return material authorization (RMA) number.

11. APPENDIX A

11.1. SPECIAL-CHIPS DATA SHEETS

Data sheets are included for the following chips used in the CPZ-186:

11.1.1. 8531 ASCC - Serial Controller
11.1.2. 8536 CIO - Parallel Port Controller
11.1.3. WD2793 - Floppy Disk Controller
11.1.4. 8259A - Interrupt Controller
11.1.5. 74LS670 - Memory Management Unit
11.1.6. 80186 CPU - 16-BIT Microprocessor

Chapter 1
General Information

1.0 INTRODUCTION

The Z8030 Serial Communications Controller (Z-SCC) and the nonmultiplexed Z8530 Serial Communications Controller (SCC) are Zilog Z8000™ peripheral components designed to provide multifunction support for handling the large variety of serial communication protocols available. The Z-SCC can be programmed to satisfy special serial communications requirements as well as to follow standard formats such as byte-oriented synchronous, bit-oriented synchronous, and asynchronous. Once programmed, the Z-SCC relieves the CPU of tasks formerly accomplished by the CPU or its associated hardware.

With access to 14 write registers and seven read registers per channel, the user can configure the Z-SCC so that it can handle all asynchronous formats regardless of data size, number of stop bits, or parity requirements. The Z-SCC also accomodates all synchronous formats including character, byte, and bit-oriented protocols.

Within each operating mode, the Z-SCC also allows for protocol variations by checking odd or even parity bits, character insertion or deletion, CRC generation and checking, break and abort generation and detection, and many other protocol-dependent features.

The Z-SCC is compatible with the Zilog Z8000 CPU or similar systems in which address and data lines are multiplexed. The SCC version may be used in microprocessor systems in which address lines and data lines are separate (such as in the Z80).

1.1 CAPABILITIES

Two independent full-duplex channels

Synchronous/Isosynchronous data rates:

- Up to 1 Megabit/second with 4 MHz clock rate
- Up to 250 Kbit/second with a 4 MHz clock rate (FM encoding)
- Up to 125 Kbit/second with a 4 MHz clock rate (NRZI encoding)

Receiver data registers quadruply buffered

Transmitter data registers double buffered

Asynchronous capabilities:

- 5, 6, 7, or 8 bits per character
- 1, 1-1/2, or 2 stop bits
- Odd or even parity
- Times 1, 16, 32, or 64 clock modes
- Break generation and detection
- Parity, overrun and framing error detection

Byte-oriented synchronous capabilities:

- Internal or external character synchronization
- 1 or 2 sync characters in separate registers
- Automatic sync character insertion and deletion
- Cyclic redundancy check (CRC) generation/detection
- 6- or 8-bit sync character

SDLC/HDLC capabilities:

- Abort sequence generation and checking
- Automatic zero insertion and deletion
- Automatic flag insertion between messages
- Address field recognition
- I-field residue handling
- CRC generation/detection
- SDLC loop mode with EOP recognition/loop entry and exit

NRZ, NRZI or FM encoding/decoding

Baud rate generator in each channel

Digital Phase-Locked Loop for clock recovery

Crystal Oscillator

1.2 BLOCK DIAGRAM

Figure 1-1 is a block diagram of the Z-SCC. Received data enters the receive data pins and follows one of several data paths, depending on the state of the control logic. The contents of the registers and the state of the external control pins establish the internal control logic. Transmitted data follows a similar pattern of control, register, and external pin definition.

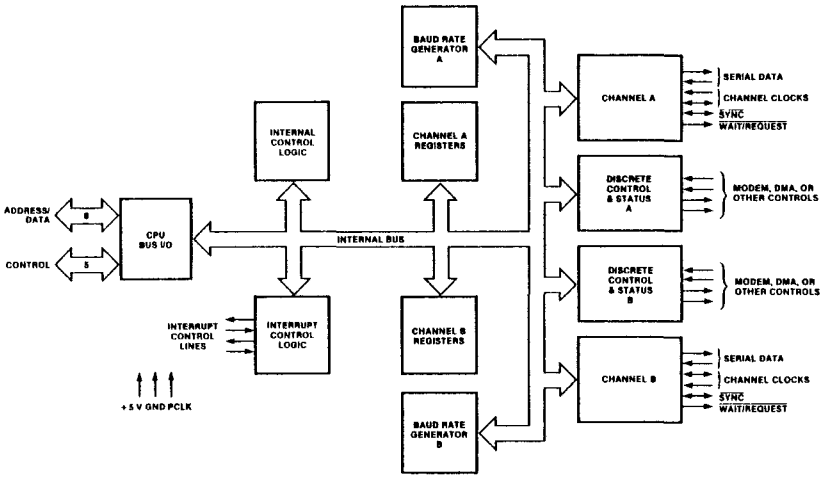


Figure 1-1. Z-SCC Block Diagram

The Z-BUS-compatible Z-SCC is suited for system applications with multiplexed address/data buses similar to Z8000 or Z8. The Z-SCC complies with the standards of the Z-BUS protocol established in the Z-BUS Technical Manual (TBS).

The nonmultiplexed SCC is designed for systems with separate address and data lines. It interfaces with any processor through Chip Select (CS) inputs, Read (RD) and Write (WR) inputs, Channel Select (A/B) input, and a Data/Control (D/C) input.

1.3 PIN FUNCTIONS

The Z-SCC pins are divided into seven functional groups: address/data, bus timing and reset, device control, interrupt, serial data (both channels), peripheral control (both channels), and clocks (both channels). Figure 1-2 shows the pins in each functional group for both the Z8030 and Z8530. Notice the pin notations unique to each version in the address/data group, bus timing and reset group, and control groups.

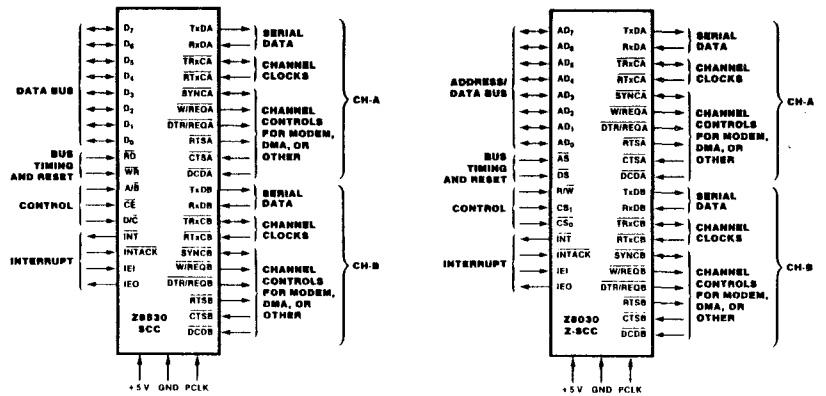


Figure 1-2. Pin Functions SCC/Z-SCC

2016-040 2016-041 2023-001

The address/data group consists of the bidirectional lines used to transfer data between the CPU and the Z-SCC. (Addresses in the Z8030 are latched by $\overline{AS}$.) The direction of these lines depends on whether the Z-SCC is selected and whether the operation is a read or a write.

The timing and control groups designate the type of transaction to occur and when this transaction will occur. The interrupt group provides inputs and outputs to conform to the Z-BUS specifications for handling and prioritizing interrupts. The remaining groups are divided into Channel A and Channel B data groups for serial data (transmit or receive), peripheral control (such as DMA or modem), and the input and output lines for the receive and transmit clocks.

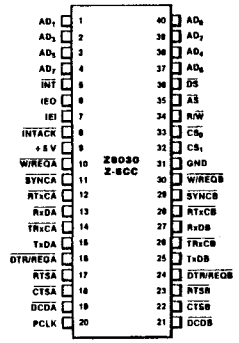


Figure 1-3. Pin Designation for Z8030 Z-SCC

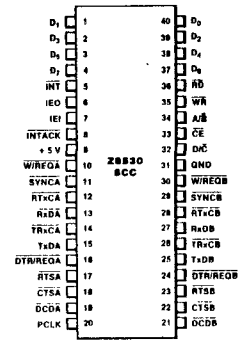


Figure 1-4. Pin Designation for Z8530 SCC

1.4 PIN DESCRIPTION

The Z-SCC is available with two sets of bus interface timings: one for multiplexed systems and one for nonmultiplexed systems.

The pin descriptions here describe the Z8030-specific pins, the Z8530-specific pins, and the pins that are identical for both versions. Figure 1-3 designates the pin locations and signal names for the Z8030 Z-SCC. Figure 1-4 designates the pin locations and signal names for the Z8530 SCC version.

PIN DESCRIPTIONS (Z8030 Z-SCC only)

AD_0-AD_7. Address/Data Bus (bidirectional, active High, 3-state). These multiplexed lines carry register addresses to the Z-SCC as well as data or control information to and from the Z-SCC.

$\overline{AS}$. Address Strobe (input, active Low). Addresses on AD_0-AD_7 are latched by the rising edge of this signal.

$\overline{CS_0}$. Chip Select 0 (input, active Low). This signal is latched concurrently with the addresses on AD_0-AD_7 and must be active for the intended bus transaction to occur.

CS_1. Chip Select 1 (input, active High). This second select signal must also be active before the intended bus transaction can occur. CS_1 must remain active throughout the transaction.

$\overline{DS}$. Data Strobe (input, active Low). This signal provides timing for the transfer of data into and out of the Z-SCC. If $\overline{AS}$ and $\overline{DS}$ are both Low, this is interpreted as a reset.

R/$\overline{W}$. Read/Write (input). This signal specifies whether the operation to be performed is a read or a write.

PIN DESCRIPTIONS (Z8530 SCC Only)

A/$\overline{B}$. Channel A/Channel B Select (input, Channel A active High). This signal selects the channel in which the read or write operation occurs.

$\overline{CE}$. Chip Enable (input, active Low). This signal selects the SCC for operation. It must remain active throughout the bus transaction.

D_0-D_7. Data Lines (bidirectional, 3-state). These I/O lines carry data or control information to and from the SCC.

D/C. Data/Control (input, data active High). This signal defines the type of information transfer performed by the SCC: data or control.

RD. Read (input, active Low). This signal indicates a read operation and when the SCC is selected, enables the SCC bus drivers. During the interrupt acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

WR. Write (input, active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of RD and WR is interpreted as a reset.

PIN DESCRIPTIONS (Both versions)

CTSA, CTSB. Clear to Send (inputs, active Low). If these pins are programmed as auto enables, a Low on these inputs enables the respective transmitters. If not programmed as auto enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The Z-SCC detects transitions on these inputs and can interrupt the CPU on both logic level transitions.

DCDA, DCDB. Data Carrier Detect (inputs, active Low). These pins function as receiver enables if they are programmed as auto enable bits; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The Z-SCC detects transitions on these pins and can interrupt the CPU on both logic level transitions.

DTR/REQA, DTR/REQB. Data Terminal Ready/Request (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as request lines for a DMA controller.

IEI. Interrupt Enable In (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High on IEI indicates that no other higher priority device has an Interrupt Under Service (IUS) or is requesting an interrupt.

IEO. Interrupt Enable Out (output, active High). IEO is High only if IEI is High and the CPU is not servicing an SCC or Z-SCC interrupt or the controller is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

INTACK. Interrupt Acknowledge (input, active Low). This signal indicates an active interrupt acknowledge cycle. During this cycle, the interrupt daisy chain settles. When RD or DS becomes active, the Z-SCC places an interrupt vector on the data bus (if IEI is High). INTACK is latched by the rising edge of AS or PCLK.

INT. Interrupt Request (output, open-drain, active Low). This signal is activated when the Z-SCC is requesting an interrupt.

PCLK. Clock (input). This is the master clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock, although the frequency of this clock must be at least 90 percent of the CPU clock frequency for a Z8000. PCLK is a TTL level signal.

RTSA, RTSB. Request To Send (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 3-7) is set, the RTS signal goes Low. When the RTS bit is reset in the Asynchronous mode and auto enable is on, the signal goes High after the transmitter is empty. In Synchronous mode or in Asynchronous mode with auto enable off, the RTS pins strictly follow the state of the RTS bit. Both pins can be used as general-purpose outputs.

RTxCA, RTxCB. Receive/Transmit Clocks (inputs, active Low). The functions of these pins are under program control. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop (refer to Section 4 for bit configurations). These pins can also be programmed for use with the respective SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

RxDA, RxDB. Receive Data (inputs, active High). These input signals receive serial data at standard TTL levels.

SYNCA, SYNCB. Synchronization (inputs/outputs, active Low). These pins can act as either inputs, outputs, or as part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to CTS and DCD. In this mode, transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 (Figure 4-18) but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, SYNC must be driven Low two receive clock cycles after the last bit in the sync character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of SYNC.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which sync characters are recognized. The sync condition is not latched, so these outputs are active each time a sync character is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

TRxCA, TRxCB. Transmit/Receive Clocks (inputs or outputs, active Low). The functions of these pins are under program control. TRxC may supply the receive clock or the transmit clock in the Input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode. (Refer to Section 4 for bit configuration.)

TxDA, TxDB. Transmit Data (outputs, active High). This output signal transmits serial data at standard TTL levels.

W/REQA, W/REQB. Wait/Request (outputs, open-drain when programmed for Wait function, driven High or Low when programmed for a Request function). These dual-purpose outputs can be programmed as Request lines for a DMA controller or as Wait lines to synchronize the CPU to the Z-SCC data rate. The reset state is Wait.

Chapter 2
Architecture

2.0 INTRODUCTION

The Z-SCC internal structure provides all the interrupt and control logic necessary to interface with multiplexed and nonmultiplexed buses. Interface logic is also provided to monitor modem or peripheral control inputs and outputs. All of the control signals are general purpose and can be applied to various peripheral devices as well as used for modem control.

The center for data activity revolves around the internal read and write registers. The programming of these registers provides the Z-SCC with a functional "personality", i.e., register values can be assigned before or during program sequencing to determine how the Z-SCC will establish a given communication protocol.

2.1 REGISTER FUNCTIONS

All modes of communication are established by the bit values of the write registers. As data is received or transmitted, read register values may change. These changed values can promote software action or internal hardware action for further register changes.

The register set for each channel includes 14 write registers and seven read registers. Ten write registers are used for control, two for sync character generation, and two for baud rate generation. The remaining two write registers are shared by both channels; one is used as the interrupt vector and one as the master interrupt control. Four read registers indicate status functions; two are used by the baud rate generator, one for the interrupt vector, one for the receiver buffer, and one for reading the interrupt pending bits.

Table 2-1 lists the assigned functions for each read and write register. The Z-SCC contains only one WR2 (interrupt vector) and one WR9 (master interrupt control). Both registers are accessed and shared by either channel. Chapter 4 provides a detailed bit legend and description of each register.

Table 2-1. Register Set

	READ REGISTER FUNCTION		WRITE REGISTER FUNCTION
RR0	Transmit/Receive buffer status, and External status	WR0	Command Register, (Register Pointers, Z8530 only), CRC initialization, resets for various modes
RR1	Special Receive Condition status, residue codes, error conditions	WR1	Interrupt conditions, Wait/DMA request control
RR2O	Modified (Channel B only) interrupt vector and Unmodified interrupt vector (Channel A only)	WR2	Interrupt vector (access through either channel)
RR3	Interrupt Pending bits (Channel A only)	WR3	Receive/Control parameters, number of bits per character, Rx CRC enable
		WR4	Transmit/Receive miscellaneous parameters and modes, clock rate, number of sync characters, stop bits, parity
		WR5	Transmit parameters and controls, number of Tx bits per character, Tx CRC enable
		WR6	Sync character or SDLC address field (1st byte)
		WR7	Sync character or SDLC flag (2nd byte)
RR8	Receive buffer	WR8	Transmit buffer
		WR9	Master interrupt control and reset (accessed through either channel), reset bits, control interrupt daisy chain
RR10	Miscellaneous XMTR, RCVR status parameters	WR10	Miscellaneous transmitter/receiver control bits, NR2I, NR2, FM encoding, CRC reset
		WR11	Clock mode control, source of Rx and Tx clocks
RR12	Lower byte of baud rate generator time constant	WR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant	WR13	Upper byte of baud rate generator time constant
		WR14	Miscellaneous control bits: baud rate generator, Phase-Locked Loop control, auto echo, local loopback
RR15	External/Status interrupt control information	WR15	External/Status interrupt control information-control external conditions causing interrupts

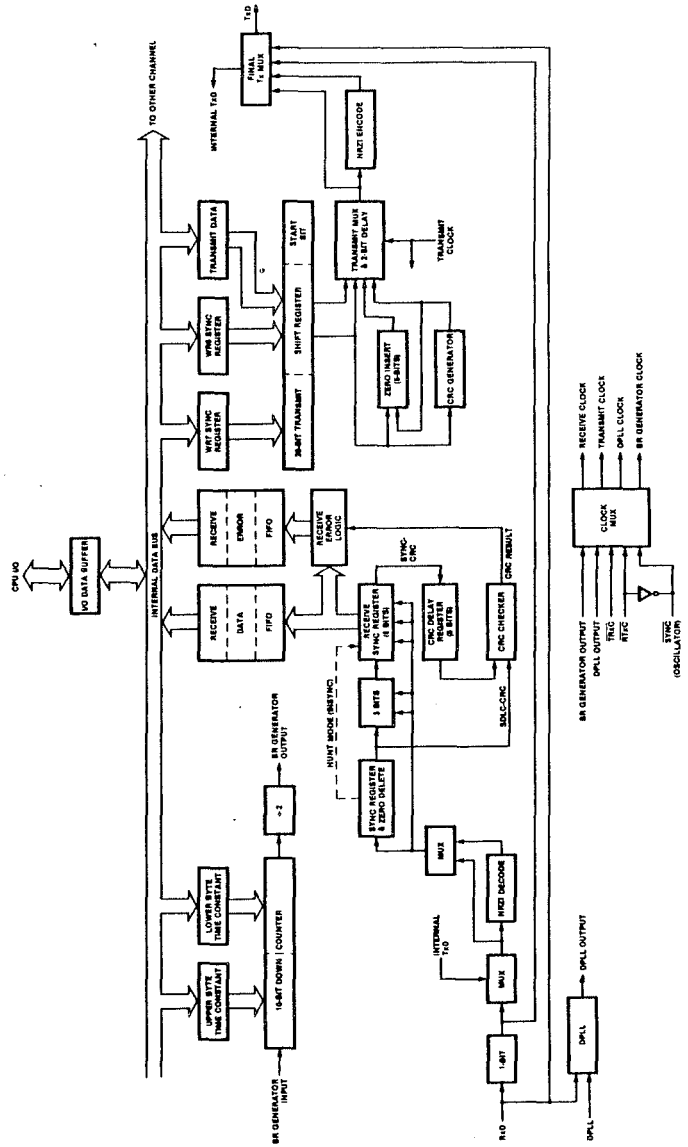


Figure 2-1. Data Paths

2.2 DATA PATHS

Figure 2.1 illustrates the data paths involved in the six major areas of the SCC:

- Receiver
- Transmitter
- Baud rate generator
- DPLL
- Clocking options
- Data encoding

All communication modes are established by programming the write registers. As data is received or transmitted, read register values may change, altering the direction of the data path. These changed values can promote software action or internal hardware action for further register changes.

2.2.1 Transmitter

The transmitter has an 8-bit Transmit Data register (WR8) loaded from the internal data bus and a Transmit Shift register loaded from either WR6, WR7, or the Transmit Data register. In byte-oriented modes, WR6 and WR7 can be programmed with sync characters. In Monosync mode, an 8-bit or 6-bit sync character is used (WR6), whereas a 16-bit sync character is used (WR6 and WR7) in Bisync mode. In bit-oriented synchronous modes, the flag contained in WR7 is loaded into the Transmit Shift register at the beginning and end of a message.

If asynchronous data is processed, WR6 and WR7 are not used and the Transmit Shift register is formatted with start and stop bits shifted out to the transmit multiplexer at the selected clock rate. Synchronous data (except SDLC/HDLC) is shifted to the CRC generator as well as to the transmit multiplexer at the X1 clock rate.

SDLC/HDLC data is shifted out through the zero insertion logic (which is disabled while the flags are being sent). A 0 is inserted in all address, control, information, and frame check fields following five contiguous 1s in the data stream. The result of the CRC generator for SDLC data is also routed through the zero insertion logic.

2.2.2. Receiver

The receiver has three 8-bit FIFO buffer registers and an 8-bit shift register. This arrangement creates a 3-byte delay time, which allows the CPU time to service an interrupt at the beginning of a block of high-speed data. With each receive FIFO, an error FIFO is provided to store parity and framing errors and other types of status information.

Incoming data is routed through one of several paths depending on the mode and character length. In Asynchronous mode, serial data enters the 3-bit delay if a character length of seven or eight bits is selected. If a character length of five or six bits is selected, data enters the receive shift register directly.

In synchronous modes, the data path is determined by the phase of the receive process currently in operation. A synchronous receive operation begins with a hunt phase in which a bit pattern that matches the programmed sync characters (6-, 8-, or 16-bit) is searched.

The incoming data then passes through the Sync register and is compared to a sync character stored in WR6 or WR7 (depending on which mode it is in). The Monosync mode matches the sync character programmed in WR7 and the character assembled in the Receive Sync register to establish synchronization.

Synchronization is achieved differently in the Bisync mode. Incoming data is shifted to the Receive Shift register while the next eight bits of the message are assembled in the Receive Sync register. If these two characters match the programmed characters in WR6 and WR7, synchronization is established. Incoming data can then bypass the Receive Sync register and enter the 3-bit delay directly.

The SDLC mode of operation uses the Receive Sync register to monitor the receive data stream and to perform zero deletion when necessary; i.e., when five continuous 1s are received, the sixth bit is inspected and deleted from the data stream if it is 0. The seventh bit is inspected only if the sixth bit equals one. If the seventh bit is 0, a flag sequence has been received and the receiver is synchronized to that flag. If the seventh bit is a 1, an abort or an EOP (end of poll) is recognized, depending on the selection of either the normal SDLC mode or SDLC Loop mode.

The same path is taken by incoming data for both SDLC modes. The reformatted data enters the 3-bit delay and is transferred to the Receive Shift register. The SDLC receive operation begins in the hunt phase by attempting to match the assembled character in the Receive Shift register with the flag pattern in WR7. When the flag character is

recognized, subsequent data is routed through the same path, regardless of character length.

Either the CRC-16 or CRC-SDLC cyclic redundancy check (CRC) polynomial can be used for both Monosync and Bisync modes, but only the CRC-SDLC polynomial is used for SDLC operation. The data path taken for each mode is also different. Bisync protocol is a byte-oriented operation that requires the CPU to decide whether or not a data character is to be included in CRC calculation. An 8-bit delay in all synchronous modes except SDLC is allowed for this process. In SDLC mode, all bytes are included in the CRC calculation.

2.3 DATA COMMUNICATIONS CAPABILITIES

Z-SCC logic handles all asynchronous, byte-oriented synchronous, and bit-oriented synchronous modes of operation. The following section briefly describes asynchronous, synchronous, and SDLC modes of communication.

2.3.1 Asynchronous

Figure 2-2 represents a typical asynchronous message format using one start bit, seven data bits, one parity bit, and one stop bit. A start bit is a High-to-Low transition detected by an asynchronous receiver and is actually an information bit notifying the receiver of an incoming message.

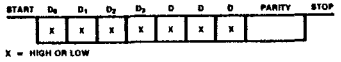


Figure 2-2. Asynchronous Message Format

The start bit also initiates a clock circuit to provide latching pulses during expected data bit intervals. The parity bit is provided for error checking and reats in a resultant state (odd or even) depending on an accumulated 1's state count of the data bits. The parity bit is calculated in both the receiver and the transmitter; the two results are compared to ensure that the expected and the actual bit values match.

The stop bit returns the message unit to the quiescent marking state; i.e., a constant high state condition lasts until the next High-to-Low start bit indicates an incoming data byte. During reception, the start and stop bits are stripped away and checked for errors, leaving only the working data for CPU interaction. The number of selected bits for each asynchronous function may vary in the transmitter and the receiver.

2.3.2 Monosync Mode

Monosync and Bisync modes require clocking information to be transmitted along with the data by a method of encoding data that contains clocking information, or by a modem to encode or decode clock information in the modulation process.

Start and stop bits are not required in synchronous modes. All bits are used to transmit data, which eliminates the "waste" characteristic of asynchronous communication.

Figure 2-3 shows the character format for synchronous transmission. For example, bits 1-8 might be one character and bits 9-13 part of another character; or bit 1 might be part of one character, bits 2-9 part of a second character, and bits 10-13 part of a third character. The framing (where each character begins) of each character is accomplished by defining a synchronization character, commonly called a "sync character."

The CPU places the receiver in Hunt mode whenever transmission begins (or whenever a data dropout has occurred and the hardware determines that resynchronization is necessary). In Hunt mode, the receiver shifts a bit into the Receive Shift register and compares the contents of the Receive Shift register with the sync character (stored in another register), repeating the process until a

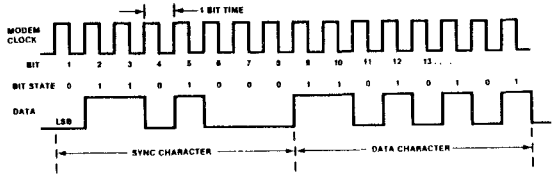


Figure 2-3. Monosync Data Character Format

match occurs. When a match occurs, the receiver begins transferring bytes to the receive FIFO.

2.3.3 Bisynchronous Mode

The Bisync mode of operation (Figure 2-4) is similar to the Monosync mode, except that two sync characters are provided instead of one. Bisync attempts a more structured approach to synchronization through the use of special characters as message "headers" or "trailers" (refer to IBM's Bisync).

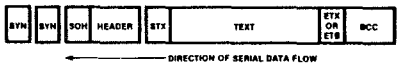


Figure 2-4. Bisynchronous Message Format

2.3.4 External Sync Mode

External Sync mode (Figure 2-5) eliminates the use of sync characters in the serial data stream by providing an external sync signal to mark the beginning of a data field; i.e., an external input pin (Sync) waits for an active state change to indicate the ensuing information field.

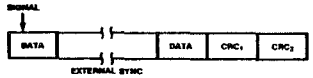


Figure 2-5. External Sync Format

2.3.5 SDLC Mode

Synchronous Data Link Control mode (SDLC) uses synchronization characters similar to Bisync and Monosync modes (such as flags and pad characters), but it is a bit-oriented protocol instead of byte-oriented protocol.

Any data communication link involves at least two stations. The station that is responsible for the data link and issues the commands to control that link is called the "primary station." The other station is a "secondary station." Not all information transfers need to be initiated by a primary station. In SDLC mode, a secondary station can be the initiator.

The basic format for SDLC is a "frame" (Figure 2-6). The information field is not restricted in format or content and can be of any reasonable length (including zero). Its maximum length is that which can be expected to arrive at the receiver error-free most of the time. Hence, the determination of maximum length is a function of communication channel error rate.

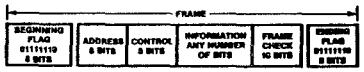


Figure 2-6. SDLC Message Format

2.3.6 SDLC Loop Mode

The Z-SCC supports SDLC Loop mode in addition to normal SDLC. SDLC Loop mode is very similar to normal SDLC but is usually used in applications where a point-to-point network is not appropriate (for example, POS terminals). In an SDLC Loop there is a primary station, called the controller, that manages the message traffic flow on the loop and there are any number of secondary stations.

A secondary station in an SDLC loop is always listening to the messages being sent around the loop, and must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can only place its own message on the loop at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages this bit pattern is unique and thus is easily recognized.

When a secondary station has a message to transmit and it recognizes an EOP on the line, the first thing that it does is to change the last 1 of the EOP to a 0 before transmitting it. This has the effect of turning the EOP into a Flag sequence. The secondary station now places its message on the loop and terminates its message with an EOP. Any secondary stations further down the loop with messages to transmit can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop, except on recognizing an EOP.

There are also restrictions as to when and how a secondary station physically becomes part of the loop. A secondary station that has just powered up must monitor the loop, without the one-bit-time delay, until it recognizes an EOP. When an EOP is recognized the one-bit-time delay is switched on. This does not disturb the loop because the line is marking idle between the time that the controller sends the EOP and the time that it receives the

2057 004 2057 003 2042-108

EOP back. The secondary station that has gone on-loop cannot place a message on the loop until the next time that an EOP is issued by the controller. A secondary station goes off-loop in a similar manner. When given a command to go off-loop, the secondary station waits until the next EOP to remove the one-bit-time delay.

To operate the Z-SCC in SDLC Loop mode, the Z-SCC must first be programmed just as if normal SDLC were to be used. The baud rate generators should be set up and the Loop mode selected by writing the appropriate control word in WR11. Since WR11 also controls the clocking mode for the receiver and the transmitter, it may be useful to write the final clocking values in WR11 without selecting the SDLC Loop mode before any of the other registers are written. This allows faster startup by allowing the clocks to settle, but is not strictly necessary. If NRZI encoding and the DPLL are being used, these options should be selected before selecting SDLC Loop mode for the same reasons. The Z-SCC is now waiting for the EOP so that it can go on loop. While waiting for the EOP, the Z-SCC ties TxD to RxD with only the internal gate delays in the signal path. When the first EOP is recognized by the Z-SCC, the Break/Abort/EOP bit is set in RR0, generating an External/Status interrupt (if so enabled). At the same time, the On-Loop bit in RR10 is set to indicate that the Z-SCC is indeed on-loop, and a one-bit time delay is inserted in the TxD to the RxD path.

The Z-SCC is now on-loop but cannot transmit a message until a flag and the next EOP are received. The requirement that a flag be received ensures that the Z-SCC cannot erroneously send messages until the controller ends the current polling sequence and starts another one. If SDLC mode is deselected before this flag is received, the Break/Abort/EOP bit resets, generating another External/Status interrupt, and the Z-SCC transmits messages in response to an EOP being received.

A secondary station on the loop is prohibited from transmitting a message during a polling sequence unless it captures the line at the moment the EOP passes by. The Z-SCC does this automatically. If the CPU in the secondary station with Z-SCC needs to transmit a message, the Go-Active-On-Poll bit in WR10 must be set. If this bit is set when the EOP is detected, the Z-SCC changes the EOP to a flag and starts sending another flag. The EOP is reported in the Break/Abort/EOP bit in RR0 and the CPU should write its data bytes to the Z-SCC, just as in normal SDLC frame transmission. When the frame is complete and CRC has been sent, the Z-SCC closes with a flag and reverts to One-Bit-Delay mode. The last zero of the flag, along with the marking line echoed from the RxD pin, form an EOP for secondary stations further down the loop. If the Go-Active-On-Poll bit is not set at the time the EOP passes by, the Z-SCC cannot send a message until a flag (terminating the current polling sequence) and another EOP are received. While the Z-SCC is actually transmitting a message, the Loop-Sending bit in R10 is set to indicate this.

If SDLC loop is deselected, the Z-SCC is designed to exit from the loop gracefully. When SDLC Loop mode is deselected by writing to WR10, the Z-SCC waits until the next polling cycle to remove the one-bit time delay. If a polling cycle is in progress at the time the command is written, the Z-SCC finishes sending any message that it may be transmitting, ends with an EOP, and disconnects TxD from RxD. If no message was in progress, the Z-SCC immediately disconnects TxD from RxD. If a polling cycle is not in progress at the time the command is given, the Z-SCC waits until an EOP is recognized to disconnect TxD from Rxd. To ensure proper loop operation after the Z-SCC goes off the loop, and until the external relays take the Z-SCC completely out of the loop, the Z-SCC should be programmed for Mark idle instead of Flag idle. When the Z-SCC goes off the loop, the On-Loop bit is reset.

The Z-SCC allows the user the option of using NRZI in SDLC Loop mode by programming WR10 appropriately. With NRZI encoding, the outputs of secondary stations in the loop may be inverted from their inputs because of messages that they have transmitted. Removing the stations from the loop (removing the one-bit time delay) may cause problems further down the loop because of extraneous transitions on the line. The Z-SCC removes this problem by making transparent adjustments at the end of each frame it sends in response to an EOP. A response frame from the Z-SCC is terminated by a flag and an EOP. Normally, the flag and the EOP share a zero, but if such sharing would cause the RxD and TxD pins to be of opposite polarity after the EOP, the Z-SCC adds another zero between the flag and the EOP. This causes an extra line transition so that RxD and Txd are identical after the EOP is sent. This extra zero is completely transparent because it only means that the flag and the EOP no longer share a zero. All that a proper loop exit needs, therefore, is the removal of the one-bit time delay.

The two flags that delineate the SDLC frame serve as reference points when positioning the address and control fields, and they initiate the trans-

mission error check. The ending flag indicates to the receiving station that the 16 bits just received constitute the frame check. The ending flag could be followed by another frame, another flag, or an idle. This means that when two frames follow one another, the intervening flag may simultaneously be the ending flag of the first frame and the beginning flag of the next frame. Since the SDLC mode does not use characters of defined length, but rather works on a bit-by-bit basis, the 01111110 flag can be recognized at any time.

To ensure that the flag is not sent accidently, SDLC procedures require a binary 0 to be inserted by the transmitter after the transmission of five contiguous 1s. The receiver then removes the 0 following a received succession of five 1s. Inserted and removed 0s are not included in the CRC calculation.

The address field is eight bits long and designates the number of the secondary station to which the commands or data from the primary station are sent. The control field is eight bits long and is used to initiate all SDLC activities (see Section 3.6).

The Z-SCC can also serve the high-level synchronous data link communication (HDLC) protocol, which is identical to SDLC except for differences in framing.

2.4 I/O CAPABILITIES

The Z-SCC can work with three basic forms of I/O operations: polling, interrupts, and block transfer. All three I/O types involve register manipulation during initialization and data transfer. However, the Interrupt mode also incorporates Z-BUS™ interrupt protocol for a faster and more efficient data transfer.

2.4.1 Polling

During a polling sequence, the status of Read Register 0 is examined in each channel. This register indicates whether or not a receive or transmit data transfer is needed and whether or not any special conditions are present, e.g., errors.

This method of I/O transfer avoids interrupts. All interrupt functions must be disabled in order to operate the device in a polled environment. With no interrupts enabled, this mode of operation must initiate a read cycle of Read Register 0 to detect an incoming character before jumping to a data handler routine.

2.4.2 Interrupts

The Z-SCC provides interrupt capability through the use of pins and a hardware scheme that increases the transfer speed of serial data. Whenever the interrupt ($\overline{INT}$) pin is active, the Z-SCC is ready to transfer data.

Read and write registers are programmed so that an interrupt vector points to an interrupt service routine. The interrupt vector can also be modified to reflect various status conditions. Therefore, as many as eight different interrupt routines can be referenced.

Transmit interrupts, receive interrupts, and external/status interrupts are the main sources of interrupts. Each interrupt source is enabled under program control, with channel A having a higher priority than channel B and with receiver, transmit, and external/status interrupts prioritized respectively within each channel. (Section 3.2.1 provides a detailed description of the interrupt scheme and the various interrupt types.)

2.4.3 Block Transfer

The Z-SCC provides a Block Transfer mode to accomodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{W/REQ}$ output in conjunction with the Wait/Request bits in Write Register 1. The $\overline{W/REQ}$ output can be defined by software as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the Z-SCC $\overline{REQUEST}$ output indicates that the Z-SCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ output indicates that the Z-SCC is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle. (Section 3.2.3 describes the registers used in block transfers.)

2.5 SUPPORT CIRCUITRY

The Z-SCC incorporates additional circuitry to aid serial communications. The designer can select an internal baud rate generator, select the frequency, and program the output to one of several circuits contained within the Z-SCC. The Z-SCC can be programmed to encode and decode in several standard formats. In addition, various clocking options can be selected for the DPLL, the baud rate generator, the receiver and transmitter.

2.5.1 Baud Rate Generator

Each channel in the Z-SCC contains a programmable baud rate generator. Each generator consists of two 8-bit, time-constant registers forming a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output that makes the output a square wave. On startup, the flip-flop on the output is set High so that it starts in a known state, the value in the time-constant register is loaded into the counter, and the counter begins counting down. When a count of zero is reached, the output of the baud rate generator toggles, the value in the time-constant register is loaded into the counter, and the process starts over. The time constant can be changed at any time, but the new value does not take effect until the next load of the counter.

No attempt is made to synchronize the loading of a new time constant with the clock used to drive the generator. When the time constant is to be changed, the generator is stopped by writing to an enable bit in WR14. This ensures the loading of a correct time constant.

If neither the transmit clock nor the receive clock are programmed to come from the TRXC pin, the output of the baud rate generator may be made available for external use on the TRXC pin. Section 3.3.1 presents the formula for determining the time constant for a given rate.

2.5.2 Digital Phase-Locked Loop (DPLL)

The Z-SCC contains a Digital Phase-Locked Loop that can be used to recover clock information from a data stream with NRZI or FM coding. The DPLL is driven by a clock nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a receive clock for the data. This clock can then be used as the Z-SCC receive clock, the transmit clock, or both. Section 3.3.1 details the clock recovery for each of the different forms of encoding.

2.5.3 Clocking Options

The Z-SCC can select several clock sources for internal and external use. Write Register 11 is the Clock Mode Control register for both the receive and transmit clocks. It determines the type of signal on the SYNC and $\overline{RTxC}$ pins and the direction of the $\overline{TRxC}$ pin.

Write Register 11 also controls the output of the baud rate generator, the DPLL output, and the selection of either a TTL or an XTAL output for the $\overline{RTxC}$ pin. (Section 3.3.4 gives a detailed description of the clocking options.)

2.5.4 Data Encoding

Figure 2-7 illustrates the four encoding methods used by the Z-SCC. In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. In FM1 (more properly, biphase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by the absence of a transition at the center of the bit cell. In FM0 (more properly, biphase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell and a 1 is represented by the absence of a transition at the center of the bit cell.

In addition to these four methods, the Z-SCC can be used to decode Manchester (biphase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is Low to High, the bit is 0. If the transition is High to Low, the bit is 1.

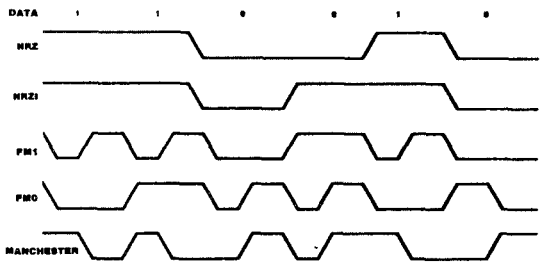


Figure 2-7. Data Encoding Methods

Chapter 3
Programming the SCC

3.0 INTRODUCTION

Functional Description

There are two versions of the SCC: the multiplexed Z8030 Z-SCC uses the Address/Data bus of a multiplexed system for data reception and transmission (e.g., Zilog's Z-BUS), whereas the non-multiplexed Z8530 SCC uses the bus dedicated to data in systems where data and address information are carried on separate buses.

Both the Z8030 Z-SCC and Z8530 SCC versions of the SCC contain read registers and write registers in each channel. There are 14 write registers, including Write Register 8, the transmit buffer; and 9 read registers, including Read Register 8, the receive buffer, in each channel. When reading the SCC, the address bits are not fully decoded, therefore the read data will appear at more than one address. This chapter provides information necessary to program both versions of the SCC for each mode of operation as well as describing the

Table 3-1. SCC Register Description

READ REGISTER	DESCRIPTION
RR0	Transmit/Receive Buffer Status and External Status
RR1	Receive Condition Status/Residue Codes
RR2	Interrupt Vector (modified in B Channel)
RR3	Interrupt Pending (Channel A only)
RR8	Receive Buffer
RR10	Loop/Clock Status
RR12	Lower Byte of Time Constant
RR13	Upper Byte of Time Constant
RR15	External Status Interrupt Enable

WRITE REGISTER	DESCRIPTION
WR0	Command Register
WR1	Transmit/Receive Interrupt and Data Transfer Mode Definition
WR2	Interrup Vector
WR3	Receive Parameters and Control
WR4	Transmit/Receive Miscellaneous Parameters and Modes
WR5	Transmit Parameter and Controls
WR6	Sync Character or SDLC Address Field
WR7	Sync Character or SDLC Flag
WR8	Transmit Buffer
WR9	Master Interrupt Control
WR10	Miscellaneous Transmitter/Receiver Control Bits
WR11	Clock Mode Control
WR12	Lower Byte of Baud Rate Generator Time Constant
WR13	Upper Byte of Baud Rate Generator Time Constant
WR14	Miscellaneous Control Bits
WR15	External Status/Interrupt Control

SCC approach to establishing each mode. In some cases, methods for initialization and data transfer are suggested and programming examples presented. These are meant to serve as aids in introducing detailed information and are not to be considered the only approach to programming the SCC.

The write registers for each channel are programmed separately to configure the channel functions. In addition to the 14 write registers there are two registers, Write Register 2 and Write Register 9, shared by the two channels and accessible through either of them. Write Register 2 contains the interrupt vector and Write Register 9 contains the control bits for both channels.

Read Registers 0, 1, 10, and 15 may be read to obtain status information. Read Registers 12 and 13 may be read to obtain the baud rate generator time constant. Read Register 2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). Read Register 3 contains the Interrupt Pending (IP) bits (Channel A).

Table 3-2. Z8030 Z-SCC Register Selection

AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	WRITE	READ	COMMENT
X	X	0	0	0	0	0	X	WR0B	RR0B	
X	X	0	0	0	0	1	X	WR1B	RR1B	
X	X	0	0	0	1	0	X	WR2	RR2B	
X	X	0	0	0	1	1	X	WR3B	RR3B	
X	X	0	0	1	0	0	X	WR4B	(RR0B)	
X	X	0	0	1	0	1	X	WR5B	(RR1B)	
X	X	0	0	1	1	0	X	WR6B	(RR2B)	
X	X	0	0	1	1	1	X	WR7B	(RR3B)	
X	X	0	1	0	0	0	X	WR8B	RR8B	B Data
X	X	0	1	0	0	1	X	WR9		Shared
X	X	0	1	0	1	0	X	WR10B	RR10B	
X	X	0	1	0	1	1	X	WR11B	(RR15B)	
X	X	0	1	1	0	0	X	WR12B	RR12B	
X	X	0	1	1	0	1	X	WR13B	RR13B	
X	X	0	1	1	1	0	X	WR14B	(RR10B)	
X	X	0	1	1	1	1	X	WR15B	RR15B	
X	X	1	0	0	0	0	X	WR0A	RR0A	
X	X	1	0	0	0	1	X	WR1A	RR1A	
X	X	1	0	0	1	0	X	WR2	RR2A	
X	X	1	0	0	1	1	X	WR3A	RR3A	
X	X	1	0	1	0	0	X	WR4A	(RR0A)	
X	X	1	0	1	0	1	X	WR5A	(RR1A)	
X	X	1	0	1	1	0	X	WR6A	(RR2A)	
X	X	1	0	1	1	1	X	WR7A	(RR3A)	
X	X	1	1	0	0	0	X	WR8A	RR8A	A Data
X	X	1	1	0	0	1	X	WR9		Shared
X	X	1	1	0	1	0	X	WR10A	RR10A	
X	X	1	1	0	1	1	X	WR11A	(RR15A)	
X	X	1	1	1	0	0	X	WR12A	RR12A	
X	X	1	1	1	0	1	X	WR13A	RR13A	
X	X	1	1	1	1	0	X	WR14A	(RR10A)	
X	X	1	1	1	1	1	X	WR15A	RR15A	

Note: Shift Right/Shift Left = 0
Ignore = X

Table 3-2 (continued). Z8030 Z-SCC Register Selection

AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	WRITE	READ	COMMENT
X	X	X	0	0	0	0	0	WR0B	RR0B	
X	X	X	0	0	0	0	1	WR0A	RR0A	
X	X	X	0	0	0	1	0	WR1B	RR1B	
X	X	X	0	0	0	1	1	WR1A	RR1A	
X	X	X	0	0	1	0	0	WR2B	RR2B	
X	X	X	0	0	1	0	1	WR2A	RR2A	
X	X	X	0	0	1	1	0	WR3B	RR3B	
X	X	X	0	0	1	1	1	WR3A	RR3A	
X	X	X	0	1	0	0	0	WR4B	(RR0B)	
X	X	X	0	1	0	0	1	WR4A	(RR0A)	
X	X	X	0	1	0	1	0	WR5B	(RR1B)	
X	X	X	0	1	0	1	1	WR5A	(RR1A)	
X	X	X	0	1	1	0	0	WR6B	(RR2B)	
X	X	X	0	1	1	0	1	WR6A	(RR2A)	
X	X	X	0	1	1	1	0	WR7B	(RR3B)	
X	X	X	0	1	1	1	1	WR7A	(RR3A)	
X	X	X	1	0	0	0	0	WR8B	RR8B	B Data
X	X	X	1	0	0	0	1	WR8A	RR8A	A Data
X	X	X	1	0	0	1	0	WR9		Shared
X	X	X	1	0	0	1	1	WR9		Shared
X	X	X	1	0	1	0	0	WR10B	RR10B	
X	X	X	1	0	1	0	1	WR10A	RR10A	
X	X	X	1	0	1	1	0	WR11B	(RR15B)	
X	X	X	1	0	1	1	1	WR11A	(RR15A)	
X	X	X	1	1	0	0	0	WR12B	RR12B	
X	X	X	1	1	0	0	1	WR12A	RR12A	
X	X	X	1	1	0	1	0	WR13B	RR13B	
X	X	X	1	1	0	1	1	WR13A	RR13A	
X	X	X	1	1	1	0	0	WR14B	(RR10B)	
X	X	X	1	1	1	0	1	WR14A	(RR10A)	
X	X	X	1	1	1	1	0	WR15B	RR15B	
X	X	X	1	1	1	1	1	WR15A	RR15A	

Note: Shift Right/Shift Left = 1
Ignore = X

3.1 ADDRESSING THE REGISTERS

3.1.1 Addressing the Z8030 Z-SCC Registers

The Z8030 Z-SCC registers are addressed with a single instruction where address and data are presented to the chip within the same instruction cycle. This direct accessing of registers is accomplished by decoding selected address bits on the address/data lines when the Address Strobe ($\overline{AS}$) makes a Low to High transition. The actual data will be written into the SCC during the Low to High transition of the Data Strobe ($\overline{DS}$) signal.

Different registers may be accessed depending on the setting of the Shift Right/Shift Left bit in Write Register 0 in Channel B . Table 3-2 shows the address used to select each register for each channel and shows which address lines are decoded depending on the setting of the Shift Right/Shift Left bit. If the Shift Right/Shift Left bit is 0, AD_1 through AD_5 will be decoded (this bit is reset by the hardware reset). The Z8001 and Z8002 CPUs use this mode. If the Shift Right/Shift Left bit is 1, then AD_0 through AD_4 will be decoded.

Each subsequent read and write cycle will follow this pattern of register selection during $\overline{AS}$ and

of bit manipulation during $\overline{DS}$. For example, if Write Register 3 is to be selected in Channel A using decoded address/data lines AD_0 through AD_4, the following bit configuration is defined:

Shift Right/Shift Left = 1
Decoded Address Lines:
AD_0 = 1
AD_1 = 1
AD_2 = 1
AD_3 = 0
AD_4 = 0

3.1.2 Addressing the Z8530 SCC Registers

Unlike the multiplexed Z-SCC, the nonmultiplexed Z8530 SCC requires two sequential accesses for register reads or writes. The first access is a write to a pointer and is then followed by a read (or write) to the actual register. The Z8530 SCC employs direct addressing only for the data registers (Write Register 8 and Read Register 8). All other registers (with the exception of Write Register 0 and Read Register 0) are accessed with the aid of two pins ($A/\overline{B}$ and $D/\overline{C}$) and six Command Register bits (D_0 through D_5).

Table 3-3 lists the bit configurations used to select registers for each state of the input select pins and the Command Register bits of a Z8530 SCC. If the $A/\overline{B}$ pin is High, a register in Channel A is selected and, if Low, a register in Channel B is selected. If the $D/\overline{C}$ pin is High (indicating a data transfer rather than a command), Write Register 8 is selected if in a write cycle and Read Register 8 if in a read cycle.

If the $D/\overline{C}$ pin is Low, the SCC performs the first access of the two sequential accesses required for the Z8530 SCC. Bits D_5 through D_0 of the Command Register (Write Register 0) are referred to for the point to the selected register. Bits D_3-D_0 point to the selected register as shown in Figure 3-1.

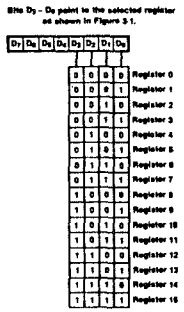


Figure 3-1. Z8530 SCC Register Selection

Table 3-3 maps the conditions for register selection for the Z8530.

Table 3-3. Z8530 SCC Register Selection

$A/\overline{B}$ PIN	$D/\overline{C}$ PIN	(WR0) D3	D2	D1	D0	WRITE	READ	COMMENTS
0	0	0	0	0	0	WR0B	RR0B	
0	0	0	0	0	1	WR1B	RR1B	
0	0	0	0	1	0	WR2	RR2B	
0	0	0	0	1	1	WR3B	RR3B	
0	0	0	1	0	0	WR4B	(RR0B)	
0	0	0	1	0	1	WR5B	(RR1B)	
0	0	0	1	1	0	WR6B	(RR2B)	
0	0	0	1	1	1	WR7B	(RR3B)	
0	1	0	X	X	X	WR8B	RR8B	B Data
1	0	0	0	0	0	WR0A	RR0A	
1	0	0	0	0	1	WR1A	RR1A	
1	0	0	0	1	0	WR2	RR2A	
1	0	0	0	1	1	WR3A	RR3A	
1	0	0	1	0	0	WR4A	(RR0A)	
1	0	0	1	0	1	WR5A	(RR1A)	
1	0	0	1	1	0	WR6A	(RR2A)	
1	0	0	1	1	1	WR7A	(RR3A)	
1	1	X	X	X	X	WR8A	RR8A	A Data
0	0	1	0	0	0	WR8B	RR8B	B Data
0	0	1	0	0	1	WR9		Shared
0	0	1	0	1	0	WR10B	RR10B	
0	0	1	0	1	1	WR11B	(RR15B)	
0	0	1	1	0	0	WR12B	RR12B	
0	0	1	1	0	1	WR13B	RR13B	
0	0	1	1	1	0	WR14B	(RR10B)	
0	0	1	1	1	1	WR15B	RR15B	
0	1	1	X	X	X	WR8B	RR8B	B Data
1	0	1	0	0	0	WR8A	RR8A	A Data
1	0	1	0	0	1	WR9		Shared
1	0	1	0	1	0	WR10A	RR10A	
1	0	1	0	1	1	WR11A	(RR15A)	
1	0	1	1	0	0	WR12A	RR12A	
1	0	1	1	0	1	WR13A	RR13A	
1	0	1	1	1	0	WR14A	(RR10A)	
1	0	1	1	1	1	WR15A	RR15A	
1	1	X	X	X	X	WR8A	RR8A	A Data

3.2 I/O INTERFACE CAPABILITIES

Regardless of the version of the SCC, all communication modes are established by programming various bit configurations into the control register set of the write registers. Once the registers have been initialized, data can be transferred. The direction of the data transfer depends on whether a read or a write cycle is in progress, and, as data is received or transmitted, read register values may change, altering the direction of the data path.

In the Z8030 Z-SCC, the R/$\overline{W}$ input is held High to read data from a selected register. The data appears on the address/data lines and is transferred in parallel to the CPU (or another parallel device).

In the Z8530 SCC, separate read ($\overline{RD}$) and write ($\overline{WR}$) inputs are used in conjunction with the Data/Control (D/$\overline{C}$) inputs to determine the type of data and the direction of data flow (Table 3-4).

The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU.

3.2.1 Polling

In a polled environment, bits 3 and 4 of Write Register 1 are configured with 0s to disable all receiver interrupts. This allows Read Register 0 (status bit) and Read Register 2 (modified interrupt vector, Channel B) to be monitored and allows the software to determine when a received character is available.

3.2.2 Interrupt Operation

As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one; i.e., when IEI is High. Through a combination of internal register bits and external interrupt lines, each peripheral is configured by both the system hardware and the software to communicate with the CPU on a priority basis. Figure 3-2 represents the interrupt structure of a Z8000 peripheral that contains four external I/O lines (IEI, $\overline{INT}$, $\overline{INTACK}$, IEO) and five internal register bits. These bits include:

- Interrupt Pending (IP)
- Interrupt Enable (IE)
- Master Interrupt Enable (MIE)
- Disable Lower Chain (DLC)
- Interrupt Under Service (IUS).

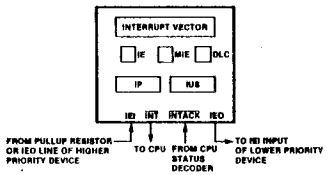


Figure 3-2. Peripheral Interrupt Structure

Table 3-4. Data Direction Control

A/$\overline{B}$	D/$\overline{C}$	Channel	$\overline{WR}$	$\overline{RD}$
0	0	B	Write Register Commands	Read Register Status
0	1	B	Write Register B Data	Read Register B Data
1	0	A	Write Register Commands	Read Register Status
1	1	A	Write Register B Data	Read Register B Data

In order for the SCC to request an interrupt the first condition that must be met is that the Master Interrupt Enable bit is set (MIE bit 3 in WR9). Then, the specific Interrupt Enable (IE) bit in WR9 for one of the six internal nterrupt sources must be set. Table 3-5 lists these six interrupt sources in their order of priority from high to low.

Table 3-5. Interrupt Source Priority

Receiver Channel A	High
Transmit Channel A	
External/Status Channel A	
Receiver Channel B	
Transmit Channel B	
External/Status Channel B	Low

If an interrupt source IE bit is set, the SCC Interrupt Pending (IP) bits can then be set (read in Read Register 3A), signalling the SCC's need for interrupt servicing. If IEI is High, informing the SCC that no higher priority device is requesting interrupt servicing, the $\overline{INT}$ output is pulled Low, which is the SCC's interrupt request to the CPU. The CPU acknowledges the interrupt by driving the Interrupt Acknowledge signal ($\overline{INTACK}$) Low.

When a Z8030 Z-SCC responds to an Interrupt Acknowledge signal from the CPU, an interrupt vector from the channel requiring service is placed on the multiplexed Address/Data bus. The Z8530 SCC responds to the $\overline{INTACK}$ low by placing the vector on the data bus during a read. This vector is written in Write Register 2 and may be read in Read Register 2A or Read Register 2B. To speed interrupt response time, the SCC can also modify three bits in this vector to indicate status. If the vector is read from Channel A, status is never included; if it is read from Channel B, status is always included. While the interrupt is being serviced (and if no higher priority device is requesting an interrupt) the SCC sets the Interrupt Under Service (IUS) bit. This action pulls the IEO Low to inhibit all lower priority interrupts, both internal and external to the SCC. Figure 3-3 is a flowchart of the SCC interrupt protocol, and Figure 3-4 is the Interrupt Acknowledge State chart.

3.2.2.1 Receiver Interrupt

Receiver Interrupt on All Characters

Each time a character reaches the top of the receive FIFO, an interrupt is generated. Error and special receive conditions generate special vectors if the Status Affects Vector bit (bit 4 of Write Register 9) is set. Optionally, a parity error can be directed to generate the special receive condition vector.

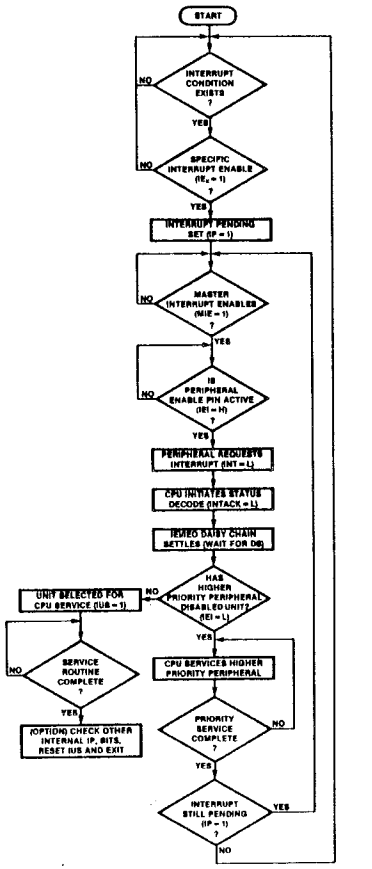


Figure 3-3. Interrupt Flowchart

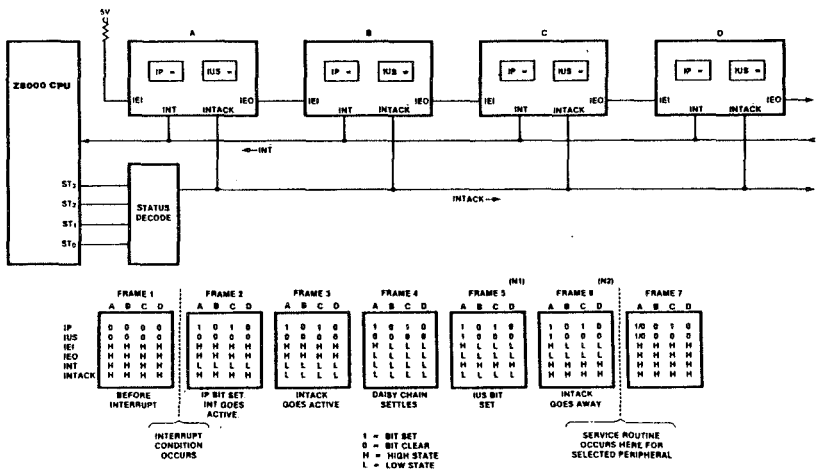


Figure 3-4. Interrupt Acknowledge State Chart

Bits 3 and 4 of Write Register 1 are written with a 0 and a 1, respectively, to allow interrupts on all characters. Special conditions such as receiver overrun or parity errors generate interrupts in this mode. (Section 4.1.2 gives further information on special conditions.)

Receiver Interrupt on First Character

This mode is normally used to start a Polling Loop or Block Transfer instruction using the $\overline{W/REQ}$ signal to synchronize the CPU or DMA device to the incoming data rate. In this mode, the SCC generates an interrupt on the first received character and thereafter generates an interrupt only if special receive conditions are detected. The mode is reinitialized with the Enable Interrupt On Next Receive Character command, which allows the next character received to generate an interrupt. In this mode special receive condition interrupts are cleared by issuing an Error Reset Command (Write Register 0). Any received data is held in the receive FIFO until this command is issued.

Receiver Interrupt on Special Condition Only

In this mode only special receive conditions cause interrupts. A special receive condition is one of the following: End of Frame (SDLC), overrun error and, optionally, parity error. Whenever a special receive condition occurs while in this mode the data in the receive FIFO is held until the Error Reset command (Write Register 0) is issued. This ensures that the CPU has enough information to handle the error.

Bits 3 and 4 in Write Register 1 are both configured with 1s to allow only special conditions to cause interrupts.

3.2.2.2 Transmit Interrupt

If the Transmit interrupt is enabled, bit 1 of Write Register 1 is set to 1. $\overline{INT}$ is pulled Low when the transmit buffer is empty.

3.2.2.3 External/Status Interrupt

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{SYNC}$ pins, however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode), or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode this feature allows secondary stations to recognize the wishes of the primary station to regain control of the loop during a poll sequence.

3.2.3 Block Transfer

The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{WAIT/REQUEST}$ output in conjuction with the Wait/Request bits in WR1. The $\overline{WAIT/REQUEST}$ output can be defined under software control as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the SCC $\overline{REQUEST}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{DTR/REQUEST}$ line allows full-duplex operation under DMA control.

3.2.3.1 Wait

Wait Low extends the current CPU execution cycle by adding null (WAIT) periods to the CPU cycle. The Wait function is selected by setting bit 6 of Write Register 1 to 0 and then setting bit 7 of Write Register 1 to 1. In the Wait mode, the $\overline{W/REQ}$ pin is open-collector when inactive and Low when active.

Bit 5 = 0	Bit 5 = 1
$\overline{W/REQ}$ is Low when the transmit buffer is full and a write to Write Register 8 is attempted.	$\overline{W/REQ}$ is Low when the receive buffer is empty and a read of Read Register 8 is attempted.

3.2.3.2 DMA Request

The DMA Request function is selected by setting bit 6 of Write Register 1 to 1 and then setting bit 7 of Write Register 1 to 1. This indicates that the SCC is ready to transfer data to or from memory. In DMA Request mode the $\overline{W/REQ}$ pin is High when inactive and Low when active. The SCC gives only one falling edge on $\overline{W/REQ}$ per request.

Bit 5 = 0	Bit 5 = 1
$\overline{W/REQ}$ is High when the transmit buffer is full.	$\overline{W/REQ}$ is High when the recieve buffer is empty.

3.3 SUPPORT CURCUITRY INITIALIZATION

In addition to the many available modes of communication, the SCC provides support for a baud rate generator, a Digital Phase-Lock Loop (DPLL) for clock recovery, and message encoding.

3.3.1 Baud Rate Generator

The baud rate generator must first be disabled for the time constant to be loaded. The formula for determining the time constant for a given rate is given below with the desired rate in bits per second and the baud rate clock frequency in seconds.

$$\frac{\text{Clock Frequency}}{2\ (\text{Desired Baud Rate})} - 2 = \text{Time Constant}$$

For example, using a 3.9936 MHz signal driving the baud rate generator, the time constants listed in Table 3-6 are loaded to obtain the desired baud rate.

$$\text{Example 1:} \left(\frac{3993600}{2(9600)}\right) - 2 = 206$$

$$\text{Example 2:} \left(\frac{3993600}{2(7200)}\right) - 2 = 275.3333$$

(Example 2 has an error of .12% because the Time Constant can only be an integer value.)

Once the baud rate generator is loaded with the Time Constant, it may be enabled by setting bits 1 and 0 of Write Register 14 to 1.

Table 3-6. Time-Constant Values

Rate	Time Constant	Error
19200	102	-
9600	206	-
7200	275	.12%
4800	414	-
3600	553	.06%
2400	830	-
2000	996	.04%
1800	1107	.03%
1200	1662	-
600	3326	-
300	6654	-
150	13310	-
134.5	14844	.0007%
110	18151	.0015%
75	26622	-
50	39934	-

3.3.2 Digital Phase-Locked Loop

The SCC contains a Digital Phase-Locked Loop that recovers clock information from a data stream with

NRZI or FM coding. The DPLL is driven by a clock nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a receive clock for the data.

NRZI encoding (DPLL)

The DPLL counts the 32x clock to create nominal bit times. While counting, the DPLL is searching the data stream. Whenever an edge is detected, the DPLL adjusts its count (during the next counting cycle) so that the terminal count is closer to the center of the bit cell. To determine how much of an adjustment must be made, the DPLL divides each nominal bit time into three regions.

Region 1 extends from the DPLL count of 0 until halfway through count 15. Region 2 extends from halfway through count 15 until halfway through count 16. Region 3 extends from halfway through count 16 until the next count of 0. The data edge normally occurs between counts 15 and 16 or in region 2, with no subsequent adjustment. No adjustment is necessary if there is no edge detection during the 0 to 31 counting cycle. If the data edge appears between counts 15 and 16, the falling edge output of the DPLL is centered on the bit cell and is used by the receiver to sample the RxD input.

If the transition in the incoming data stream did not occur during region 2, the DPLL needs to make an adjustment to the counting sequence in order to bring its terminal count closer to the center of the bit cell. If the transition of the incoming data stream occurs during region 1, the DPLL output is causing the data to be sampled too late in the bit cell. To remedy this situation, the DPLL shortens the count by one during the next counting cycle. If the transition in the incoming data stream occurs during region 3, the DPLL output is causing the data to be sampled too early in the bit cell. In this case, the count is extended by one during the next counting cycle. The small adjustments are chosen to prevent instabilities in the output of the DPLL, which cannot tell a noise edge from a data edge. By making the adjustments small, the system becomes highly overdamped, which is good for rejecting noise but increases the time required to lock up to the data stream.

Without a special startup procedure, the DPLL could require up to 16 transitions on the line to count the sampling edge. This is clearly unacceptable because in SDLC, there may be only three flags preceding the first frame. With NRZI, there are only three transitions on the line. To avoid this problem, the DPLL has a special Startup mode controlled by a command in Write Register 10. In this mode, the DPLL sits at count 16 and waits for an edge. The first edge detected is presumed to be a valid data edge and the DPLL begins counting from that point. If the first edge was in fact a valid data edge, the DPLL begins sampling correctly in the middle of a bit cell. If the first edge the DPLL detected was a noise edge, then the DPLL will lock on, although it will take longer to do so. When the DPLL recognizes the first edge, the Startup mode is automatically cancelled.

The maximum of one 32x clock correction per bit time places a constraint on the maximum errors allowed in the 32x clock rate. In SDLC, the longest time between 0s is seven bit times, corresponding to flags with shared 0s. To remain locked on the data stream, the number of errors in the 32x clock must be less than the maximum number of corrections allowed in one step. A larger correction might seem appropriate, but other considerations enter into the choice. If the DPLL is also supplying the transmit clock, large corrections to the DPLL output introduce undesirable amounts of jitter into the transmitted data stream.

The 32x clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or from the output of the baud rate generator.

3.3.3 Clocking Options

All transmit and receive clocking options are selected via Write Register 11 (see Figure 3-5). Bits 0 and 1 are used to select output source for the $\overline{TRxC}$ pin. Bit 2 determines whether the $\overline{TRxC}$ pin is an output or an input. Bits 3 and 4 select one of four sources of transmit clocks:

- $\overline{RTxC}$ pin
- $\overline{TRxC}$ pin
- Baud rate generator output
- DPLL output

Bits 5 and 6 select one of the four clock sources for the receive clock. Finally, bit 7 selects whether or not a crystal (XTAL) is connected to the $\overline{RTxC}$ pins.

The Receive/Transmit Clock ($\overline{RTxC}$) input may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed to use their respective $\overline{SYNC}$ pins as a crystal oscillator.

The $\overline{TRxCA}/\overline{TRxCB}$ pins may be programmed as either inputs or outputs. As an input, the pin supplies

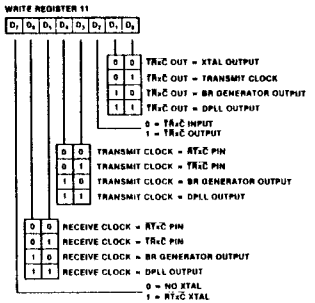


Figure 3-5. Transmit and Clocking Options

either the receive or transmit clock. As an output, it may supply the DPLL, crystal oscillator, baud rate generator, or the transmit clock.

The output of the baud rate generator can be used as either the transmit clock, the receive clock, or both. It can also drive the the Digital Phase-Locked Loop. If the receive clock or transmit clock is not programmed to come from the $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

3.3.4 CRC Error Checking

A Cyclic Redundancy Character (CRC) error check on the receive message can be performed on a per character basis under program control. The Receive CRC enable bit 3 in Write Register 3 must be established by the program before the next character is transferred from the Receive Shift register into the Receive buffer. This ensures proper inclusion or exclusion of data characters in the CRC check.

To allow the CPU ample time to enable or disable the CRC check on a particular character, the SCC calculates CRC eight bit times after the character has been transferred to the receive buffer. If CRC is enabled before the next character is transferred, CRC is calculated on the transferred character. If CRC is disabled before the time of the next transfer, calculation proceeds on the word in progress, but the word just transferred to the buffer is not included. When this happens, the 3-byte Receive data buffer is unusable in Bisync operation. CRC may be enabled or disabled as many times as necessary for a given message calculation.

Bit 6 of Read Register 1 contains the result of the CRC checker (after a 16-bit time delay for the data and CRC shift). The result should be zero, indicating an error-free transmission. The result is valid only at the end of CRC calculation. If the result is examined before this time, it usually indicates an error. Also, the comparison is made with each transfer and is valid only as long as the character remains in the receive FIFO.

The following description is an example of the CRC checking operation when four characters (A,B,C, and D) are received in that order.

Characters A and B are loaded into the Receive buffer. If CRC is disabled before character C is in the buffer, CRC is not calculated on B. After character C is loaded, the CRC Framing Error bit shows the result of the comparison through character A only. After character D is in the buffer, the CRC error bit shows the result of the comparison through character B, whether or not B was included in the CRC calculation.

Due to the serial nature of CRC calculation, the Receive clock (RxC) must cycle 16 times (to make up for the delay) after the second CRC character has been loaded into the Receive buffer. Or it must cycle 20 times (because of a three-bit buffer and 1-bit input delay) after the last bit is at the RxD input before CRC calculation is complete. A faster external clock can be gated into the Receive clock input to supply the required 16 cycles.

3.4 ASYNCHRONOUS MODE

The examples described in the following sections are not to be considered the final word on SCC operation. The design engineer or programmer is free to manipulate the SCC in a large number of system configurations and is encouraged to view the following example as a programming method and not as an element of constraint.

Many types of asynchronous operation are implemented in the SCC, thereby freeing the CPU of data communication functions. On the modem interface side of the SCC are programmable pins, which provide RS-232 interface capability. On the processor interface end is an 8-bit bus connected directly to the CPU data lines, which transfers usable data to and from the CPU. On the SCC interface, all external and interrupt provisions are available to ensure interrupt priority, interrupt vectors, and shared CPU data access.

In the Asynchronous mode, the receiver strips away expected start and stop bits at a programmed clock rate, provides error checking for overrun, parity and carrier-loss errors, and, if desired, provides interrupts for these conditions. Conversely, the transmit process inserts start, stop, and parity bits to a variable data format and supplies a serial data stream to the transmit data output.

3.4.1 Transmitter

In the following transmit initialization discussion, register assignments establish an example of an asynchronous message format with one start bit, seven data bits, one parity bit, and two stop bits. Other asynchronous transmit parameters are established during the following discussion. Figure 3-6 details the registers and their associated bit options for programming asynchronous operations. Both the receive and transmit parameters can be established during the initialization routine.

The following paragraphs give an example of a programming scenario and refer to the register programmed. More information on specific bit functions is presented in Chapter 4, where each register is explained in detail.

The user should disable all interrupts with the Master Interrupt Enable (Write Register 9, bit 3). Write Register 9 is also programmed at the beginning of the initialization routine to reset both Channels A and B, deselect the vector, control daisy-chain functions,and disable all interrupts. This register is important during the beginning and end of software routines when register assignments are performed in anticipation of data transfer.

For an example, the SCC transmit sequence will use the polled mode of operation; i.e., as the transmit buffer becomes empty, the program is forced to jump to a transmit handler routine. In a polled environment all appropriate interrupt modes are disabled, so the Transmit Interrupt Enable function in Write Register 1 is disabled. The Wait/DMA Request lines in Write Register 1 are temporarily disabled and are re-enabled when the initialization routine is complete.

The bits in Write Register 4 are extremely important to both the transmit and the receiver operations of the SCC and should be established first in the initialization routine. In the program example, two stop bits, odd parity, and the X16 asynchronous clock are selected. Further transmit parameters are selected in Write Register 5 as seven bits per character, and the RTS is enabled. (The synchronous transmission attributes of Write Register 5 are simply left unprogrammed.)

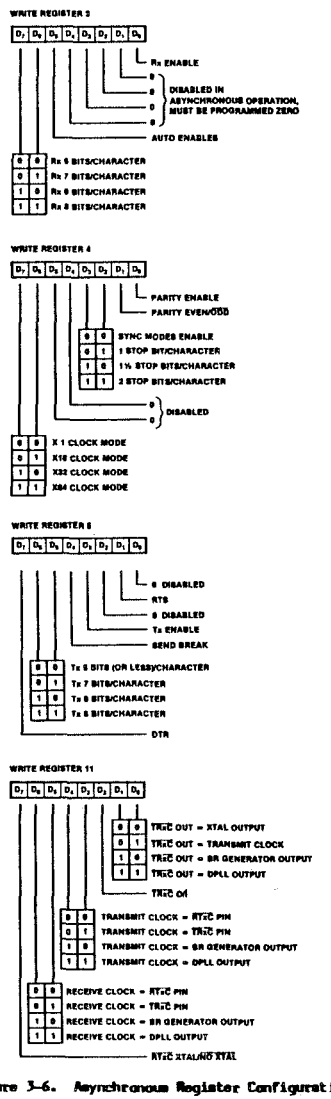


Figure 3-6. Asynchronous Register Configurations

Write Register 11 selects a clock source for the transmit data. For example, if the baud rate generator is selected as that source, the time constant selected for the baud rate generator will provide the desired clock rate. Write Register 12 contains the lower byte of the time constant and Write Register 13 contains the upper byte. Write Register 14 disables DPLL. Write Register 10 enables the selection of data coding (NRZ, NRZI, FM1, and FM0). The SCC transmitter is now initialized and is ready to transfer CPU data to the external world.

Data Transfer

The SCC inserts one start bit, calculates and inserts odd parity, and appends two stop bits to the character to be transmitted. The entire character is then shifted out of the SCC to the modem at the preselected clock rate.

3.4.2 Asynchronous Receive

In the initialization routine for transmitter operation, many of the functions are common to receive initialization (Figure 3-6). The number of start, stop, and parity bit assignments are assumed to be identical for two-way communication with an asynchronous device. Therefore, certain common elements are combined within the write register. This example initializes the SCC receiver to transfer data using interrupts.

Write Register 0 is configured to enable interrupts on the next received character and to reset the Rx CRC Checker bit. Write Register 1 is instrumental in setting the various modes of interrupts. Receiver interrupts are defined by bits 3 and 4. In the following example, interrupts are generated on all received characters or on a special condition. Notice that parity errors are programmed as special conditions. All modem interface signals are initialized at the end of the routine.

Example:

Since the Interrupt mode is selected, an interrupt vector is programmed to Write Register 2. Since there is only one interrupt vector, a way of distinguishing between the various interrupts available would be valuable to the programmer. The vector will be modified if the Vector Include Status (VIS) bit is set in Write Register 9. The VIS bit enables the SCC to modify either the three most significant bits or the three least significant bits of the vector in Write Register 2, depending on the setting of the Status High/Status Low bit in Write Register 9. In our initialization program for receiver operation in Channel A and with the Status High/Status Low bit equal to 0, the lower three vector bits are modified. Write Register 15 is also vital to the interrupt driven SCC because all of the miscellaneous interrupt sources are enabled (DCD or CTS transitions and Break detect).

The register values of Write Registers 3, 4,and 5 are established in the transmit initialization routine and are common to those of the receiver operation. These values establish the Clock, parity and stop bits, and the number of bits per character. Write Registers 6 and 7 are not used in asynchronous modes of operation. Write Register 11 is used to establish the source for the receive and transmit clocks. The baud rate generator is selected as the clock source with the identical time constant selected in the transmit initialization routine via Write Registers 12 and 13. The generator must be enabled in Write Register 14.

After the entire write register file is programmed, all of the enables are set as their associated routines become apparent; i.e., Tx Enable (Write Register 5), Receiver Enable (Write Register 3), and all of the interrupt source enables in Write Register 1 are set.

Receiver Data Transfer

The SCC automatically strips the start and stop bits from the character leaving only the right-justified data bits in the receive buffer (Read Register 8). Any errors occurring during the serial-to-parallel conversion are processed in a separate error-handling routine.

The receive buffer can then be read by the CPU. A data read program module resets the IUS bit with a command in Write Register 0 so that a new interrupt can be generated in the event of another received character. In a polled environment, the only significant difference in receive operation is the inclusion of a software idle loop to inspect Read Register 0 for a receive character available.

3.5 SYNCHRONOUS MODE

The SCC incorporates all of the internal logic necessary to handle synchronous modes of operation. Through the use of write registers allocated for sync character storage as well as on-chip logic for CRC checking, the SCC minimizes software and hardware design tasks for any communications

system using byte-oriented synchronous messages. This section details the steps involved in configuring the SCC to handle byte-oriented modes of communication.

Figure 3-7 illustrates the differences between monosync, bisync, and external sync modes. Monosync mode transmits a single sync character, which is compared to an identical sync character in the receiver. When the receiver recognizes this sync character, synchronization is complete and the receiver transfers subsequent received characters to the receive FIFO. Bisync mode uses a 16-bit or 12-bit sync character to obtain synchronization. External Sync mode uses an external synchronization signal to mark the beginning of a data field; i.e., an external input pin ($\overline{SYNC}$) indicates the start of the information field. In all synchronous modes, two cyclic redundancy check (CRC) bytes are added to the message to provide information about possible data transmission errors. The CRC bytes are calculated and inserted in the transmitted message to be compared to the CRC bytes calculated in the receiver. The actual CRC bytes received are then compared to the expected CRC bytes, and the results of this comparison are held in the receive error FIFO.

3.5.1 Transmitter

The system program must initialize the transmitter with the following:

- Odd or even parity
- The X1 Clock mode
- Transmitter enable
- Request to send
- Data terminal ready
- CRC polynomial
- Transmitted character length
- Interrupt modes, sync characters and enables

One of two methods can be used for data transfer: interrupts or block transfer using $\overline{WAIT}$ or $\overline{REQUEST}$. The External/Status Interrupt mode is used to monitor the status of the $\overline{CTS}$ input as well as of the Transmit Underrun/EOM latch. Optionally, the auto enable feature can be used to enable the transmitter when $\overline{CTS}$ is active.

The TxD pin is held marking after reset or if the transmitter is not enabled. A break can be programmed to generate a spacing line that begins with the next transmit clock after the send break is set. With the transmitter fully initialized and enabled, the default condition is continuous transmission of the 6- or 8- bit sync character.

3.5.1.1 Bisynchronous Transmit

Figure 3-8 lists the registers and their bits that are concerned directly with establishing a bisync message format. The bit configuration shown in the diagram serves only as an example to aid in the description of unique features of the Bisync mode of operation. The SCC registers will be programmed for no parity, the X1 Clock mode, one 16-bit sync character, the CRC-16 CRC polynomial, transmit enables, request to send, data terminal ready, interrupt modes, and transmit character length.

3.5.1.2 Monosynchronous Transmit

Monosync Transmit mode features initialization, status monitoring and data transfer procedures identical to those of the Bisync mode with the following exceptions: only one sync character is programmed in Write Registers 6 and 7, and the 6-bit or 8-bit select function in Write Register 10 must be programmed. All other features of synchronous operation are identical to the Transmit Bisync mode of operation.

In Write Register 0, the transmit CRC generator must be reset, as well as any external status interrupts, and then Write Register 4 must be initialized before other bisync functions are programmed. Write Register 4 is configured to select the Bisync mode and also set bits for no parity,

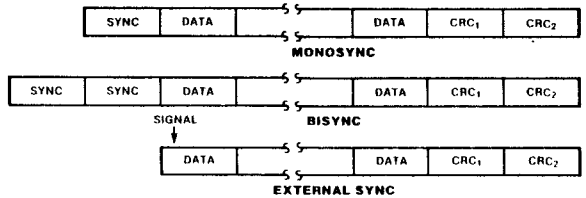


Figure 3-7. Synchronous Modes

sync mode enable, and the X1 Clock mode.

Write Register 5 is used to establish the number of bits per character transmitted. Write Register 5 also selects the the CRC-16 polynomial (the SDLC CRC polynomial could be selected), the transmit enable and the transmit CRC enable.

Write Registers 6 and 7 contain the transmit sync characters. Each time the CPU instructs the SCC to transmit a Bisync message, two sync characters are sent prior to the actual data. Write Register 6 and Write Register 7 are programmed to store these sync characters. The receiver portion of the Bisync device must use the identical sync characters or else the transmitted message will never be accepted.

Write Register 9 must be accessed several times during initialization; initially to reset the programmed channel, and at the end for setting the Master Interrupt Enable (MIE). Interrupt and vector variables are also programmed here.

Write Register 10 contains a few transmitter control bits that are important for Bisync operation. The programmer has the option of presetting the CRC generator to either all 0s or all 1s.

The transmit clock is established in Write Register 11. If the baud rate generator or external clocks are used as a clock source, Write Register 14 must be configured accordingly. Write Register 1 is the last register to be programmed in the initialization routine; it allows the programmed interrupt conditions to operate. At this point, bisync transmit initialization is complete and the controller is ready to transfer data.

Figure 3-8 shows the registers selected in Bisync mode that are used to establish the serial transmit functions.

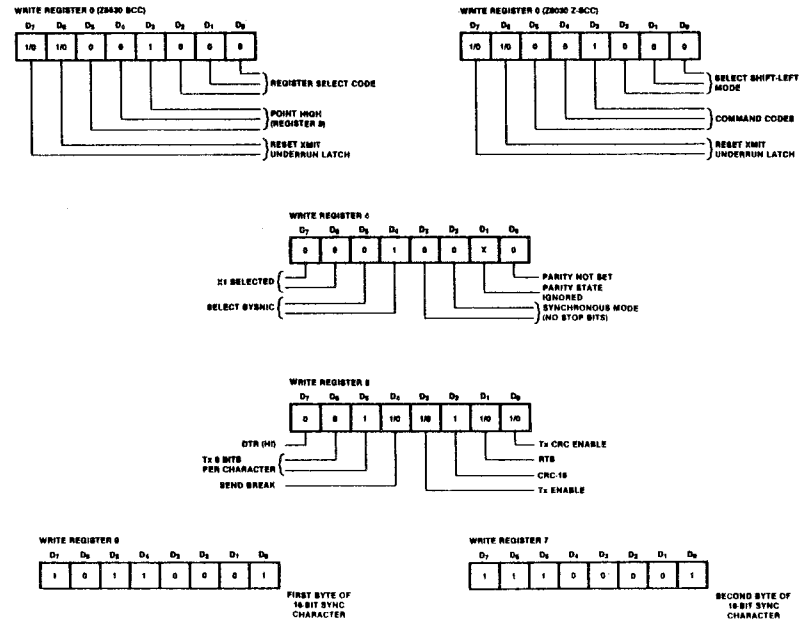


Figure 3-8. Transmit Bisync Mode of Operation

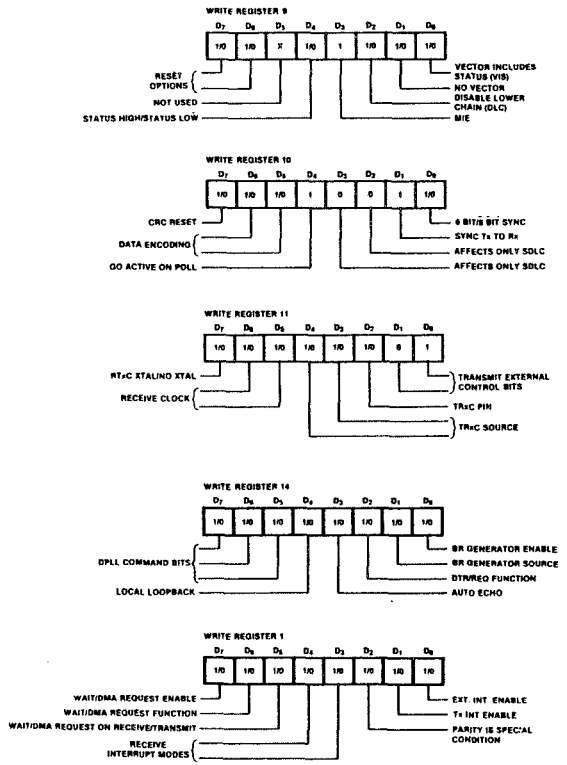


Figure 3-8. Transmit Bisync Mode of Operation (continued)

Transmit Data Transfer

When the initialization process is completed, several methods of data transfer are available using various combinations of interrupt or $\overline{\text{Wait}}$/$\overline{\text{DMA Request}}$ schemes. The Z-SCC continually monitors all transmit conditions and sets various bits in the read registers, which may or may not generate interrupts.

Bisync Transmit Termination

The SCC is equipped with a special termination feature that maintains data integrity and validity. If the transmitter is disabled while a data or sync character is being sent, the character is sent as usual but is followed by a marking line rather than sync or CRC characters. When the transmitter is disabled, a character in the buffer remains in the buffer. If the transmitter is disabled while CRC is being sent, the 16-bit transmission is completed, but sync is sent instead of CRC.

If the External/Status Interrupt Enable bit is set, transmitter conditions such as Transmit Underrun/EOM and a $\overline{\text{CTS}}$ state change cause interrupts and the data transfer is halted to allow the program to service the interrupt.

Bisync Transmit Underrun

When the transmitter has no further data to transmit, the SCC inserts bisync filler characters to maintain synchronization. The SCC has two programmable options for handling this situation: sync characters can be inserted or the CRC characters generated so far can be sent (followed by sync characters). These options are controlled by the Reset Transmit Underrun/End of Message (EOM) command in Write Register 0.

Following a chip or channel reset, the Transmit Underrun/EOM status bit in Read Register 0 (bit 6) is in a set condition and allows sync characters to be inserted when there is no data to send. CRC is not calculated on the automatically inserted sync characters. When the CPU detects the end of the message, a Reset Transmit Underrun/EOM command can be issued. This allows the CRC to be sent when the transmitter has no data. In this case, the SCC sends CRC followed by sync characters that terminate the message.

There is no restriction as to when in the message the Transmit Underrun/EOM bit can be reset. If the reset is issued after the first data character has been loaded, the 16-bit CRC is sent and followed by sync characters the first time the transmitter has no characters to send. Because of the transmit underrun condition, an External/Status interrupt is generated whenever the Transmit Underrun/EOM bit is set.

In the case of sync character insertion, an interrupt is generated only after the first automatically inserted sync character is loaded. The status bits indicate that the Transmit Underrun/EOM bit and the Transmit Buffer Empty bit are set.

In the case of CRC insertion, the Transmit Underrun/EOM bit is set and the Transmit Buffer Empty bit is reset while CRC is being sent. When CRC is completely sent, the Transmit Buffer Empty status bit is set and an interrupt is generated, indicating to the CPU that another message can be sent. This interrupt occurs when CRC has been sent and the sync character is loaded. If no more messages are to be sent, the program can terminate transmission by resetting Request To Send (RTS) and resetting bit 3 in Write Register 5. Pad characters (inserted to meet character count requirements) can be sent by setting the SCC to eight bits per transmit character and by writing all 1s to the transmitter while CRC is being sent. Alternatively, the sync charcters can be redefined as pad characters during this time.

After the last data byte has been written to the Z-SCC, the CPU issues the Reset Transmit Underrun/EOM Latch command and satisfies the interrupt with the Reset Tx Interrupt Pending command, which prevents the SCC from requesting more data. The SCC then sends CRC (because of the transmit underrun condition), which causes the External/Status interrupt with the Transmit Underrun/EOM latch set. The CPU satisfies this interrupt by loading pad characters into the transmit buffer in Write Register 8 and then issuing the Reset/External Status Interrupt command. In this sequence, CRC is followed by a pad character instead of a sync character. Note that the SCC issues an interrupt when CRC is completely sent and that the pad character is loaded in the Transmit Shift register. The CPU can now send more pad characters or sync characters.

3.5.1.3 External Sync Mode

All initialization procedures, data transfer methods, and status monitoring of the External Sync mode are identical to those of the Monosync mode except the $\overline{\text{SYNC}}$ pin on the SCC is used as an input for synchronization in the receiver. The transmitter is in Monosync mode when External Sync mode is selected for the receiver.

CRC Generation Mode

Setting the Transmit CRC enable bit 0 in Write Register 5 initiates CRC accumulation when the program sends the first data character to the SCC. Although the sync characters are inserted automatically, manually inserting a few additional sync characters ensures synchronization at the receiving end.

The Transmit CRC enable bit can be changed on the fly at any point in the message to include or exclude a particular data character from CRC accumulation. The Transmit CRC enable bit should be in the state desired when the data character is loaded from the transmit data buffer into the Transmit Shift register. To ensure the proper state of this bit, the Transmit CRC enable bit must be issued before sending the data character to the SCC.

3.5.1.4 CRC Transmission (Transparent Mode)

A Transparent mode of operation is made possible with the SCC's ability to change the Transmit CRC enable bit at any time during program sequencing and the additional capability of inserting 16-bit sync characters. Exclusion of DLE (Data Link Escape) characters from CRC calculation can be

achieved by disabling CRC calculations immediately preceding the DLE character transfer to the SCC. In the case of a transmit underrun condition in the Transparent mode, a pair of DLE-SYNC characters are sent. The SCC can be programmed to send the DLE-SYNC sequence by loading a DLE character into Write Register 6 and a SYNC character in Write Register 7.

3.5.1.5 Transmitter to Receiver Synchronization

The SCC offers a method of forcing character-synchronization between the transmitter and the receiver. That is, if both receiver and transmitter are programmed for the same number of bits per character, the character boundaries will be aligned. This feature works in Monosync with six or eight bit sync characters. To force synchronization both transmitter and receiver must be disabled and the Loop mode and Go Active On Poll (GAOP) bits in WR10 are set. Then the transmitter and receiver are enabled, in that order. While the transmitter is disabled, and until it goes active, it sends continuous 1s unless Break is programmed, in which case it sends all 0s. Once the receiver is enabled it is in Hunt until a sync character is recognized. Once a sync character is recognized the transmitter waits one character time and then goes active in character synchronization with the receiver. If Break was programmed it is automatically removed when the transmitter is activated. The On Loop bit in Write Register 10 is set at this time and the Loop Mode and GAOP bits may now be reset at any time, as the transmitter and receiver are now operating independently.

3.5.2 Receiver

Byte-oriented receiver programs are usually initialized with the following parameters: odd or even parity, 8- or 16-bit sync characters, the X1 Clock mode, the CRC polynomial, and the receiver character length. The sync characters must be loaded into Write Registers 6 and 7. The receivers should be enabled only after all of the receive parameters have been established.

When this is done, the receiver enters the hunt phase and remains in Hunt mode until character synchronization is achieved. The following sections describe receive parameters unique to the various modes of operation. The most extensive discussion is applied to the more common Bisync mode of operation. Monosync and external sync discussions are confined to their unique elements. For information on special conditions, interrupts, and data transfer methods, refer to the bisync section.

3.5.2.1 Bisync Receiver Initialization

During transmitter initialization for the Bisync mode, many of the receiver parameters are also established. Figure 3-9 highlights those elements unique to receiver initialization.

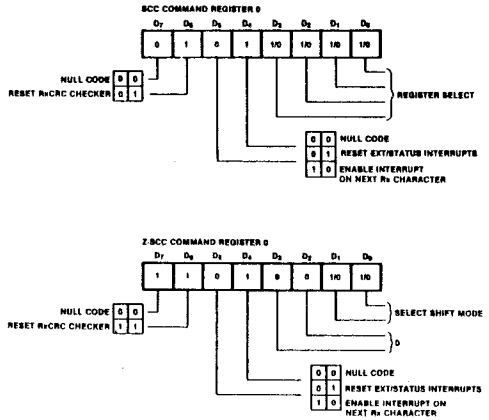


Figure 3-9. Receive Bisync Mode

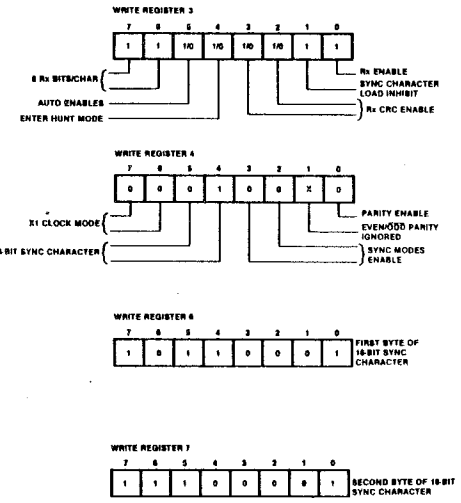


Figure 3-9. Receive Bisync Mode (continued)

The following receive parameters are programmed: no parity, 16-bit sync characters, the X1 Clock mode, the CRC-16 polynomial, receive character length, and sync characters. The receivers can be enabled only after all parameters are established along with the interrupt and special condition enables.

3.5.2.2. Monosync

Since monosync operation implies a single synchronization character (as opposed to two sync characters for bisync), all operating parameters are identical to Bisync. These include the resetting of the CRC checker, End Of Message (EOM) latch, the enabling of interrupt and receive character conditions, and enables for CRC, synchronous modes, and the monosync receiver. The 6- or 8-bit sync character is established in Write Register 6 and selected by bit 0 of Write Register 10.

Data transfer and status monitoring in Monosync mode is identical to that in Bisync mode.

Initialization

The Command Register (Write Register 0) is used to reset the external/status interrupts, reset any errors, and reset the receive CRC checker.

Write Register 3 selects the number of bits per character, enables the receiver, and enables the CRC checker. Hunt is automatically entered when the receiver is enabled, which initiates a comparison of incoming sync characters to the programmed value of Write Register 6 and Write Register 7. Comparison continues until character synchronization is complete. Before data is transferred, the receiver must leave Hunt. If desired, the leading sync characters of the message can be prevented from being loaded into the receive FIFO (and the CPU) by setting the Sync Character Load Inhibit bit in Write Register 3. Enabling the receiver is the last operation before data transfer begins.

Write Register 4 selects the Bisync mode of operation, enables synchronous operation, and establishes parity and clocking functions. Since the Bisync mode was selected, Write Registers 6 and 7 are programmed with two bytes of sync characters (usually the same as the transmitted bisync characters).

Data Transfer

After character synchronization is achieved, the assembled characters are transferred to the re-

ceive data FIF . During the monitoring period, interrupts are checked and CRC is calculated.

3.5.2.3 External Sync

In External Sync mode, the SCC utlizes the $\overline{SYNC}$ input pins. In this mode, $\overline{SYNC}$ must be driven Low two receive clock cycles after the last bit in the received sync character. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{SYNC}$.

The External Sync mode must be selected in Write Register 4. All other synchronous qualities are identical to those in the Bysync mode of operation.

3.6 SDLC MODE

SDLC mode differs considerably from Monosync and Bisync modes; it is bit-oriented rather than character-oriented and can handle transparent operation naturally. Bit-orientation makes SDLC flexible in terms of message length and bit patterns (Figure 3-10).

The SDLC message, called the frame, is opened and closed by flags which are a bit pattern of 01111110. The SCC handles the transmission and recognition of the flag characters that mark the beginning and end of a frame. The SCC can receive shared-zero flags, but cannot transmit them. The 8-bit address field of an SDLC frame contains the secondary station address. The SCC has an Address Search mode used to recognize a secondary station address. In this manner, the SCC can determine which frames to send to the CPU.

Since the control field of the SDLC frame is transparent to the SCC, it is simply tranferred to the CPU. The way in which the SCC handles the CRC simplifies programming by incorporating such features as initializing the CRC generator, resetting the CRC checker when the opening flag is detected in the Receive mode, and sending the CRC flag sequence in the Transmit mode. Zeros are automatically inserted and deleted by logic contained in the SCC.

3.6.1 SDLC Transmitter

To program the SCC for SDLC operation, the write registers are loaded with specific SDLC parameters: SDLC mode, SDLC-CRC polynomial, request-to-send, data terminal ready, transmit character length, Transmit Interrupt mode, transmit enable, auto enables, and external/status interrupts enable.

The SDLC-CRC polynomial must be set in SDLC mode. The CRC generator and checker may be preset to all 1s or all 0s. After reset and initialization, TxD is held in marking state. After the transmitter is enabled, continuous flags are transmitted.

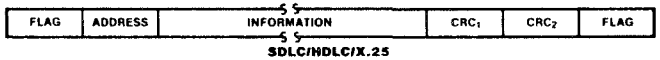
The Reset Tx CRC Generator command in Write Register 0 should be issued after the transmitter is enabled to initialize the CRC generator. Write Register 1 is used to establish the Wait/DMA Request on Receive/Transmit.

Write Register 4 must be programmed before any other registers to select the SDLC mode by writing a 10 to bits 5 and 4. Bits 6 and 7 of the same register are used to select the 1X Clock mode. The CRC polynomial is selected in Write Register 5 by programming a 0 into bit 2, while the Transmitter enable bit 3 is set following the initialization routine.

Write Register 6 is programmed to contain the secondary address field (control field), used to compare against the address field of the SDLC frame, and Write Register 7 is programmed to contain a flag character (01111110).

Write Register 9 is the Master Interrupt Control register. Write Register 10 contains the Go-Active-On-Poll bit and the Loop Mode bit for configuring SDLC operation. Bit 3 is the Mark/Idle Flag bit used in SDLC mode to control the idle line condition. Write Register 11 selects the source of the transmit clock while the selected time constant is programmed in Write Registers 12 and 13 (if the baud rate generator is selected).

The source for the baud rate generator is selected in Write Register 14 from either the Crystal (XTAL), PCLK, or $\overline{RTxC}$ pins. Write Register 15 establishes the enable conditions for all interrupts.



SDLC/HDLC/X.25

Figure 3-10. SDLC Message Format

3.7 SDLC DATA TRANSFER MODES

The actual data exchange between the CPU, the SCC, and the selected peripheral depends on the write register configuration of either the Interrupt or Polled mode of operation and the state of the external pins.

3.7.1 Interrupt Mode

If transmit interrupts are enabled, an interrupt is generated each time the transmit buffer becomes empty. This interrupt can be satisfied either by writing another character into the transmitter or by issuing the Reset Tx Interrupt Pending command. If nothing more is to be written into the transmitter, no further interrupts occur. This results in a transmitter underrun condition. When another character is written and sent out, the transmitter can become empty again and interrupt the CPU. (When the transmitter is first enabled, it is already empty and cannot interrupt until the first data character is written.)

3.7.2 Data Transfer Using Wait/DMA Request

The $\overline{WAIT}$ signal indicates to the CPU that the SCC is not ready to accept data and that the CPU must extend the I/O cycle. When information is sent to a DMA controller, $\overline{REQ}$ indicates an empty buffer. Care must be taken to ensure the arrival of a new data character from the CPU before the Transmit Shift register becomes empty. If this does not happen, the SCC enters a transmit underrun condition.

3.7.3 SDLC Transmit Characteristics

The SCC can transmit SDLC frames of any length. This is accomplished by changing the number of bits per character to be transmitted (in Write Register 5) on the fly. Any transmit data written to the SCC after the bits per character field is changed are affected by this change. The same is true of any characters in the buffer at the time the bits per character field is changed. A transmitted frame can be terminated by CRC and a flag, by a flag only, or by an abort. This is controlled by the Tx Underrun/EOM latch and the Abort/Flag on Underrun bit in Write Register 10. The idle line condition may be flags or continuous 1s. This is controlled by the Mark/Flag Idle bit in Write Register 10.

Transmit Underrun, CRC Transmission

The state of the Transmit Underrun/EOM command determines SCC action in handling underrun situations which occur when no bits are remaining in the Shift register and the buffer is empty. The SCC can terminate an SDLC frame by sending two CRC bytes followed by one or more flag characters. This procedure allows high speed DMA or block I/O instructions without CPU intervention.

When the Transmit Underrun/EOM bit is set, CRC characters cannot be inserted into the message stream. However, the SCC sends the frame as soon as data is written into the buffer, and between the time the first data byte is written and the end of the message, the Transmit Underrun/EOM bit must be reset. Resetting this bit allows CRC and flags to be sent when there is no data to send.

When CRC transmission begins, the Tx Underrun/EOM latch is set causing an External/Status interrupt if enabled. At the same time, the Tx Buffer Empty bit is reset, indicating that CRC transmission has begun. At the end of CRC transmission, when the closing flag is loaded into the Transmit Shift register, the Tx Buffer Empty bit is set. If transmit interrupts are enabled, a transmit interrupt is generated. If $\overline{REQ}$ has been enabled,it will pulse Low for one PCLK cycle when the Tx Buffer Empty bit is set, requesting the first byte of the next frame.

The CRC preset I/O and Transmit CRC enable bits must be programmed before the first data is written to the SCC. Channel resets and hardware resets do not preset the CRC generator, so the Reset Tx CRC Generator command must also be issued after the transmitter is enabled but before the first data is written to the SCC.

3.7.4 SDLC Receiver

Once SDLC mode is selected in Write Register 4, the other registers pertinent to SDLC mode operation can be programmed. The SDLC CRC polynomial must be selected in Write Register 5. Write Register 3 selects the bits per character as well as Address Search mode and Rx CRC enable. The SDLC flag and receiver address field should be programmed into Write Register 6 and Write Register 7, respectively. The underrun, idle line, and CRC preset options in Write Register 10 are programmed, and then the receiver should be enabled by programming Write Register 3. The receiver is in Hunt mode when it is first enabled.

3.7.4.1 SDLC Receive CRC Checking

Control of the receive CRC checker is automatic. It is reset by the leading flag and CRC is calculated up to the final flag. The byte with the

End-of-Frame bit set is the byte that contains the result of the CRC check. The last byte of a frame transferred to the receive FIFO is not bits 0 to 7 of the CRC, but is bits 2 through 9 of the CRC. The last two bits before the closing flag are never transferred to the CPU. If the CRC error bit is not set, a correct CRC was received. A special check sequence is used for the SDLC check, because the SCC inverts the CRC character before transmission. The final check must match a binary 0001110100001111.

3.7.4.2 Hunt mode, Address Search

The SDLC receive sequence begins when the SCC enters the Hunt mode. This mode is entered upon receive enable, abort detect, or by software command. In the Hunt mode, the SCC searches for flag characters, and when it detects a flag, an interrupt is generated. The SCC then exits the Hunt mode and enters Address Search mode.

If the Address Search bit in Write Register 3 is set, the first non-flag character following a flag is compared to the programmed address in Write Register 6 and the hardwired global address (11111111). If the SDLC frame address field matches either of these addresses, data transfer begins. Upon receipt of a valid flag, the assembled characters are transferred to the receive data FIFO.

3.7.4.3 SDLC Loop Mode

When the frame is complete and CRC has been sent, the SCC closes with a flag and reverts to One-Bit Delay mode. The last zero of the flag and marking line echoed from the RxD pin form an EOP for secondary stations further down the loop. If the Go-Active-On-Poll bit (Write Register 10) is not set at the time the EOP passes by, the SCC cannot send a message until a flag (terminating the current polling sequence) and another EOP are received. The loop sending bit in Read Register 10 is set to indicate when the SCC is transmitting a message.

If the SDLC Loop mode is deselected, the SCC is designed to exit from the loop gracefully. When SDLC Loop mode is deselected by writing to Write Register 11, the SCC waits until the next polling cycle to remove the 1-bit delay. If a polling cycle is in progress at the time the command is written, the SCC finishes sending any messages in progress, ends with an EOP, and disconnects TxD from RxD. If no message is in progress, the SCC immediately disconnects TxD from RxD. If a polling cycle is not in progress at the time the command is given, the SCC waits until an EOP is recognized to disconnect TxD from RxD. To ensure proper loop operation after the SCC goes off-loop, and until the external relays take the SCC completely out of the loop, the SCC should be programmed for mark idle instead of flag idle. When the SCC goes off-loop, the on-loop bit is reset. SDLC Loop mode is illustrated in Figure 3-11.

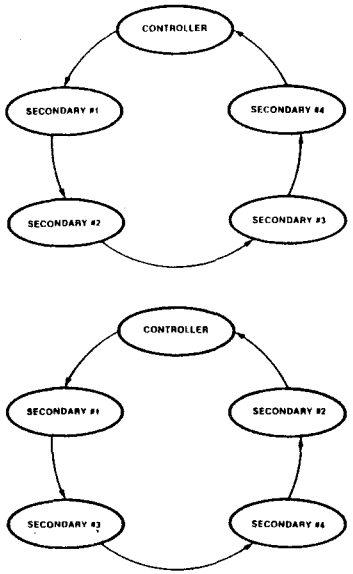


Figure 3-11. SDLC Loop

NRZI and FM in SDLC Loop Mode

The SCC gives the user the option of using NRZI or FM in SDLC Loop mode by programming Write Register 10 appropriately. The SCC monitors the incoming data stream and automatically adjusts its encoding logic so that noise is not introduced into the loop.

The off-loop procedure can be complicated. With NRZI or FM encoding, the outputs of secondary stations in the loop can be inverted from their inputs because of messages they have transmitted. Removing the stations from the loop (removing the 1-bit delay) may cause problems further down the loop because of extraneous transitions on the line. The SCC avoids this problem by making transparent adjustments at the end of each frame it sends in response to an EOP. A response frame from the SCC is terminated by a flag and an EOP. Normally, the flag and EOP share a zero, but if doing so would cause the RxD and TxD pins to be of opposite polarity after the EOP, the SCC adds another zero between the flag and EOP. This causes an extra line transition so that RxD and TxD will be identical after the EOP is sent. This extra zero is transparent and means only that the flag and the EOP no longer share a zero. Loop exit is thus simplified so that all that is necessary for a proper loop exit is the removal of the 1-bit delay.

3.8 Auto Echo Mode

The SCC can automatically echo everything received from the serial data stream. This feature is useful in asynchronous modes and also works in byte-oriented and bit-oriented modes. Auto Echo mode is selected by writing the appropriate control word to Write Register 11. In Auto Echo mode, TxD is connected directly to RxD. In Auto Echo mode, the $\overline{CTS}$ input is ignored as a transmitter enable, although transitions on this input can still cause interrupts if programmed to do so. In this mode, the transmitter is actually bypassed and the programmer should disable transmitter interrupts and the $\overline{WAIT/REQUEST}$ pin on transmit.

The SCC also has a Local Loopback mode. In Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored. Transmit data still goes to $\overline{TxC}$. In Local Loopback mode, the $\overline{CTS}$ and $\overline{DCD}$ inputs are ignored as transmit and receive enables, but transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, byte-oriented and bit-oriented synchronous modes with NRZ, NRZI and FM coding of the data stream. In either of these modes the full range of clocking options is still available to the user.

Chapter 4
Register Description

4.0 REGISTER DESCRIPTION

The following sections describe the Z-SCC registers. Each register is detailed in terms of bit configuration, the active states of each bit, their definitions, their functions, and their effects upon the internal hardware and external pins.

4.1 WRITE REGISTERS

The Z-SCC write register set includes ten control registers, two sync character registers, two baud rate time constant registers and a transmit buffer in each channel, and one master interrupt register. The only difference in register definition between the Z8030 and Z8530 versions of the Z-SCC exists in the command decode structure. The following sections describe in detail each write register and the associated bit configuration for each.

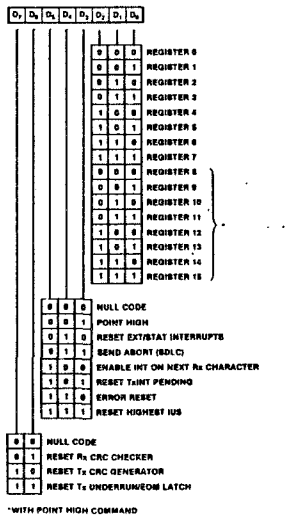


Figure 4-1. Command Register (8530)

4.1.1 Write Register 0 (Command Register)

WR0 is the command register and the CRC reset code register. WR0 in the Z8030 version varies slightly from that in the Z8530 version. Figure 4-1 shows the bit configuration for the Z8530 version and includes register select bits in addition to command and reset codes. Figure 4-2 shows the bit configuration for the Z8030 version and includes (in Channel B only) the address decoding select described in the Programming section. The following bit description for WR0 is identical for both versions except where specified.

Bits D7-D6: CRC RESET CODES 0 AND 1

Null Codes (00). This command has no effect on the Z-SCC and is used when a write to WR0 is necessary for some reason other than a CRC Reset command.

Reset Receive CRC Checker (01). This command is used to initialize the receive CRC circuitry. It is necessary in synchronous modes (except SDLC) if the Enter Hunt Mode command in Write Register 3 is not issued between received messages. Any action

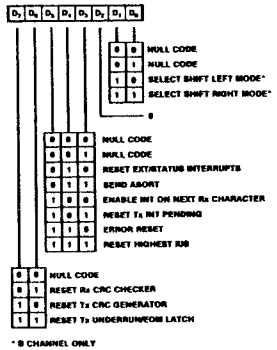


Figure 4-2. Command Register (8030)

that disables the receiver initializes the CRC circuitry. Resetting the Receive CRC Checker command is accomplished automatically in SDLC mode.

Reset Transmit CRC Generator (10). This command initializes the CRC generator. It is usually issued in the initialization routine and after the CRC has been transmitted. A Channel Reset will not initialize the generator and this command should not be issued until after the transmitter has been enabled in the initialization routine.

Reset Transmit Underrun/EOM Latch (11). This command controls the transmission of CRC at the end of transmission (EOM). If this latch has been reset and a transmit underrun occurs, the Z-SCC automatically appends CRC to the message. In SDLC mode with Abort On Underrun selected, the SCC sends an abort, and Flag on underrun if the Tx Underrun/EOM latch has been reset.

At the start of the CRC transmission, the Tx Underrun/EOM latch is set. The Reset command can be issued at any time during a message. If the transmitter is disabled, this command will not reset the latch. However, if no External Status interrupt is pending or if a Reset External Status Int command accompanies this command while the transmitter is disabled, an External/Status interrupt is generated with the Tx Underrun/EOM bit rest in RR0.

Bits D_5-D_3: COMMAND CODES

Null Code (000). The Null command has no effect on the Z-SCC.

Point High (001). This command effectively adds eight to the Register Pointer (B2-B0) by allowing WR8 through WR15 to be accessed. The Point High command and the Register Pointer bits are written simultaneously. This command is used only in the Z8530 version of the Z-SCC. In the Z8030 version, the registers are accessed as described in the **Programming** section.

Reset External/Status Interrupts (010). After an External/Status interrupt (a change on a modem line or a break condition, for example), the status bits in RR0 are latched. This command re-enables the bits and allows interrupts to occur again as a result of a status change. Latching the status bits captures short pulses until the CPU has time to read the change. The SCC contains simple queueing logic associated with most of the external status bits in RR0. If another External/Status condition changes while a previous condition is still pending (Reset External/Status Interrupts has not yet been issued) and this condition persists until after the command is issued, this second change causes another External/Status interrupt. However, if this second status change does not persist (there are two transitions), another interrupt is not generated. Exceptions to this rule are detailed in the RR0 description.

Send Abort (011). This command is used in SDLC mode to transmit a sequence of eight to thirteen 1s. This command always empties the transmit buffer and sets the Tx Underrun/EOM bit in Read Register 0.

Enable Interrupt on Next Rx Character (100). If the interrupt on the First Received Character mode is selected, this command is used to reactivate that mode after each message is received. The next character to enter the receive FIFO causes a Receive interrupt. Alternatively, the first previously stored character in the FIFO will cause a Receive interrupt.

Reset Tx Interrupt Pending (101). This command is used in cases where there are no more characters to be sent; e.g., at the end of a message. This command prevents further transmit interrrupts until after the next character has been loaded into the transmit buffer or until CRC has been completely sent. This command is necessary to prevent the transmitter from requesting an interrupt when the transmit buffer becomes empty (with Transmit Interrupt Enabled).

Error Reset (110). This command resets the error bits in RR1. If Interrupt on First Rx Character or Interrupt on Special Condition modes are selected and a special condition exists, the data with the special condition is held in the receive FIFO until this command is issued. If either of these modes is selected and this command is issued before the data has been read from the receive FIFO, the data is lost.

Reset Highest IUS (111). This command resets the highest priority Interrupt Under Service (IUS) bit, allowing lower priority conditions to request interrupts. This command allows the use of the internal daisy chain (even in systems without an external daisy chain) and should be the last operation in an interrupt service routine.

Bits 2 through 0: REGISTER SELECTION CODE

These three bits select Registers 0 through 7. With the Point High command, Registers 8 through 15 are selected. The Register Selection Code bits are used only in the Z8530 version. In the Z8030 version, these bits are always 0.

The following is a summary of the bit descriptions for each write register (WR1-WR15) used in both the Z8530 and the Z8030 Z-SCC.

4.1.2 Write Register 1 (Transmit/Receive Interrupt and Data Transfer Mode Definition)

Write Register 1 is the control register for the various Z-SCC interrupt and Wait/Request modes. Figure 4-3 shows the bit assignments for WR1.

Bit 7: WAIT/DMA REQUEST ENABLE

This bit enables the Wait/Request function in conjunction with the Request/Wait Function Select bit (B6). If bit 7 is set to 1, the state of bit 6 determines the activity of the WAIT/REQUEST pin (Wait or Request). If bit 7 is set to 0, the selected function (bit 6) forces the WAIT/REQUEST pin in to the appropriate inactive state (High for Request, floating for Wait).

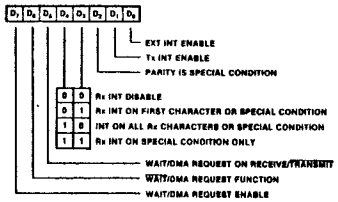


Figure 4-3. Write Register 1

Bit 6: WAIT/DMA REQUEST FUNCTION

The request function is selected by setting this bit to 1. In the Request mode, the WAIT/REQUEST pin switches from High to Low when the SCC is ready to transfer data. When this bit is 0, the wait function is selected. In the Wait mode, the WAIT/REQUEST pin switches from floating to Low when the CPU attempts to transfer data before the SCC is ready.

Bit 5: WAIT/DMA REQUEST ON RECEIVE TRANSMIT

This bit determines whether the WAIT/REQUEST pin operates in the Transmit mode or the Receive mode. When set to 1, this bit allows the wait/request function to follow the state of the receive buffer; i.e., depending on the state of bit 6, the WAIT/REQUEST pin is active or inactive in relation to the empty or full state of the receive buffer. Conversely, if this bit is set to 0, the state of the WAIT/REQUEST pin is determined by bit 6 and the state of the transmit buffer. (Note that a transmit request function is available on the DTR/REQUEST pin. This allows full-duplex operation under DMA control for both channels).

The request function may occur only when the Z-SCC is not selected; e.g., if the internal request becomes active while the Z-SCC is in the middle of a read or write cycle, the external request will not become active until the cycle is complete. An active request output causes a DMA controller to initiate a read or write operation. If the request on Transmit mode is selected in either SDLC or Synchronous Mode, the Request pin is pulsed Low for one PCLK cycle at the end of CRC transmission to allow the immediate transmission of another block of data.

In the Wait On Receive mode, the WAIT pin is active if the CPU attempts to read Z-SCC data that has not yet been received. In the Wait On Transmit mode, the WAIT pin is active if the CPU attempts to write data when the transmit buffer is still full. Both situations can occur frequently when block transfer instructions are used.

Bits 4 and 3: Receive Interrupt Modes

These two bits specify the various character-available conditions that may cause interrupt requests.

Receive Interrupts Disabled (00). This mode prevents the receiver from requesting an interrupt and is normally used in a polled environment where either the status bits in RR0 or the modified vector in RR2 (Channel B) can be monitored to initiate a service routine. Although the receiver interrupts are disabled, a special condition can still provide a unique vector status in RR2.

Receive Interrupt on First Character or Special Condition (01). The receiver requests an interrupt in this mode on the first available character (or stored FIFO character) or on a special condition. Sync characters to be stripped from the message stream do not cause interrupts.

Special receive conditions are: receiver overrun, framing error, end of frame, or parity error (if

selected). If a special receive condition occurs, the data containing the error is stored in the receive FIFO until an Error Reset command is issued by the CPU.

This mode is ususally selected when a Block Transfer mode is used. In this interrupt mode, a pending special receive condition remains set until either an Error Reset command, a channel or hardware reset, or until receive interrupts are disabled.

The Receive Interrupt on First Character or Special Condition mode can be re-enabled by the Enable Rx Interrupt on Next Character command in WR0.

Interrupt on All Receive Characters or Special Condition (10). This mode allows an interrupt for every character received (or character in the receive FIFO) and provides a unique vector when a special condition exists. The Receiver Overrun bit and the Parity Error bit in RR1 are two special conditions that are latched. These two bits must be reset by the Error Reset command. Receiver overrun is always a special receive condition, and parity can be programmed to be a special condition.

Data conditions with special receive conditions are not held in the receive FIFO in the Interrupt On All Receive Characters or Special Conditions Mode.

Receive Interrupt on Special Condition (11). This mode allows the receiver to interrupt only on characters with a special receive condition. When an interrupt occurs, the data containing the error is held in the receive FIFO until an Error Reset command is issued. When using this mode in conjunction with a DMA, the DMA can be initialized and enabled before any characters have been received by the Z-SCC. This eliminates the time-critical section of code required in the Receive Interrupt on First Character or Special condition mode; i.e., all data can be transferred via the DMA so that the CPU need not handle the first received character as a special case.

Bit 2: Parity Is Special Condition

If this bit is set to 1, any received characters with parity not matching the sense programmed in WR4 give rise to a Special Receive Condition. If parity is disabled (WR4), this bit is ignored. A special condition modifies the status of the interrupt vector stored in WR2. During an interrupt acknowledge cycle, this vector can be placed on the data bus.

Bit 1: Transmitter Interrupt Enable

If this bit is set to 1, the transmitter requests an interrupt whenever the transmit buffer becomes empty.

Bit 0: External/Status Master Interrupt Enable

This bit is the master enable for External/Status interrupts including $\overline{DCD}$, $\overline{CTS}$, $\overline{SYNC}$ pins, break, abort, the beginning of CRC transmission when the Transmit/Underrun/EOM latch is set, or when the counter in the baud rate generator reaches 0. Write Register 15 contains the individual enable bits for each of these sources of External/Status interrupts. This bit is reset by a channel or hardware reset.

4.1.3 Write Register 2 (interrupt vector)

WR2 is the interrupt vector register. Only one register vector exists in the Z-SCC, but it can be accessed through either channel. The interrupt vector can be modified by status information. This is controlled by the Vector Includes Status (VIS) and the Status High/Status Low bits in register W/R 9. The bit positions for WR2 are shown in Figure 4-4.

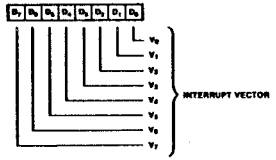


Figure 4-4. Write Register 2

4.1.4 Write Register 3 (Receive Parameters and Control)

This register contains the control bits and parameters for the receiver logic illustrated in Figure 4-5.

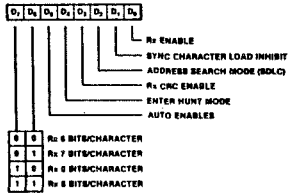


Figure 4-5. Write Register 3

2016 006 2016:006

Bits 7 and 6: Receiver Bits/Character

The state of these two bits determines the number of bits to be assembled as a character in the received serial data stream. The number of bits per character can be changed while a character is being assembled but only before the number of bits currently programmed is reached. Unused bits in the Received Data Register (RR8) are set to 1 in asynchronous modes. In synchronous modes and SDLC modes, the Z-SCC merely transfers an 8-bit section of the serial data stream to the receive of the serial data stream to the receive FIFO at the appropriate time. Table 4-1 lists the number of bits per character in the assembled character format.

Table 4-1. Receive Bits/Character

B_7	B_6	
0	0	5 Bits/Character
0	1	7 Bits/Character
1	0	6 Bits/Character
1	1	8 Bits/Character

Bit 5: Auto Enables

This bit programs the function for both the DCD and CTS pins. CTS becomes the transmitter enable and DCD becomes the receiver enable when this bit is set to 1. However, the Receiver Enable and Transmit Enable bits must be set before the DCD and CTS pins can be used in this manner. When the Auto Enables bit is set to 0, the DCD and CTS pins are merely inputs to the corresponding status bits in Read Register 0. The state of DCD is ignored in the Local Loopback mode. The state of CTS is ignored in both Auto Echo and Local Loopback modes.

Bit 4: Enter Hunt Mode

This command forces the comparison of sync characters or flags to assembled receive characters for the purpose of synchronization. After reset, the Z-SCC automatically enters the Hunt mode (except asynchronous). Whenever a flag or sync character is matched, the Sync Hunt bit in Read Register 0 is reset and, if External/Status Interrupt Enable is set, an interrupt sequence is initiated. The Z-SCC automatically enters the Hunt mode when an abort condition is received or when the receiver is disabled.

Bit 3: Receiver CRC Enable

This bit is used to initiate CRC calculation at the beginning of the last byte transferred from the Receiver Shift register to the receive FIFO. This operation occurs independently of the number of bytes in the receive FIFO. When a particular byte is to be excluded from CRC calculation, this bit should be reset before the next byte is transferred to the receive FIFO.

This bit is internally set to 1 in SDLC mode and the SCC calculates CRC on all bits except between the opening and closing character flags. This bit is ignored in asynchronous modes. If this feature is used, care must be taken to ensure that eight bits per character is selected in the receiver because of an inherent delay from the Receive Shift register to the CRC checker.

Bit 2: Address Search Mode (SDLC)

Setting this bit in SLDC mode causes messages with addresses not matching the address programmed in WR6 to be rejected. No receiver interrupts can occur in this mode unless there is an address match. The address that the Z-SCC attempts to match can be unique (1 in 256) or multiple (16 in 256), depending on the state of Sync Character Load Inhibit bit. The Address Search mode bit is ignored in all modes excpet SDLC.

Bit 1: Sync Character Load Inhibit

If this bit is set to 1 in any synchronous mode except SDLC, the Z-SCC compares the byte in WR6 with the byte about to be stored in the FIFO, and it inhibits this load if the bytes are equal. The Z-SCC does not calculate the CRC on bytes stripped from the data stream in this manner. If the 6-bit sync option is selected while in Monosync mode, the compare is still across eight-bits, so WR6 must be programmed for proper operation.

If the 6-bit sync option is selected with this bit set to 1, all sync characters except the one immediately preceding the data are stripped from the message. If the 6-bit sync option is selected while in the Bisync mode, this bit is ignored.

The address recognition logic of the receiver is modified in SDLC mode if this bit is set to 1; i.e., only the four most significant bits of WR6 must match the receiver address. This procedure allows the Z-SCC to receive frames from up to 16 separate sources without programming WR6 for each source (if each station address has the four most significant bits in common). The address field in the frame is still eight bits long.

This bit is ignored in SDLC mode if Address Search mode has not been selected.

Bit 0: Receiver Enable

When this bit is set to 1, receiver operation begins. This bit should be set only after all other receiver parameters are established and the receiver is completely initialized. This bit is reset by a channel or hardware reset command, and it disables the receiver.

4.1.5 Write Register 4 (Transmit/Receiver Miscellaneous Parameters and Modes)

WR4 contains the control bits for both the receiver and the transmitter. These bits should be set in the transmit and receiver initialization routine before issuing the contents of WR1, WR3, WR6, and WR7. Bit positions for WR4 are shown in Figure 4-6.

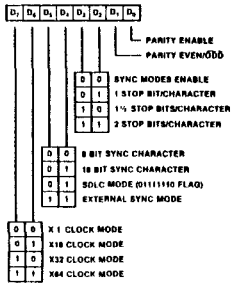


Figure 4-6. Write Register 4

Bits 7 and 6: Clock Rate 1 and 0

These bits specify the multiplier between the clock and data rates. In synchronous modes, the 1X mode is forced internally and these bits are ignored unless External Sync mode has been selected.

1X Mode (00). The clock rate and data rate are the same. In External Sync mode, this bit combination specifies that only the $\overline{SYNC}$ pin can be used to achieve character synchronization.

16X Mode (01). The clock rate is 16 times the data rate. In External Sync mode, this bit combination specifies that only the $\overline{SYNC}$ pin can be used to achieve character synchronization.

32X Mode (10). The clock rate is 32 times the data rate. In External Sync mode, this bit combination specifies that either the $\overline{SYNC}$ pin or a match with the character stored in WR7 will signal character synchronization. The sync character can be either six or eight bits long as specified by the 6-bit/8-bit Sync bit in WR10.

64X Mode (11). The clock rate is 64 times the data rate. With this bit combination in External Sync mode, both the receiver and transmitter are placed in SDLC mode. The only variation from normal SDLC operation is that the $\overline{SYNC}$ pin can be used to start or stop the reception of a frame by forcing the receiver to act as though a flag had been received.

Bits 5 and 4: Sync Modes 1 and 0

These two bits select the various options for character synchronization. They are ignored unless synchronization modes are selected in the stop bits field of this register.

Monosync (00). In this mode, the receiver achieves character synchronization by matching the character stored in WR7 with an identical character in the received data stream. The transmitter uses the character stored in R6 as a time fill. The sync character can be either six or eight bits, depending on the state of the 6-bit/8-bit Sync bit in WR10. If the Sync Character Load Inhibit bit is set, the receiver strips the contents of WR6 from the data stream if received within character boundaries.

Bisync (01). The concatenation of WR7 with WR6 is used for receiver synchronization and as a time fill by the transmitter. The sync character can be 12 or 16 bits in the receiver, depending on the state of the 6-bit/8-bit Sync bit in WR10. The transmitted character is always 16 bits.

SDLC Mode (10). In this mode, SDLC is selected and requires a FLag (01111110) to be written to WR7. The receiver address field should be written to WR6. The SDLC CRC polynomial must also be selected (WR5) in SDLC mode.

External Sync Mode (11). In this mode, the SCC expects external logic to signal character synchronization via the $\overline{SYNC}$ pin. If the crystal oscillator option is selected (in WR11), the internal $\overline{SYNC}$ signal is forced to 0. In this mode, bits B7-B6 of this register select special versions of External Sync mode. In this mode, the transmitter is in Monosync mode using the contents of WR6 as the time fill with the sync character length specified by the 6-bit/8-bit Sync bit in WR10.

Bits 3 and 2: Stop Bits 1 and 0

These bits determine the number of stop bits added to each asynchronous character that is transmitted. The receiver always checks for one stop bit in Asynchronous mode. A Special mode specifies that a Synchronous mode is to be selected. B2 is always set to 1 by a channel or hardware reset to ensure that the SYNC pin is in a known state after a reset.

Synchronous Modes Enable (00). This bit combination selects one of the synchronous modes specified by bits B4, B5, B6, and B7 of this register and forces the 1X Clock mode internally.

1 Stop Bit/Character (01). This bit selects Asynchronous mode with one stop bit per character.

1 1/2 Stop Bits/Character (10). These bits select Asynchronous mode with 1 1/2 stop bits per character. This mode can not be used with the 1X clock mode.

2 Stop Bits/Character (11). These bits select Asynchronous mode with two stop bits per transmitted character and check for one received stop bit.

Bit 1: Parity Even/$\overline{Odd}$

This bit determines whether parity is checked as even or odd. A 1 programmed here selects even parity, and a 0 selects odd parity. This bit is ignored if the Parity bit is not set.

Bit 0: Parity Enable

When this bit is set, an additional bit position beyond those specified in the bits/character control is added to the transmitted data and is expected in the receive data. The Received Parity bit is transferred to the CPU as part of the data unless eight bits per character is selected in the receiver.

4.1.6 Write Register 5 (Transmit Parameter and Controls)

WR5 contains control bits that affect the operation of the transmitter. B2 affects both the transmitter and the receiver. Bit positions for WR5 are shown in Figure 4-7.

Bit 7: Data Terminal Ready

This is the control bit for the $\overline{DTR}/\overline{REQ}$ pin while the pin is in the $\overline{DTR}$ mode (selected in WR14). When set, $\overline{DTR}$ is Low; when reset, $\overline{DTR}$ is High. This bit is ignored when $\overline{DTR}/\overline{REQ}$ is programmed to act as a $\overline{REQUEST}$ pin. This bit is reset by a channel or hardware reset.

Bits 6 and 5: Tx Bits/Character 1 and 0

These bits control the number of bits in each byte transferred to the transmit buffer. Bits sent must be right justified with least significant bits first.

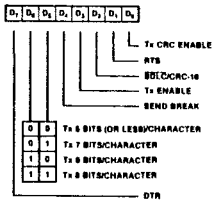


Figure 4-7. Write Register 5

The Five Or Less mode allows transmission of one to five bits per character; however, the CPU should format the data character as shown below in Table 4.2. In the Six or Seven Bits/Character modes, unused data bits are ignored.

Table 4-2. Tx Bits/Character 1 and 0

Tx BITS/ CHAR 1	Tx BITS/ CHAR 0	
0	0	5 or less bits/character
0	1	7 bits/character
1	0	6 bits/character
1	1	8 bits/character

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	0	0	0	0	Sends one data bit
1	1	1	0	0	0	D	D	Sends two data bits
1	1	0	0	0	D	D	D	Sends three data bits
1	0	0	0	D	D	D	D	Sends four data bits
0	0	0	D	D	D	D	D	Sends five data bits

Bit 4: Send Break

When set, this bit forces the TxD output to send continuous 0s beginning with the following transmit clock, regardless of any data being transmitted at the time. This bit functions whether or not the transmitter is enabled. When reset, TxD continues to send the contents of the Transmit Shift register, which might be syncs,

data, or all 1s. If this bit is set while in the X21 mode (Monosync and Loop mode selected) and character synchronization is achieved in the receiver, this bit is automatically reset and the transmitter begins sending syncs or data. This bit can also be reset by a channel or hardware reset.

Bit 3: Transmit Enable

Data is not transmitted until this bit is set, and the TxD output sends continuous 1s unless Auto Echo mode or SDLC Loop mode is selected. If this bit is reset after transmission started, the transmission of data or sync characters is completed. If the transmitter is disabled during the transmission of a CRC character, sync or flag characters are sent instead of CRC. This bit is reset by a channel or hardware reset.

Bit 2: $\overline{\text{SDLC}}$/CRC-16

This bit selects the CRC polynomial used by both the transmitter and receiver. When set, the CRC-16 polynomial is used; when reset, the SDLC polynomial is used. The SDLC/CRC polynomial must be selected when SDLC mode is selected. The CRC generator and checker can be preset to all 0s all 1s, depending on the state of the Preset 1/Preset 0 bit in WR10.

Bit 1: Request to Send

This is the control bit for the $\overline{\text{RTS}}$ pin. When the $\overline{\text{RTS}}$ bit is set, the $\overline{\text{RTS}}$ pin goes Low; when reset, $\overline{\text{RTS}}$ goes High. In the Asynchronous mode with the Auto Enables bit set, $\overline{\text{RTS}}$ goes High only after all bits of the character have been sent and the transmit buffer is empty. In synchonous modes or the Asychronous mode with auto enables off, the pin directly follows the state of this bit. This bit is reset by a channel or hardware reset.

Bit 0: Transmit CRC Enable

This bit determines whether or not CRC is calculated on a transmit character. If this bit is set at the time the character is loaded from the transmit buffer to the Transmit Shift register, CRC is calculated on that character. CRC is not automatically sent unless this bit is set when the transmit underrun exists.

4.1.7 Write Register 6 (Sync Characters or SDLC Address Field)

WR6 is programmed to contain the transmit sync character in the Monosync mode, the first byte of a 16-bit sync character in the External Sync mode. WR6 is not used in asynchronous modes. In the SDLC modes, it is programmed to contain the secondary address field used to compare against the address field of the SDLC Frame. In SDLC mode, the SCC does not automatically transmit the station address at the beginning of a response frame. Bit positions for WR6 are shown in Figure 4-8.

4.1.8 Write Register 7 (SYNC Character or SDLC Flag)

WR7 is programmed to contain the receive sync character in the Monosync mode, a second byte (the last eight bits) of a 16-bit sync character in the Bisync mode, or a Flag character (01111110) in the SDLC modes. WR7 may hold the receive sync character or a flag if one of the special versions of the External Sync mode is selected. WR7 is not used in Asynchronous mode. Bit positions for WR7 are shown in Figure 4-9.

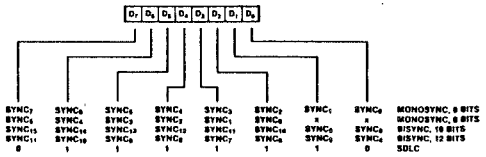


Figure 4-8. Write Register 6

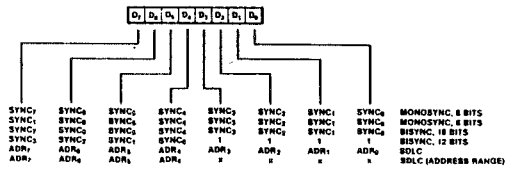


Figure 4-9. Write Register 7

4.1.9 Write Register 8 (transmit buffer)

WR8 is the transmit buffer register.

4.1.10 Write Register 9 (Master Interrupt Control)

WR9 is the Master Interrupt Control register and contains the Reset command bits. Only one WR9 exists in the SCC and can be accessed from either channel. The interrupt control bits can be programmed at the same time as the Reset command because these bits are only reset by a hardware reset. Bit positions for WR9 are shown in Figure 4-10.

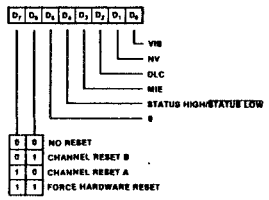


Figure 4-10. Write Register 9

Bit 7-6: Reset Command Bits

Together, these bits select one of the reset commands for the SCC. Setting either of these bits to 1 disables both the receiver and the transmitter in the corresponding channel, forces TxD for that channel marking, forces the modem control signals High in that channel, resets all IPs and IUSs and disables all interrupts in that channel. Four extra PCLK cycles must be allowed beyond the usual cycle time after any of the active reset commands is issued before any additional commands or controls are written to the channel affected. In the Z8530, four extra PCLK cycles must be allowed beyond the usual cycle time before any additional command or controls are written to the SCC.

No Reset (00). This command has no effect. It is used when a write to WR9 is necessary for some reason other than an SCC Reset command.

Channel Reset B (01). Issuing this command causes a channel reset to be performed on Channel B.

Channel Reset A (10). Issuing this command causes a channel reset to be performed on Channel A.

Force Hardware Reset (11). The effects of this command are identical to those of a hardware reset, except that the Shift Right/Shift Left bit is not changed and the MIE, Status High/Status Low and DLC bits take the programmed values that accompany this command.

Bit 5 Not Used

Bit 4: Status High/$\overline{\text{Status Low}}$

This bit controls which vector bits the Z-SCC will modify to indicate status. When set to 1, the Z-SCC modifies bits V6, V5, and V4 according to the table. When set to 0, the Z-SCC modifies bits V1, V2, and V3 according to the table above. This bit controls status in both the vector returned during an interrupt acknowledge cycle and the status in RR2B. This bit is reset by a hardware reset.

Bit 3: Master Interrupt Enable

The Master Interrupt Enable bit is used to globally inhibit Z-SCC interrupts. Setting this bit to 0 has the same effect on the internal interrupt logic as does a low on the IEI pin, except that IEO is not forced Low. When this bit is set to 0, IEO is not forced Low. When this bit is set to 0, IEO follows the state of IEI unless there is an IUS set in the Z-SCC. No IUS can be set after the MIE bit is set to 0. This bit is reset by a hardware reset.

Bit 2: Disable Lower Chain

The Disable Lower Chain bit can be used by the CPU to control the interrupt daisy chain. Setting this bit to 1 forces the IEO pin Low, preventing lower-priority devices on the daisy chain from requesting interrupts. This bit is reset by a hardware reset.

Bit 1: No Vector

The No Vector bit controls whether or not the Z-SCC will respond to an interrupt acknowledge

cycle by placing a vector on the data bus if the Z-SCC is the highest-priority device requesting an interrupt. If this bit is set, no vector is returned; i.e., AD_0-AD_7 remain 3-stated during an interrupt acknowledge cycle, even if the Z-SCC is the highest-priority device requesting an interrupt.

Bit 0: Vector Includes Status

The Vector Includes status bit controls whether or not the Z-SCC will include status information in the vector it places on the bus in response to an interrupt acknowledge cycle. If this bit is set, the vector returned is variable, with the variable field depending on the highest-priority IP that is set. Table 4-3 shows the encoding of the status information. This bit is ignored if the No Vector (NV) bit is set.

Table 4-3. Status Vector Mode Table

V_3	V_2	V_1	Status High/Status Low = 0
V_4	V_5	V_6	Status High/Status Low = 1
0	0	0	Ch B transmit buffer Empty
0	0	1	Ch B External/Status Change
0	1	1	Ch B Receive Character Available
0	1	1	Ch B Special Receive Condition
1	0	0	Ch A transmit buffer Empty
1	0	1	Ch A External/Status Change
1	1	0	Ch A Receive Character Available
1	1	1	Ch A Special Receive Condition

4.1.11 Write Register 10 (Miscellaneous Transmitter/Receiver Control Bits)

WR10 contains miscellaneous control bits for both the receiver and the transmitter. Bit positions for WR10 are shown in Figure 4-11.

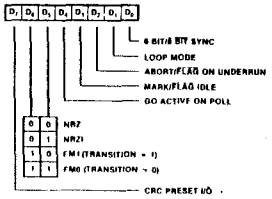


Figure 4-11. Write Register 10

Bit 7: CRC Preset I/$\overline{O}$

This bit specifies the initialized condition of the receive CRC checker and the transmit CRC generator. If this bit is set to 1, the CRC generator and checker are preset to 1. If this bit is set to 0, the CRC generator and checker are preset to 0. Either option can be selected with either CRC polynomial. In SDLC mode, the transmitted CRC is inverted before transmission and the received CRC is checked against the bit pattern 0001110100001111. This bit is reset by a channel or hardware reset. This bit is ignored in Asynchronous mode.

Bits 6 and 5: Data Encoding 1 and 2

These bits control the coding method used for both the transmitter and the receiver, as illustrated in Table 4-4. All of the clocking options are available for all coding methods. The DPLL in the Z-SCC is useful for recovering clocking information in NRZI and FM modes. Any coding method can be used in any 1x mode. A hardware reset forces NRZ mode. Timing for the various modes is shown in Figure 4-12.

Table 4-4. Data Encoding

Data	Data Encoding	Encoding
0	0	NRZ
0	1	NRZI
1	0	FM1 (transition = 1)
1	1	FM0 (transition = 1)

Bit 4: Go Active On Poll

When Loop mode is first selected during SDLC operation, the Z-SCC connects RxD to TxD with only gate delays in the path. The Z-SCC does not go on-loop and insert the 1-bit delay between RxD and TxD until this bit has been set and an EOP received. When the Z-SCC is on-loop, the transmitter cannot go active unless this bit is set at the time an EOP is received. The Z-SCC examines this bit whenever the transmitter is active in SDLC Loop mode and is sending a flag. If this bit is set at the time the flag is leaving the Transmit Shift register, another flag or data byte (if the transmit buffer is full) is transmitted. If the Go Active on Poll bit is not set at this time, the transmitter finishes sending the flag and reverts to the 1-Bit Delay mode. Thus, to transmit only one response frame, this bit should be reset after the first data byte is sent to the Z-SCC but before CRC has been transmitted. If the bit is not reset before CRC is transmitted, extra flags are sent, slowing down response time on the loop. If this bit is reset before the first data is written, the Z-SCC completes the transmission of the present flag and reverts to the 1-Bit Delay mode. After gaining control of the loop, the Z-SCC is not able to transmit again until a flag and another EOP have been received. Though not strictly necessary, it is good practice to set this bit only upon receipt of a poll frame to ensure that the Z-SCC does not go on loop without the CPU noticing it.

In synchronous modes other than SDLC with the Loop Mode bit set, this bit must be set before the transmitter can go active in response to a received sync character.

This bit is always ignored in Asynchronous mode and synchronous modes unless the Loop Mode bit is set. This bit is reset by a channel or hardware reset.

Bit 3: Mark/$\overline{\text{Flag}}$ Idle

This bit affects only SDLC operation and is used to control the idle line condition. If this bit is set to 0, the transmitter sends flags as an idle line. If this bit is set to 1, the transmitter sends continuous 1s after the closing flag of a frame. The idle line condition is selected byte by byte; i.e., either a flag or eight 1s are transmitted. The primary station in an SDLC loop should be programmed for Mark Idle to create the EOP sequence. Mark Idle must be deselected at the beginning of a frame before the first data is written to the Z-SCC, so that an opening flag can be transmitted. This bit is ignored in Loop mode, but the programmed value takes effect upon exiting the Loop mode. This bit is reset by a channel or hardware reset.

Bit 2: Abort/$\overline{\text{Flag}}$ On Underrun

This bit affects only SDLC operation and is used to control how the Z-SCC responds to a transmit underrun condition. If this bit is set to 1 and a transmit underrun occurs, the Z-SCC sends an abort and a flag instead of CRC. If this bit is reset, the Z-SCC sends CRC on a transmit underrun. At the beginning of this 16-bit transmission, the Transmit Underrun/EOM bit is set, causing an External/Status interrupt. The CPU uses this status, along with the byte count from memory or the DMA, to determine whether the frame must be retransmitted. A transmit buffer Empty interrupt occurs at the end of this 16-bit transmission to start the next frame. If both this bit and the Mark/$\overline{\text{Flag}}$ Idle bit are set to 1, all 1s are transmitted after the transmit underrun. This bit should be set after the first byte of data is sent to the Z-SCC and reset immediately after the last byte of data so that the frame will be terminated properly with CRC and a flag. This bit is ignored in Loop mode, but the programmed value is active upon exiting Loop mode. This bit is reset by a channel or hardware reset.

Bit 1: Loop Mode

In SDLC mode, the initial set condition of this bit forces the Z-SCC to connect TxD to RxD and to begin searching the incoming data stream so that it can go on loop. All bits pertinent to SDLC mode operation in other registers must be set before this mode is selected. The transmitter and receiver should not be enabled until after this mode has been selected. As soon as the Go Active On Poll bit is set and an EOP is received, the Z-SCC goes on loop. If this bit is reset after the Z-SCC is on loop, the Z-SCC waits for the next EOP to go off loop.

In synchronous modes, the Z-SCC uses this bit, along with the Go Active On Poll bit, to synchronize the transmitter to the receiver. The receiver should not be enabled until after this mode is selected. The TxD pin is held marking when this mode is selected unless a break condition is programmed. The receiver waits for a sync character to be received and then enables the transmitter on a character boundary. The break condition, if programmed, is removed. This mode works properly with sync characters of six, eight, or 16 bits. This bit is ignored in Asynchonous mode and is reset by a channel or hardware reset.

Bit 0: 6 Bit/$\overline{\text{8 Bit}}$ Sync

This bit is used to select a special case of synchronous modes. If this bit is set to 1 in Monosync mode, the receiver and transmitter sync characters are six bits long instead of the usual eight. If this bit is set to 1 in Bisync mode, the received sync will be twelve bits and the transmitter sync character will remain 16 bits long. This bit is ignored in SDLC and Asynchronous modes but still has effect in the special external sync modes. This bit is reset by a channel or hardware reset.

4.1.12 Write Register 11 (Clock Mode Control)

WR11 is the Clock Mode Control register. The bits in this register control the sources of both the reeive and transmit clcoks, the type of signal on the $\overline{\text{SYNC}}$ and $\overline{\text{RTxC}}$ pins, and the direction of the $\overline{\text{TRxC}}$ pin. Bit positions for WR11 are shown in Figure 4-13.

Bit 7: $\overline{\text{RTxC}}$ XTAL/$\overline{\text{NO XTAL}}$

This bit controls the type of input signal the Z-SCC expects to see on the $\overline{\text{RTxC}}$ pin. If this bit is set to 0, the Z-SCC expects a TTL-compatible signal as an input to this pin. If this bit is set to 1, the Z-SCC connects a high-gain amplifier between the $\overline{\text{RTxC}}$ and $\overline{\text{SYNC}}$ pins in expectation of a quartz crystal being placed across the pins.

The output of this oscillator is available for use as a clocking source. In this mode of operation, the $\overline{\text{SYNC}}$ pin is unavailable for other use. The $\overline{\text{SYNC}}$ signal is forced to 0 internally. A hardware reset forces $\overline{\text{NO XTAL}}$. (At least 20ms should be allowed after this bit is set to allow the oscillator to stabilize.)

Bit 6-5: Receive Clock 1 and 0

These bits determine the source of the receive clock as shown in Table 4-5. They do not interfere with any of the modes of operation in the Z-SCC but simply control a multiplexer just before the internal receive clock input. A hardware reset forces the receive clock to come from the $\overline{\text{RTxC}}$ pin.

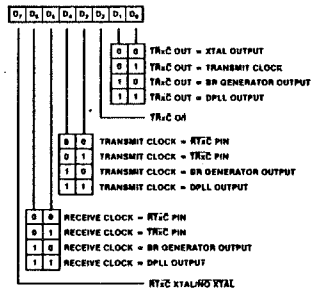


Figure 4-13. Write Register 11

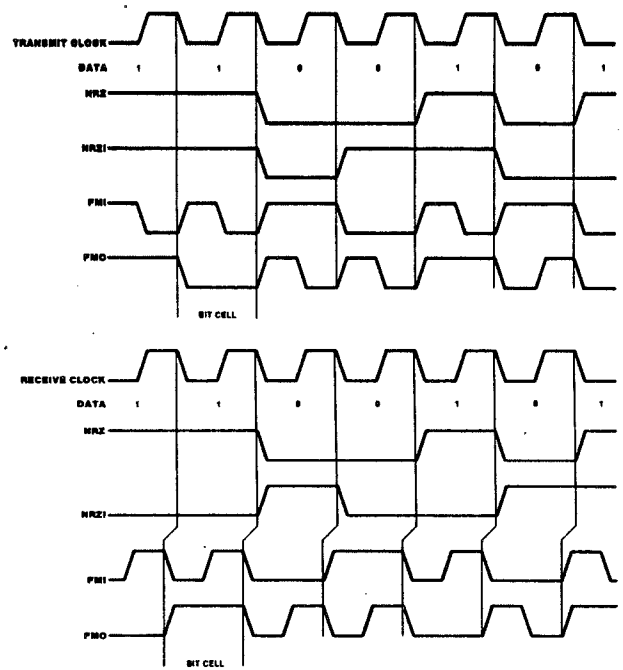


Figure 4-12. NRZ(NRZI)/FM1(FM0) Timing

Table 4-5. Receive Clock Source

Receive Clock 1	Receive Clock 0	
0	0	Receive Clock = $\overline{\text{RTxC}}$ pin
0	1	Receive Clock = $\overline{\text{TRxC}}$ pin
1	0	Receive Clock = BR output
1	1	Receive Clock = DPLL output

Bit 4-3: Transmit Clock 1 and 0

These bits determine the source of the transmit clock as shown in Table 4-6. They do not interfere with any of the modes of operation of the Z-SCC but simply control a multiplexer just before the internal transmit clock input. The DPLL output that may be used to feed the transmitter in FM modes lags by 90 the output of the DPLL used by the receiver. This makes the received and transmitted bit cells occur simultaneously, neglecting delays. A hardware reset selects the $\overline{\text{TRxC}}$ pin as the source of the transmit clock.

Table 4-6. Transmit Clock Source

Transmit Clock 1	Transmit Clock 0	
0	0	Transmit Clock = $\overline{\text{RTxC}}$ pin
0	1	Transmit Clock = $\overline{\text{TRxC}}$ pin
1	0	Transmit Clock = BR output
1	1	Transmit Clock = DPLL output

Bit 2: TRxC O/I

This bit determines the direction of the $\overline{\text{TRxC}}$ pin. If this bit is set to 1, the $\overline{\text{TRxC}}$ pin is an output and carries the signal selected by D1 and D0 of this register. However, if either the receive or the transmit clock is programmed to come from the $\overline{\text{TRxC}}$ pin, $\overline{\text{TRxC}}$ will be an input, regardless of the state of this bit. The $\overline{\text{TRxC}}$ pin is also an input if this bit is set to 0. A hardware reset forces this bit to 0.

Bits 1 and 0: $\overline{\text{TRxC}}$ Output Source 1 and 0

These bits determine the signal to be echoed out of the Z-SCC via the $\overline{\text{TRxC}}$ pin. No signal is produced if $\overline{\text{TRxC}}$ has been programmed as the source of either the receive or the transmit clock. If $\overline{\text{TRxC}}$ O/I (bit 2) is set, these bits are ignored.

If the XTAL oscillator output is programmed to be echoed, the the XTAL oscillator has not been enabled, the $\overline{\text{TRxC}}$ pin goes High. The DPll signal that is echoed is the DPLL signal used by the receiver. Hardware reset selects the XTAL oscillator as the output source.

Table 4-7. Transmit External Control Selection

Output Signal	Output Signal	
0	0	TRxC = XTAL oscillator output
0	1	TRxC = Transmit Clock
1	0	TRxC = BR output
1	1	TRxC = DPLL output (receive)

4.1.13 Write Register 12 (Lower Byte of Baud Rate Generator Time Constant)

WR12 contains the lower byte of the time constant for the baud rate generator. The time constant can be changed at any time, but the new value does not take effect until the next time the time constant is loaded into the down counter. No attempt is made to synchronize the loading of the time constant into WR12 and WR13 with the clock driving the down counter. For this reason, it is advisable to disable the baud rate generator while the new time constant is loaded into WR12 and WR13. Ordinarily, this is done anyway to prevent a load of the down counter between the writing of the upper and lower bytes of the time constant.

The formula for determining the appropriate time constant for a given baud rate is shown below with the desired rate in bits per seconds and the BR clock period in seconds. This formula is derived because the counter decrements from N down to 0-plus-one-cycle for reloading the time constant and is then fed to a toggle flip-flop to make the output a square wave. Bit positions for WR12 are shown in Figure 4-14.

Time constant =
[1/2 * desired rate *BR clock period)] -2

4.1.14 Write Register 13 (Upper Byte of Baud Rate Generator Time Constant)

WR13 contains the upper byte of the time constant for the baud rate generator. Bit positions for WR13 are shown in Figure 4-15.

4.1.15 Write Register 14 (Miscellaneous Control Bits)

WR14 contains some miscellaneous control bits. Bit positions for WR14 are shown in Figure 4-16.

Bits 7 and 5: Digital Phase-Locked Loop Command Bits

These three bits encode the eight commands for the Digital Phase-Locked Loop. A channel or hardware reset disables the DPLL, resets the missing clock latches, sets the source to the $\overline{RTxC}$ pin and selects NRZI mode. The Enter Search Mode command enables the DPLL after a reset.

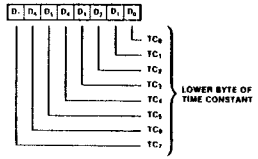


Figure 4-14. Write Register 12

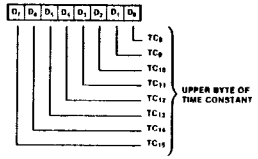


Figure 4-15. Write Register 13

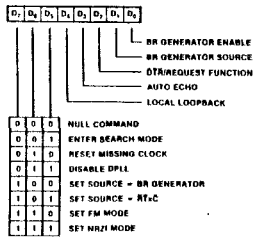


Figure 4-16. Write Register 14

Null Command (000). This command has no effect on the DPLL.

Enter Search Mode (001). Issuing this command causes the DPLL to enter the Search mode, where the DPLL searches for a locking edge in the incoming data stream. The action taken by the DPLL upon receipt of this command depends on the operating mode of the DPLL.

In NRZI mode, the output of the DPLL is High while the DPLL is waiting for an edge in the incoming data stream. After the Search mode is entered, the first edge the DPLL sees is assumed to be a valid data edge, and the DPLL begins the clock recovery operation from that point. The DPLL clock rate must be 32 times the data rate in NRZI mode. Upon leaving the Search mode, the first sampling edge of the DPLL occurs 16 of these 32X clocks after the first data edge and the second sampling edge occurs 48 of these 32X clocks after the first data edge. Beyond this point, the DPLL begins normal operation, adjusting the output to remain in sync with the incoming data.

In FM mode, the output of the DPLL is Low while the DPLL is waiting for an edge in the incoming data stream. The first edge the DPLL detects is assumed to be a valid clock edge. For this to be the case, the line must contain only clock edges; i.e. with FM1 encoding, the line must be continuous 0s. With FM0 encoding the line must be continuous 1s, whereas Manchester encoding requires alternating 1s and 0s on the line. The DPLL clock rate must be 16 times the data rate in FM mode. The DPLL output causes the receiver to sample the data stream in the nominal center of the two halves of the bit cell to decide whether the data was a 1 or a 0. After this command is issued, as in NRZI mode, the DPLL starts sampling immediately after the first edge is detected. (In FM mode, the DPLL examines the clock edge of every other bit cell to decide what correction must be made to remain in sync). If the DPLL does not see an edge during the expected window, the one clock missing bit in RR10 is set. If the DPLL does not see an edge after two successive attempts, the two clocks missing bit in RR10 is set and the DPLL automatically enters the Search mode. This command resets both clock missing latches.

Reset Clock Missing (010). Issuing this command resets the clock missing latches in RR10 and arms the latches to detect the next missing clock edge on the line. The clock missing latches are only used in FM mode.

Disable DPLL (011). Issuing this command disables the DPLL, resets the clock missing latches in RR10, and forces a continuous Search mode state.

Set Source = BR Gen (100). Issuing this command forces the clock for the DPLL to come from the output of the baud rate generator.

Set Source = $\overline{RTxC}$ (101). Issuing this command forces the clock for the DPLL to come from the $\overline{RTxC}$ pin or the crystal oscillator, depending on the state of the XTAL/no XTAL bit in WR11. This mode is selected by a channel or hardware reset.

Set FM Mode (110). This command forces the DPLL to operate in the FM mode and is used to recover the clock from FM or Manchester-encoded data (Manchester is decoded by placing the receiver in NRZ mode while the DPLL is in FM mode).

Set NRZI mode (111). Issuing this command forces the DPLL to operate in the NRZI mode. This mode is also selected by a hardware or channel reset.

Bit 4: Local Loopback

Setting this bit to 1 selects the Local Loopback mode of operation. In this mode, the internal transmitted data is routed back to the receiver, as well as to the TxD pin. The $\overline{CTS}$ and $\overline{DCD}$ inputs are ignored as enables in Local Loopback mode, even if auto enables is selected. (If so programmed, transitions on these inputs still cause interrupts.) This mode works with any Transmit/Receive mode except Loop mode. For meaningful results, the frequency of the transmit and receive clocks must be the same. This bit is reset by a channel or hardware reset.

Bit 3: Auto Echo

Setting this bit to 1 selects the Auto Echo mode of operation. In this mode, the TxD pin is connected to RxD, as in Local Loopback mode, but the receiver still listens to the RxD input. Transmitted data is never seen inside or outside the Z-SCC in this mode, and $\overline{CTS}$ is ignored as a transmit enable. This bit is reset by a channel or hardware reset.

Bit 2: $\overline{DTR}$/Request Function

This bit selects the function of the $\overline{DTR}/\overline{REQ}$ pin. If this bit is set to 0, the $\overline{DTR}/\overline{REQ}$ pin follows the state of the DTR bit in WR5. If this bit is set to 1, the $\overline{DTR}/\overline{REQ}$ pin goes Low whenever the transmit buffer becomes empty and in any of the synchronous mode when CRC has been sent at the end of a message. The request function on the $\overline{DTR}/\overline{REQ}$ pin differs somewhat from the transmit request function available on the $\overline{W}/\overline{REQ}$ pin in that $\overline{REQUEST}$ does not go inactive until the internal operation satisfying the request is complete, which occurs four to five PCLK cycles after the rising edge of $\overline{DS}$, $\overline{READ}$ or $\overline{WRITE}$. If the DMA used is edge-triggered, this difference is unimportant. This bit is reset by a channel or hardware reset.

Bit 1: Baud Rate Generator Source

This bit selects the source of the clock for the baud rate generator. If this bit is set to 0, the baud rate generator clock comes from either the $\overline{RTxC}$ pin or the XTAL oscillator (depending on the state of the XTAL/no XTAL bit). If this bit is set to 1, the clock for the baud rate generator is the Z-SCC's PCLK input. Hardware reset sets this bit to 0, selecting the $\overline{RTxC}$ pin as the clock source for the baud rate generator.

Bit 0: Baud Rate Generator Enable

This bit controls the operation of the baud rate generator. The counter in the baud rate generator in enabled for counting when this bit is set to 1, and counting is inhibited when this bit is set to 0. When this bit is set to 1, change in the state of this bit is not reflected by the output of the baud rate generator for two counts of the counter. This allows the command to be synchronized. However, when set to 0, disabling is immediate. This bit is reset by a hardware reset.

4.1.16 Write Register 15 (External/Status Interrupt Control)

WR15 is the External/Status Source Control register. If the External/Status interrupts are enabled as a group via WR1, bits in this register control which External/Status conditions can cause an interrupt. Only the External/Status conditions that occur after the controlling bit is set to 1 will cause an interrupt. This is true even if an External/Status condition is pending at the time the bit is set. Bit positions for WR15 are shown in Figure 4-17.

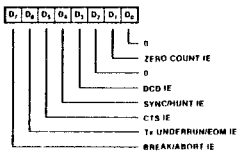


Figure 4-17. Write Register 15

Bit 7: Break/Abort IE

If this bit is set to 1, a change in the Break/Abort status of the receiver causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 6: Tx Underrun/EOM

If this bit is set to 1, a change of state by the Tx Underrun/EOM latch in the receiver causes an External/Status interrupt. This bit is set to 1 by a channel or hardware reset.

Bit 5: CTS IE

If this bit is set to 1, a change of state on the CTS pin causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 4: Sync/Hunt IE

If this bit is set to 1, a change of state on the SYNC pin causes an External/Status interrupt in Asynchronous mode, and a change of state in the Hunt bit in the receiver causes an External/Status interrupt in synchronous modes. This bit is set by a channel or hardware reset.

Bit 3: DCD IE

If this bit is set to 1, a change of state on the DCD pin causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 2: Zero count IE. If this bit is set to 1, an External/Status interrupt is generated whenever the counter in the baud rate generator reaches 0. This bit is set to 0 by a channel or hardware reset.

4.2 READ REGISTERS

Both the Z8030 and the Z8530 versions of the Z-SCC contains nine read registers in each channel. The status of these registers is continually changing and depends on the mode of communication, received and transmitted data, and the manner in which this data is transferred to and from the CPU. The following description details the bit assigments for each registers.

4.2.1 Read Register 0 (Transmit/receive buffer Status and External Status)

Read Register 0 contains the status of the receive and transmit buffers. RR0 also contains the status bits for the six sources of External/Status interrupts. The bit configuration is illustrated in Figure 4-18.

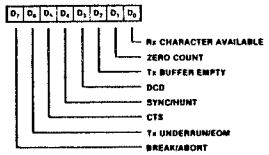


Figure 4-18. Read Register 0

Bit 7: Break/Abort

In the Asynchronous mode, this bit is set when a Break sequence (null character plus framing error) is detected in the receive data stream. This bit is reset when the sequence is terminated, leaving a single null character in the receive FIFO. This character should be read and discarded. In SDLC mode, this bit is set by the detection of an Abort sequence (seven or more 1s), then reset automatically at the termination of the Abort sequence. In either case, if the Break/Abort IE bit is set, an External/Status interrupt is initiated. Unlike the remainder of the External/status bits, both transitions are guaranteed to cause an External/Status interrupt, even if another External/Status interrupt is pending at the time these transitions occur. This procedure is necessary because Abort or Break conditions may not persist.

Bit 6: Tx Underrun/EOM

This bit is set by a channel or hardware reset and when the transmitter is disabled or a Send Abort command is issued. This bit can only be reset by the Reset Tx Underrun/EOM Latch command in WR0. When the Transmit Underrun occurs, this bit is set and causes an External/Status interrupt (if the Tx Underrun/EOM IE bit is set).

Only the 0-to-1 transition of this bit causes an interrupt. This bit is always 1 in Asynchronous mode, unless a reset Tx Underrun/EOM Latch command has been erroneously issued. In this case, the Send Abort command can be used to set the bit to one and at the same time cause an External/Status interrupt.

Bit 5: Clear to Send

If the CTS IE bit in WR15 is set, this bit indicates the state of the CTS pin the last time any

of the enabled External/status bits changed. Any transition on the $\overline{CTS}$ pin while no interrupt is pending latches the state of the $\overline{CTS}$ pin and generates an External/Status interrupt. Any odd number of transitions on the $\overline{CTS}$ pin while another External/Status interrupt is pending also causes an External/Status interrupts condition. If the CTS IE bit is reset, it merely reports the current unlatched state of the $\overline{CTS}$ pin.

Bit 4: SYNC/HUNT

The operation of this bit is similar to that of the $\overline{CTS}$ bit, except that the condition monitored by the bit varies depending on the mode in which the SCC is operating.

When the XTAL oscillator option is selected in asynchronous modes, this bit is forced to 0 (no External/Status interrupt is generated). Selecting the XTAL oscillator in synchronous or SDLC modes had no effect on the operation of this bit. The XTAL oscillator should not be selected in External Sync mode.

In Asynchronous mode, the operation of this bit is identical to that of the CTS status bit, except that this bit reports the state of the $\overline{SYNC}$ pin.

In External sync mode the $\overline{SYNC}$ pin is used by external logic to signal character synchronization. When the Enter Hunt Mode command is issued in External Sync mode, the $\overline{SYNC}$ pin must be held High by the external sync logic until character synchronization is achieved. A High on the $\overline{SYNC}$ pin holds the Sync/Hunt bit in the reset condition.

When external synchronization is achieved, $\overline{SYNC}$ must be driven Low on the second rising edge of the Receive Clock after the last rising edge of the Receive Clock on which the last bit of the receive character was received. Once $\overline{SYNC}$ is forced Low, it is good practice to keep it Low until the CPU informs the external sync logic that synchronization has been lost or that a new message is about to start. Both transitions on the $\overline{SYNC}$ pin cause External/Status interrupts if the Sync/Hunt IE bit is set to 1.

The Enter Hunt Mode command should be issued whenever character synchronization is lost. At the same time, the CPU should inform the external logic that character synchronization has been lost and that the SCC is waiting for $\overline{SYNC}$ to become active.

In the Monosync and Bisync Receive modes, the Sync/Hunt status bit is initially set to 1 by the Enter Hunt Mode command. The Sync/Hunt bit is reset when the SCC established character synchronization. Both transitions cause External/Status interrupts if the Sync/Hunt IE bit is set. When the CPU detects the end of message or the loss of character synchronization, the Enter Hunt Mode command should be issued to set the Sync/Hunt bit and cause an External/Status interrupt. In this mode, the $\overline{SYNC}$ pin is an output, which goes Low every time a sync pattern is detected in the data stream.

In the SDLC modes, the Sync/Hunt bit is initially set by the Enter Hunt Mode command or when the receiver is disabled. It is reset when the opening flag of the first frame is detected by the SCC. An External/Status interrupt is also generated if the Sync/Hunt IE bit is set. Unlike the Monosync and Bisync modes, once the Sync/Hunt bit is reset in SLDC mode, it does not need to be set when the end of the frame is detected. The SCC automatically maintains synchronization. The only way the Sync/Hunt bit can be set again is by the Enter Hunt Mode command or by disabling the receiver.

Bit 3: Data Carrier Detect

If the DCD IE bit in WR15 is set, this bit indicates the state of the $\overline{DCD}$ pin the last time the Enabled External/status bits changed. Any transition on the $\overline{DCD}$ pin while no interrupt is pending latches the state of the $\overline{DCD}$ pin and generates an External/Status interrupt. Any odd number of transitions on the $\overline{DCD}$ pin while another External/Status interrupt is pending also causes an External/Status interrupt condition. If the DCD IE bit is reset, this bit merely reports the current, unlatched state of the $\overline{DCD}$ pin.

Bit 2: Tx Buffer Empty

This bit is set to 1 when the transmit buffer is empty. It is reset while CRC is sent in a synchronous or SDLC mode and while the transmit buffer is full. The bit is reset when a character is loaded into the transmit buffer. This bit is always in the set condition after a hardware or channel reset.

Bit 1: Zero count

If the Zero Count Interrupt Enable bit is set in WR15, this bit is set to 1 and latched when the counter in the baud rate generator reaches the count of zero. If there is no other External/Status interrupt condition pending at the time this bit is set, an External/Status interrupt is

generated. However, if there is another External/Status interrupt pending at this time, no interrupt is initiated until interrupt service is complete. This bit can be reset by the Reset External/Status Interrupts command or by a channel or hardware reset. If the Zero Count IE bit is reset, this bit is always in the zero state.

Bit 0: Rx Character Available

This bit is set to 1 when at least one character is available in the receive FIFO and is reset when the receive FIFO is completely empty. A channel or hardware reset empties the receive FIFO.

4.2.2 Read Register 1

RR1 contains the Special Receive Condition status bits and the residue codes for the I-field in SDLC mode. Figure 4-19 shows the bit positions for RR1.

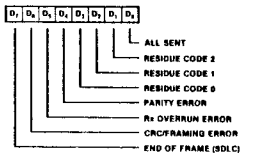


Figure 4-19. Read Register 1

Bit 7: End of Frame (SDLC)

This bit is used only in SDLC mode and indicates that a valid closing flag has been received and that the CRC Error bit and residue codes are valid. This bit can be reset by issuing the Error Reset command. It is also updated by the first character of the following frame. This bit is reset in any mode other than SDLC.

Bit 6: CRC/Framing Error

If a framing error occurs (in Asynchronous mode), this bit is set (and not latched) for the receive character in which the framing error occurred. Detection of a framing error adds an additional one-half bit to the character time so that the framing error is not interpreted as a new Start bit. In Synchronous and SDLC modes, this bit indicates the result of comparing the CRC checker to the appropriate check value. This bit is reset by issuing an Error Reset command, but the bit is never latched. Therefore, it is always updated when the next character is received. When used for CRC error status in Synchronous or SDLC modes, this bit is usually set since most bit combinations, except for a correctly completed message, result in a non-zero CRC.

Bit 5: Receive Overrun Error

This bit indicates that the receive FIFO has overflowed. Only the character that has been written over is flagged with this error, and when the character is read, the Error condition is latched until reset by the Error Reset command. The overrun character and all subsequent characters received until the Error Reset command is issued causes a Special Receive Condition vector to be returned.

Bit 4: Parity Error

When parity is enabled, this bit is set for the characters whose parity does not match the programmed sense (even/odd). This bit is latched so that once an error occurs, it remains set until the Error Reset command is issued. If the parity in Special Condition bit is set, a parity error causes a Special Receive Condition vector to be returned on the character containing the error and on all subsequent characters until the Error Reset command is issued.

Bits 3 - 1: Residue Codes 2, 1, and 0

In those cases in SDLC mode where the received I-Field is not an integral multiple of the character length, these three bits indicate the length of the I-Field and are meaningful only for the transfer in which the end of frame bit is set. This field is set to 011 by a channel or hardware reset and is forced to this state in Asynchronous mode. These three bits can leave this state only if SDLC is selected and a character is received. The codes signify the following (Reference Table 4-8) when a receive character length is eight bits per character.

Table 4-8. I-Field Bit Selection (8 Bits Only)

Residue	Residue	Residue	I-Field Bits in Previous Byte	I-Field Bits in Previous Byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

I-Field bits are right-justified in all cases. If a receive character length other than eight bits is used for the I-Field, a table similar to Table 4-8 can be constructed for each different character length. Table 4-9 shows the residue codes for no residue (the I-Field boundary lies on a character boundary).

Table 4-9. Residue Bits/Character

Bits/Char	Residue	Residue	Residue
8	0	1	1
7	0	0	0
6	0	1	0
5	0	0	1

Bit 0: All Sent

In Asynchronous mode, this bit is set when all characters have completely cleared the transmitter. Most modems contains additional delay in the data path, which requires the modem control signals to remain active until after the data has cleared both the transmitter and the modem. This bit is always set in synchronous and SDLC modes.

4.2.3 Read Register 2

RR2 contains the interrupt vector written into WR2. When the register is accessed in Channel A, the vector returned is the vector actually stored in WR2. When this register is accessed in Channel B, the vector returned includes status information in bits 1, 2, and 3 or in bits 6, 5, and 4, depending on the state of the Status High/Status Low bit in WR9 and independent of the state of the VIS bit in WR9. The vector is modified according to Table 4-3 shown in the explanation of the VIS bit in WR9. If no interrupts are pending, the status is V3, V2, V1 = 011, or V6, V5, V4 = 110. Figure 4-20 shows the bit positions for RR2.

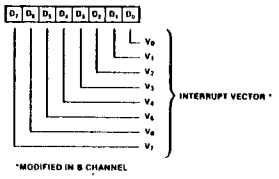


Figure 4-20. Read Register 2

4.2.4 Read Register 3

RR3 is the Interrupt Pending register. The status of each of the Interrupt Pending bits in the Z-SCC is reported in this register. This register exists only in Channel A. If this register is accessed in Channel B, all 0s are returned. The two unused bits are always returned as 0. Figure 4-21 shows the bit positions for RR3.

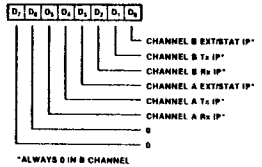


Figure 4-21. Read Register 3

4.2.5 Read Register 8

RR8 is the Receive Data register.

4.2.6 Read Register 10

RR10 contains some miscellaneous status bits. Unused bits are always 0. Bit positions for RR10 are shown in Figure 4-22.

Bit 7: One Clock Missing

While operating in the FM mode, the DPLL sets this bit to 1 when it does not see a clock edge on the incoming line in the window where it expects one. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR10. In the NRZI mode of operation and while the DPLL is disabled, this bit is always 0.

Bit 6: Two Clocks Missing

While operating in the FM mode, the DPLL sets this bit to 1 when it does not see a clock edge in two successive tries. At the same time the DPLL enters the Search mode. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR10. In the NRZI mode of operation and while the DPLL is disabled, this bit is always 0.

Bit 4: Loop Sending

This bit is set to 1 in SDLC loop mode while the

transmitter is in control of the loop, that is, while the SCC is actively transmitting on the loop. This bit is reset at all other times.

Bit 1: On Loop

This bit is set to 1 while the SCC is actually on-loop in SDLC Loop mode. This bit is set to 1 in the X.21 mode (Loop mode selected while in monosync) when the transmitter goes active. This bit is 0 at all other times.

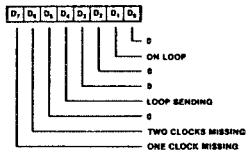


Figure 4-22. Read Register 10

4.2.7 Read Register 12

RR12 returns the value stored in WR12, the lower byte of the time constant for the baud rate generator. Figure 4-23 shows the bit positions for RR12.

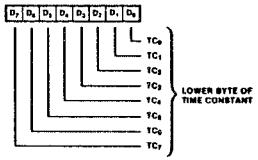


Figure 4-23. Read Register 12

4.2.8 Read Register 13

RR13 returns the value stored in WR13, the upper byte of the time constant for the baud rate generator. Figure 4-24 shows the bit positions for RR13.

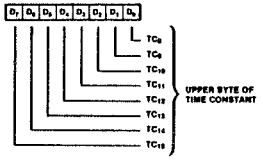


Figure 4-24. Read Register 13

4.2.9 Read Register 15

RR15 reflects the value stored in WR15, the External/Status IE bits. The two unused bits are always returned as 0s. Figure 4-25 shows the bits positions for RR15.

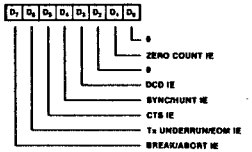


Figure 4-25. Read Register 15

Appendix A
Hardware Information

Absolute Maximum Ratings

Voltages on all inputs and outputs with respect to GND . . . . . -0.3 V to +7.0 V

Operating Ambient Temperature . . . . . As Specified in Ordering Information

Storage Temperature . . . . . -65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

- $+4.75\ V \leq V_{CC} \leq +5.25\ V$
- GND = 0 V
- T_A as specified in Ordering Information

All ac parameters assume a load capacitance of 50 pF max.

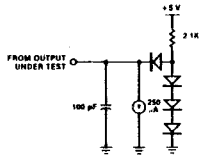


Figure 15. Standard Test Load

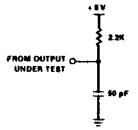


Figure 16. Open-Drain Test Load

DC Characteristics

Symbol	Parameter	Min	Max	Unit	Condition
V_{IH}	Input High Voltage	2.0	$V_{CC}+0.3$	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -250\ \mu A$
V_{OL}	Output Low Voltage		0.4	V	$I_{OL} = +2.0$ mA
I_{IL}	Input Leakage		±10.0	µA	$0.4 \leq V_{IN} \leq +2.4V$
I_{OL}	Output Leakage		±10.0	µA	$0.4 \leq V_{OUT} \leq +2.4V$
I_{CC}	V_{CC} Supply Current		250	mA	

V_{CC} = 5 V ± 5% unless otherwise specified, over specified temperature range.

Capacitance

Symbol	Parameter	Min	Max	Unit	Test Condition
C_{IN}	Input Capacitance		10	pF	Unmeasured Pins Returned to Ground
C_{OUT}	Output Capacitance		15	pF	
$C_{I/O}$	Bidirectional Capacitance		20	pF	

f = 1 MHz, over specified temperature range.

Chapter 1
General Description

1.1 INTRODUCTION

The Z8036 Z-CIO and Z8536 CIO Counter/Timer and Parallel I/O devices are general-purpose peripheral circuits that satisfy most counter/timer and parallel I/O needs encountered in system design, and are therefore helpful in real-time situations and for interrupt control. The Z8036 Z-CIO is designed for systems using the Z-BUS or any other multiplexed Address/Data bus. The Z8536 CIO is designed for CPUs using a nonmultiplexed bus, like that of the Z80 CPU. The differences between the two devices are found in the CPU interface, pin-outs, and timing.

NOTE

All material in this manual referring to "the CIO" applies to both the Z8036 and the Z8536, unless specifically designated by reference to either Z8036 or Z8536. All references to the Z-CIO refer only to the Z8036.

1.2 FEATURES

The Z-CIO and CIO devices satisfy a wide range of applications because of their extensive list of features:

- Two independent 8-bit, double-buffered, bidirectional I/O ports, plus a 4-bit special-purpose I/O port. The I/O ports feature programmable polarity, programmable direction (Bit mode), 1's catchers, and programmable open-drain outputs.
- Four handshake modes, including 3-Wire (like the IEEE-488).
- REQUEST/$\overline{\text{WAIT}}$ signal for high-speed data transfer.
- Flexible pattern-recognition logic, programmable as a 16-vector interrupt controller.
- Three independent 16-bit counter/timers, each with three output duty cycles (pulsed, one-shot, and square-wave) and up to four external access lines (count input, output, gate, and trigger). The counter/timers are programmable as retriggerable or non-retriggerable.
- All registers are read/write. In the Z8036, the registers are directly addressable; in the Z8536, the registers are accessed in two steps.

1.3 OVERVIEW

The CIO (Figure 1-1) consists of a CPU interface, three I/O ports (two general-purpose 8-bit ports and one special-purpose 4-bit port), three 16-bit counter/timers, an interrupt control logic block, and an internal control logic block. A large number of programmable options allow users to tailor the configuration to suit specific applications.

1.3.1 I/O Ports

There are three I/O ports: two general-purpose 8-bit ports (which are linkable into one 16-bit port), and one special-purpose 4-bit port.

1.3.1.1 Ports A and B

The two general-purpose 8-bit I/O ports, Ports A and B (Figure 1-2), are identical, except that Port B can be programmed to provide external access to Counter/Timers 1 and 2. Either port can be programmed to be either a handshake-driven, single- or double-buffered port (input, output, or bidirectional), or a control port with the direction of each bit individually programmable.

Both ports include pattern-recognition logic, which allows interrupt generation when a specific pattern is detected. The pattern-recognition logic can be programmed to make the port function like a priority interrupt controller. Ports A and B can also be linked to form a 16-bit I/O port with handshake.

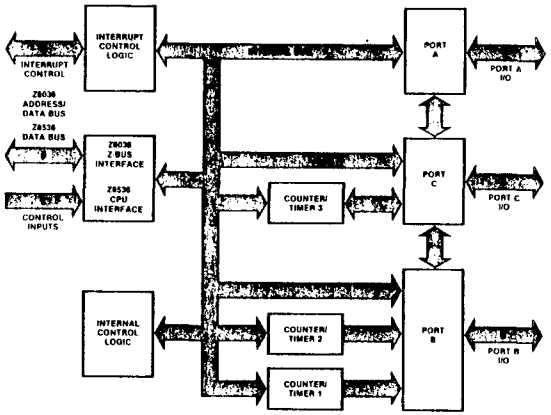


Figure 1-1. Z8036/Z8536 Z-CIO/CIO Block Diagram

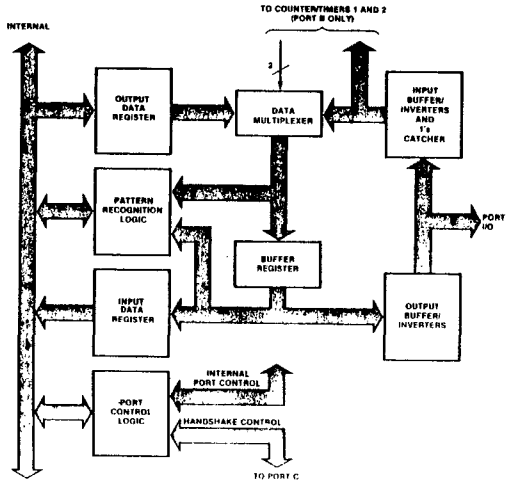


Figure 1-2. Ports A and B Block Diagram

Each port has 12 control and status registers, which control these capabilities. The data path of each port is composed of three internal registers: the Input Data register, the Output Data register, and the Buffer register. The Input Data register is accessed by writing the Port Data register; similarly, the Output Data register is accessed by reading the Port Data register. Two registers, the Mode Specification register and the Handshake Specification register, are used to define the mode of the port and to specify which type of handshake, if any, is to be used. The reference pattern for the pattern-recognition logic is specified by the contents of three registers: the Pattern Polarity register, Pattern Transition register, and Pattern Mask register. The detailed characteristics of each bit path (for example, the direction of data flow or whether a path is inverting or non-inverting) are programmed using the Data Path Polarity register, Data Direction register, and Special I/O Control register.

For each port, the primary control and status bits are grouped in a single register, the Command and Status register. After the port is configured, this is the only register that needs to be accessed frequently. To facilitate initialization, the port logic is designed so that registers associated with an unrequired capability are ignored and do not have to be programmed.

1.3.1.2 Port C

The function of the special-purpose 4-bit port, Port C (Figure 1-3), depends upon the roles of Ports A and B. Port C provides the handshake lines when required by the other two ports. A REQUEST/WAIT line can also be provided by Port C so that transfers by Ports A and B can be synchronized with DMAs or CPUs. Any bits of Port C not used as handshake lines can be used as I/O lines or as external access to Counter/Timer 3.

Since Port C's function is defined primarily by Ports A and B (besides the internal Input Data and Output Data registers, which are accessed as in Ports A and B), only the three bit path registers are needed: the Data Path Polarity register, the Data Direction register, and the Special I/O Control register.

1.3.2 Counter/Timers

The three counter/timers (Figure 1-4) are all identical. Each is composed of a 16-bit down-counter, a 16-bit Time Constant register (which holds the value loaded into the down-counter), a 16-bit Current Count register (used to read the contents of the down-counter), and two 8-bit registers for control and status (the Mode Specification and the C/T Command and Status registers).

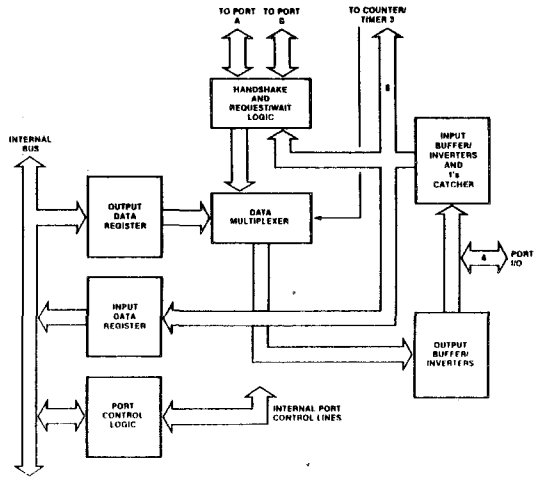


Figure 1-3. Port C Block Diagram

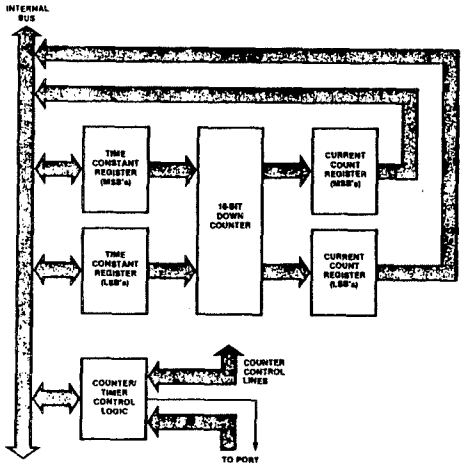


Figure 1-4. Counter/Timer Block Diagram

Up to four port pins (counter input, gate input, trigger input, and counter/timer output) can be used as dedicated external access lines for each counter/timer. Three different counter/timer output duty cycles are available: pulse, one-shot, and square-wave. The operation of the counter/timers can be programmed as either retriggerable or non-retriggerable.

1.3.3 Interrupt Control Logic

The Z8036 and Z8536 interrupt control logic provides the basis for standard Z-BUS and non-Z-BUS interrupt handling capabilities. (See Z-BUS Component Interconnect Summary, Zilog Data Book.) There are five registers (the Master Interrupt Control register, the Current Vector register, and the three Interrupt Vector registers) associated with the interrupt logic. In addition, each Port and Counter/Timer Command and Status register includes three bits associated with the interrupt logic: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE).

Chapter 2
Register Description

2.1 INTRODUCTION

This chapter provides brief descriptions of the command, status and data registers contained in the CIO. Each description includes the register address, the operation of the individual bits, and the state of the register after a reset (hardware or software).

For simplicity, the descriptions assume that the data path polarity of each bit is programmed to be non-inverting. Table 2-1 is a summary of the 48 CIO registers arranged in functional and numerical order. The binary internal addresses are the 6 bits written to an internal Pointer register as the addresses A_0-A_5. The details of the addressing schemes are described in Section 2.2 for the Z8036 and Section 2.3 for the Z8536.

For more complete discussions of the features and modes of operation of the CIO specified by these bits, refer to the appropriate chapters:

Chapter 3 Port Operation
Chapter 4 Counter/Timer Operation
Chapter 5 Interrupt Operation
Chapter 6 Initialization

Table 2-1. Z8036/Z8536 Z-CIO/CIO Register Summary

Internal Address (Binary)	Read/Write	Register Name
$A_5..A_0$		**Main Control Registers**
000000	R/W	Master Interrupt Control
000001	R/W	Master Configuration Control
000010	R/W	Port A Interrupt Vector
000011	R/W	Port B Interrupt Vector
000100	R/W	Counter/Timer Interrupt Vector
000101	R/W	Port C Data Path Polarity
000110	R/W	Port C Data Direction
000111	R/W	Port C Special I/O Control
		Most Often Accessed Registers
001000	*	Port A Command and Status
001001	*	Port B Command and Status
001010	*	Counter/Timer 1 Command and Status
001011	*	Counter/Timer 2 Command and Status
001100	*	Counter/Timer 3 Command and Status
001101	R/W	Port A Data**
001110	R/W	Port B Data**
001111	R/W	Port C Data**

Table 2-1. Z8036/Z8536 Z-CIO/CIO Register Summary--Continued

Internal Address (Binary)	Read/Write	Register Name	
		Counter/Timer Related Registers	
010000	R	Counter/Timer 1 Current Count	MS Byte
010001	R	Counter/Timer 1 Current Count	LS Byte
010010	R	Counter/Timer 2 Current Count	MS Byte
010011	R	Counter/Timer 2 Current Count	LS Byte
010100	R	Counter/Timer 3 Current Count	MS Byte
010101	R	Counter/Timer 3 Current Count	LS Byte
010110	R/W	Counter/Timer 1 Time Constant	MS Byte
010111	R/W	Counter/Timer 1 Time Constant	LS Byte
011000	R/W	Counter/Timer 2 Time Constant	MS Byte
011001	R/W	Counter/Timer 2 Time Constant	LS Byte
011010	R/W	Counter/Timer 3 Time Constant	MS Byte
011011	R/W	Counter/Timer 3 Time Constant	LS Byte
011100	R/W	Counter/Timer 1 Mode Specification	
011101	R/W	Counter/Timer 2 Mode Specification	
011110	R/W	Counter/Timer 3 Mode Specification	
011111	R	Current Vector	
		Port A Specification Registers	
100000	R/W	Port A Mode Specification	
100001	R/W	Port A Handshake Specification	
100010	R/W	Port A Data Path Polarity	
100011	R/W	Port A Data Direction	
100100	R/W	Port A Special I/O Control	
100101	R/W	Port A Pattern Polarity	
100110	R/W	Port A Pattern Transition	
100111	R/W	Port A Pattern Mask	
		Port B Specification Registers	
101000	R/W	Port B Mode Specification	
101001	R/W	Port B Handshake Specification	
101010	R/W	Port B Data Path Polarity	
101011	R/W	Port B Data Direction	
101100	R/W	Port B Special I/O Control	
101101	R/W	Port B Pattern Polarity	
101110	R/W	Port B Pattern Transition	
101111	R/W	Port B Pattern Mask	

* All bits can be read and some bits can be written.
** Also directly addressable in Z8536 using pins A_0 and A_1. (See Table 2-2 and Figures B-1 and B-2.)

2.2 REGISTER ADDRESSING FOR THE Z8036 (Z-CIO)

Register addressing in the Z8036 is accomplished through the use of an internal Pointer register. The Z8036 takes the contents of the multiplexed Address/Data bus and gates a subset of them into the internal Pointer register when $\overline{AS}$ is Low. The internal Pointer register identifies which register will be accessed during the subsequent part of this cycle.

The Z8036 provides two schemes for selecting the desired six of the eight address bits. The scheme to be used is determined by the Right Justify Address (RJA) bit in the Master Interrupt Control register. When RJA = 0, Address bus bits 0 and 7 are ignored, and bits 1 through 6 are decoded for the register address (A_0 derives from AD_1, A_5 derives from AD_6). When RJA = 1, address bits 0 through 5 are decoded for the register address (A_0 derives from AD_0, A_5 derives from AD_5). In the following register descriptions, only six bits are shown for addressing--they represent Address/Data bus bits 5 through 0 or 6 through 1, depending on the state of the RJA bit.

2.3 REGISTER ADDRESSING FOR THE Z8536 (CIO)

The registers in the Z8536 are accessed in a two-step sequence with pins A_0 and A_1 = 1. In the first step, a 6-bit address (the least-significant 6 bits of the Data bus) is written to an internal Pointer register. In the second step, the register identified by the Pointer register is read from or written to.

The data registers for the Z8536 Ports A, B, and C can be accessed by this sequence. The data registers can also be directly addressed by use of device pins A_0 and A_1, as shown in Table 2-2.

Table 2-2. Port Data Register Addressing for the CIO

Data Register	Address Line A_1	Address Line A_0
Port C	0	0
Port B	0	1
Port A	1	0
Control	1	1

The Z8536 contains a state machine which determines if accesses with A_0 and A_1 = 1 (see Table 2-2) are to the Pointer register or to an internal control register. (Refer to Figure 2-1 for the following discussion.) Reads in State 0 leave the state machine in State 0. Writes to the Z8536 in State 0 update the Pointer register and put the state machine into State 1. Accesses in State 1 are to the register addressed by the Pointer register, and cause the state machine to revert to State 0. State changes occur only when pin A_0 = pin A_1 = 1. Direct accesses of the data registers have no effect on state machine operation.

After any control read operation (pin A_0 = pin A_1 = 1), the state machine is in State 0 (the next control access is to the Pointer register). This can be used to force the state machine into a known state. Control reads in State 0 return the contents of the last register pointed to. Therefore, a register can be read continuously without writing to the pointer. While the Z8536 is in State 1 (next control access is to the register pointed to), many internal operations are suspended, Interrupt Pending (IP) cannot be set, and internal status is frozen. Therefore, to minimize interrupt latency and to allow continuous status updates, the Z8536 should not be left in State 1.

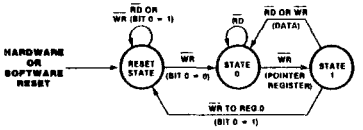


Figure 2-1. Z8536 State Machine Operation

2.4 MASTER CONTROL REGISTERS

The Master Control registers consist of the Master Interrupt Control register and Master Configuration Control register. These registers provide primary controls for the interrupt logic, port and counter/timer enable bits, port and counter/timer link bits and the RESET bit.

2.4.1 Master Interrupt Control Register

The Master Interrupt Control register contains the primary control bits for the interrupt control logic. When the device is reset all bits in all device registers are forced to 0 except RESET, which is set to 1. The RJA bit (D_1) is only

applicable to the Z8036 Z-CIO. All bits in the Master Interrupt Control register are Read/Write.

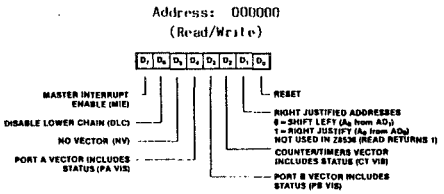


Figure 2-2. Master Interrupt Control Register

Master Interrupt Enable--MIE (D_7). Clearing this bit to 0 inhibits the device from requesting an interrupt or responding to an Interrupt Acknowledge. Its effect is the same as pulling the Interrupt Enable In (IEI) input Low, except that the daisy-chain is left intact. A 1 in this bit allows the interrupt logic to operate normally.

The MIE bit also affects whether or not status is included when reading interrupt vectors. If MIE = 0, interrupt vector reads do not include status. If MIE = 1, vector reads always include status, independent of the state of the corresponding Vector Includes Status (VIS) bit.

Disable Lower Chain--DLC (D_6). If DLC is set to 1, the Interrupt Enable Out (IEO) output of the device is forced Low, disabling interrupts from all lower-priority devices on the daisy-chain. When DLC is 0, IEO operates normally.

No Vector--NV (D_5). When NV is set to 1, the device is inhibited from outputting an interrupt vector during an Interrupt Acknowledge cycle. This allows the vector to be provided by external hardware. It has no effect on the setting of the Interrupt Under Service (IUS) bit. If NV is written with 0, the interrupt vector is output as usual.

Port A Vector Includes Status--PA VIS (D_4). If this bit is 0 when a Port A interrupt is acknowledged, the interrupt vector that is output is the unmodified content of the Port A Interrupt Vector register. If this bit is written with a 1, the Port A base vector is modified to include status, which indicates the cause of the interrupt. Vector modification is described in Section 5.3.4. The state of this bit has no effect on the value returned when the Port A Interrupt Vector register is read. When reading the vector, the MIE bit determines if status is included in the vector, (that is, no status is included if MIE = 0).

Port B Vector Includes Status--PB VIS (D_3). This bit controls whether or not the Port B interrupt vector includes status. It operates the same way that the PA VIS bit controls the Port A interrupt vector.

Counter/Timer Vector Includes Status--CT VIS (D_2). This bit controls whether or not the base interrupt vector shared by the three counter/timers includes status. It operates the same way that the other two VIS bits (PA VIS and PB VIS) operate.

Right Justified Address--RJA (D1). (Z8036 only). When this bit is 0, the register address is shifted left one bit (see Section 2.2). Address bit A_0 is derived from Address/Data bus bit AD_1. When set to 1, the address is right justified, (for example, $A_0 = AD_0$).

The Z8536 does not use RJA--this bit is always set to 1, which causes the address to always be right-justified.

RESET--(D_0). Setting the RESET bit to 1 by a software write resets the device. The bit can also be set by a hardware reset on the Z8036 by forcing Address Strobe ($\overline{AS}$) and Data Strobe ($\overline{DS}$) Low simultaneously; or on the Z8536 by forcing Read ($\overline{RD}$) and Write ($\overline{WR}$) Low simultaneously. While RESET is 1, reads of all other registers will be 0 and writes to other registers are ignored. This bit is cleared only by writing a 0 to the RESET bit (see Sections 6.2 and 6.3).

2.4.2 Master Configuration Control Register

The Master Configuration Control register contains the control bits used to enable different sections of the device after they are initially configured, as well as the bits used to link the ports together and the timers together. All bits are cleared to 0 by resetting the device. The register is read/write.

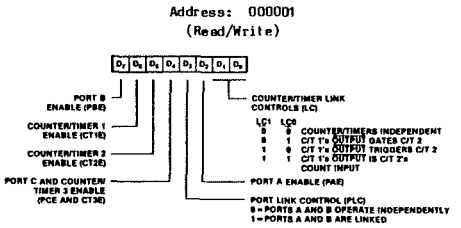


Figure 2-3. Master Configuration Control Register

Port B Enable--PBE (D_7). This bit, when set to 1, allows Port B to operate normally. When cleared to 0, it inhibits the Port B logic from issuing an interrupt request (its IP cannot be set); however, if IP was already set, clearing PBE does not clear IP. While cleared to 0, PBE inhibits READY/$\overline{WAIT}$ assertion, holds all 1's catchers in a transparent condition, and forces the Port B I/O lines into a high-impedance state. The purpose of this bit is to allow Port B to be configured initially without setting its IP erroneously or having its I/O lines go low-impedance until it is safe to do so.

Counter/Timer 1 Enable--CT1E (D_6). When cleared to 0, Counter/Timer 1 is put into an initialized state: its IP cannot be set (however, if IP was already set, clearing CT1E does not clear IP), the Count In Progress (CIP) flag is cleared, Read Counter Control (RCC) is forced to 0, and all trigger inputs are ignored. Setting CT1E to 1 allows the counter/timer to function normally.

Counter/Timer 2 Enable--CT2E (D_5). The CT2E bit performs the same function for Counter/Timer 2 that CT1E performs for Counter/Timer 1.

Port C and Counter/Timer 3 Enable--PCE and CT3E (D_4). This bit enables both Port C and Counter/Timer 3. The function is the same as D_7 (PBE) and D_6 (CT1E) for Port B and Counter/Timer 1, respectively. In addition, while this bit is cleared to 0, the handshake logic for Ports A and B is forced into an idle state and the internal Acknowledge Input ($\overline{ACKIN}$) signal is forced High. This allows the start-up of handshake operations to be precisely controlled.

Port Link Control--PLC (D_3). When PLC is set to 1, Ports A and B are linked to form a 16-bit port. In this mode, only the Port A Handshake Specification and Command and Status registers are used. Port B must be specified as a bit port and its pattern match capability must be disabled. Also, when linked, the Port B data register must be read or written before the Port A data register. A 0 in the PLC bit allows the ports to operate independently. If the ports are to be linked, this bit must be set before the ports are enabled.

Port A Enable--PAE (D_2). The Port A Enable bit performs the same function for Port A that the Port B Enable bit (D_7) performs for Port B.

Counter/Timer Link Controls--LC_1 & LC_0 (D_1 & D_0). These two bits specify if and how Counter/Timers 1 and 2 are linked. The Counter/Timers must be linked before they are enabled. The various configurations are shown in Table 2-3.

Table 2-3. Counter/Timer Link Controls

LC_1	LC_0	Configuration
0	0	Counter/Timers are independent
0	1	Counter/Timer 1's output (inverted) gates Counter/Timer 2
1	0	Counter/Timer 1's output (inverted) triggers Counter/Timer 2
1	1	Counter/Timer 1's output (inverted) is Counter/Timer 2's count input (Counter/Timer 2's External Count Enable* bit must be cleared to 0)

* (See Section 2.9.1 for description of External Count Enable bit.)

2.5 PORT SPECIFICATION REGISTERS

Each of these registers define the port operating mode, specify the type of handshake (if one is used), and contain the command and status bits used to affect data transfers of its port. There is a set of Port Specification registers for both Port A and Port B.

2.5.1 Port Mode Specification Registers

Each Port Mode Specification register contains the bits that define the operating mode of its port and specify the operation of pattern match logic of the port. A reset forces all bits to be cleared to 0. All bits are read/write.

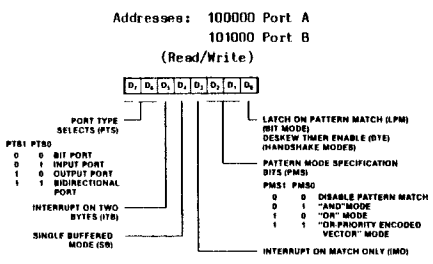


Figure 2-4. Port Mode Specification Registers

Port Type Selects--PTS_1 & PTS_0 (D_7 & D_6). The port type is specified by these two bits, as shown in Table 2-4.

Table 2-4. Port Type Selects

PTS_1	PTS_0	Port Type
0	0	Bit port (no handshake)
0	1	Input port with one of four handshakes*
1	0	Output port with one of four handshakes*
1	1	Bidirectional port with one of two handshakes**

* The four handshakes are: Interlocked, Pulsed, Strobed, or 3-wire.

** The two handshakes are: Interlocked or Strobed.

Interrupt on Two Bytes--ITB (D_5). For a port programmed with handshake, this bit indicates when an interrupt should be requested. If ITB is set to 1, IP is set when two bytes of data can be read or written. For an input port, IP is set when both the Input Data register and Buffer register are full. For an output port, IP is set when both the Output Data and Buffer register are empty. When ITB is cleared to 0, IP is set whenever a single byte of data is available to be moved (the Input Data register is full or the Output Data register is empty). This bit must always be cleared to 0 for ports specified either as bit ports, single-buffered ports (SB = 1), or bidirectional ports.

ITB also affects the operation of the Request line. When ITB = 0, the Request line will go active as soon as the device is ready for a data transfer. For input ports, the Request line will go High when the Input Data register is full. If ITB = 1, both the Buffer register and Input Data register must be full for Request to go active. For output ports with ITB = 0, the Request line will go High when the Output Data register is empty. If ITB = 1, the Request line will go High when both the Buffer register and Output Data registers are empty. In either case, the Request line will stay active as long as a byte is available to be read or written.

Single Buffer--SB (D_4). For a port programmed with handshake, this bit specifies if the port should be single- or double-buffered. When SB is cleared to 0, the port is double-buffered. When SB is set to 1 (ITB must be 0), the port is single-buffered: an input byte is loaded into both the Buffer and Input Data registers, or an output byte is loaded into both the Output Data and Buffer registers. This bit must always be cleared to 0 for bit ports.

Interrupt on Match Only--IMO (D_3). For ports with handshake (when this bit is set to 1) an interrupt will be generated only when the data moved into the Input Data register or out of the Output Data register matches the pattern specification. When cleared to 0, the port operates normally. The purpose of this bit is to allow the generation of CPU interrupts only on bytes which match the pattern specification. It is useful, for example, when the data is being moved under Direct Memory Access (DMA) control. IMO must be 0 if either SB or ITB are set to 1 or if the port is a bit port.

Pattern Mode Specification Bits--PMS_1 & PMS_0 (D_2 & D_1). These two bits define the mode of operation of the pattern match logic, as is shown in Table 2-5.

Table 2-5. Pattern Mode Specification Bits

PMS_1	PMS_0	Pattern Mode
0	0	Disable Pattern Match
0	1	AND Mode
1	0	OR Mode
1	1	OR-Priority Encoded Vector mode

The OR-Priority Encoded Vector mode must not be specified for ports configured as bit ports with the Latch on Pattern Match (LPM) bit set to 1 or for ports with handshake.

Latch On Pattern Match--LPM, or Deskew Timer Enable--DTE (D_0). The LPM/DTE bit is a dual-function bit. The LPM function is active when the port is specified in bit mode; the DTE function is active when the port is specified as an output port with handshake.

The LPM bit, when set to 1, causes the port to latch the input data present at the port when a pattern match is detected. If LPM is 0, pattern matches are still detected, but the data read back from the port follows the port pins.

The DTE bit, when set to 1, activates the deskew timer to perform delay functions as set in the Port Handshake Specification register. When cleared to 0, no delay is activated (see Section 3.4.3.2).

LPM/DTE must be cleared to 0 for input ports with handshake or for bit ports whose pattern match logic is in the OR-Priority Encoded Vector mode.

2.5.2 Port Handshake Specification Registers

Each of the Port Handshake Specification registers contain the bits that specify the type of handshake, the utilization of the REQUEST/$\overline{\text{WAIT}}$ line, and the Deskew Timer Time Constant for ports programmed with handshake. These bits are ignored if the port is a bit port. A RESET forces all bits to 0. All bits are read/write.

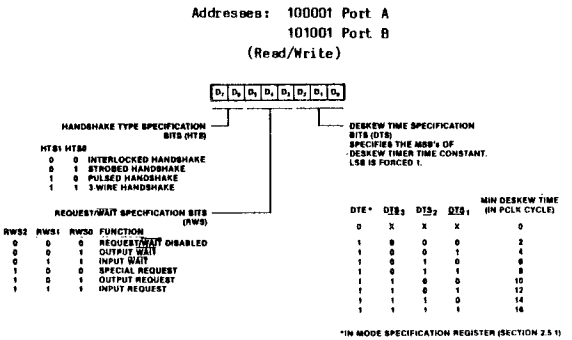


Figure 2-5. Port Handshake Specification Registers

Handshake Type Specification Bits--HTS_1 & HTS_0 (D_7 & D_6). These two bits specify the handshake type that a port with handshake will use, as is shown in Table 2-6.

Table 2-6. Handshake Type Specification Bits

HTS_1	HTS_0	Handshake Type
0	0	Interlocked Handshake
0	1	Strobed Handshake
1	0	Pulsed Handshake
1	1	3-Wire Handshake

The Pulsed Handshake and the 3-Wire Handshake must not be specified for bidirectional ports. Only one port at a time can use the Pulsed Handshake configuration. If one port uses the 3-Wire Handshake, the other port must be a bit port.

REQUEST/$\overline{WAIT}$ Specification Bits--RWS_2-RWS_0 (D_5-D_3). These three bits specify the utilization of the REQUEST/$\overline{WAIT}$ line, as is shown in Table 2-7.

Table 2-7. REQUEST/WAIT Specification Bits

RWS_2	RWS_1	RWS_0	Function
0	0	0	REQUEST/$\overline{WAIT}$ Disabled
0	0	1	Output $\overline{WAIT}$
0	1	1	Input $\overline{WAIT}$
1	0	0	Special REQUEST
1	0	1	Output REQUEST
1	1	1	Input REQUEST

If a port uses the REQUEST/$\overline{WAIT}$ capability, the other port must be programmed as a bit port, because three pins of Port C are required. (See Table 3-1.)

Deskew Time Specification Bits--DTS_3 Through DTS_1 (D_2-D_0). These three bits are the most significant bits of the Deskew Timer Time Constant. They specify the minimum amount of deskew time to be provided for output data. They define the minimum number of Peripheral Clock (PCLK) cycles of delay, 0 to 16, between the output of a new byte of data and the handshake logic indicating that new data is available ($\overline{DAV}$ falling). This logic is particularly useful in systems where large amounts of skew can exist between the data and the handshake signals or where the receiver of the data has a large set-up time requirement. The amount of deskew provided is shown in Figure 2-5.

NOTE

0 PCLK cycles deskew time is obtained by not enabling the deskew timer (DTE = 0 in the Port Mode Specification register).

2.5.3 Port Command and Status Registers

Each of these registers contain the primary command and status bits for its port. Other than the data bits themselves, these are the bits most often accessed in normal port operation. A reset forces ORE to 1 and all other bits to 0. All bits are readable and four are writeable.

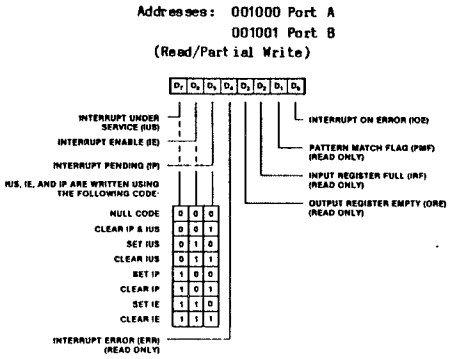


Figure 2-6. Port Command and Status Registers

Interrupt Under Service--IUS (D_7). This status bit is automatically set to 1 if its corresponding IP is the highest-priority interrupt request pending when an Interrupt Acknowledge sequence takes place. It can also be set directly by CPU command. While the IUS is set, the same and lower priority sources of interrupt are prohibited from requesting interrupts via the internal and external daisy-chains. The IUS can be cleared to 0 only by CPU command. This bit is read/write. It is changed by writing to the Command and Status register of the port using the code shown in Figure 2-6.

Interrupt Enable--IE (D_6). This bit enables or disables the port's interrupt logic. While IE is cleared to 0, the port is unable to request an interrupt or to respond to an Interrupt Acknowledge. The normal operation of IP or IUS is not affected--the IP is simply masked off from the rest of the device. A 1 in IUS still affects the interrupt daisy-chain. If IE is programmed to be 1, the interrupt logic operates normally. This bit is read/write. It is changed by writing to the Command and Status register of the port using the code shown in Figure 2-6.

Interrupt Pending--IP (D_5). IP is a status bit which, when set to 1, indicates that the port requires servicing due to a pattern match, a handshake, or an error. It is set to 1 by the port logic (or by the CPU command). If IE is also 1 and no higher-priority interrupts are under service, then the $\overline{INT}$ line is pulled Low to request an interrupt. It is cleared to 0 either automatically or by a CPU command, depending on port configuration. It is changed by writing to the Port Command and Status register using the code shown in Figure 2-6.

Interrupt Error--ERR (D_4). This status bit is automatically set to 1 along with IP when, for a bit port with pattern match enabled, a second match occurs before a previous match is acknowledged (IP is still set). If the port Interrupt On Error (IOE) bit is 0, errors are ignored and this bit is held at 0. This bit can be cleared only by clearing the corresponding IP. This bit is a read-only bit; writes to it are ignored.

Output Data Register Empty--ORE (D_3). ORE is a status bit used in conjunction with ports, specified either as output or bidirectional ports, to indicate whether or not the Output Data register is full. It is set to 1 when a byte of data is moved out of the Output Data register as part of an output handshake. The bit can only be cleared by writing to the data register. As a bit port, ORE is forced to 1 unless OR-PEV pattern match mode is specified--in which case, ORE is forced to 0. This bit is a read-only bit; writes to it are ignored. RESET empties the Output Data register, so after a RESET the ORE is set.

Input Data Register Full--IRF (D_2). IRF is a status bit used in conjunction with ports, specified either as input or bidirectional ports, to indicate whether or not the Input Data register is full. It is automatically set to 1 when a new byte of data is available to be read as the result of an input handshake. The bit can only be cleared by reading the port data register, thus cleared by reading the port data register, thus "emptying" the Input Data register and forcing the bit to 0. If the port is an output port or a bit port, this bit is always forced to 0. IRF is a read-only bit; writes to it are ignored.

Pattern Match Flag--PMF (D_1). The PMF is a status bit set to 1 when a pattern match is detected. If the port is a bit port, PMF is not latched. It reflects the state of the pattern match logic just before it is read. For the Z8036, it is updated each $\overline{AS}$. For the Z8536, it is updated every second PCLK cycle while the CIO is in State 0 (See Section 2.3). For ports with handshake, the state of the PMF is updated each time a byte of data is moved into the Input Data register or out of the Output Data register. If the port pattern match logic is not enabled (PMS_1 = PMS_0 = 0), the PMF is forced to 0. This is a read-only bit. Writes to it are ignored.

Interrupt on Error--IOE (D_0). While IOE is cleared to 0, error conditions in bit ports using pattern-recognition logic (a second match before a previous match is acknowledged) are ignored. However, if IOE is 1, such errors will cause IP to be set and will halt normal operation of the port until the error condition is dealt with. This bit has no meaning for ports with handshake and must be cleared to 0.

2.6 BIT PATH DEFINITION REGISTERS

The Bit Path Definition registers are used to specify the details of each bit path of each port. They define:

- whether a bit path is inverting or non-inverting
- if an output is normal or open-drain
- if a bit port input has a 1's catcher inserted in its path
- which direction the data is flowing for each bit of a bit port

Each port has a set of these registers. The four most-significant bits of each register do not exist in the registers associated with Port C (writes are ignored, reads return 1s).

2.6.1 Data Path Polarity Registers

The Data Path Polarity registers each define whether the bits in its port are inverting or non-inverting on a bit-by-bit basis.

Addresses: 100010 Port A
101010 Port B
000101 Port C
(4 LSBs only)
(Read/Write)

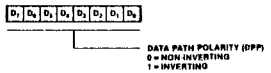


Figure 2-7. Data Path Polarity Registers

A 0 in a particular bit position of this register specifies the corresponding bit path of the port as non-inverting (that is, a High level at the port pin is 1). If a bit in this register is written with 1, the data path is programmed inverting (that is, a Low level at the pin is 1). A reset clears all bits to 0 (the port is non-inverting). The bits are read/write.

2.6.2 Data Direction Registers

Each of the Data Direction registers define the direction of data flow for the individual bits of its port if configured as a bit port. The state of this register is ignored for ports with handshake.

Addresses: 100011 Port A
101011 Port B
000110 Port C
(4 LSBs only)
(Read/Write)

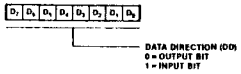


Figure 2-8. Data Direction Registers

A 0 in a bit position of this register specifies the corresponding bit of the port as an output bit, while a 1 specifies it as an input. The value programmed in this register for Ports A and B is overridden if the port is one with handshake. An input bit specification is overridden for bits in Port C used as outputs for handshake signals or a REQUEST/$\overline{\text{WAIT}}$ line. Bits used as handshake inputs must be specified as inputs.

A reset forces all bits in these registers to 0. All bits are read/write.

2.6.3 Special I/O Control Registers

Each of the Special I/O Control registers is a dual-function register which specifies special characteristics about its port's data path. Its exact function depends on the direction of data flow defined for the path.

Addresses: 100100 Port A
101100 Port B
000111 Port C
(4 LSBs only)
(Read/Write)

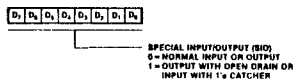


Figure 2-9. Special I/O Control Registers

If a bit is an input bit, a 1 in this register's corresponding bit position invokes a 1's catcher. A 1's catcher functions by automatically latching a 1 if its input goes to 1. It is cleared only by writing a 0 to the Input Data register. A 1's catcher is inserted into the input path after the bit's invert/non-invert logic. If the bit is programmed 0, it is a normal input bit. The 1's catcher is available only for input bit port bits.

If a bit is an output bit, a 0 in the corresponding bit position of this register specifies the output as a normal output with both a pull-up and a pull-down transistor. A 1 in this register defines the output as open-drain; no pull-up transistor is provided. The value programmed in this register applies to all output modes, independent of utilization.

A reset forces all bits to 0. All bits are read/write.

2.7 PATTERN DEFINITION REGISTERS

These registers collectively specify the match pattern for the port. As the registers must be taken together to define the pattern, they are described differently than the previous registers.

Addresses: 100101 Port A
101101 Port B
(Read/Write)

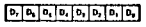


Figure 2-10. Pattern Polarity Registers

Addresses: 100110 Port A
101110 Port B
(Read/Write)

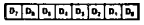


Figure 2-11. Pattern Transition Registers

Addresses: 100111 Port A
101111 Port B
(Read/Write)

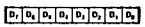


Figure 2-12. Pattern Mask Registers

A reset forces all of these registers to 0. All are read/write.

The pattern specification for each bit is defined as shown in Table 2-8.

The pattern specified by the Pattern Definition registers is a logical (not a physical) specification--this concept is important in understanding the interaction between the pattern match logic and the invert/non-invert logic. An example which shows the logical (as opposed to physical) nature of the specification is: a High level (V_{CC}) on an input pin programmed as inverting matches a 0 specification. Similarly, an output written with a 1 matches a 1 specification even if it is programmed inverting and the output pin is at a Low voltage level.

If the port is programmed as a port with handshake, or if the pattern match mode is OR-Priority Encoded Vector, the transition detection patterns should not be specified (PTn should be set to 0). If the AND mode is specified, no more than one bit should be specified to detect transitions.

2.8 PORT DATA REGISTERS

Ports A and B each have a data path that is composed of three registers: an Input Data register, an Output Data register, and a Buffer register (See Figure 1-2). Output data written to the data register is stored in the Output Data register. Reading the data register returns the contents of the Input Data register. The Buffer register is used to buffer the input and output data if the port is configured as a port with handshake. If so enabled, it is used by the bit port to latch data when a pattern match is detected.

Addresses: 001101 Port A
001110 Port B
(Read/Write)

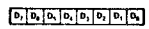


Figure 2-13. Port A and B Data Registers

Table 2-8. Pattern Specification Definition

Pattern Mask Register$_n$	Pattern Transition Register$_n$	Pattern Polarity Register$_n$	Pattern Specification
0	0	0	Bit Masked Off (X)
0	1	0	Any Transition (⤫)
1	0	0	Zero (0)
1	0	1	One (1)
1	1	0	One to Zero Transition (↘)
1	1	1	Zero to One Transition (↗)

The individual bits of the port data registers map directly onto the port I/O pins (bit 0 of the Port A Data register corresponds to the PA_0 pin, etc.).

The Port C Data register consists of two registers: an Input Data register and an Output Data register (see Figure 1-3). Output data written to the data register is stored in the Output Data register. Reading the data register returns the contents of the Input Data register. Because Port C is only four bits wide, the four least-significant bits of an 8-bit register are used for the Port C Data register. The four most-significant bits are used as a write protect mask for the four least-significant bits (bit D_7 is the write protect mask for bit D_3, etc.), as shown in Figure 2-14. Writing a 0 to the write protect mask bit enables writing to the corresponding bit in Port C. Writing a 1 inhibits writing the corresponding bit in Port C. Reading Port C always returns 1's in the upper four bits.

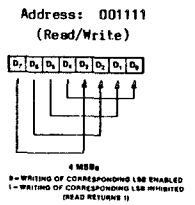


Figure 2-14. Port C Data Register

Details of the operation of these registers in the various configurations are given in Chapter 3. The data registers in the Z8536 can also be directly accessed by pin A_0 and pin A_1 (see Table 2-2).

NOTE

A reset does not effect the contents of the data registers.

2.9 COUNTER/TIMER CONTROL REGISTERS

Each counter/timer has a set of Counter/Timer Control registers, which perform several functions for the counter/timers:

- specify the mode of operation
- monitor the status
- provide control
- allow access to the down-counter so that it can be preset and read

2.9.1 Counter/Timer Mode Specification Registers

Each Counter/Timer Mode Specification register contains the bits that define its counter/timer's mode of operation and specify the external control and status lines to provide for it. A reset forces all bits to 0. All bits are read/write.

Addresses: 011100 Counter/Timer 1
011101 Counter/Timer 2
011110 Counter/Timer 3
(Read/Write)

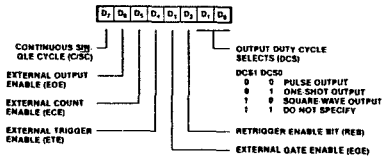


Figure 2-15. Counter/Timer Mode Specification Registers

Continuous/Single Cycle--C/$\overline{SC}$ (D_7). If C/$\overline{SC}$ is set to 1, then each time the down-counter reaches the count of 1, the time constant value is reloaded (on the next count) and the countdown sequence is repeated. If C/$\overline{SC}$ is 0 when the count of 1 is encountered (and, for square-wave outputs, if the output is a 1), the counter is allowed to count down to 0 and the countdown sequence is terminated.

External Output Enable--EOE (D_6). By programming this bit to be 1, the output of the counter/timer is provided on the I/O line of the port associated with that particular counter/timer (see Table 4-1). This bit should not be set to 1 unless the corresponding bit is available, (it is not being used as part of an input, output, or bidirectional port, or it is not being used as a handshake or REQUEST/$\overline{WAIT}$ line). The bit must be programmed to be an output bit in the Data Direction register of its port.

External Count Enable--ECE (D_5). When ECE is set to 1, the counter/timer is put into the counter mode. The I/O line of the port associated with the counter/timer (Table 4-1) is used as an external counter input. On each rising edge of the count input (when the data path is specified non-inverting), the down-counter is decremented. The bit must be available and it must be specified to be an input. (Even if the port bit is programmed as an output bit, the port pin [if enabled] is used as the counter/timer input, allowing the CPU to write this input directly.)

External Trigger Enable--ETE (D_4). When ETE is set to 1, the I/O line of the port associated with the counter/timer (see Table 4-1) is used as a trigger input to the counter/timer. A rising edge (when the data path is specified non-inverting) on this line will cause the down-counter to be loaded. To guarantee that the counter/timer will be triggered on a particular rising edge of the clocking signal (PCLK/2 or counter input), the trigger rising edge must satisfy a setup time to the preceding falling edge of the clocking signal. As in the external count input, the bit of the port must be available for use by the counter/timer, and must be programmed as an input bit. (Even if the port bit is programmed as an output bit, the port pin is used as the counter/timer input [if enabled], allowing the CPU to write this input directly.)

External Gate Enable--EGE (D_3). By setting EGE to 1, the I/O line of the port associated with the counter/timer (see Table 4-1) is used as an external gate input to the counter/timer. If the external gate input is a 0 (assuming the data path is programmed non-inverting), the countdown sequence is suspended; forcing it to a 1 enables the countdown sequence to continue. To guarantee the enabling or disabling of the counter/timer for a particular rising edge of the clocking signal (PCLK/2 or counter input), the gate input must satisfy a setup time to the preceeding falling edge of the clocking signal. Like external trigger input, the bit must be available and it must be programmed to be an input. (Even if the port bit is programmed as an output bit, the port pin is used as the counter/timer input if enabled. This allows the CPU to write this input directly.)

Retrigger Enable Bit--REB (D_2). If REB is set to 0, triggers (internal or external) which occur during a countdown sequence are ignored. If REB is 1, each trigger causes the time constant value to be reloaded and a new countdown sequence to be initiated. When a counter/timer is programmed in square-wave mode, a retrigger will cause the Time Constant value to be reloaded and the new countdown will start on the first half of the square-wave cycle.

Output Duty Cycle Selects--DCS_1 & DCS_0 (D_1 & D_0). These two bits select the output duty cycle according to the information indicated in Table 2-9.

Table 2-9. Output Duty Cycle Selects

DCS_1	DCS_0	Output Duty Cycle
0	0	Pulse Output
0	1	One-Shot Output
1	0	Square Wave Output
1	1	- DO NOT USE -

(See Section 4.2.5 for a description of each output duty cycle type.)

Addresses: 001010 Counter/Timer 1
001011 Counter/Timer 2
001100 Counter/Timer 3
(Read/Partial Write)

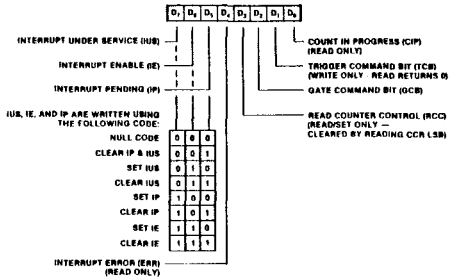


Figure 2-16. Counter/Timer Command and Status Registers

2.9.2 Counter/Timer Command and Status Registers

Each Counter/Timer Command and Status register contains the primary command and status bits for its counter/timer and (in most cases) will be the register most often accessed. A reset forces all bits to 0. The detailed bit descriptions will discuss whether or not a bit can be read or written.

Interrupt Under Service--IUS (D_7). The operation is the same as the port IUS bit.

This status bit is automatically set to 1 if its corresponding IP is the highest-priority interrupt request pending when an Interrupt Acknowledge sequence takes place. It can also be set directly by CPU command. As long as it is set, the same and lower-priority sources of interrupt are inhibited from requesting interrupts via the internal and external daisy-chains. It can be cleared only by CPU command. This bit is read/write. It is changed by writing to the Counter/Timer Command and Status register of the port using the code shown in Figure 2-16.

Interrupt Enable--IE (D_6). The operation is the same as the port IE bit.

This bit enables or disables the counter/timer's interrupt logic. When IE is cleared to 0, the counter/timer is unable to request an interrupt or to respond to an Interrupt Acknowledge. It does not affect the normal operation of IP or IUS, but simply masks IP off from the rest of the device. A 1 in IUS still affects the interrupt daisy-chain. If IE is programmed to be 1, the interrupt logic operates normally. This bit is read/write. It is changed by writing to the Counter/Timer Command and Status register of the port using the code shown in Figure 2-16.

Interrupt Pending--IP (D_5). The operation is similar to the port IP bit.

IP is a status bit which, when set to 1, indicates that the counter/timer requires servicing. It is automatically set to 1 each time the counter/timer reaches its terminal count (or by the CPU command). If IE is also 1 and no higher-priority interrupts are under service, then the $\overline{INT}$ line is pulled Low to request an interrupt. This bit is read/write. It is changed by writing to the Counter/Timer Command and Status register using the code shown in Figure 2-16.

Interrupt Error--ERR (D_4). This status bit is set along with IP to indicate that an error has occurred. An error occurs for a counter/timer whenever terminal count is reached and IP is still set from a previous terminal count. ERR can be cleared only by having software clear the IP it corresponds to. ERR is a read-only bit.

Read Counter Control--RCC (D_3). RCC is a command bit that enables the counter/timer to be read reliably while it is in a countdown sequence. Writing a 1 to RCC causes the contents of the Counter/Timer Current Count register (CCR), which normally follows the down-counter, to be frozen until the least-significant byte of the CCR is read. Reading the RCC bit indicates when the CCR is frozen. RCC can only be set directly and cannot be set unless the Counter/Timer is enabled in the Master Configuration Control register (CT1E, CT2E, or CT3E). RCC can be cleared automatically by reading the least-significant byte of the CCR or by disabling the counter/timer via the corresponding enable bit.

Gate Command Bit--GCB (D_2). GCB is a command bit that can be used to halt a countdown sequence. By writing GCB with a 0, the countdown sequence is halted. Returning GCB to 1 allows the sequence to resume where it left off. The state of the GCB bit does not affect the operation of the trigger inputs. GCB is a read/write bit.

Trigger Command Bit--TCB (D_1). Writing a 1 to the TCB triggers the counter/timer. It causes the down-counter to be loaded with the time constant value and a countdown sequence to be initiated. It can also retrigger the counter/timer if the Retrigger Enable bit (REB) is set to 1. TCB is a write-only bit. When read, it always returns 0. In this way, erroneous trigger comands are not issued when bit set or clear operations are performed on the other bits in this register.

Count In Progress--CIP (D_0). CIP is a status bit that indicates if a countdown sequence is in progress. It is automatically set to 1 when the counter/timer is triggered and the down-counter is loaded with the time constant value. It is automatically reset to 0 when the down-counter reaches a count of 0. The state of the gate inputs (internal and external) has no effect on this bit. CIP is read-only.

2.9.3 Counter/Timer Time Constant Registers

Each of the Time Constant registers is 16-bits and holds the value loaded into the down-counter of its counter/timer when a trigger is detected. It is accessed by the CPU as two consecutive 8-bit registers (bit 7 of the most-significant byte is bit 15 of the Time Constant register). These registers can be read and written at any time. However, care must be taken when writing them so that a trigger does not occur while the time constant value is changing. A reset does not effect the Time Constant register.

Addresses: 010110 Counter/Timer 1's MSB
010111 Counter/Timer 1's LSB
011000 Counter/Timer 2's MSB
011001 Counter/Timer 2's LSB
011010 Counter/Timer 3's MSB
011011 Counter/Timer 3's LSB
(Read/Write)

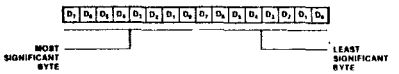


Figure 2-17. Counter/Timer Time Constant Registers

2.9.4 Counter/Timer Current Count Registers

Each of the Counter/Timer Current Count registers (CCR) is a 16-bit register used to read the contents of its counter/timer down-counter. The CCR follows the down-counter until the RCC bit in the Counter/Timer Command and Status register is written with a 1. The value present when the write occurs is held until the least-significant byte is read. Then, the CCR follows the down-counter again. The countdown sequence is not affected. The CCR is accessed as two consecutive 8-bit registers (bit 7 of the most-significant byte is bit 15 of the Time Constant register). They can be read at anytime, whether or not the

Addresses: 010000 Counter/Timer 1's MSB
010001 Counter/Timer 1's LSB
010010 Counter/Timer 2's MSB
010011 Counter/Timer 2's LSB
010100 Counter/Timer 3's MSB
010101 Counter/Timer 3's LSB
(Read Only)

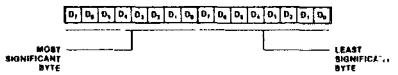


Figure 2-18. Counter/Timer Current Count Registers

value is frozen. Writes to the CCR are ignored. A reset forces the CCR to follow the down-counter (neither are forced to a specific value).

2.10 INTERRUPT RELATED REGISTERS

These registers contain the interrupt vectors output during Interrupt Acknowledge sequences. Three vector registers are provided: one for Port A, one for Port B, and one shared by the three counter/timers. Another register is provided, which facilitates using this device in a polled environment.

2.10.1 Interrupt Vector Registers

Each of the Interrupt Vector registers holds the interrupt vector returned when the source of interrupt associated with its port is acknowledged. The interrupt vector value is user-defined by writing the desired 8-bit identification code to this register when initializing the CIO. A modified version of the value written to the Interrupt Vector register can be returned if the vector is programmed to include status. This does not affect the value written to the Interrupt Vector register.

Addresses: 000010 Port A
000011 Port B
000100 Counter/Timers
(Read/Write)

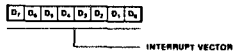


Figure 2-19. Interrupt Vector Register

The Interrupt Vector register is a read/write register. When read, the value returned always includes the status if MIE = 1 (whether or not the associated Vector Includes Status bit is 1). If MIE = 0, the unmodified vector is returned independent of the state of the VIS bit. A reset does not affect the Interrupt Vector register. The status bit outputs are as shown in Table 2-10.

Table 2-10. Interrupt Vector Register Status Bits

Port Vector Status			
OR-Priority Encoded Vector Mode:			
D_3	D_2	D_1	
x	x	x	Number of highest-priority bit with a match
All Other Modes:			
D_3	D_2	D_1	
ORE	IRF	PMF	Normal
0	0	0	Error
Counter/Timer Status			
D_2	D_1		
0	0		Counter/Timer 3
0	1		Counter/Timer 2
1	0		Counter/Timer 1
1	1		Error*

*The error status indicates that the highest-priority counter/timer with an interrupt pending also has its ERR flag set. The CPU must poll the Command and Status registers to determine which counter/timer has its ERR flag set.

2.10.2 Current Vector Register

When the Current Vector register is read, it returns the interrupt vector that would have been output by the device during an Interrupt Acknowledge cycle if its IEI input had been High. The vector returned corresponds to the highest priority IP independent of the IUS. The order of priority (highest to lowest) is: Counter/Timer 3, Port A, Counter/Timer 2, Port B, Counter/Timer 1. If no enabled interrupts are pending, a pattern of all 1s is output. This is useful in a polled environment or when CPU does not read vectors. This register is a read-only register. Since a reset disables all interrupts, reading the Current Vector register after a reset will return all 1s.

Address: 011111
(Read Only)

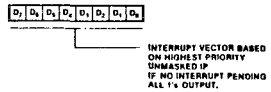


Figure 2-20. Current Vector Register

Chapter 3
I/O Port Operation

3.1 OVERVIEW

There are three I/O ports provided by the CIO device. Ports A and B are 8-bit general-purpose ports; Port C is a 4-bit special-purpose port. There are two port configurations: bit port and port with handshake. All three ports can be programmed as bit ports; only Ports A and B can function as handshake ports.

In general, bit ports are used to provide status input lines and control output lines. When the I/O ports are configured as bit ports, data can be moved in either direction on an individual, pin-by-pin basis. There are up to twenty pins available for this kind of data handling by the three ports.

By configuring Ports A and B as ports with handshake (input, output, or bidirectional), the data can be moved in either direction on a byte-by-byte (parallel 8-bit or 16-bit) basis. Four different handshakes are available: Interlocked, Strobed, Pulsed, or 3-Wire.

Port C is a 4-bit wide, special-purpose port that provides the handshake control lines for Ports A and B, when required. A REQUEST/$\overline{\text{WAIT}}$ line can also be provided to synchronize Port A and B data transfers with DMAs or CPUs. Any Port C bits not used as handshake lines can be used as I/O lines.

Another I/O Port function is to provide external access for the control of three independent counter/timers and distribution of their outputs. Port B provides access for Counter/Timers 1 and 2. Port C provides access to Counter/Timer 3.

Pattern-recognition capability is provided in Ports A and B. In general, it is possible to test data for specified patterns and to generate interrupt requests based on the match obtained.

3.2 PATTERN-RECOGNITION LOGIC OPERATION

Both Ports A and B can be programmed to generate interrupts when a specific pattern is recognized at the port. The pattern-recognition logic is independent of the port application, thereby allowing the port to recognize patterns in all of its configurations. The pattern can be independently specified for each bit as: 1, 0, 0-to-1 transition, 1-to-0 transition, or any transition. Individual bits can be masked off. Three modes of pattern-recognition operation are supported: AND, OR, and OR-Priority Encoded Vector (OR-PEV). A pattern match is defined as the simultaneous satisfaction of all nonmasked bit specifications in the AND mode or the satisfaction of any nonmasked bit specifications in either the OR or OR-PEV modes.

The pattern specified in the Pattern Definition register assumes that the data path is programmed to be non-inverting. If an input bit in the data path is programmed to be inverting, the pattern detected is the opposite of the one specified. Output bits used in the pattern match logic are internally sampled before the invert/non-invert logic.

The operation of the pattern-recognition logic in the various port modes will be described in detail in the following sections.

3.3 BIT PORT OPERATION

Bit ports are used to provide the CPU with input lines to monitor status, and with output lines to provide control. There are up to twenty bits available for this type of data handling provided by the three ports of the CIO: eight each by Ports A and B and four by Port C.

Writing the data register of a bit port updates the value being output by all output bits in the port. Reading the data register of the bit port returns the state of all bits, outputs as well as inputs.

3.3.1 Bit Port Simple Operation

The port's Data Direction register specifies the direction of data flow for each bit of a bit port. A 1 specifies an input bit; a 0 specifies an output bit.

The Data Path Polarity register provides the capability of inverting the data path. A 1 specifies inverting, and a 0 specifies non-inverting. All discussions of the port operations assume that the path is programmed non-inverting.

The value returned when reading an input bit reflects the state of the input just prior to the read. A 1's catcher can be inserted into the input data path by programming a 1 to the corresponding bit position of the port's Special I/O Control register. When a 1 is detected at the 1's catcher input, its output is automatically set to 1 until it is cleared by software. The 1's catcher is cleared by writing a 0 to the corresponding bit in the data register. In all other cases, attempted writes to input bits are ignored. The 1's catcher is level-sensitive. If the input is still a 1 when it is cleared, the output will again be set to a 1. Also, the input to the 1's catcher follows the invert/non-invert logic. If the bit is programmed inverting, a low voltage level at the pin will cause the 1's catcher output to go to a 1.

When Ports A and B include output bits, reading the data register returns the value being output. Reads of Port C return the state of the pins. Outputs can be specified as open-drain by writing a 1 to the corresponding bit of the port's Special I/O Control register. Port C has the additional feature of bit-addressable writes. When writing to Port C, the four most-significant bits are used as a write protect mask for the least-significant bits (0-4, 1-5, 2-6, and 3-7). With this feature, any combination of bits can be set or cleared (while other bits remain undisturbed), without first reading the register.

3.3.2 Bit Port Pattern-Recognition Operation

Ports A and B contain pattern-recognition logic, which enables the port to detect a user-specified pattern and to generate an interrupt request when the pattern is detected. Pattern-recognition may be performed on all bits, including those used as I/O for the counter/timers. For input bits, the input to the pattern-recognition logic reflects the value on the pins (through the invert/non-invert logic) in all cases except for inputs with 1's catchers. In this case, the output of the 1's catcher is used. For output bits, this is the value being output before the invert/non-invert logic is used. When operating in the AND or OR mode, the transition from a no-match to a match state causes the interrupt. In the OR mode, if a second match occurs before the first match goes away, it does not cause a second interrupt. Bit ports specified in the OR-PEV mode generate interrupts as long as a match state exists. A transition from a no-match to a match state is not required. Since a match condition only lasts a short time when transition patterns are specified, care must be taken--no more than one bit should be programmed with a transition match specification in a port operating in the AND mode.

The pattern-recognition logic of bit ports operates in two basic modes: Transparent and Latched. When the Latch on Pattern Match (LPM) bit is set to 0 (Transparent mode), the interrupt indicates that a specified pattern has occurred, but a read of the data register does not necessarily indicate the state of the port at the time the interrupt was generated. In the Latched mode (LPM = 1), the state of all the port inputs at the time the match was detected is latched in the Buffer register and held until IP is cleared. In all cases, the Pattern Match Flag (PMF) in the port's Command and Status register indicates the state of the port at the time the PMF is read. Only Transparent mode (LPM = 0) is supported when OR-PEV is specified. In all modes, the port's IP bit is set and an interrupt generated (if enabled) when the pattern match is detected. The IP can only be cleared by a command to the Port Command and Status register.

If a second match occurs while IP is already set, an error condition exists. If the Interrupt On Error bit (IOE) is 0, the match is ignored. However, if IOE is 1 after the first IP is cleared, the IP is automatically set to 1 along with the Interrupt Error (ERR) flag. Matches occurring while ERR is set are ignored. ERR is automatically cleared when the corresponding IP is cleared by software.

When a pattern-match is present in the OR-Priority Encoded Vector mode, IP is set to 1. The IP cannot be cleared until a match is no longer present. If the interrupt vector is allowed to include status, the vector returned during Interrupt Acknowledge indicates the highest-priority bit matching its specification at the time of the Acknowledge cycle. Bit 7 is the highest-priority bit and bit 0 is the lowest-priority bit. The bit initially causing the interrupt may not be the one indicated by the vector if a higher-priority bit matches before the Acknowledge. Once the Interrupt Acknowledge cycle is initiated, the vector is frozen until the corresponding Interrupt Under Service (IUS) is cleared. If an input that causes interrupts changes before the interrupt is serviced, the 1's catcher can be used to hold the value. Bits should not be specified with transition detection, because the match will no longer be valid at the time of the Interrupt Acknowledge. If no match is present at the time of the Acknowledge, the vector will indicate the lowest-priority bit (Bit 0).

Because a no-match-to-match transition is not required, the source of the interrupt must be cleared before IP is cleared or else a second interrupt is generated. No programmer error detection is performed in this mode and the Interrupt on Error bit should be 0.

One application of the OR-PEV pattern match mode is to use the CIO as a Programmable Interrupt Controller (PIC). This facilitates using non-Z-BUS peripherals with a Z-BUS CPU. (See Chapter 5 for further discussion.)

3.4 HANDSHAKE PORT OPERATION

Ports A and B can be specified as 8-bit input, output, or bidirectional ports with handshake. The CIO provides four different handshakes for its ports: Interlocked, Strobed, Pulsed, and 3-Wire. When specified as a port with handshake, the transfer of data into or out of the port and interrupt generation is under the control of the handshake logic. Port C provides the handshake lines, as shown in Table 3-1.

When Ports A and B are configured as ports with handshake, they are single- or double-buffered according to the setting of the Single Buffered Mode (SB) bit of their respective Port Mode Specification registers.

The double-buffered mode (SB = 0) allows for more relaxed interrupt service routine response time. A second byte can be input to or output from the port before the interrupt for the first byte is serviced. The Single-Buffered mode (SB = 1) is useful if the handshake line must be stopped on a byte-by-byte basis.

Normally, the Interrupt Pending (IP) bit is set and an interrupt is generated when data is moved into the Input Data register (input port) or out of the Output Data register (output port). For input and output ports, the IP is normally cleared automatically when the data is read or written. In bidirectional ports, IP is cleared only by software command to the port Command and Status register. When the Interrupt on Two Bytes (ITB) control bit is set to 1, interrupts are generated only when two bytes of data are available to be read or written. This allows 16 bits (two bytes) of information to be transferred on each interrupt. With ITB set, the IP is not automatically cleared until the second byte of data is read or written.

Pattern recognition logic is also available for use with the port with handshake. Each time a byte is moved into the Input Data register or out of the Output Data register, the pattern match flag indicates whether a match has occurred.

Ports A and B can be linked to form a 16-bit port by programming a 1 in the Port Link Control (PLC) bit of the Master Configuration Control register. In this mode, only the Port A Handshake Specification and Command and Status registers are used, and Port B must be specified as a bit port. When linked, only Port A has pattern match capability. Port B's pattern match capability must be disabled. Also, when the ports are linked, the Port B Data register must be read or written before the Port A Data register.

When a port is specified as a port with handshake, its mode (input, output, or bidirectional) determines the direction of data flow. The data direction for the bidirectional port is determined by a bit in Port C (See Table 3-1). In all cases, the contents of the port's Data Direction register are ignored. The contents of the Special I/O Control register apply only to output bits (normal or open-drain). Input ports with handshake do not have 1's catchers; therefore, those bits in the Special I/O Control register are ignored. Port C lines used for handshake can all be programmed as inputs because the handshake specification overrides the Port C Data Direction register for bits that must be outputs. All other Port C options are available (polarity, 1's catcher, open-drain outputs, etc.).

Table 3-1. Port C Pin Utilization

Port A/B Configuration	Port C Bits			
	Pin C_3	Pin C_2	Pin C_1	Pin C_0
Ports A & B = Bit Ports	Bit I/O	Bit I/O	Bit I/O	Bit I/O
Port A = Input or Output port (Interlocked, Strobed, or Pulsed Handshake)*	RFD or $\overline{DAV}$	$\overline{ACKIN}$	REQUEST/$\overline{WAIT}$ or Bit I/O	Bit I/O
Port B = Input or Output port (Interlocked, Strobed, or Pulsed Handshake)*	REQUEST/$\overline{WAIT}$ or Bit I/O	Bit I/O	RFD or $\overline{DAV}$	$\overline{ACKIN}$
Port A or B = Input port (3-Wire Handshake)	RFD (Output)	$\overline{DAV}$ (Input)	REQUEST/$\overline{WAIT}$ or Bit I/O	DAC (Output)
Port A or B = Output port (3-Wire Handshake)	$\overline{DAV}$ (Output)	DAC (Input)	REQUEST/$\overline{WAIT}$ or Bit I/O	RFD (Input)
Port A or B = Bidirectional port (Interlocked or Strobed Handshake)	RFD or $\overline{DAV}$	$\overline{ACKIN}$	REQUEST/$\overline{WAIT}$ or Bit I/O	IN/$\overline{OUT}$

* Both Ports A & B can be specified input or output with Interlocked, Strobed, or Pulsed Handshake at the same time if neither uses REQUEST/$\overline{WAIT}$. However, only one port can use the Pulsed Handshake at a time.

3.4.1 Four Handshake Modes

There are four handshake modes: Interlocked, Strobed, Pulsed, and 3-Wire.

3.4.1.1 Interlocked Handshake

In the Interlocked Handshake mode, the action of the CIO must be acknowledged by the external device before the next action can take place. An output port does not indicate that new data is available until the external device indicates it is ready for the data. Similarly, an input port does not indicate that it is ready for new data until the data source indicates that the previous byte of data is no longer available, thereby acknowledging the input port's acceptance of the last byte. This handshake allows the CIO to interface directly to the port of a Z8 microcomputer, a UPC, an FIO, an FIFO, or to another CIO port, etc., with no external logic.

3.4.1.2 Strobed Handshake

In the Strobed Handshake mode, data is "strobed" into or out of the port by the external logic. The falling edge of the Acknowledge Input ($\overline{ACKIN}$) strobes data into or out of the port. In contrast to the Interlocked Handshake, the signal indicating that the port is ready for another data transfer operates independently of the $\overline{ACKIN}$ input. The external logic must ensure that data does not transfer at too fast or too slow a rate.

3.4.1.3 Pulsed Handshake

The Pulsed Handshake mode is designed to interface to mechanical-type devices which require data to be held for long periods of time and need relatively wide pulses to gate the data into or out of the device. The logic is the same as the Interlocked Handshake mode, except that an internal counter/timer (Counter/Timer 3) is linked to the handshake logic. If the port is specified in the input mode, the timer is inserted in the $\overline{ACKIN}$ path. The external $\overline{ACKIN}$ input triggers the timer, and its output (an internal delayed $\overline{ACKIN}$) is used as the Interlock Handshake's normal Acknowledge input. If the port is an output port, the timer is placed in the Data Available ($\overline{DAV}$) output path. The timer is triggered when the normal Interlocked Handshake $\overline{DAV}$ output goes Low and the timer output is used as the actual $\overline{DAV}$ output. The counter/timer maintains all of its normal capabilities. This handshake is not available to bidirectional ports.

3.4.1.4 3-Wire Handshake

The 3-Wire Handshake mode is designed for situations in which one output port is communicating with many input ports simultaneously. It is essentially the same as the Interlocked Handshake, except that two signals are used to indicate if an input port is ready for new data or if it has accepted the present data. In the 3-Wire Handshake, the rising edge of one status line indicates that the port is ready for data (RFD), and the rising edge of another status line indicates that the data has been accepted (DAC). With the 3-Wire Handshake, the output lines on many input ports can be bused together (wire-AND) with open-drain drivers; the output port knows when all the ports have accepted the data and are ready. This is the same handshake used on the IEEE-488 bus. Because this handshake requires three lines, only one port (either A or B) can be a 3-Wire Handshake port at a time. The 3-Wire Handshake is not available in the bidirectional mode. However, because the direction of the port can be changed under software control, bidirectional IEEE-488-type transfers can be performed.

3.4.2 Input Port With Handshake

An input port handles data movement from the CIO port pins to the CPU. This allows 8-bit data (or 16-bit if the Ports A and B are linked) to be read from external devices. (See Figure 3-1.)

Only one of the three Bit Path Definition registers affects input port operations: the data path is modified as specified by the Data Path Polarity register (Section 2.5.1). Both the Data Direction and Special I/O registers are ignored.

Since the port mode of operation is independent of the handshakes, the port operation modes will be examined first, independent of the handshake types. This will then be followed by an examination of the four handshake types (Interlocked, Strobed, Pulsed, and 3-Wire) in the input port context.

3.4.2.1 Basic Modes of Operation

There are three independent modes of operation that, taken together, characterize a particular input port configuration. These modes of operation are:

- double- or single-buffered
- interrupted on one or two bytes
- pattern match logic used or not used

Double-Buffered (SB = 0).

When the input port is specified as double-buffered (SB = 0) in the Port Mode Specification register, input data is latched in the Buffer register by the handshake logic. The falling edge of Acknowledge Input ($\overline{ACKIN}$) latches the incoming data and the Ready For Data (RFD) output signal goes Low, indicating that the Buffer register is full. When the Input Data register is empty (IRF = 0), the data is moved out of the Buffer register (RFD may now go High depending on the particular handshake) and into the Input Data register causing it to be "filled" (IRF = 1).

The Interrupt on Two Bytes (ITB) command bit of the Port Mode Specification register determines when IP is set and when an interrupt request is generated. When programmed to interrupt on every byte (ITB = 0), Interrupt Pending (IP) of the Port Command and Status register is set (along with Input Data register Full (IRF) of the Port Command and Status register) when the data is shifted into the Input Data register. Reading the Port Data register "empties" the Input Data register and automatically clears the IP, hence, IRF = 0 and IP = 0. While IP can be cleared (IP = 0) by software command (by writing bits D_7-D_5 of the Port Command and Status register), the Input Data register is not "emptied" (cleared) until the data is read.

When programmed to interrupt on two bytes (ITB = 1), IP is not set until both the Input Data and Buffer registers are full (the Buffer register becomes full while the Input Data register is full). IP is automatically cleared (IP = 0) when the second byte of data is read by the CPU as

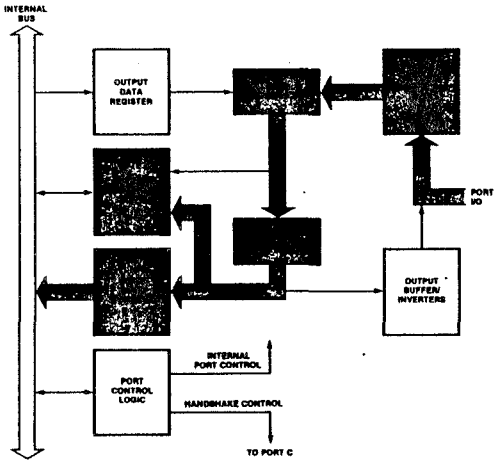


Figure 3-1. Input Port Data Path

follows: the first data read empties the Input Data register, allowing the Buffer register data to be moved into the Input Data register. The second read empties the Input Data register again and automatically clears the IP (IP = 0).

NOTE

When IIB = 1, the Input Data register should not be read unless IP = 1 even if IRF = 1. Otherwise, the data may--at that instant--be read just as a byte is being latched in the Buffer register and IP is being set. Then an interrupt may be generated, but only one byte can be read, because the first byte has already been read.

NOTE

The IP can be cleared on command. This suggests the following possible sequence for providing byte-by-byte control of the RFD output: reading data and then clearing IP. This allows an interrupt on the next byte that is moved into the Buffer register.

Single-Buffered (SB = 1)

When the input port is specified as single-buffered (SB = 1), input data is latched in the Buffer register as in double-buffered mode. However, when the data is moved from the Buffer register to the Input Data register, the Buffer register is not emptied; consequently, RFD stays Low. Reading the Port Data register empties both the Input Data and Buffer registers. As in double-buffered mode, IP is set (IP = 1) when the data is moved into the Input Data register and the IP is automatically cleared when the data is read.

NOTE

IIB = 1 does not make sense in single-buffered operation. Thus, when SB = 1, IIB must = 0.

With Pattern Match Added (PMS_1, $PMS_0 \neq 0$)

The port's built-in pattern match logic can be used to test the incoming data as it is moved into or through the Input Data register. The available pattern match modes operate independently of handshake type and are specified by the Pattern Mode

Specification bits, (PMS_1 and PMS_0). In the input port operation, the AND and the OR modes of Pattern Match operation are available for use, but the OR-PEV is not. This pattern availability for AND and OR logic is a consequence of the data pattern being tested as the data is moved into the Input Data register, eliminating access to the transition information. Because of this, transition patterns cannot be used.

The Pattern Match Flag (PMF) of the Port Command and Status register is automatically set to 1 or cleared to 0 as the data is moved from the Buffer register to the Input Data register. When a byte of data is moved from the Buffer register to the Input Data register, IRF is set to 1, indicating that another byte is available to be read. If the data matches the specified pattern, the PMF is set along with IRF; otherwise, PMF is cleared. If the interrupt vector includes status, it indicates that a match has been detected. Each time data is transferred to the Input Data register, the PMF is updated.

When IIB = 0, IP is set normally when data is moved into the Input Data register. However, if PMF = 1, reading the Input Data register does not automatically clear the IP; the IP can only be cleared by writing to the Port Command and Status register. Also, reading the Input Data register does not "empty" it; the Input Data register can be "emptied" (and IRF cleared to 0) only if it is read and the IP is cleared (in any order).

When IIB = 1, the pattern match logic can override the IIB logic. If the byte moved into the Input Data register matches the specified pattern, the IP will be set immediately. IP is cleared and the Input Data registers are "emptied" in the same manner as when IIB = 0.

In this mode of operation (IIB = 1 and Pattern Match is enabled) care must be taken, because an IP can mean that either one or two bytes are available to be read (depending upon whether the match occurred on the first or second input byte). There is also the possibility that, after a data register read and a clear IP, a second byte matches. There are three conditions which can cause an interrupt:

Condition 1: Two bytes have been received--neither match the pattern

Condition 2: Two bytes have been received--the second byte matches the pattern

Condition 3: One byte has been received--it matches the pattern

Given the above information, the following set of operations will determine the cause of the interrupt and properly process it.

```
Poll PMF

  If PMF is set (a match occurred)          The cause is condition 3
  • Read data
  • Clear IP
  • Return

  If PMF is not set
  • Read data (reads the first byte)
  • Poll PMF (to test the second byte)

      If PMF is set (a match occured)       The cause is condition 2
      • Read data
      • Clear IP
      • Return

      If PMF not set                        The cause is condition 1
      • Read data
      • Return
```

In summary, then, careful interrupt testing and handling is required if ITB = 1 and the pattern match logic is enabled.

Interrupt on Match Only (IMO = 1) Specified

When the Interrupt on Match Only (IMO) bit of the Port Mode Specification register is set to 1, an interrupt will be generated only when the data moved into the Input Data register matches the pattern specification. For input ports, the IMO capability is especially useful when data transfer is under the control of an external device (for example, a DMA controller). In this way the bulk of data transfers can be accomplished without interrupts (that is, without involvement of the CPU), by having an interrupt generated only when the match pattern in encountered.

NOTE

IMO must be 0 if either ITB or SB = 1, or if the port is a bit port.

3.4.2.2 Handshake Types

The operation of the Port A and B input handshakes is explained in this section by describing in detail the sequence of operations performed by an input port programmed with the Interlocked Handshake. Any differences encountered when using the other handshakes will then be described. Table 3-1 identifies the handshake lines furnished by Port C bits for Ports A and B.

Interlocked Input Handshake

As noted in Section 3.3, the Interlocked Input Handshake requires the input port to not indicate that it is ready for data until the data source indicates that the previous byte of data is no longer available, thereby acknowledging that the input port has accepted the previous byte. A primary benefit of Interlocked Handshake port configuration is that it allows the CIO to communicate directly with a variety of other devices without the need for intervening external logic. Devices such as another Z-CIO/CIO, an FIO, an FIFO, a Z8 Port, etc., can be directly connected and serviced. Figure 3-2 shows two interconnected CIOs: output port's $\overline{DAV}$ output connects to input port's $\overline{ACKIN}$ input and input port's RFD output connects to output port's $\overline{ACKIN}$ input.

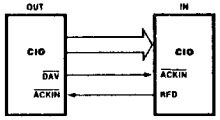


Figure 3-2. Two Interconnected CIOs Using Interlocked Handshake

In Interlocked Handshake mode (Figure 3-3), on the falling edge of the Acknowledge Input ($\overline{ACKIN}$), the data on the port input lines is latched in the Buffer register. This fills the Buffer register and the Ready for Data (RFD) output is pulled Low. If the Input Data register is empty, the data is moved to it ("emptying" the Buffer register) and the Input Data register Full (IRF) flag is automatically set to 1. When the Buffer register becomes empty (and if $\overline{ACKIN}$ is High), the RFD line will return High only if the $\overline{ACKIN}$ input is High. This achieves the interlock.

The following example provides a step-by-step analysis of a double-buffered input port using Interlocked Handshake. (This description uses Figure 3-3 as reference).

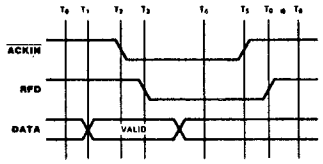


Figure 3-3. Interlocked Input Handshake Timing Diagram

T_0. $\overline{ACKIN}$ and RFD are both High. $\overline{ACKIN}$ High indicates that the data is not valid; RFD High indicates that the Buffer register is empty and ready for data.

T_1. Data on port pins becomes valid.

T_2. $\overline{ACKIN}$ goes Low, indicating that the data is valid, and causing it to be latched into the Buffer register.

T_3. RFD goes Low, indicating that the Buffer register is full and the port is not ready for more data.

T_4. The data is transferred into the Input Data register, the Input Data register Full (IRF) flag goes High and the Buffer register is emptied. The port is now ready for the next byte of data. RFD could go High if $\overline{ACKIN}$ is High; but because $\overline{ACKIN}$ is Low, RFD stays Low.

T_5. $\overline{ACKIN}$ goes High.

T_6 becomes T_0. RFD goes High, concluding the handshake process; the cycle is ready to repeat.

Strobed Input Handshake

The Strobed Handshake (Figure 3-4) operates in the same way as the Interlocked Handshake, except that the rising edge of the RFD output is independent of $\overline{ACKIN}$ going High. As soon as the Buffer register is emptied, RFD goes High, even if $\overline{ACKIN}$ is still Low. In all other respects, the two handshakes are the same. The falling edge of the $\overline{ACKIN}$ input "strobes" the data into the port.

The following example provides a step-by-step analysis of an input port configured as double-buffered and using Strobed Handshake. (This description uses Figure 3-4 as reference.)

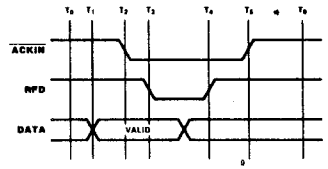


Figure 3-4. Strobed Input Handshake Timing Diagram

T_0. $\overline{ACKIN}$ and RFD are both High. $\overline{ACKIN}$ High indicates that the data is not valid; RFD High indicates that the Buffer register is empty and ready for data.

T_1. Data on port pins becomes valid.

T_2. $\overline{ACKIN}$ goes Low, indicating that the data is valid, and causing it to be latched in the Buffer register.

T_3. RFD goes Low, indicating that the Buffer register is full and the port is not ready for more data.

T_4. The data is moved into the Input Data register, the Input Data register Full (IRF) flag goes High, the Buffer register is emptied, and RFD goes High.

T_5 becomes T_0. $\overline{ACKIN}$ is High; the cycle is ready to repeat.

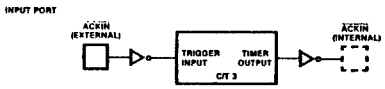


Figure 3-5. Pulsed Input Handshake Counter/Timer Insertion

Pulsed Input Handshake

The Pulsed Handshake operates exactly like the Interlocked Handshake with Counter/Timer 3 inserted in the $\overline{\text{ACKIN}}$ input path (see Figure 3-5). The counter/timer is triggered on the falling edge of $\overline{\text{ACKIN}}$. The output of the timer is inverted and used as the Acknowledge input for the handshake. The falling edge of the internal $\overline{\text{ACKIN}}$ latches the data, and RFD cannot go High until the internal $\overline{\text{ACKIN}}$ goes High. Because all of the programmable capabilities of Counter/Timer 3 are available, many different operations are possible. However, since only Counter/Timer 3 is used, only one port can have Pulsed Handshake at a time.

If the counter/timer output duty cycle is programmed in the pulse mode, TC cycles (where TC is the value programmed in the counter/timer Time Constant register) after the $\overline{\text{ACKIN}}$ falling edge is detected, the Internal Acknowledge falls, latching the data in the Buffer register; the internal $\overline{\text{ACKIN}}$ rises a cycle later. If the counter is programmed with the one-shot duty cycle, then as soon as the external falling edge on the $\overline{\text{ACKIN}}$ input is detected, the Internal Acknowledge falls; it rises TC cycles later. When the counter/timer duty cycle is selected to be square-wave, the Internal Acknowledge goes Low TC clock cycles after $\overline{\text{ACKIN}}$ falls and stays Low for TC cycles. See Figure 3-6 for timing diagrams of the three different duty cycles (pulsed, one-shot, and square-wave) available with Counter/Timer 3.

Because the handshake is interlocked with the internal $\overline{\text{ACKIN}}$ (not the $\overline{\text{ACKIN}}$ port pin), the RFD output is held Low as long as the Internal Acknowledge is Low. This can be used to guarantee a minimum RFD Low time without CPU intervention. Many simple interfaces can be made with the Pulsed Handshake by connecting the RFD output (inverted) to the $\overline{\text{ACKIN}}$ input.

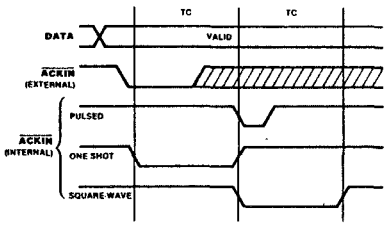


Figure 3-6. Pulsed Input Handshake Counter/Timer Duty Cycles

3-Wire Input Handshake

The 3-Wire Handshake (Figure 3-7) is the same as the Interlocked Handshake, except that the role of the Ready for Data (RFD) output is replaced by two signals: RFD and Data Accepted (DAC). The name of the $\overline{\text{ACKIN}}$ input is changed to Data Available ($\overline{\text{DAV}}$) to be consistent with the IEEE-488 specification, but its function is the same. The RFD output goes High when the Buffer register is empty and the $\overline{\text{DAV}}$ input is High. When $\overline{\text{DAV}}$ falls, the input data is latched, RFD goes Low (the Buffer register is full), and DAC goes High, indicating that the data was received. When the $\overline{\text{DAV}}$ input goes High, DAC is forced Low. When the Buffer register is emptied, RFD goes High, indicating that the port is ready for the next byte. Like the Interlocked Handshake, RFD will not go High until the $\overline{\text{DAV}}$ input goes High. The operation of the interrupt logic is the same as for the Interlocked Handshake.

One Talker--Many Listeners

The 3-Wire Handshake is useful in the situation in which there is one source of data (the "Talker") and many receivers ("Listeners") for the data.

Each Listener has one input line called Data Available ($\overline{\text{DAV}}$) and two output lines called Ready For Data (RFD) and Data Accepted (DAC). The RFD and DAC lines of all Listeners are connected together in a wire-AND.

(A wired-AND requires all inputs [Listener's signals] to go High before the output [signal to the Talker] goes High.) The RFD signal to the Talker

tells the Talker when the last Listener is ready to receive data (final Listener's RFD goes High). The Talker then tells the Listeners that data is now available by bringing $\overline{\text{DAV}}$ Low.

Each Listener, working at its individual pace, signals when data reception is done by letting its DAC line go High. The wired-AND DAC signal will tell the Talker when the last Listener has accepted the current data (the last Listener's DAC finally goes High).

The Talker then tells the Listeners that the data is no longer valid ($\overline{\text{DAV}}$ goes High). Each Listener puts DAC Low and, when ready for new data, puts RFD high.

The wired-AND RFD and the Low DAC tells the Talker that all Listeners are ready for new data, and the cycle is ready to begin again.

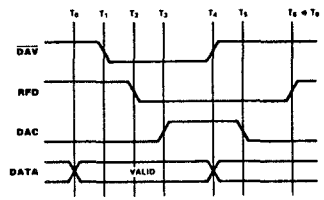


Figure 3-7. 3-Wire Input Handshake Timing Diagram

This procedure is in agreement with the IEEE-488 3-Wire Handshake.

The following example provides a step-by-step analysis of an input port configured as double-buffered and using the 3-Wire Handshake. (This description uses Figure 3-7 as reference).

T_0. $\overline{\text{DAV}}$ and RFD are both High. $\overline{\text{DAV}}$ High indicates that the data is not valid.

T_1. $\overline{\text{DAV}}$ goes Low, indicating that the data is ready to be read and that the input ports are ready for data (the interlock requires RFD to be High before the output port can lower $\overline{\text{DAV}}$. The data is latched into the input port Buffer register.

T_2. RFD goes Low, indicating that the Buffer register is full and the port is not ready for more data.

T_3. The individual input ports allow DAC to go High. The wire-ANDed DAC goes High, indicating that the data was received by all input ports.

T_4. $\overline{\text{DAV}}$ goes High as the start of the completion of data handling handshake, acknowledging that the data was received.

T_5. DAC goes Low as the data transfer is completed.

T_6 becomes T_0. When their Buffer registers are empty, the individul input ports allow RFD to go High. The wire-ANDed RFD goes High, indicating that the port is ready for the next byte; the cycle is ready to repeat.

3.4.3 Output Port With Handshake

Output ports handle data movement from the CPU to the CIO port pins. This allows the writing of 8-bit (or 16-bit if Ports A and B are linked) data to external devices. (See Figure 3-8.)

There are two Bit Path Definition registers that can affect output port operation: The data path

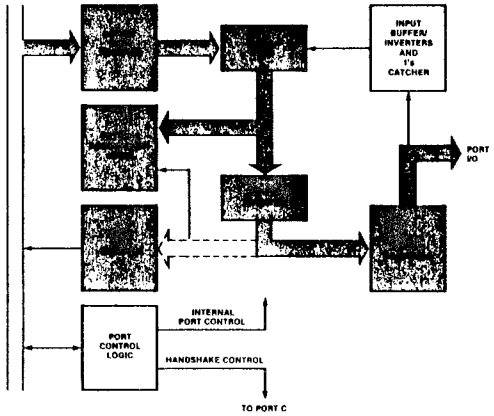


Figure 3-8. Output Port Data Path

is modified as specified by the Data Path Polarity register (see Section 2.6.1); and the Special I/O Control register allows selection of either normal or open-drain outputs (see Section 2.6.3).

When a port is programmed in the output mode, its Input Data Register Full (IRF) flag of the Port Command and Status register is automatically held at 0.

Because the port operation is independent of the handshake, the port operation modes will be examined in this section independent of handshake types. This will be followed by an examination of the four handshake types (Interlocked, Strobed, Pulsed, and 3-Wire) in the output port context.

3.4.3.1 Basic Modes of Operation

There are three independent modes of operation that, taken together, characterize a particular output port configuration. These modes of operation are:

- double- or single-buffered
- interrupt on one or two bytes
- using or not using pattern match logic

Double-Buffered (SB = 0)

The CPU writes data to the Output Data register. The data is moved to the Buffer register if it is empty. When the output port is specified as double-buffered (SB = 0) in the Port Mode Specification register, the data move to the Buffer register "empties" the Output Data register, setting the Output Data Register Empty (ORE) flag; the CPU can then write another byte into the Output Data register. The falling edge of $\overline{\text{ACKIN}}$ indicates that the data has been taken, and empties the Buffer register. Reading the Input Data register will return the current value in the Buffer register.

The $\overline{\text{DAV}}$ output tells receivers that the output port data is available and valid when this signal is Low.

The Interrupt on Two Bytes (ITB) control bit of the Port Mode Specification register determines when IP is set and when an interrupt should be requested.

While programmed to interrupt on every byte (ITB = 0), Interrupt Pending (IP) of the Port Command and Status register is automatically set to 1 along with Output Data Register Empty (ORE) of the Port Command and Status register when the data is moved out of the Output Data register and into the Buffer register. Writing to the port data register "fills" the Output Data register and automatically clears the IP (hence, ORE = 0 and IP = 0). IP can be cleared (IP = 0) by software command; that is, by writing bits D_7-D_5 of the Port Command and Status register. However, the Output Data register is not "filled" until the data is written. Since IP is set only when data is moved out of the Output Data register, then if the port is enabled (PAE or PBE is set to 1) without writing any data, IP will not be set even though the Output Data register is empty.

While programmed to interrupt on two bytes (ITB = 1), IP is not automatically set to 1 until both the Output Data and Buffer registers are empty; that is, the Buffer register becomes empty while the Output Data register is empty. IP is automatically cleared (IP = 0) when the second byte of data is written by the CPU to the data register. The first data write fills the Output Data register and allows the data to be moved into the Buffer register. The second data write fills the now empty Output Data register and automatically clears the IP. As when ITB = 0, when the port is initially enabled, the IP will not be set nor will an interrupt request be generated until a byte of data is written to it. Data can be written to it after it is initially configured and before it is enabled.

When ITB = 1, the Output Data register should not be written unless IP = 1, even if ORE = 1. Otherwise, the data may be written just as the Buffer register is going empty and as IP is being set. In this case, an interrupt may be requested for a two-byte write, when only one byte is needed, because the first byte has already been written but not output.

NOTE

The IP can be cleared on command. This suggests the following possible sequence for providing byte-by-byte control of the DAV output: writing one byte of data and then clearing IP by command allows an interrupt when the Buffer register is "emptied" by the handshake logic.

Single-Buffered (SB = 1)

When the output port is specified as single-buffered (SB = 1), output data is moved from the Output Data register to the Buffer register as in the double-buffered case. However, when the data is moved into the Buffer register, the Output Data register is not emptied. A copy of the data is maintained there. In this mode, the handshake logic "empties" both registers: when $\overline{\text{ACKIN}}$ falls, the Output Data and Buffer registers are emptied, and both ORE and IP are automatically set to 1. Writing the port data register fills the Output Data register and clears IP (ORE = IP = 0).

NOTE

Unlike the double-buffered operation, ITB = 1 does not make sense in single-buffered operation. Thus, when SB = 1, ITB must = 0.

With Pattern Match Added (PMS_1 = $PMS_0 \neq 0$)

The port's built-in pattern match logic can be used to test the data as it is coming into or through the data register. The available pattern match modes operate independently of handshake type and are specified by the Pattern Mode Specification bits (PMS_1 and PMS_0). In the output port operation, the AND and the OR modes are available for use, but the OR-PEV is not. This pattern availability for AND and OR logic is a consequence of the pattern being tested as the data is moved into the Input Data register, eliminating access to the transition information. Because of this, transition patterns cannot be used.

The Pattern Match Flag (PMF) of the Port Command and Status register is set to 1 or is cleared to 0 as the data is moved from the Output Data register to the Buffer register.

When ITB = 0, IP is set when data is moved out of the Output Data register. If PMF = 1, writing the data register does not automatically clear the IP--the IP can only be cleared by writing to the Port Command and Status Register. Also, writing the Output Data register does not "fill" it; the Output Data register can be "filled" (and ORE cleared to 0) only if it is written and the IP is cleared, (in any order). Data that is written

before IP is cleared, however, is latched in the Output Data register.

While IIB = 1, the pattern match logic can override the IIB logic. If the byte moved into the Buffer register matches the specified pattern, the IP goes to 1 immediately. IP is cleared and the Output Data register is filled in the same manner as when IIB = 0.

In the Pattern Match mode of operation, where IIB = 1 and Pattern Match is enabled, care must be taken, because an IP = 1 can mean that either one or two bytes may be written (depending upon whether a match occurred and if the match occured on the first or second output byte). There is also the possibility that, if IP is cleared by writing to Port Command and Status register after a single data register write, no further interrupts will be generated if the byte matched.

A normal assumption when IIB = 1 is that both the Output Data and Buffer registers are empty when an interrupt occurs, and two consecutive writes will fill both registers and clear IP. However, if the first byte written matches the pattern, the second write will not "fill" the Output Data register or clear IP because PMF was set to 1 when the first byte was moved into the Buffer register. If the second byte matches, IP and PMF will be set when it is moved into the Buffer register and an interrupt request will be generated--however, only one byte can be written since only the Output Data register is empty.

When IIB = 1 and both PMS_1 and PMS_0 are not 0, there are three conditions which can cause an interrupt:

Condition 1: 2 bytes were written--neither match the pattern

Condition 2: 2 bytes were written--the second byte matches the pattern

Condition 3: 1 byte was written--it matches the pattern

Given the above information, the following set of operations will determine the cause of the interrupt and properly process it.

```
Poll PMF (this could be Status In Vector)

  If PMF = 0 (both registers are empty)        The cause is condition 1
  • Write 1st byte
  • Poll PMF

      If PMF = 0 (1st byte doesn't match)
      • Write 2nd byte (IP automatically cleared)
      • Return

      If PMF = 1 (1st byte matches)            The cause is condition 3
      • Clear IP
      • Write 2nd byte (if any)
      • Return

If PMF = 1 (2nd byte matches)                  The cause is condition 2
• Clear IP
• Write a byte (if any)
• Return
```

In summary, careful interrupt handling is required if both the IIB and pattern match logic are enabled.

Interrupt on Match Only (IMO = 1) Specified

When the Interrupt on Match Only bit of the Port Mode Specification register is set to 1, an interrupt request will be generated only when the data moved out of the Output Data register into the Buffer register matches the pattern specification. For output ports, the IMO capability is especially useful when data transfer is under the control of an external device (for example, a DMA controller). In this way the bulk of the data transfers can be accomplished without interrupts (that is, without involvement of the CPU) by having an interrupt generated only when the match pattern is encountered. The CIO, through the IMO, allows a long string of bytes to be output without interrupts. This is accomplished by simply waiting on a pattern which is inserted at the end of a string to signal the end of the transmission by way of a single CPU interrupt request. This way, many bytes can be moved with only one interrupt request being generated.

NOTE

IMO must be 0 if either IIB or SB = 1, or if the port is a bit port.

3.4.3.2 Handshake Types

The operation of Port A and B output handshakes will be explained by describing in detail the sequence of operations performed by an output port programmed with Interlocked Handshake. Any differences encountered when using the other handshakes will then be described. See Table 3-1 for identification of the handshake lines furnished by Port C bits for Ports A and B.

Deskew Timer Description

Because external devices may require that the data be valid for a certain minimum amount of time prior to the $\overline{DAV}$ signal being pulled Low (to indicate data available), the CIO provides a separate deskew timer for each port. As data is transferred to the Buffer register, the deskew timer is triggered (if the timer is enabled (DTE = 1). After the number of PCLK cycles (up to 16) specified by the deskew timer Time Constant (see Figure 2-5), $\overline{DAV}$ is allowed to go Low (the interlock may keep $\overline{DAV}$ High). The deskew timer does not extend the time from $\overline{ACKIN}$ rising to $\overline{DAV}$ falling. The deskew timer therefore guarantees that the output data is valid for a specified minimum amount of time before $\overline{DAV}$ goes Low.

Deskew timers are available for output ports independent of the type of handshake employed. Each port has a separate 4-bit deskew timer. Thus, the CIO can provide the proper timing to interface to those external devices that require a large data valid to $\overline{DAV}$ falling setup time. For example, the IEEE-488 specification requires data to be valid for 2 microseconds before $\overline{DAV}$ can go Low.

Interlocked Output Handshake

The Interlocked Output Handshake requires that the output port does not indicate that it has data available ($\overline{DAV}$ goes Low) until the receiving port indicates that the previous byte of data has been accepted (indicated by $\overline{ACKIN}$ being High).

Interlocked Handshake port configuration allows the CIO to communicate directly with a variety of other devices without the need for external logic. Devices such as another CIO, FIO, FIFO, Z8 Port, etc., can be interfaced directly. Figure 3-2 shows two interconnected CIOs: the output port's $\overline{DAV}$ output connects to the input port's $\overline{ACKIN}$ input and the input port's RFD output connects to the output port's $\overline{ACKIN}$ input.

In Interlocked Output Handshake mode (Figure 3-9), the CPU writes data into the Output Data register. If the Buffer register is empty, the data is moved to it and on to the port output pins ("emptying" the Output Data register). The Output Data register Empty (ORE) bit in the port Command and Status register is then set to 1 and Data Available ($\overline{DAV}$) handshake line goes Low (if $\overline{ACKIN}$ is High). The CPU can now write another byte into the Output Data register, setting ORE to 0. When

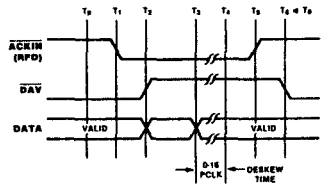


Figure 3-9. Interlocked Output Handshake Timing Diagram

the receiving port has accepted the data it lowers $\overline{ACKIN}$; the Buffer register is "emptied" and $\overline{DAV}$ goes High, indicating that data is no longer available. The data in the Output Data register is moved to the Buffer register and out of the port. At this time the next byte of data is available. However, the $\overline{DAV}$ signal cannot go Low until the $\overline{ACKIN}$ input is High, indicating that the receiving port is ready to accept another byte of data. This achieves the interlock.

The following example is a step-by-step analysis of a double-buffered output port using Interlocked Handshake. (This description uses Figure 3-9 as reference).

T_0. The Buffer register is full, data is valid on the pins, $\overline{DAV}$ is Low, and $\overline{ACKIN}$ is High.

T_1. $\overline{ACKIN}$ goes Low, indicating that the receiver has taken the data and is not ready for more data. This empties the Buffer register.

T_2. $\overline{DAV}$ goes High, indicating that the Buffer register is empty and the data is no longer valid.

T_3. The next byte to be output is moved into the Buffer register from the Output Data register. Data is valid at the port pins. The Deskew Timer is triggered if enabled.

T_4. The Deskew Timer times out, but $\overline{DAV}$ remains High because $\overline{ACKIN}$ is still Low (this is the interlock).

T_5. $\overline{ACKIN}$ goes High, indicating that the input port is ready for data.

T_6 becomes T_0. $\overline{DAV}$ goes Low, indicating that the data is valid and has been valid for Deskew time programmed; the cycle is ready to repeat.

Strobed Output Handshake

The Strobed Handshake (Figure 3-10) operates in the same way as the Interlocked Handshake, except that $\overline{DAV}$ goes Low independent of $\overline{ACKIN}$. That is, the output port can indicate that new data is available by $\overline{DAV}$ going Low (independent of $\overline{ACKIN}$). In all other respects, the two handshakes are the same. The falling edge of the $\overline{DAV}$ output indicates that the data is ready to be read (just like in Interlocked Handshake). The deskew timers can be used as described in output port with Interlocked Handshake above.

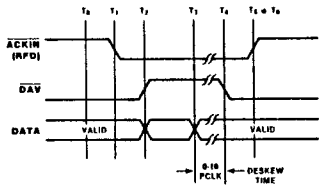


Figure 3-10. Strobed Output Handshake Timing Diagram

The following example is a step-by-step analysis of a double-buffered output port using Strobed Handshake. (This description uses Figure 3-10 as reference).

T_0. The Buffer register is full, data is valid on the pins, $\overline{DAV}$ is Low, and $\overline{ACKIN}$ is High.

T_1. $\overline{ACKIN}$ goes Low, indicating that the receiver has taken the data and is not ready for more data. This empties the Buffer register.

T_2. $\overline{DAV}$ goes High, indicating that the Buffer register is empty and the data is no longer valid.

T_3. The next byte to be output is moved into the Buffer register from the Output Data register. Data is valid at the port pins. The deskew timer is triggered if enabled.

T_4. The deskew timer times out, and $\overline{DAV}$ goes Low, independent of $\overline{ACKIN}$ (there is no $\overline{ACKIN}$ interlock).

T_5 becomes T_0. $\overline{ACKIN}$ goes High, indicating that the input port is ready for data; the cycle is ready to repeat.

Pulsed Output Handshake

The Pulsed Handshake operates exactly like the Interlocked Handshake with Counter/Timer 3 inserted internally in the $\overline{DAV}$ output path (see Figure 3-11). The timer is triggered on the falling edge of an internal $\overline{DAV}$ signal. The output of the timer is inverted and used as the Data Available output for the handshake. The interlock is between the $\overline{ACKIN}$ input and the internal $\overline{DAV}$ signal (the internal $\overline{DAV}$ cannot go Low until $\overline{ACKIN}$ is High). Because all the capabilities of Counter/Timer 3 are available for use, many operations are possible depending on how the timer is programmed. However, since only Counter/Timer 3 is used, only one port can have Pulsed Handshake at a time. The deskew timers can be used to delay the internal $\overline{DAV}$ signal as described before.

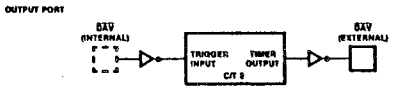


Figure 3-11. Pulsed Output Handshake Counter/Timer Insertion

If Counter/Timer 3's output duty cycle is programmed in the pulse mode, then TC cycles (where TC is the value programmed in the counter/timer Time Constant register) after the internal $\overline{DAV}$ falling edge is detected, the $\overline{DAV}$ output falls and the external $\overline{DAV}$ rises a cycle later.

If the counter is programmed with the one-shot duty cycle, then the $\overline{DAV}$ output falls as soon as the external falling edge on the internal $\overline{DAV}$ signal is detected; it rises TC cycles later. When the counter duty cycle is selected to be square-wave, the $\overline{DAV}$ output goes Low TC cycles after internal $\overline{DAV}$ falls and it stays Low for TC cycles. The duty cycle selected for Counter/Timer 3 determines which $\overline{DAV}$ outputs are available (see the timing diagrams in Figure 3-12).

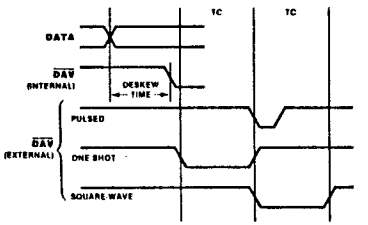


Figure 3-12. Pulsed Output Handshake Counter/Timer Duty Cycles

Many simple interfaces can be made with the Pulsed Handshake by linking the $\overline{DAV}$ output to the $\overline{ACKIN}$ input. For example, if the duty cycle selected for Counter/Timer 3 is square-wave, the port will provide valid data with a setup of TC clock cycles to the external $\overline{DAV}$ falling edge (even longer if the deskew timer is enabled). Also the $\overline{DAV}$'s Low time will be TC clock cycles. This could be used for interfacing to a device which requires a large data setup time to a strobe and a minimum time between characters.

3-Wire Output Handshake

The 3-Wire Handshake (Figure 3-13) is the same as the Interlocked Handshake, except that the role of the $\overline{ACKIN}$ input is replaced by two signals: Ready for Data (RFD) and Data Accepted (DAC). This nomenclature is consistent with the IEEE-488 specification.

When the output port Buffer register is full, its data is available to send. However, the 3-Wire Interlock requires that the receiver(s) first signal that it is ready for data by having DAC Low and raising RFD High (the interlock).

If the deskew timer is enabled, the deskew countdown starts with data moved into the Buffer register. On deskew timeout, the $\overline{DAV}$ signal goes Low; the data has been valid for the whole Deskew count. If the deskew timer is not enabled, the output port then immediately lowers $\overline{DAV}$, signaling that the data is available.

The input port puts RFD Low and, after accepting the data, DAC goes High. The rising edge of DAC "empties" the Buffer register (like $\overline{ACKIN}$ falling in the Interlocked and Strobed Handshakes). The output port then raises $\overline{DAV}$ High which causes the input port's DAC to return Low. New data can now move into the Buffer register in preparation for a new cycle. The new cycle begins when the input port(s) signals that it is ready for data by RFD going High (the interlock).

An output port using 3-Wire can be used as the source of data in a communication network that has one source (Talker) and many receivers (Listeners). The following is a description of the operation of such a network from the Talker point of view.

In the 3-Wire Handshake (Figure 3-13), separate signals are used to indicate when the Listeners are ready to receive new data (RFD), and when the data has been accepted (DAC). (In the Interlocked Handshake, the $\overline{ACKIN}$ input indicates both [High = Ready for Data, Low = Data Accepted]). Since two signals are used, many Listeners can be connected simultaneously to a single Talker by wire-ANDing their RFD and DAC outputs. In this way RFD will rise only when all Listeners are ready, and DAC will rise only after all Listeners have received the data.

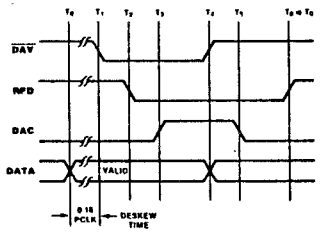


Figure 3-13. 3-Wire Output Handshake Timing Diagram

When $\overline{DAV}$ falls, the Listeners can receive the data at their own individual pace, indicating that they have received the data by bringing RFD Low, and letting their own DAC go High. Since DAC is a wired-AND, the DAC signal to the Talker will indicate when the last Listener has accepted the current data.

The Talker then tells the Listeners that the data is no longer valid ($\overline{DAV}$ goes High). The Listeners then respond by pulling DAC Low and, when they are ready, letting RFD go High. When the last Listener's RFD goes High, the wired-AND tells the Talker that all Listeners are again ready for new data, and the cycle is ready to begin again.

This procedure is in agreement with the IEEE-488 3-Wire Handshake. See the 3-Wire Handshake timing example (Section 3.4.2.2) for a step-by-step analysis of a port configured as double-buffered which uses 3-Wire Handshake.

3.4.4 Bidirectional Port Operation

The bidirectional port is both an input and an output port. That is, it handles data movement in either direction between the CPU and the CIO pins. The direction of data flow is controlled by a single line, the IN/$\overline{OUT}$ line, provided by Port C (see Table 3-1). If the line of Port C used as the direction control is programmed to be an input bit, the port is a "slave" bidirectional port. The direction of data flow is controlled by an external "master," and the IN/$\overline{OUT}$ input controls the port. (If the bit is programmed to be an output bit, the port is a "master" bidirectional port.) In this configuration, the CPU controls the direction of data flow and the IN/$\overline{OUT}$ line is a status line that indicates the direction of data flow. The operation of data transfer is the same for both "master" and "slave" ports--only the mechanism of changing direction is different. The details of the differences will be described in the following paragraphs.

The bidirectional port uses two lines in addition to the IN/$\overline{OUT}$ line for handshaking. One line, the RFD/$\overline{DAV}$ line, has multiple functions depending on the data direction. When acting as an input port, this line is the Ready For Data (RFD) output, like the simple input port. In the output mode, this line is the Data Available ($\overline{DAV}$) output, like the simple output port. The other line is the Acknowledge Input ($\overline{ACKIN}$). Since three of Port C's four lines are required for handshake and direction control lines, only one port may be bidirectional at any one time.

In the bidirectional mode, a port's Input register and Output register are both operational and independent. Even when the IN/$\overline{OUT}$ line is Low (output mode), reading the Port's data register gives the contents of Input Data register, not the value being output at the time (like a simple output port gives). Reading the port's data register empties the Input register; writing the data register fills the Output register, regardless of the direction of the port at the time of the read or write.

Both double- and single-buffered operations function in the bidirectional mode like simple input and output ports, depending on the direction of data flow.

Since the direction of data flow can be changed at almost any time and because a single Buffer register is multiplexed between the input and output path, any data in the Buffer register is somewhat vulnerable. The user must take precautions that data is not lost.

Only two handshakes are supported by a port in the bidirectional mode: the Interlocked Handshake and the Strobed Handshake. The Pulsed Handshake and the 3-Wire Handshake must not be specified. Like the simple input and output ports, the only difference between the Interlocked and the Strobed handshakes is whether or not the $\overline{ACKIN}$ input influences the operation of the RFD output (input mode) or $\overline{DAV}$ output (output mode). The operation of the bidirectional port will be explained by describing in detail the input operation, input-to-output transition, output operation, and output-to-input transition for a port using the Interlocked Handshake. Differences encountered using the Strobed Handshake will be described as they occur.

NOTE

When in the bidirectional operation condition, the Interrupt on Two Bytes (ITB) bit of the Port Mode Specification register must be set to 0 (zero).

3.4.4.1 Input Operation

The Bidirectional port operating in the input direction operates like a simple input port, except for how IP is cleared and how the Input register is emptied.

On the falling edge of the Acknowledge Input ($\overline{ACKIN}$) the input data is latched in the Buffer register and RFD/$\overline{DAV}$ output is pulled Low, indicating that the data has been accepted. If the Input Data register is empty, the data is transferred to the Input Data register, the Input Register Full (IRF) flag is set to 1, the Buffer register is emptied (if not single-buffered), and IP is set to 1. When the Buffer register becomes empty, the RFD/$\overline{DAV}$ returns High, if $\overline{ACKIN}$ is High. If the Strobed Handshake is specified, it goes High independent of $\overline{ACKIN}$'s state.

In bidirectional input operation, the IP cannot be cleared automatically. It must be cleared by writing to the Port Command and Status Register. Also, the Input register (and Buffer register, if single-buffered) will remain "full" until the Input Data register is read and IP is cleared (in any order).

3.4.4.2 Input to Output Direction Change

The direction can safely be changed at any time except for the period between the falling edge of $\overline{ACKIN}$ and the time that RFD/$\overline{DAV}$ falls, indicating the reception of input data. As long as any input data is in the Buffer register or any input interrupts are pending, the port remains in the input mode internally. As the data is shifted into the Input Data register, IP is set; the port remains an input port until that IP is cleared. While the internal state of the port differs from the IN/$\overline{OUT}$ line, the RFD/$\overline{DAV}$ line is forced High, indicating that no output data is available. If the IN/$\overline{OUT}$ line is returned High during this time, RFD/$\overline{DAV}$ may go Low to indicate that the port is not ready for input data and that a false RFD/$\overline{DAV}$ falling edge may be generated. This problem can be alleviated if the input-to-output transition is not made while the RFD/$\overline{DAV}$ line is Low. When all input interrupts have been recognized, (IP's cleared), the port becomes an output port and operates normally. Until this time, all port data lines remain input (the output drivers will not go active).

If the port becomes an output port and the Output Data register is empty, IP is automatically set just as if the Output register had become empty as a result of an output handshake.

3.4.4.3 Output Operations

A bidirectional port, operating in the output direction, functions like a simple output port

except for the way in which IP is cleared and the Output register is "filled".

The output operation starts by writing a byte to the port's data register. If IP = 0, this fills the Output Data register and clears the Output Register Empty (ORE) flag to 0. If the Buffer register is empty, the data is moved to the Buffer register (and, therefore, to output on the port pins), and, if specified, the optional deskew timer is triggered. If the port is double-buffered (SB = 0), the Output Data register is "emptied" and both ORE and IP are set to 1. If the port is single-buffered (SB = 1), the Output Data register is not "emptied" when the data is moved into the Buffer register. If $\overline{ACKIN}$ is High, the RFD/$\overline{DAV}$ output is pulled Low either immediately, or after the deskew timer times out. If the Strobed Handshake is specified, RFD/$\overline{DAV}$ goes Low independent of the state of $\overline{ACKIN}$. When $\overline{ACKIN}$ goes Low, indicating that the data has been received, the Buffer register is emptied, and the RFD/$\overline{DAV}$ line is forced High. If the port is single-buffered, this also "empties" the Output Data register and causes both IP and ORE to be set to 1.

In bidirectional output operation, IP cannot be cleared automatically. It can only be cleared by writing to the Port's Command and Status register. Also, the Output Data register remains "empty" until both the Output Data register is written and IP is cleared (in any order).

3.4.4.4 Output to Input Direction Change

The change in direction from output to input can safely take place at any time, except for the time between the falling edge of $\overline{ACKIN}$ and the rising edge of RFD/$\overline{DAV}$ (which indicates that the receipt of the data is recognized). As soon as the direction change to input is recognized, the port becomes an input port. The output drivers are forced to high impedence and the Buffer register is emptied (if full) to prepare it for an input data byte. Any data that is in the Buffer register is lost. If the direction change could occur at any time (and possibly destroy Buffer register data), the port should be specified to be single-buffered. In this case, a copy of the data to be output is contained in the Output Data register, and the data is not lost. When the Buffer register is cleared, the port is ready to receive new data immediately and RFD/$\overline{DAV}$ is output High. Although the port is ready to receive data in the Buffer register, it will not be moved to the Input Data register until IP is cleared. This guarantees that no interrupts are missed on direction change.

It is recommended that a "master" bidirectional port only change directions to input when the RFD/$\overline{DAV}$ output is High. This can be done by programming the port to be single-buffered and then changing directions in response to an output handshake interrupt. At this time, the previous output handshake is completed and RFD/$\overline{DAV}$ is High.

If the Input Data register is full when the transition to an input port takes place, an interrupt sequence is initiated just as if the register had become full as a result of an input handshake. (This is delayed until any output IP is cleared by software.)

3.4.4.5 Pattern Match

The operation of the pattern match logic (if enabled) in the bidirectional mode is essentially the same as combining the input and output modes. Each time IP is set, the PMF is updated, based on whether or not the byte transferred into the Input Data register or out of the Output Data register matches the pattern. IP is set and cleared like a normal bidirectional port.

NOTE

The ITB bit must be 0 during bidirectional operation. If the port is programmed to be single-buffered (SB = 1), the Interrupt on Match Only (IMO) bit must not be 1.

3.4.5 REQUEST/$\overline{WAIT}$ Line Operation

When used as a port with handshake, Port C can be programmed to provide a status signal output in addition to the normal handshake lines for either Port A or B. The additional signal is either a REQUEST or $\overline{WAIT}$ signal. The REQUEST signal indicates when a port is ready to perform a data transfer via the Z-BUS. It is intended for use by a DMA-type device. The $\overline{WAIT}$ signal provides synchronization for transfers with a CPU. Three bits in the Port Handshake Specification register provide controls for the REQUEST/$\overline{WAIT}$ logic. Because an extra Port C line is used, only one port can be specified to be a port with a handshake and a REQUEST/$\overline{WAIT}$ line. The other port must be a bit port.

3.4.5.1 REQUEST Line Operation

Operation of the REQUEST line is dependent on the state of the port's Interrupt on Two Bytes (ITB) control bit. When ITB = 0, the REQUEST line goes active as soon as the CIO is ready for a data transfer. If the port is used for the input, the REQUEST line goes High when the Input Data register is full. If the port is used for output, the REQUEST line goes High when the Output Data register is empty. If ITB = 1, REQUEST goes active only if two bytes can be moved. For input, both Input Data and Buffer registers are full, and for output, both Buffer and Output Data registers are empty. REQUEST stays active as long as a byte is available to be read or written. However, if the port is single-buffered or if the Pattern Match Flag is set, REQUEST goes Low when the data is read or written.

In the bidirectional mode, ITB must be 0. Therefore, the REQUEST line reflects the state of the Input or Output Data register, depending on the type of REQUEST line specified.

The DMA-type transfer is facilitated by the use of the Interrupt on Match Only (IMO) control bit in the Port Mode Specification register. In most DMA transfers, the peripheral does not generate an interrupt request. However, the Interrupt on Match Only (IMO) capability of the CIO allows it to interrupt the CPU when there is a specified byte match, such as an end of transfer flag, etc. Except for the specified pattern match, the CPU is not involved in the transfer; the data movement is consequently accomplished much faster. This REQUEST line/IMO pattern match operation can function in all port operating modes.

The SPECIAL REQUEST function is reserved for use with bidirectional ports only. In this case, the REQUEST line indicates the status of the register not being used in the data path at that time. If the IN/$\overline{OUT}$ line is High (input port), the REQUEST line is High when the Output Data register is empty. If IN/$\overline{OUT}$ is Low (output port), the REQUEST line goes High when the Input Data register is full. In this mode, both the Input and the Output Data registers can be monitored. The RFD/$\overline{DAV}$ indicates the state of the register in the data path, and the REQUEST line monitors the register not in the data path.

This line can be used to indicate when to change the CIO direction. For example, if the CIO is a "slave" bidirectional port in the input direction (IN/$\overline{OUT}$ = High), then when the SPECIAL REQUEST line goes Low (indicating that the Output Data register has a byte to be transferred), the master port can use this as the signal to turn the port around so that the byte can be output.

3.4.5.2 $\overline{WAIT}$ Line Operation

The REQUEST/$\overline{WAIT}$ line configured in the $\overline{WAIT}$ mode is useful to synchronize CPU-to-CIO transactions. For example, when the CPU wants the data transfer to be performed as rapidly as possible, it does not want to wait for an interrupt service routine to tell it that the CIO is ready to receive or supply a byte of data. Rather, the CPU attempts a read or write to the CIO. If $\overline{WAIT}$ is enabled, and the CIO is not ready for a read or write to it, $\overline{WAIT}$ is pulled Low and the CPU is forced to wait until the CIO is ready (the Input register becomes full or the Output register becomes empty), signified by the $\overline{WAIT}$ line going High. The CPU is now released from the $\overline{WAIT}$ pause and can complete the data transaction.

For an input port, $\overline{WAIT}$ is pulled Low when an attempt is made to read the Input Data register and the port is empty. For an output port, $\overline{WAIT}$ is pulled Low when an attempt is made to write to the Output Data register that is still full; the data integrity is maintained in the case of the output port--the data is not overwritten. Action is merely suspended until the write can take place.

In the Z8036, $\overline{WAIT}$ falling is caused by $\overline{DS}$ falling. In the Z8536, $\overline{WAIT}$ is synchronized with either $\overline{RD}$ or $\overline{WR}$ falling. $\overline{WAIT}$, however, may be required to be valid at the CPU prior to this. A practical way to use $\overline{WAIT}$ with the CIO is to have external logic generate a $\overline{WAIT}$ cycle automatically, which can then be extended as needed by the CIO.

The release of $\overline{WAIT}$ in both the Z8036 and the Z8536 is synchronized with the PCLK input. Thus for the Z8036 REQUEST/$\overline{WAIT}$ function, a PCLK input is required. Also, while the Z8036 is asserting $\overline{WAIT}$, internal status is updated using PCLK (not $\overline{AS}$ as it normally does), because $\overline{AS}$ stops as long as the CPU is WAITed.

3.4.6 Linked Port Operation

Ports A and B can be linked to form a 16-bit port by programming a 1 in the Port Link Control (PLC) bit of the Master Configuration Control register. In this mode, only Port A's Handshake Specification and Status registers are used. Port B must be specified as a bit port. When linked, only Port A has pattern-match capability. Port B's pattern-match capability must be disabled. Also, when the ports are linked, Port B's Data register must be read or written before Port A's. The PLC bit must be set to 1 before the ports are enabled.

Chapter 4
Counter/Timer Operation

4.1 COUNTER/TIMER ARCHITECTURE

The three independent 16-bit counter/timers each consist of a presettable 16-bit down-counter, a 16-bit Time Constant register, a 16-bit Current Counter register, an 8-bit Mode Specification register, an 8-bit Command and Status register, and the associated control logic that links these registers.

The flexibility of the counter/timers is enhanced by the provision of up to four lines per counter/timer (counter input, gate input, trigger input, and counter/timer output) for direct external control and status. Counter/Timer 1's external I/O lines are provided by the four most-significant bits of Port B. Counter/Timer 2's external I/O lines are provided by the four least-significant bits of Port B. Counter/Timer 3's external I/O lines are provided by the four bits of Port C. The utilization of these lines (Table 4-1) is programmable on a bit-by-bit basis via the Counter/Timer Mode Specification registers.

When external counter/timer I/O lines are to be used, the associated port lines must be vacant and programmed in the proper data direction. Lines used for counter/timer I/O have the same characteristics as simple input lines. They can be specified as inverting or non-inverting, and can be read and used with the pattern-recognition logic. They can also include the 1's catcher input.

4.2 COUNTER/TIMER SEQUENCE OF EVENTS

The following discussion assumes that the inputs and outputs are programmed non-inverting.

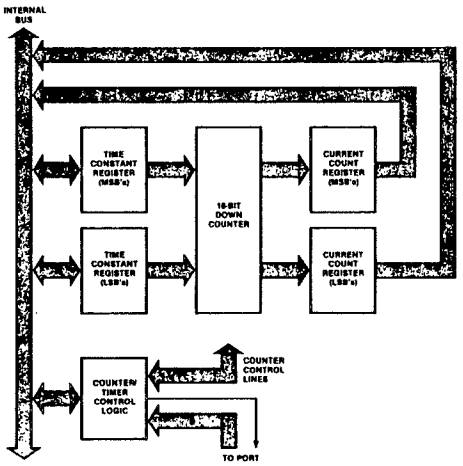


Figure 4-1. Counter/Timer Block Diagram

Table 4-1. Counter/Timer External Access

Function	Counter/ Timer 1	Counter/ Timer 2	Counter/ Timer 3
Counter/Timer Output	Port B 4	Port B 0	Port C 0
Counter Input	Port B 5	Port B 1	Port C 1
Trigger Input	Port B 6	Port B 2	Port C 2
Gate Input	Port B 7	Port B 3	Port C 3

4.2.1 Initializing the Counter/Timer

Before starting a counter/timer sequence:

First, the Counter/Timer Mode Specification register and the Counter/Timer Command and Status register of the desired counter/timer must be initialized. Initialization requires several things to be specified, for example, the external lines to be used, the output duty cycle, and whether the cycle is continuous or single-cycle.

Second, the Time Constant must be specified by writing the desired value to the Time Constant register. The Time Constant register is accessed as two 8-bit registers. The registers are readable as well as writeable, and can be accessed in any order. A 0 in the Time Constant register specifies a Time Constant of 65,536.

Third, if external access is going to be provided, the port to be used must be programmed as a bit port and the necessary bits must be programmed in the proper direction (see Section 3.3).

Finally, the Counter/Timer Enable bit in the Master Configuration Control register is set. This initialization sequence can best be understood by examining the function of the various enable bits. This bit, while cleared to 0, prevents spurious counter/timer operation:

- IPs can not be set.
- Counter/Timers can not be triggered.
- The Read Current Count bit that freezes the value in the Current Count register will be held cleared to 0.
- The Counter/Timer output is forced to 0.

Clearing an enable bit will not clear an existing IP that is set--it will only inhibit the IP from being set again. Clearing the enable bit will clear the Read Counter Control bit, causing the Current Count register to follow the down-counter.

4.2.2 Starting the Counter/Timer

The countdown sequence is initiated when the counter/timer is triggered and the down-counter is loaded with the contents of the Time Constant register. The down-counter is normally loaded on the rising edge of the external trigger input, or by writing a 1 to the Trigger Command Bit (TCB) of the Command and Status register. But, for Counter/Timer 2 only, triggering can occur on the falling edge of Counter/Timer 1's internal output if the counters are linked via the trigger input. Also, Counter/Timer 3 can be triggered by the handshake logic when it is used with the Pulsed Handshake (see Section 3.4.1.3).

The trigger functions as the logical OR of all the potential triggers (see Figure 4-2). Since the trigger function is an OR function, and since it is rising-edge sensitive, any input remaining in its active state will mask off other trigger sources as it stays High.

NOTE

In order to ensure the loading of a Trigger Constant on a particular rising edge of the clocking signal, sufficient setup time must be allowed--the trigger must occur prior to the immediately preceding falling edge of the clocking signal. (The clocking signal equals the count input if in Counter mode or PCLK/2 if in Timer mode.)

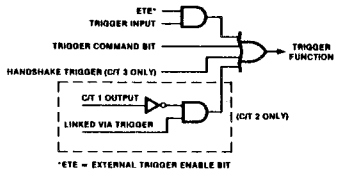


Figure 4-2. Trigger OR-Function Diagram

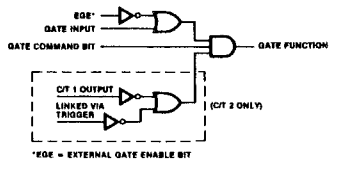


Figure 4-3. Gate AND-Function Diagram

4.2.3 Countdown Sequence

The rate at which the down-counter counts is determined by the mode of the counter/timer. In the Timer mode (the External Count Enable [ECE] bit is 0), the down-counter is clocked internally by a signal that is half the frequency of the PCLK input to the chip. In the Counter mode (ECE is 1), the down-counter is decremented on the rising edge of the counter/timer's counter input.

Once the down-counter is loaded, the countdown sequence continues toward terminal count as long as all of the counter/timers' hardware and software gate inputs are High. The gate inputs are: the Gate Command bit of the Counter/Timer Command and Status register, and the external gate input if enabled in the External Gate Enable bit of the counter/timer Mode Specification register. Also, for Counter/Timer 2 use only, the counter/timer output (inverted) can be used as a gate if linked via the gate in the Counter/Timer Link Controls bits of the Master Configuration Control register. If any of the gate inputs go Low (0), the countdown halts. It resumes when all gate inputs are 1 again. The gate function does not affect the trigger function.

The gate functions as the logical AND of all the potential gates (see Figure 4-3).

NOTE

In order to ensure the enabling or disabling of the counter/timer on a particular rising edge of the clocking signal, sufficient setup time must be allowed. The gate signal must be valid prior to the immediately preceding falling edge of the clocking signal.

The reaction to triggers occurring during a countdown sequence is determined by the state of the Retrigger Enable Bit (REB) in the Mode Specification register. If REB is 0, retriggers are ignored and the countdown continues normally. If REB is 1, each trigger causes the down-counter to be reloaded and the countdown sequence starts over again. If the output is programmed in the Square-Wave mode, a retrigger causes the sequence to start over from the initial load of the time constant.

The state of the down-counter can be determined in two ways: by reading the contents of the down-counter via the Current Count register or by testing the Count In Progress (CIP) status bit in the Command and Status register. The CIP status bit is set when the down-counter is loaded; it is reset when the down-counter reaches 0. The Current Count register is a 16-bit register, accessible as two 8-bit registers, which mirrors the contents of the down-counter. This register can be read anytime. However, reading the register is asynchronous to the counter's counting, and the value returned can be guaranteed as valid only if the counter is stopped. The down-counter can be read reliably while it is counting by first writing a 1 to the Read Counter Control (RCC) bit in the counter/timer's Command and Status register. This freezes the value in the Current Count register until a read of the least-significant byte is performed. A read of RCC indicates if the CCR is holding a value, or if it is following the down-counter.

4.2.4 Ending Condition

The Continuous/$\overline{\text{Single Cycle}}$ (C/$\overline{\text{SC}}$) bit in the Mode Specification register controls operation of the down-counter when it reaches terminal count (the count following the count of 1). If C/$\overline{\text{SC}}$ is 0

when a terminal count is reached, the countdown sequence stops. If the C/$\overline{SC}$ bit is 1 each time the count-down counter reaches 1, the next cycle causes the time constant value to be reloaded. The time constant value may be changed by the CPU, and on reload, the new time constant value is loaded. **This must be done with care.**

Each time the counter reaches terminal count, its Interrupt Pending (IP) bit is set to 1, and if interrupts are enabled (IE = 1), an interrupt request is generated. If a terminal count occurs while IP is already set, an internal error flag is set. As soon as IP is cleared, it is forced to a 1 along with the Interrupt Error (ERR) flag. Errors that occur after the internal flag is set are ignored. ERR is cleared to 0 when the corresponding IP is cleared.

4.2.5 Counter/Timer Output

There are three duty cycles available for the timer/counter output: pulse, one-shot, and square-wave. Figure 4-4 shows the counter/timer timing diagrams. When the Pulse mode is specified, the output goes High for one cycle, beginning when the down-counter leaves the count of 1. In the One-Shot mode, the output goes High when the counter/timer is triggered and goes Low when the down-counter reaches 0. When the square-wave output duty cycle is specified, the counter/timer goes through two full sequences for each cycle. The initial trigger causes the down-counter to be loaded and the normal count-down sequence to begin. When a 1 count is detected on the down-counter's clocking edge, the output goes High and the time constant value is reloaded. On the clocking edge, when both the down-counter and the output are 1's, the output is forced low.

4.2.6 Linked Sequence

Counter/timers 1 and 2 can be linked internally in three different ways. Counter/timer 1's output (inverted) can be used as Counter/timer 2's trigger, gate, or counter input. When linked, the counter/timers have the same capabilities as when used separately. However, when they are linked, they should be linked before they are enabled. The only restriction is that when Counter/timer 1 drives Counter/timers 2's count input, Counter/timer 2 must be programmed with its external count input disabled (ECE = 0).

The initialization procedure, then, is the same as for individual counter/timers, except that the linking bits need to be appropriately set.

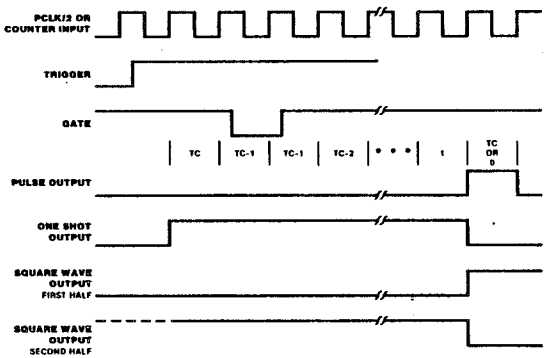


Figure 4-4. Counter/Timer Timing Diagram

Chapter 5
Interrupt Operation

5.1 OVERVIEW

Interrupts are generated whenever CPU intervention is required by a peripheral device. Three examples of interrupt request sources in the CIO are: a pattern match occurring in a bit port; another byte becoming available in an input port with handshake; and a counter/timer reaching its terminal count.

The operation of the Z8036 is compatible with the Z-BUS specification (see Zilog Data Book) and all members of the Z8000 family. The operation of the Z8536 is similar to the Z-BUS specification, which, with a minimum amount of external logic, can be interfaced to a Z80 and its family of peripherals.

Interrupt operation is affected by three things: the external device pins, internal register bit settings, and Interrupt Acknowledge transaction including vector response.

5.2 PRIORITY HANDLING AND THE CIO

The CIO is designed to provide interrupt priority resolution in situations where there may be competing interrupt requestors.

Interrupt priority resolution can be accomplished by using either a separate interrupt controller device or a daisy-chain. The interrupt controller device handles priority resolution and interrupt request generation for all the peripherals in a system. In contrast, the daisy-chain structure uses serial daisy-chain pin connections to establish the priority of the interrupt devices, thus distributing the decision-making among these peripheral devices. The interrupt priority of a device is determined by its position in the daisy-chain.

One of the features of the CIO is that it allows both of the above interrupt schemes to be used in the same CPU environment. The CIO contains all the logic necessary for it to be used in both the Z-BUS and the Z80 BUS daisy-chain interrupt structures. This logic generates interrupts, resolves interrupt priority, inhibits preemption by lower-priority interrupts, and identifies the source of interrupt. The OR-PEV mode pattern recognition logic of the ports (see Section 3.2) allows the CIO to act as an interrupt controller, facilitating the use of devices which do not have the necessary interrupt logic. (See Figure 5-1.)

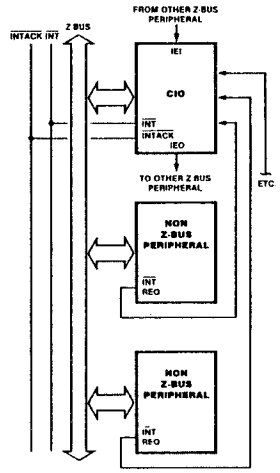


Figure 5-1. The CIO as an Interrupt Controller

The CIO has five potential sources of interrupts: the three counter/timers and Ports A and B. The priorities of these sources are fixed in the following order (highest to lowest): Counter/timer 3, Port A, Counter/timer 2, Port B, and Counter/timer 1. Since the counter/timers all have equal

capabilities and Ports A and B have equal capabilities, there is no adverse impact from the fixed priorities.

The CIO interrupt priority, relative to other components within the system, is determined by the interrupt daisy-chain. Two pins, Interrupt Enable In (IEI) and Interrupt Enable Out (IEO), provide the input and output necessary to implement the daisy-chain. When IEI is pulled Low by a higher-priority device, the CIO cannot request an interrupt of the CPU. The IEO output is connected the IEI input of the next lower-priority device on the daisy-chain. IEO is forced Low to inhibit interrupts from all lower-priority devices. The following discussion assumes that the IEI input is High.

5.3 THE FOUR INTERRUPT LOGIC FUNCTIONS

The CIO has the logic necessary to: generate interrupts, resolve priority when there is more than one interrupt requestor, inhibit preemptive interrupts by the lower-priority requestors, and clearly identify the exact source of interrupt.

5.3.1 Generating the Interrupt Request

Each source interrupt in the CIO contains three bits for the control and status of the interrupt logic: an Interrupt Pending (IP) bit, an Interrupt Under Service (IUS) bit, and an Interrupt Enable (IE) bit. IP is automatically set when an event requiring CPU intervention occurs. (Chapters 3 and 4 describe in detail how a particular IP is normally set and cleared.) The setting of IP results in an Interrupt Request ($\overline{INT}$) output pulled Low if all other conditions are met. (IP can also be set by a command. This is useful when debugging interrupt handler software.)

The IE bit provides the CPU with a means of masking off individual sources of interrupts. When IE is set to 1, an interrupt request is generated normally. When IE is set to 0 the IP is masked off. The IP bit is still set when an event occurs that would normally require service; however, the $\overline{INT}$ output is not pulled Low.

The IUS status bit is set by the CPU as a result of the Interrupt Acknowledge cycle if, at the time of the Interrupt Acknowledge cycle, the corresponding IP is the highest-priority unmasked IP. (The details of setting and clearing IUS are described in Section 5.3.3.) When IUS is 1, it indicates that the corresponding IP has been recognized by the CPU and is being serviced. As long as IUS is set the corresponding IP is masked off and the IEO output is forced Low.

The Master Interrupt Enable (MIE) bit allows all sources of interrupts within the CIO to be disabled without having to individually clear each IE to 0. If MIE is set to 0, all IPs are masked off and no interrupt can be requested or acknowledged.

The IEI input is also involved in the control of interrupt generation. If IEI is Low, it indicates that a higher-priority interrupt is being serviced (that is, a higher-priority IUS is set to 1). An interrupt request can be made only when IEI is High.

In summary, for a device with one source of interrupt, $\overline{INT}$ is pulled Low and an interrupt request is generated only when IP and IE are 1 and IUS is 0, MIE is 1, and IEI input is 1. For a device with many sources of interrupt (for example, the CIO), there is an internal IEI-IEO daisy-chain that determines the internal interrupt priority. The internal IEI for the particular interrupt source must also be a 1.

5.3.2 Priority Resolution

The CPU responds to an interrupt request by generating an Interrupt Acknowledge cycle to determine the source of the interrupt. The first part of the cycle (from Interrupt Acknowledge [$\overline{INTACK}$] falling, until Data Strobe [$\overline{DS}$] [in the Z8036] or Read [$\overline{RD}$] [in the Z8536] goes Low) is used to determine which requestor has the highest priority. More than one device may have all the conditions satisfied for pulling $\overline{INT}$ Low. As soon as the Interrupt Acknowledge begins, all devices that have an unmasked (IE = MIE = 1, IUS = 0) IP set will pull their IEO Low. The Low will ripple down the daisy-chain, disabling all lower devices by forcing their IEIs Low. When the daisy-chain settles, only one source of interrupt will have an unmasked IP with its IEI High--this is the interrupt source being acknowledged.

During the Interrupt Acknowledge cycle, IPs cannot be set, so the daisy-chain has an opportunity to stabilize. To satisfy this restriction in the Z8036, IPs are set only when Address Strobe ($\overline{AS}$) goes Low. In the Z8536, IPs are set by PCLK during State 0. (This is why the $\overline{ACKIN}$ input must be synchronous with PCLK in the Z8536.)

5.3.3 Inhibiting Preemption by Lower-Priority Sources

When $\overline{DS}$ (Z8036) or $\overline{RD}$ (Z8536) falls during an Interrupt Acknowledge cycle, the IUS corresponding to the highest unmasked IP is automatically set to 1. As long as IUS is set, IEO is held Low, prohibiting interrupt requests from lower-priority interrupt sources. This guarantees that an interrupt service routine will not be interrupted to service a lower-priority interrupt. IUS can be reset to 0 only by writing to the corresponding Counter/Timer or Port Command and Status register. It is not cleared automatically. IUS can be cleared before interrupt servicing is complete if lower-priority interrupts wish to be recognized. However, IP must be cleared or a second interrupt request will be generated.

The Disable Lower Chain (DLC) bit is included to allow the CPU to modify the system daisy-chain. When the DLC bit is set to 1, the CIO's IEO is forced Low (independent of the state of the CIO or its IEI input) and interrupts from all lower-priority devices are disabled.

Daisy-chain operation is handled differently between the Z8000 peripherals and the Z80 peripherals--however, they are compatible. (Refer to Interfacing 8500 Peripherals to the Z80, Microcomputer Applications Reference Book, document #00-2145-01). The CIO forces IEO Low when IEI is Low, or when an IUS is 1 (except during an Interrupt Acknowledge cycle with an unmasked IP = 1 when IEO is also forced Low).

The Z80 peripherals (CTC, PIO, DMA, and SIO) normally force IEO Low if IEI is Low, or if either IP or IUS is set. However, they use the Z80 Return from Interrupt Instruction (RETI ED_H - $4D_H$) to automatically clear the highest IUS set. To implement this when an ED is decoded as the first byte of an instruction fetch, Z80 peripherals inhibit IP from affecting the daisy-chain.

Although the daisy-chains are different, during critical times (during an Interrupt Acknowledge or when a RETI instruction is executed); they are the same and are therefore compatible.

5.3.4 Identification of the Highest-Priority Interrupt Request; The Use of Vectors.

As part of the Interrupt Acknowledge cycle, the CIO is capable of responding with an 8-bit interrupt vector that specifies the highest-priority interrupt requestor (see Table 5-1). The vector is output when $\overline{DS}$ (Z8036) or $\overline{RD}$ (Z8536) goes Low and IUS is set. The identification vector is a key item of the Z8000 family interrupt handling logic. It speeds the information passing and can, if desired, include additional status information identifying the cause of the interrupt as well as the source identification.

Table 5-1. Interrupt Vector Encoding if Vector Includes Status

D_3	D_2	D_1	
Port Vector Status			
OR-Priority Encoded Vector Mode:			
x	x	x	Number of highest-priority bit with a match
All Other Modes:			
D_3	D_2	D_1	
ORE	IAF	PMF	Normal
0	0	0	Error
Counter/Timer Status			
	D_2	D_1	
	0	0	Counter/Timer 3
	0	1	Counter/Timer 2
	1	0	Counter/Timer 1
	1	1	Error

The CIO contains three vector registers: one for Port A, one for Port B, and one shared by the three counter/timers. Unique identification information can be placed by the user in the Interrupt Vector register for each interrupt source needed during initialization. The vector output can be modified to include status information to pinpoint the cause of interrupt. A Vector Includes Status (VIS) control bit controls whether or not the vector includes status.

Each base vector has its own VIS bit and is controlled independently. When MIE = 1, reading the base vector register always includes status, independent of the state of the VIS bit. All the information obtained by the vector, including status, can thus be obtained with one additional instruction when VIS is set to 0. When MIE = 0,

reading the vector register returns the unmodified base vector so that it can be verified.

Another register, the Current Vector register, facilitates the use of the CIO in a polled environment. When read, the data returned is the same as the interrupt vector that is output in an Interrupt Acknowledge, based on the highest-priority IP set. If no unmasked IPs are set, the value FF_H is returned. The Current Vector register provides a simple way to poll all IPs in a single read.

The No Vector (NV) control bit of the Master Interrupt Control register, when set to 1, inhibits the outputting of an interrupt vector during an $\overline{INTACK}$ cycle. The NV bit does not affect the setting of the IUS operation. The only thing that the NV does is prevent the vector from being output on the bus.

5.4 Z-BUS INTERRUPT OPERATION

Figure 7-5 displays Interrupt Acknowledge timing. The Z8036 generates an interrupt request by lowering the $\overline{INT}$ line only if:

- Such interrupt requests are enabled (IE is 1, MEI is 1).
- It has an interrupt pending (IP = 1).
- It does not have an interrupt under service (IUS is 0).
- No higher-priority interrupt is being serviced (IEI is 1).

Figure 5-2 shows a typical Z-BUS interrupt arbitration setting.

IEO is not pulled down by the Z8036 at this time; IEO continues to follow IEI until an Interrupt Acknowledge transaction occurs.

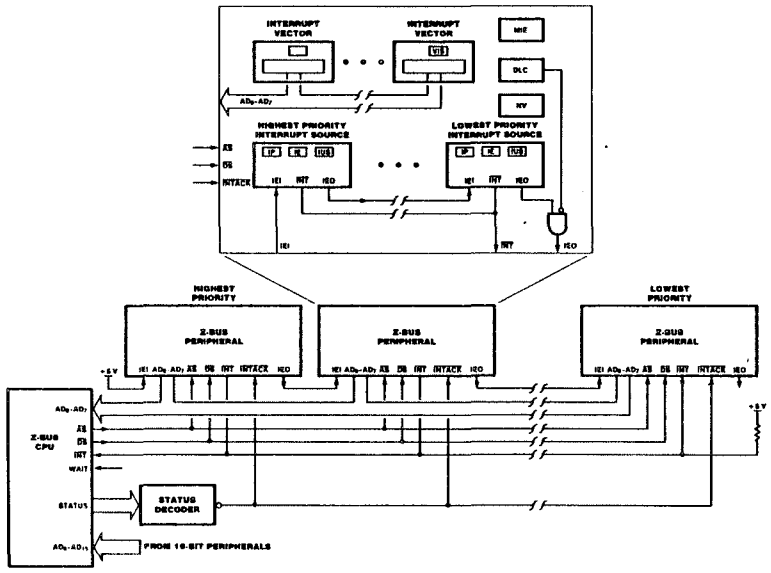


Figure 5-2. Z-BUS Interrupt Arbitration

Some time after $\overline{INT}$ has been pulled Low, the CPU initiates an Interrupt Acknowledge transaction. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the IEI/IEO daisy-chain settles. Any Z-BUS peripheral with one of its interrupts pending (IP is 1) or one of its interrupts under service (IUS is 1) holds its IEO line Low; all others make IEO follow IEI.

When $\overline{DS}$ falls, only the highest-priority interrupt source with a pending interrupt (IP is 1) has its IEI input High, its IE bit set to 1, and its IUS bit set to 0. This is the interrupt source being acknowledged, and at this time it sets its IUS bit to 1. If its NV bit is 0, the Z8036 identifies itself by placing its interrupt vector from the corresponding interrupt vector register onto address/data lines AD_0-AD_7. If NV is 1, the Z8036 AD_0-AD_7 lines remain floating, allowing external logic to supply a vector. (All Z-BUS interrupts require a vector to identify the requesting device.)

If the corresponding Z8036 VIS is 1, the vector also contains status information, (see Table 5-1) which further identifies the source of the interrupt within the Z8036. If VIS is 0, the vector held in the interrupt vector register is output without status included (base vector).

5.5 NON-Z-BUS INTERRUPT OPERATION

Figure 8-5 shows the non-Z-BUS Interrupt Acknowledge cycle timing. Figure 5-3 displays a Z80 system using Z8500 and Z80 devices. (Section 8.5 describes generation of $\overline{INTACK}$ from Z80 signals.) Figure 5-4 shows the external logic required for interfacing Z80 to Z8500 peripherals.

For the Z8536, the IP bit is not set while the device is in State 1 (Refer to Section 2.3 for a description of state conditions.) Therefore, to minimize interrupt latency, **the Z8536 should not be left in State 1.**

The Z8536 generates an interrupt request by lowering the $\overline{INT}$ line only if:

- Such interrupt requests are enabled (IE is 1, MIE is 1).
- It has an interrupt pending (IP = 1).
- It does not have an interrupt under service (IUS is 0).
- No higher-priority interrupt is being serviced (IEI is 1).

IEO is not pulled down by the Z8536 at this time; IEO continues to follow IEI until an Interrupt Acknowledge transaction occurs.

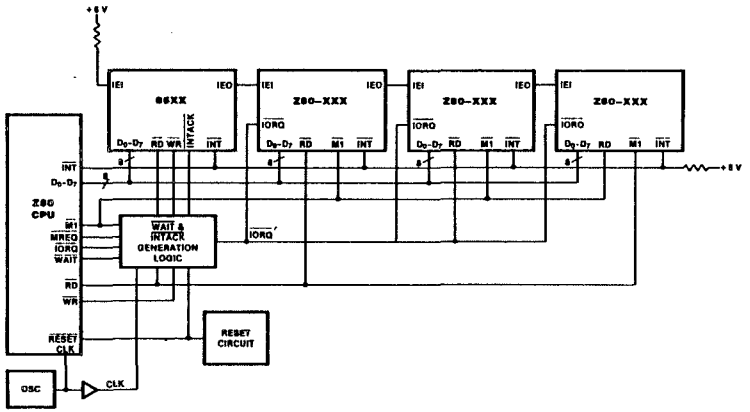


Figure 5-3. Non Z-Bus Interrupt Arbitration

Some time after $\overline{INT}$ has been pulled Low, the CPU initiates an Interrupt Acknowledge transaction. Between the falling edge of $\overline{INTACK}$ and the falling edge of $\overline{RD}$, the IEI/IEO daisy-chain settles. Any peripheral with one of its interrupts pending (IP is 1) or one of its interrupts under service (IUS is 1) holds its IEO line Low; all other conditions make IEO follow IEI.

When $\overline{RD}$ falls, only the highest-priority interrupt source with a pending interrupt (IP is 1) has its IEI input High, its IE bit set to 1, and its IUS bit set to 0. This is the interrupt source being acknowledged, and at this point it sets its IUS bit to 1. If its NV bit is 0, the Z8536 identifies itself by placing its interrupt vector from the corresponding interrupt vector register on data lines D_0-D_7. If NV is 1, the Z8536's D_0-D_7 lines remain floating, allowing external logic to supply a vector.

If the Z8536 VIS is 1, the vector also contains status information (see Table 5-1) which further identifies the source of the interrupt within the Z8536. If VIS is 0, the vector held in the interrupt vector is output without status included (base vector). The bit codes are in Section 2.9.1.

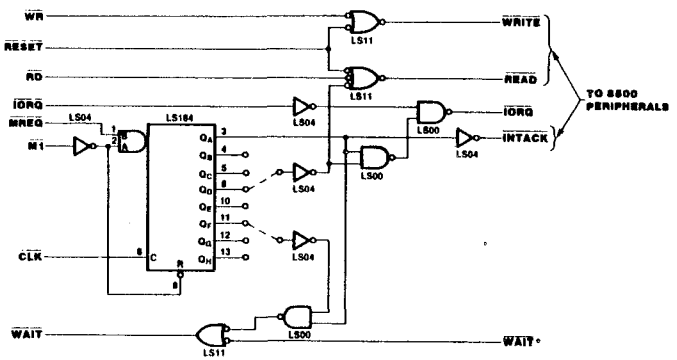


Figure 5-4. $\overline{WAIT}$ and $\overline{INTACK}$ Generation Logic

Chapter 6
Z-CIO/CIO Initialization

6.1 INTRODUCTION

This chapter discusses reset, initialization, and programming for both the Z8036 Z-CIO and the Z8536 CIO.

The normal sequence for initializing this device is simple and straightforward:

1. Reset the device
2. Clear RESET
3. Program the various functions
4. Set the appropriate Enable bits

NOTE

Reset operations are substantially different in the Z8036 and the Z8536 and are therefore discussed separately.

6.2 Z8036 (Z-CIO) RESET

The Z8036 is reset by forcing $\overline{AS}$ and $\overline{DS}$ Low simultaneously (normally an illegal operation), or by writing a 1 to the Reset bit (D_0) in the Master Interrupt Control register. After the Z8036 is reset, a Read or Write to the Reset bit is the only responsive command available--writes to all other bits (in all other registers) are ignored and all reads return 0s. In this state, all control bits are forced to 0 (see Chapter 2 for details), all port I/O lines are high-impedance, the interrupt pin is not asserted, and the Interrupt Enable Out (IEO) pin follows the Interrupt Enable In (IEI) pin. Only after clearing the Reset bit (by writing a 0 to it) can the other command bits be programmed.

6.3 Z8536 (CIO) RESET

The Z8536 is reset by forcing $\overline{RD}$ and $\overline{WR}$ Low simultaneously (normally an illegal condition), or by writing a 1 to the Reset bit (D_0) in the Master Interrupt Control register. RESET disables all functions except a read or write to the Reset bit. In the reset state, the pointer always points to the Master Interrupt Control register (see the state diagram shown in Figure 6-1). Writes to all other bits are ignored, and all reads return 01_H. In this state, all control bits are forced to 0 (see Chapter 2 for details), all port I/O lines are high-impedance, the interrupt pin is not asserted, and the IEO pin follows the IEI pin. The other command bits can be programmed only after clearing the Reset bit by writing a 0 to it.

Even if the state of the Z8536 is not known, the following sequence will reset it and put it in State 0.

```
IN   A, (CIOCTL)  ;INSURES STATE 0 OR RESET STATE
LD   A, 0
OUT  (CIOCTL), A  ;WRITE POINTER OR CLEAR RESET
IN   A, (CIOCTL)  ;STATE 0
LD   A,0          ;REG 0--MASTER INTERRUPT CONTROL
OUT  (CIOCTL), A  ;WRITE POINTER
LD   A,1
OUT  (CIOCTL), A  ;WRITE RESET
LD   A,0
OUT  (CIOCTL), A  ;CLEAR RESET
```

6.4 ENABLE BITS OPERATION

As the different functions of the device are being initially programmed, it is possible for erroneous interrupt requests to be generated, or for an illegal combination of modes to be temporarily specified. To alleviate this problem without imposing severe restrictions on the sequence of events required to initialize the device, five internal enable control bits are provided: Port A Enable, Port B Enable, Counter/Timer 1 Enable, Counter/Timer 2 Enable, and one enable shared by Counter/Timer 3 and Port C. While these bits are

cleared to 0, the corresponding logic sections are in an initialization mode. All of the registers can be read and written, but the normal operation of the sections is inhibited. The Port A and Port B Enables, when cleared to 0, force their respective I/O lines into a high-impedance state, hold the 1's catchers in a reset condition, inhibit REQUEST/$\overline{\text{WAIT}}$ generation, and prevent the setting of their Interrupt Pending (IP) bits (the states of IP and Interrupt Under Service (IUS) are not affected). Additionally, output data can be written (the first data output is valid when the output drivers go active), but the data direction for these bits must be properly specified before the data is written. The Port C Enable operates in the same way, and, until set to 1, the handshake logic for Ports A and B is forced into an idle state. The Counter/Timer Enables, when set to 0, terminate any countdown sequence in progress, inhibit the counter/timer from being triggered, and force the counter output to 0. While the enable is 0, the Read Counter Control (RCC) bit in the Counter/Timer Command and Status register is forced to 0. Independent enable bits are provided for the different sections of the device so that the individual sections can be reconfigured without disturbing the status of the unchanged sections. By using these enable bits, the device can be initialized in any sequence as long as the desired configuration for a section is specified before its enable bit is set to 1. When ports or counter/timers are to be linked, the bits which specify linking must be programmed before the functions are enabled. In this case two writes are required to the Master Configuration Control register.

6.5 PROGRAMMING

Programming the CIO entails loading control registers with bits to implement the desired operation. As discussed above, individual enable bits are provided for the various major blocks so that erroneous operations do not occur while the port is being initialized. Before the ports are enabled: IPs cannot be set, REQUEST and $\overline{\text{WAIT}}$ cannot be asserted, and all outputs remain high-impedance; the handshake lines are ignored until Port C is enabled; and the counter/timers cannot be triggered until their enable bits are set.

6.5.1 Programming the Z8036

Programming the Z8036 is simple, because every register is directly addressable--a key advantage of the multiplex Address/Data bus.

The Z8036 allows two schemes for register addressing. Both schemes use only six of the eight bits of the Address/Data bus. The scheme used is determined by the Right Justify Address (RJA) bit in the Master Interrupt Control register. When RJA equals 0, Address bus bits 0 and 7 are ignored, and bits 1 through 6 are decoded for the register address (A_0 is derived from AD_1). When RJA equals 1, bits 0 through 5 are decoded for the register address (A_0 is derived from AD_0).

6.5.2 Programming the Z8536

The Data Registers of Ports A, B, and C are directly addressed by pins A_0 and A_1, as shown in Table 6-1.

Table 6-1. Z8536 Data Register Addressing

A_1	A_0	Register
0	0	Port C Data Register
0	1	Port B Data Register
1	0	Port A Data Register
1	1	Control Registers

All other internal registers are accessed by the following two-step sequence (with pins $A_0 = A_1 = 1$). First write the address of the target register to an internal 6-bit Pointer register, then read from or write to the target register. The Data registers can also be accessed by this method.

In the Z8536, an internal state machine determines if access (with pins $A_0 = A_1 = 1$) is to the Pointer register or to an internal control register (See Figure 6-1). Following any control read operation the state machine is in State 0, and the next control access is to the Pointer register. This can be used to force the state machine into a known state. Control reads in State 0 return the contents of the last register pointed to. Therefore, a register can be read continuously without writing to the pointer.

While the Z8536 is in State 1, the next control access is to the register pointed to, which returns the state machine to State 0. Note that when in State 1, many internal operations are suspended--no IPs are set and internal status is frozen. Therefore to minimize interrupt latency and to allow continuous status updates, **the Z8536 should NOT be left in State 1.**

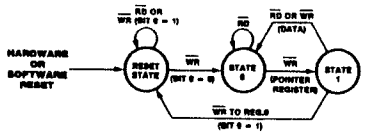


Figure 6-1. Z8536 State Machine Operation

Chapter 7
Z8036 (Z-CIO) Interfacing

7.1 INTRODUCTION

This section provides information on pin functions and assignments and functional timing diagrams for the Z8036 Z-CIO.

7.2 FEATURES

The following features of the Z8036 are not obvious without reference to the ac timing diagrams in the Z8036 Product Specification, document #00-2014-A0.

- The Address Strobe ($\overline{AS}$) input functions as the clock of the Z8036. If the $\overline{AS}$ stops, then data does not get clocked in or through the device, IPs are not set, etc. Care should be taken in the design of the system to ensure that $\overline{AS}$ to the Z8036 is not blocked.
- The assertion of REQUEST is synchronous with PCLK.
- The release of $\overline{WAIT}$ is sychronous with PCLK.
- PCLK is only used with the counter/timers (in timer mode), the deskew timers, and the REQUEST/$\overline{WAIT}$ logic. If these functions are not used, the PCLK input can be held Low.

7.3 PIN FUNCTIONS AND ASSIGNMENTS

The Z8036 is configured for Z-BUS interface controls and timing. The pin functions and assignments are shown in Figures 7-1 and 7-2. Section 7.4 is a description of the pin functions for the Z8036.

7.4 PIN DESCRIPTIONS

AD_0–AD_7. Z-BUS Address/Data lines (bidirectional/3-state). These multiplexed Address/Data

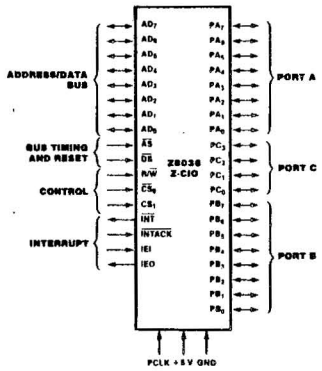


Figure 7-1. Z8036 (Z-CIO) Pin Functions

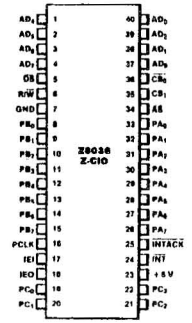


Figure 7-2. Z8036 (Z-CIO) Pin Assignments

lines are used for transfers between the CPU and Z-CIO.

$\overline{AS}$*. Address Strobe (input, active Low). Addresses, $\overline{INTACK}$, and $\overline{CS}_0$ are sampled while $\overline{AS}$ is Low.

$\overline{CS}_0$ and CS_1. Chip Select 0 (input, active Low) and Chip Select 1 (input, active High). $\overline{CS}_0$ and CS_1 must be Low and High, respectively, in order to select a device. $\overline{CS}_0$ is latched by $\overline{AS}$.

$\overline{DS}$. Data Strobe (input, active Low). $\overline{DS}$ provides timing for the transfer of data into or out of the Z8036.

IEI. Interrupt Enable In (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher-priority device has an interrupt under service or is requesting an interrupt.

IEO. Interrupt Enable Out (output, active High). IEO is High only if IEI is High and either:

(1) the CPU is not servicing an interrupt from the CIO, or

(2) during an Interrupt Acknowledge Cycle, the CIO is not requesting an interrupt.

IEO is connected to the next lower-priority device's IEI input and thus inhibits interrupts from lower-priority devices.

$\overline{INT}$. Interrupt Request (output, open-drain, active Low). This signal is pulled Low when the Z8036 requests an interrupt.

$\overline{INTACK}$. Interrupt Acknowledge (input, active Low). This signal indicates to the Z8036 that an Interrupt Acknowledge cycle is in progress. $\overline{INTACK}$ is sampled while $\overline{AS}$ is Low.

PA_0-PA_7. Port A I/O lines (bidirectional, 3-state, or open-drain). These eight I/O lines transfer information between the Z8036's Port A and external devices.

PB_0-PB_7. Port B I/O lines (bidirectional, 3-state, or open-drain). These eight I/O lines transfer information between the Z8036's Port B and external devices. They may also be used to provide external access to Counter/Timers 1 and 2.

PC_0-PC_3. Port C I/O lines (bidirectional, 3-state, or open-drain). These four I/O lines are used to provide handshake, $\overline{WAIT}$, and REQUEST lines for Ports A and B; to provide external access to Counter/Timer 3; or to access Port C of the Z8036.

PCLK. (input, TTL-compatible). This is a peripheral clock that may be, but is not necessarily, the CPU clock. It is used with timers and REQUEST/$\overline{WAIT}$ logic.

R/$\overline{W}$. Read/Write (input). R/$\overline{W}$ indicates that the CPU is reading from (High) or writing to (Low) the Z8036.

* When AS and DS are detected Low at the same time (normally an illegal condition), the Z-CIO is reset.

7.5 Z8036 (Z-CIO) READ CYCLE TIMING

The CPU places an address on the Address/Data bus. The most-significant bits and status information are combined and decoded by external logic to provide two Chip Selects ($\overline{CS}_0$ and CS_1). Six bits of the least-significant byte of the address are latched within the Z8036 and used to specify a Z8036 register. The data from the register specified is strobed onto the Address/Data bus when the CPU issues a $\overline{DS}$. If the register indicated by the address does not exist, the Z8036 remains high-impedance.

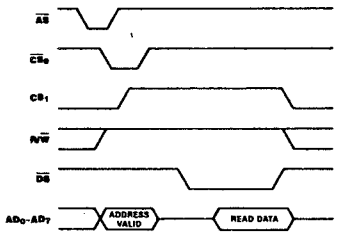


Figure 7-3. Z8036 (Z-CIO) Read Cycle Timing

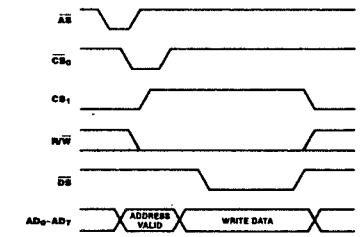


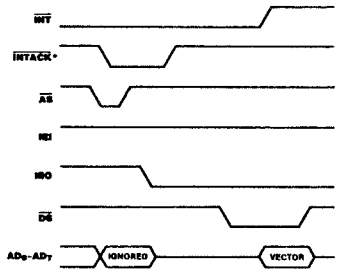
Figure 7-4. Z8036 (Z-CIO) Write Cycle Timing

7.6 Z8036 (Z-CIO) WRITE CYCLE TIMING

The CPU places an address on the Address/Data bus. The most-significant bits and status information are combined and decoded by external logic to provide two Chip Selects ($\overline{CS}_0$ and CS_1). Six bits of the least-significant byte of the address are latched within the Z8036 and used to specify a Z8036 register. The CPU places the data on the Address/Data bus and strobes it into the Z8036 register by issuing a $\overline{DS}$.

7.7 Z8036 (Z-CIO) INTERRUPT ACKNOWLEDGE TIMING

When one of the IP bits in the Z8036 goes High and interrupts are enabled, the Z8036 pulls its $\overline{INT}$ output line Low, requesting an interrupt. The CPU responds with an Interrupt Acknowledge cycle. When $\overline{INTACK}$ goes Low with IP set, the Z8036 pulls its IEO Low, disabling all lower-priority devices on the daisy-chain. The CPU reads the Z8036 interrupt vector by issuing a Low $\overline{DS}$, thereby strobing the interrupt vector onto the Address/Data bus. The IUS that corresponds to the IP is also set, which causes IEO to remain Low.



*INTACK IS DECODED FROM Z8000 STATUS

Figure 7-5. Z8036 (Z-CIO) Interrupt Acknowledge Timing

Chapter 8
Z8536 (CIO) Interfacing

8.1 INTRODUCTION

This section provides information on pin functions and assignments and functional timing diagrams for the Z8536.

8.2 FEATURES

The following features of the Z8536 are not obvious without reference to the ac timing diagrams in the Z8536 Product Specification, document #00-2021-A0.

- The state machine conventions relating to programming and register addressing (see Section 6.5.2) must be followed.
- PCLK can be asynchronous with respect to the CPU--it does not have to be the same as the CPU. However, a minimum of three PCLK cycles must occur between two successive accesses of the Z8536 (that is, between the end of the first access and the beginning of the second access).
- The $\overline{\text{INTACK}}$ input is synchronous, that is, $\overline{\text{INTACK}}$ and PCLK have a relationship that must be maintained.
- The assertion of REQUEST is synchronous with PCLK.
- The release of $\overline{\text{WAIT}}$ is synchronous with PCLK.

8.3 PIN FUNCTIONS AND ASSIGNMENTS

The Z8536 is configured for general microcomputer interface controls and timing. The pin functions and assignments are shown in Figures 8-1 and 8-2. Section 8.4 is a description of the pin functions for the Z8536 CIO.

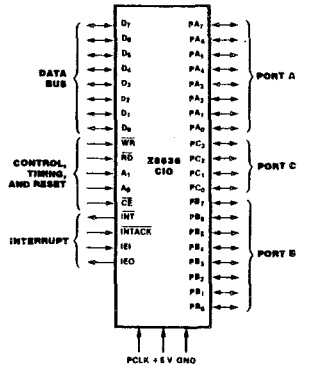


Figure 8-1. Z8536 (CIO) Pin Functions

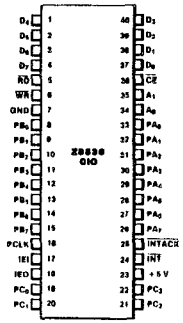


Figure 8-2. Z8536 (CIO) Pin Assignments

8.4 PIN DESCRIPTIONS

A_0–A_1. Address Lines (input). These two lines are used to select the register involved in the CPU transaction: Port A's Data register, Port B's Data register, Port C's Data register, or a control register.

$\overline{CE}$. Chip Enable (input, active Low). A Low level on this input enables the Z8536 to be read from or written to.

D_0–D_7. Data Bus (bidirectional, 3-state). These eight data lines are used for transfers between the CPU and the CIO.

IEI. Interrupt Enable In (input, active High). IEI is used with IEO to form an interrupt daisy-chain when there is more than one interrupt-driven device. A High IEI indicates that no other higher-priority device has an interrupt under service or is requesting an interrupt.

IEO. Interrupt Enable Out (output, active High). IEO is High only if IEI is High and either:

(1) the CPU is not serving an interrupt from the CIO, or

(2) during an Interrupt Acknowledge cycle, the CIO is not requesting an interrupt.

IEO is connected to the next lower-priority device's IEI input and thus inhibits interrupts from lower-priority devices.

$\overline{INT}$. Interrupt Request (output, open-drain, active Low). This signal is pulled Low when the Z8536 requests an interrupt.

$\overline{INTACK}$. Interrupt Acknowledge (input, active Low). This input indicates to the Z8536 that an Interrupt Acknowledge cycle is in progress. $\overline{INTACK}$ must be synchronized to PCLK, and it must be stable throughout the Interrupt Acknowledge cycle.

PA_0–PA_7. Port A I/O lines (bidirectional, 3-state, or open-drain). These eight I/O lines transfer information between the CIO's Port A and external devices.

PB_0–PB_7. Port B I/O lines (bidirectional, 3-state, or open-drain). These eight I/O lines transfer information between the Z8536's Port B and external devices. They may also be used to provide external access to Counter/Timers 1 and 2.

PC_0–PC_3. Port C I/O lines (bidirectional, 3-state, or open-drain). These four I/O lines are used to provide handshake, $\overline{WAIT}$, and REQUEST lines for Ports A and B; external access to Counter/Timer 3; or access to the Z8536's Port C.

PCLK. Peripheral Clock (input, TTL-compatible). This is the clock used by the internal control logic and the counter/timers in timer mode. (It does not have to be the CPU clock.)

$\overline{RD}$*. Read (input, active Low). This signal indicates that a CPU is reading from the CIO. During an Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the Data bus if the Z8536 is the highest-priority device requesting an interrupt.

$\overline{WR}$*. Write (input, active Low). This signal indicates a CPU write to the Z8536.

* When $\overline{RD}$ and $\overline{WR}$ are detected Low at the same time (normally an illegal condition), the Z8536 is reset.

8.5 Z8536 (CIO) READ CYCLE TIMING

At the beginning of a read cycle, the CPU places an address on the Address bus. Bits A_0 and A_1 specify a Z8536 register; the remaining address bits and status information are combined and decoded to generate a Chip Enable ($\overline{CE}$) signal that selects the Z8536. When Read ($\overline{RD}$) goes Low, data from the specified register is gated onto the Data bus.

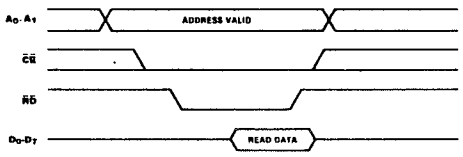


Figure 8-3. Z8536 (CIO) Read Cycle Timing

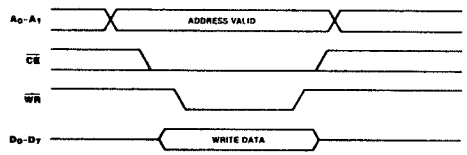


Figure 8-4. Z8536 (CIO) Write Cycle Timing

8.6 Z8536 (CIO) WRITE CYCLE TIMING

At the beginning of a write cycle, the CPU places an address on the Data bus. Bits A_0 and A_1 specify a Z8536 register; the remaining address bits and status information are combined and decoded to generate a Chip Enable ($\overline{CE}$) signal that selects the Z8536. When $\overline{WR}$ goes Low, data placed on the bus by the CPU is strobed into the specified Z8536 register.

8.7 Z8536 (CIO) INTERRUPT ACKNOWLEDGE TIMING

The Z8536 pulls its Interrupt Request ($\overline{INT}$) line Low, requesting interrupt service from the CPU, if an Interrupt Pending (IP) bit is set and interrupts are enabled. The CPU responds with an Interrupt Pending (IP) bit is set and interrupts are enabled. The CPU responds with an Interrupt Acknowledge cycle. When Interrupt Acknowledge ($\overline{INTACK}$) goes active and the IP is set, the Z8536 forces Interrupt Enable Out (IEO) Low, disabling all lower priority devices in the interrupt daisy-chain. If the CIO is the highest priority device requesting service (IEI is High), it places its interrupt vector on the Data bus and sets the Interrupt Under Service (IUS) bit when Read ($\overline{RD}$) goes Low.

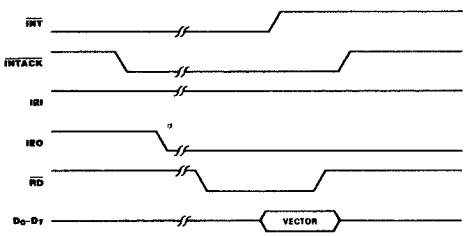


Figure 8-5. Z8536 (CIO) Interrupt Acknowledge Timing

Appendix A

CIO MNEMONICS

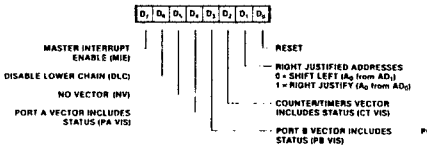
$\overline{\text{ACKIN}}$	Acknowledge Input
$\overline{\text{AS}}$	Address Strobe
C/$\overline{\text{SC}}$	Continuous/$\overline{\text{Single Cycle}}$
CIO	Counter/Timer, Parallel Input/Output Unit
CIP	Count In Progress
CTnE	Counter/Timer n Enable
CT3E and PCE	Counter/Timer 3 and Port C Enable
CT VIS	Counter/Timer Vector Includes Status
DAC	Data Accepted
$\overline{\text{DAV}}$	Data Available
DCS	Duty Cycle Selects
DD	Data Direction register
DLC	Disable Lower Chain
DMA	Direct Memory Access
DPP	Data Path Polarity register
$\overline{\text{DS}}$	Data Strobe
DTE	Deskew Timer Enable
DTE/LPM	Deskew Timer Enable/Latch on Pattern Match
ECL	External Count Enable
EGE	External Gate Enable
EOE	External Output Enable
ERR	Interrupt Error
ETE	External Trigger Enable
GCB	Gate Command Bit
HTS	Handshake Type Specification bits
IE	Interrupt Enable
IEI	Interrupt Enable In
IEO	Interrupt Enable Out
IMO	Interrupt on Match Only
$\overline{\text{INT}}$	Interrupt
$\overline{\text{INTACK}}$	Interrupt Acknowledge
I/O	Input/Output
IP	Interrupt Pending
IRF	Input Register Full
ITB	Interrupt on Two Bytes
IUS	Interrupt Under Service
LC	Counter/Timer Link Controls
LPM	Latch on Pattern Match
LSB	least-significant bit
MIE	Master Interrupt Enable
MSB	most-significant bit
NV	No Vector
OR-PEV	OR-Priority Encoded Vector
ORE	Output Register Empty
PA	Port A
PA VIS	Port A Vector Includes Status
PAE	Port A Enable
PB	Port B
PB VIS	Port B Vector Includes Status
PBE	Port B Enable
PC	Port C
PCE	Port C Enable
PCLK	Peripheral Clock
PLK	Port Link Control
PM	Pattern Mask registers
PMF	Pattern Match Flag
PMS	Pattern Mode Specification bits
PP	Pattern Polarity registers
PT	Pattern Transition registers
PTS	Port Type Selects
RCC	Read Counter Control
REB	Retrigger Enable Bit
RFD	Ready For Data
RJA	Right Justified Address
R/$\overline{\text{W}}$	Read/$\overline{\text{W}}$rite
RWS	REQUEST/$\overline{\text{WAIT}}$ Specification bits
SB	Single-Buffered
SIO	Special Input/Output
TCB	Trigger Command Bit
VIS	Vector Includes Status
1's Catcher	One's Catcher
3-Wire	3-Wire Handshake (IEEE-488)

Appendix B

Registers

Master Interrupt Control Register

Address: 000000
(Read/Write)



Master Configuration Control Register

Address: 000001
(Read/Write)

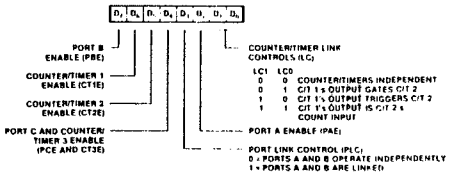
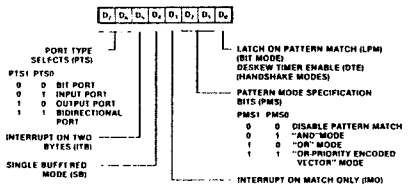


Figure B-1. Master Control Registers

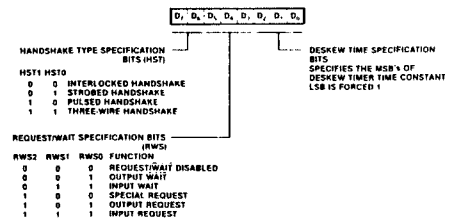
Port Mode Specification Registers

Addresses: 100000 Port A
101000 Port B
(Read/Write)



Port Handshake Specification Registers

Addresses: 100001 Port A
101001 Port B
(Read/Write)



Port Command and Status Registers

Addresses: 001000 Port A
001001 Port B
(Read/Partial Write)

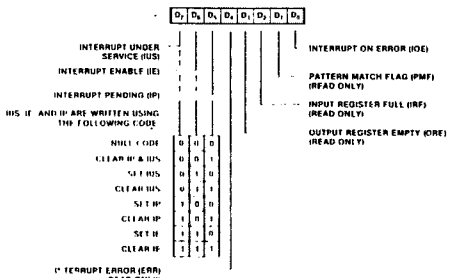
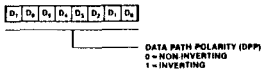


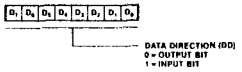
Figure B-2. Port Specification Registers

Registers
(Continued)

Data Path Polarity Registers
Addresses: 100010 Port A
101010 Port B
000101 Port C (4 LSBs only)
(Read/Write)



Data Direction Registers
Addresses: 100011 Port A
101011 Port B
000110 Port C (4 LSBs only)
(Read/Write)



Special I/O Control Registers
Addresses: 100100 Port A
101100 Port B
000111 Port C (4 LSBs only)
(Read/Write)

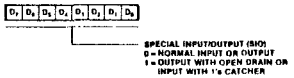
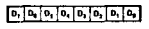


Figure B-3. Bit Path Definition Registers

Port Data Registers
Addresses: 001101 Port A
001110 Port B
(Read/Write)



Port C Data Register
Address: 001111
(Read/Write)

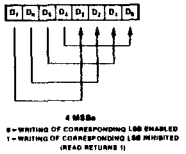


Figure B-4. Port Data Registers

Pattern Polarity Registers (PP)
Addresses: 100101 Port A
101101 Port B
(Read/Write)

Pattern Transition Registers (PT)
Addresses: 100110 Port A
101110 Port B
(Read/Write)

Pattern Mask Registers (PM)
Addresses: 100111 Port A
101111 Port B
(Read/Write)

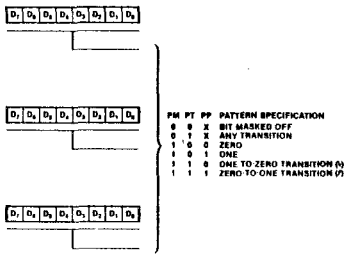
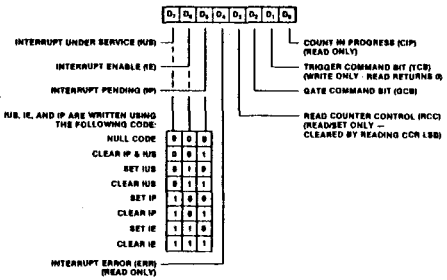


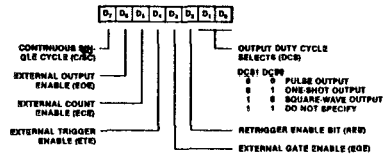
Figure B-5. Pattern Definition Registers

Registers
(Continued)

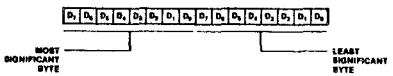
Counter/Timer Command and Status Registers
Addresses: 001010 Counter/Timer 1
001011 Counter/Timer 2
001100 Counter/Timer 3
(Read/Partial Write)



Counter/Timer Mode Specification Registers
Addresses: 011100 Counter/Timer 1
011101 Counter/Timer 2
011110 Counter/Timer 3
(Read/Write)



Counter/Timer Current Count Registers
Addresses: 010000 Counter/Timer 1's MSB
010001 Counter/Timer 1's LSB
010010 Counter/Timer 2's MSB
010011 Counter/Timer 2's LSB
010100 Counter/Timer 3's MSB
010101 Counter/Timer 3's LSB
(Read Only)



Counter/Timer Time Constant Registers
Addresses: 010110 Counter/Timer 1's MSB
010111 Counter/Timer 1's LSB
011000 Counter/Timer 2's MSB
011001 Counter/Timer 2's LSB
011010 Counter/Timer 3's MSB
011011 Counter/Timer 3's LSB
(Read/Write)

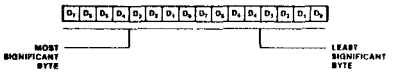
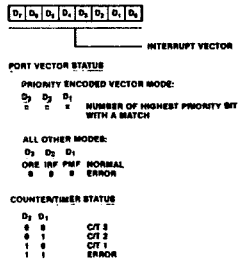


Figure B-6. Counter/Timer Registers

Registers (Continued)

Interrupt Vector Register
Addresses: 000010 Port A
000011 Port B
000100 Counter/Timers
(Read/Write)



Current Vector Register
Address: 011111
(Read Only)

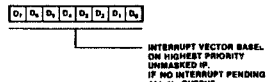


Figure B-7. Interrupt Vector Registers

Register Address Summary

Main Control Registers

Address	Register Name
000000	Master Interrupt Control
000001	Master Configuration Control
000010	Port A's Interrupt Vector
000011	Port B's Interrupt Vector
000100	Counter/Timer's Interrupt Vector
000101	Port C's Data Path Polarity
000110	Port C's Data Direction
000111	Port C's Special I/O Control

Most Often Accessed Registers

Address	Register Name
001000	Port A's Command and Status
001001	Port B's Command and Status
001010	Counter/Timer 1's Command and Status
001011	Counter/Timer 2's Command and Status
001100	Counter/Timer 3's Command and Status.
001101	Port A's Data
001110	Port B's Data
001111	Port C's Data

Counter/Timer Related Registers

Address	Register Name
010000	Counter/Timer 1's Current Count-MSBs
010001	Counter/Timer 1's Current Count-LSBs
010010	Counter/Timer 2's Current Count-MSBs
010011	Counter/Timer 2's Current Count-LSBs
010100	Counter/Timer 3's Current Count-MSBs
010101	Counter/Timer 3's Current Count-LSBs
010110	Counter/Timer 1's Time Constant-MSBs
010111	Counter/Timer 1's Time Constant-LSBs
011000	Counter/Timer 2's Time Constant-MSBs
011001	Counter/Timer 2's Time Constant-LSBs
011010	Counter/Timer 3's Time Constant-MSBs
011011	Counter/Timer 3's Time Constant-LSBs
011100	Counter/Timer 1's Mode Specificaiton
011101	Counter/Timer 2's Mode Specification
011110	Counter/Timer 3's Mode Specification
011111	Current Vector

Port A Specification Registers

Address	Register Name
100000	Port A's Mode Specification
100001	Port A's Handshake Specification
100010	Port A's Data Path Polarity
100011	Port A's Data Direction
100100	Port A's Special I/O Control
100101	Port A's Pattern Polarity
100110	Port A's Pattern Transition
100111	Port A's Pattern Mask

Port B Specification Registers

Address	Register Name
101000	Port B's Mode Specification
101001	Port B's Handshake Specification
101010	Port B's Data Path Polarity
101011	Port B's Data Direction
101100	Port B's Special I/O Control
101101	Port B's Pattern Polarity
101110	Port B's Pattern Transition
101111	Port B's Pattern Mask

Z-CIO and CIO pin Functions and Assignments

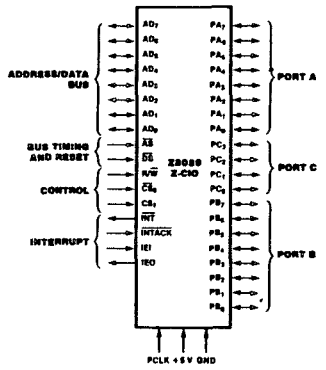


Figure 1. Pin Functions

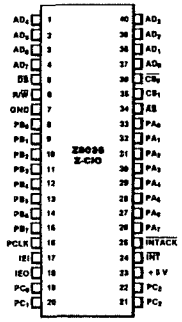


Figure 2. Pin Assignments

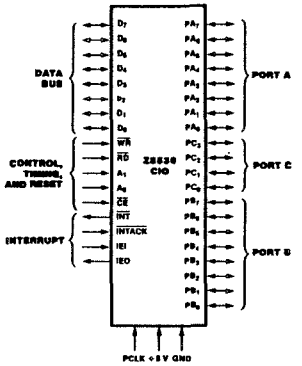


Figure 1. Pin Functions

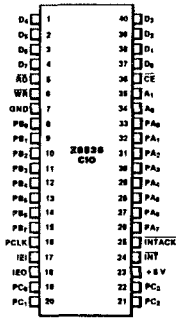


Figure 2. Pin Assignments

Absolute Maximum Ratings

Voltages on all inputs and outputs with respect to GND . . . . . -0.3 V to +7.0 V

Operating Ambient Temperature . . . . . As Specified in Ordering Information

Storage Temperature . . . . . -65°C to +150°C

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affe device reliability.

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

- +4.75 V ≤ V_{CC} ≤ +5.25 V
- GND = 0 V
- T_A as specified in Ordering Information

All ac parameters assume a load capacitance of 50 pF max.

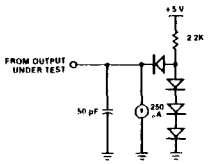


Figure 21. Standard Test Load

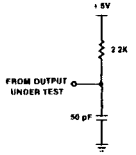


Figure 22. Open-Drain Test Load

DC Characteristics

Symbol	Parameter	Min	Max	Unit	Condition
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.3	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −250 µA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = +2.0 mA
			0.5	V	I_{OL} = +3.2 mA
I_{IL}	Input Leakage		±10.0	µA	0.4 ≤ V_{IN} ≤ +2.4 V
I_{OL}	Output Leakage		±10.0	µA	0.4 ≤ V_{OUT} ≤ +2.4 V
I_{CC}	V_{CC} Supply Current		250	mA	

V_{CC} = 5 V ± 5% unless otherwise specified, over specified temperature range.

Capacitance

Symbol	Parameter	Min	Max	Unit	Test Condition
C_{IN}	Input Capacitance		10	pF	Unmeasured Pins Returned to Ground
C_{OUT}	Output Capacitance		15	pF	
$C_{I/O}$	Bidirectional Capacitance		20	pF	

f = 1 MHz, over specified temperature range.

Ordering Information

Product Number	Package/ Temperature	Speed	Description
Z8536	CE, CM, CMB, CS, DE, DS, PE, PS	4.0 MHz	CIO (40-pin)
Z8536A	CE, CM, CMB, CS, DE, DS, PE, PS	6.0 MHz	Same as above
Z8036	CE, CS, DE, DS, PE, PS	4.0 MHz	Z-CIO (40-pin)
Z8046	CM, CMB	6.0 MHz	Same as above
Z8046A	CE, CM, CMB, CS, DE, DS, PE, PS	6.0 MHz	Same as above

BODIES: C = Ceramic, D = Cerdip, P = Plastic, E = -40°C to +85°C, M = -55°C to +125°C, MB = -55°C with MIL-STD-883 with Class B processing, S = 0°C to +70°C

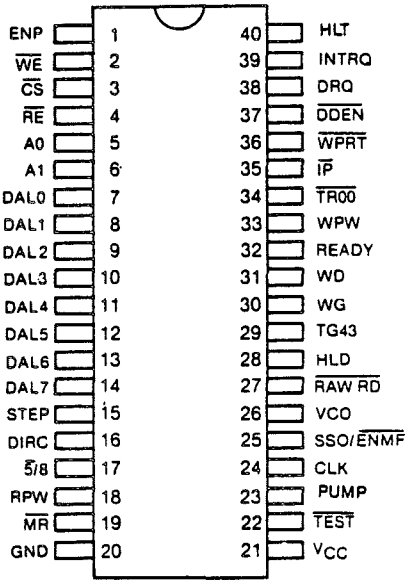
I

Zilog

WD279X-02 Floppy Disk Formatter/Controller Family

FEATURES

- ON-CHIP PLL DATA SEPARATOR
- ON-CHIP WRITE PRECOMPENSATION LOGIC
- SINGLE +5V SUPPLY
- ACCOMMODATES SINGLE AND DOUBLE DENSITY FORMATS
  - IBM 3740 (FM)
  - IBM 34 (MFM)
- AUTOMATIC SEEK WITH VERIFY
- MULTIPLE SECTOR READ/WRITE
- TTL COMPATIBLE
- PROGRAMMABLE CONTROL
  - SELECTABLE TRACK-TO-TRACK ACCESS
  - HEAD LOAD TIMING
- SOFTWARE COMPATIBLE WITH THE FD179X SERIES
- SOFT SECTOR FORMAT COMPATIBILITY



PIN DESIGNATION

DESCRIPTION

The WD279X are N-Channel Silicon Gate MOS LSI devices which perform the functions of a Floppy Disk Formatter/Controller in a single chip implementation. The WD279X, which can be considered the end result of both the FD1771 and FD179X designs, is IBM 3740 compatible in single density mode (FM) and System 34 compatible in Double Density Mode (MFM). The WD279X contains all the features of its predecessor the FD179X plus a high performance Phase-Lock-Loop Data Separator as well as Write Precompensation Logic. In Double Density mode, Write Precompensation is automatically engaged to a value programmed via an external potentiometer. In order to maintain compatibility, the FD1771, FD179X and WD279X designs were made as close as possible with the computer interface, instruction set, and I/O registers being identical. Also, head load control is identical. In each case, the actual pin assignments vary by only a few pins from any one to another.

The processor interface consists of an 8-bit bi-directional bus for data, status, and control word transfers. The WD279X is set up to operate on a multiplexed bus with other bus-oriented devices.

The WD279X is TTL compatible on all inputs and outputs. The outputs will drive one TTL load or three LS loads. The 2793 is identical to the 2791 except the DAL lines are TRUE for systems that utilize true data busses.

The 2795/7 has a side select output for controlling double sided drives.

PIN DESCRIPTION

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
1	ENABLE PRECOMP	ENP	A Logic high on this input enables write precompensation to be performed on double density Write Data output only.
19	$\overline{\text{MASTER RESET}}$	$\overline{\text{MR}}$	A logic low (50 microseconds min.) on this input resets the device and loads HEX 03 into the command register. The Not Ready (Status Bit 7) is reset during MR ACTIVE. When $\overline{\text{MR}}$ is brought to a logic high a RESTORE Command is executed, regardless of the state of the Ready signal from the drive. Also, HEX 01 is loaded into sector register.
20	POWER SUPPLIES	V_{SS}	Ground
21		V_{CC}	+5V ±5%
COMPUTER INTERFACE:			
2	$\overline{\text{WRITE ENABLE}}$	$\overline{\text{WE}}$	A logic low on this input gates data on the DAL into the selected register when $\overline{\text{CS}}$ is low.
3	$\overline{\text{CHIP SELECT}}$	$\overline{\text{CS}}$	A logic low on this input selects the chip and enables computer communication with the device.
4	$\overline{\text{READ ENABLE}}$	$\overline{\text{RE}}$	A logic low on this input controls the placement of data from a selected register on the DAL when $\overline{\text{CS}}$ is low.
5, 6	REGISTER SELECT LINES	A0, A1	These inputs select the register to receive/transfer data on the DAL lines under $\overline{\text{RE}}$ and $\overline{\text{WE}}$ control: $\overline{\text{CS}}$ A1 A0 $\overline{\text{RE}}$ $\overline{\text{WE}}$ 0 0 0 Status Reg Command Reg 0 0 1 Track Reg Track Reg 0 1 0 Sector Reg Sector Reg 0 1 1 Data Reg Data Reg
7-14	DATA ACCESS LINES	DAL0-DAL7	Eight bit bi-directional bus used for transfer of commands, status, and data. These lines are inverted (active low) on WD2791 and WD2795.
24	CLOCK	CLK	This input requires a free-running 50% duty cycle square wave clock for internal timing reference, 2 MHz ± 1% for 8" drives, 1 MHz ± 1% for minifloppies.
38	DATA REQUEST	DRQ	This output indicates that the Data Register contains assembled data in Read operations, or the DR is empty in Write operations. This signal is reset when serviced by the computer through reading or loading the DR.
39	INTERRUPT REQUEST	INTRQ	This output is set at the completion of any command and is reset when the Status register is read or the Command register is written to.
FLOPPY DISK INTERFACE:			
15	STEP	STEP	The step output contains a pulse for each step.
16	DIRECTION	DIRC	Direction Output is active high when stepping in, active low when stepping out.
17	5¼," 8" SELECT	$\overline{5/8}$	This input selects the internal VCO frequency for use with 5¼" drives or 8" drives.
18	READ PULSE WIDTH	RPW	An external potentiometer tied to this input controls the phase comparator within the data separator.
22	$\overline{\text{TEST}}$	$\overline{\text{TEST}}$	A logic low on this input allows adjustment of external resistors by enabling internal signals to appear on selected pins.

PIN DESCRIPTION (Continued)

PIN NUMBER	PIN NAME	SYMBOL	FUNCTION
23	PUMP	PUMP	High-Impedance output signal which is forced high or low to increase/decrease the VCO frequency.
25	ENABLE MINI-FLOPPY (2791, 2793)	$\overline{ENMF}$	A logic low on this input enables an internal ÷2 of the Master Clock. This allows both 5¼" and 8" drive operation with a single 2 MHz clock. For a 1 MHz clock on Pin 24, this line must be left open or tied to a Logic 1.
25	SIDE SELECT OUTPUT (2795, 2797)	SSO	The logic level of the Side Select Output is directly controlled by the 'S' flag in Type II or III commands. When U = 1, SSO is set to a logic 1. When U = 0, SSO is set to a logic 0. The SSO is compared with the side information in the Sector I.D. Field. If they do not compare Status Bit 4 (RNF) is set. The Side Select Output is only updated at the beginning of a Type II or III command. It is forced to a logic 0 upon a MASTER RESET condition.
26	VOLTAGE-CONTROLLED OSCILLATOR	VCO	An external capacitor tied to this pin adjusts the VCO center frequency.
27	$\overline{RAW\ READ}$	$\overline{RAW\ READ}$	The data input signal directly from the drive. This input shall be a negative pulse for each recorded flux transition.
28	HEAD LOAD	HLD	The HLD output controls the loading of the Read-Write head against the media.
29	TRACK GREATER THAN 43	TG43	This output informs the drive that the Read/Write head is positioned between tracks 44-76. This output is valid only during Read and Write Commands.
30	WRITE GATE	WG	This output is made valid before writing is to be performed on the diskette.
31	WRITE DATA	WD	MFM or FM output pulse per flux transition. WD contains the unique Address marks as well as data and clock in both FM and MFM formats.
32	READY	READY	This input indicates disk readiness and is sampled for a logic high before Read or Write commands are performed. If Ready is low the Read or Write operation is not performed and an interrupt is generated. Type I operations are performed regardless of the state of Ready. The Ready input appears in inverted format as Status Register bit 7.
33	WRITE PRECOMP WIDTH	WPW	An external potentiometer tied to this input controls the amount of delay in Write precompensation mode.
34	$\overline{TRACK\ 00}$	$\overline{TR00}$	This input informs the WD279X that the Read/Write head is positioned over Track 00.
35	$\overline{INDEX\ PULSE}$	$\overline{IP}$	This input informs the WD279X when the index hole is encountered on the diskette.
36	$\overline{WRITE\ PROTECT}$	$\overline{WPRT}$	This input is sampled whenever a Write Command is received. A logic low terminates the command and sets the Write Protect Status bit.
37	$\overline{DOUBLE\ DENSITY}$	$\overline{DDEN}$	This input pin selects either single or double density operation. When $\overline{DDEN}$ = 0, double density is selected. When $\overline{DDEN}$ = 1, single density is selected.
40	HEAD LOAD TIMING	HLT	When a logic high is found on the HLT input the head is assumed to be engaged. It is typically derived from a 1 shot triggered by HLD.

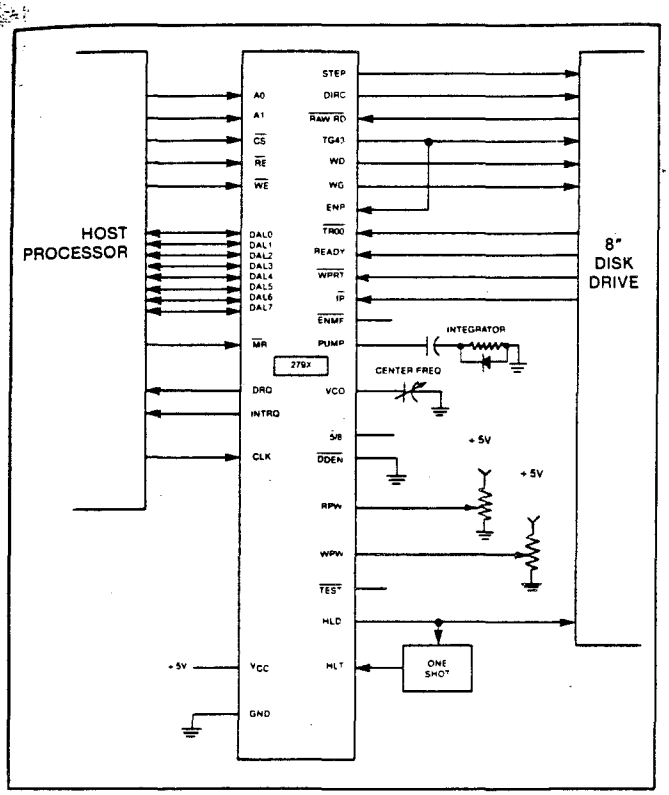


Figure 1.

APPLICATIONS

8" FLOPPY AND 5¼" MINI FLOPPY CONTROLLER
SINGLE OR DOUBLE DENSITY
CONTROLLER/FORMATTER

The WD279X Family are MOS/LSI devices which perform the functions of a Floppy Disk Controller/Formatter. Software compatible with its predecessor, the FD179X, the device also contains a high performance Phase-Lock-Loop Data Separator as well as Write Precompensation Logic.

When operating in Double Density mode, Write Precompensation may be enabled, its value predetermined by an external potentiometer. An on-chip VCO and phase comparator allows adjustable frequency range for 5¼" or 8" Floppy Disk interfacing.

The WD279X is fabricated in NMOS silicon gate technology and available in a 40 pin dual-in-line package.

FEATURES	2791	2793	2795	2797
Single Density (FM)	X	X	X	X
Double Density (MFM)	X	X	X	X
True Data Bus		X		X
Inverted Data Bus	X		X	
Side Select Out			X	X
Internal CLK Divide	X	X		

ORGANIZATION

The Floppy Disk Formatter block diagram is illustrated on page 5. The primary sections include the parallel processor interface and the Floppy Disk interface.

Data Shift Register — This 8-bit register assembles serial data from the Read Data input (RAW READ) during Read operations and transfers serial data to the Write Data output during Write operations.

Data Register — This 8-bit register is used as a holding register during Disk Read and Write operations in Disk Read operations the assembled data byte is transferred in parallel to the Data Register from the Data Shift Register. In Disk Write operations information is transferred in parallel from the Data Register to the Data Shift Register.

When executing the Seek command the Data Register holds the address of the desired Track position. This register is loaded from the DAL and gated onto the DAL under processor control.

Track Register — This 8-bit register holds the track number of the current Read/Write head position. It is incremented by one every time the head is stepped in (towards track 76) and decremented by one when the head is stepped out (towards track 00). The contents of the register are compared with the recorded track number in the ID field during disk Read, Write and Verify operations. The Track Register can be loaded from or transferred to the DAL. This Register should not be loaded when the device is busy.

Sector Register (SR) — This 8-bit register holds the address of the desired sector position. The contents of the register are compared with the recorded sector number in the ID field during disk Read or Write operations. The Sector Register contents can be loaded from or transferred to the DAL. This register should not be loaded when the device is busy.

Command Register (CR) — This 8-bit register holds the command presently being executed. This register should not be loaded when the device is busy unless the new command is a force interrupt. The command register can be loaded from the DAL, but not read onto the DAL.

Status Register (STR) — This 8-bit register holds device Status information. The meaning of the Status bits is a function of the type of command previously executed. This register can be read onto the DAL, but not loaded from the DAL.

CRC Logic — This logic is used to check or to generate the 16-bit Cyclic Redundancy Check (CRC). The polynomial is: $G(x) = x^{16} + x^{12} + x^5 + 1$.

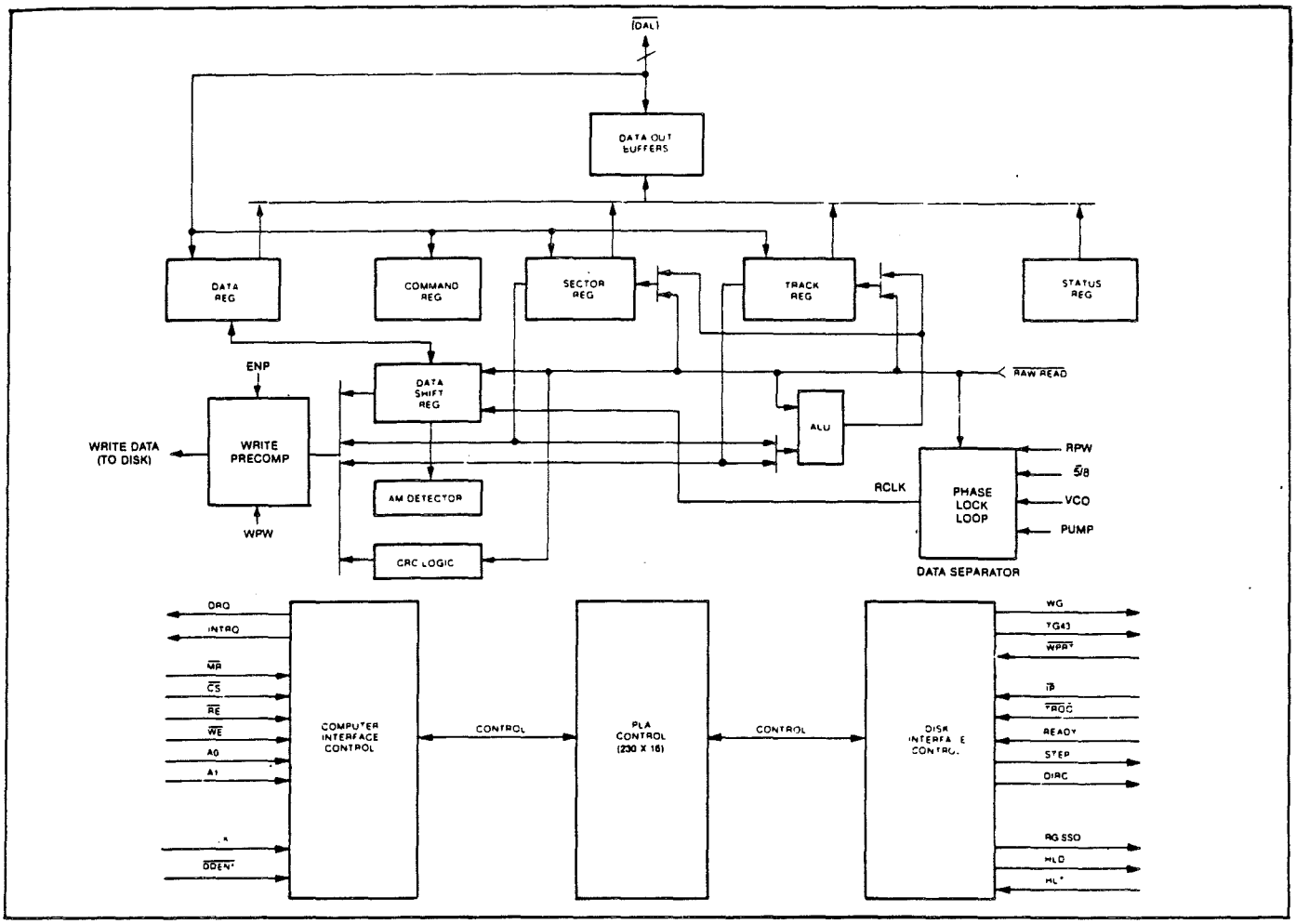
The CRC includes all information starting with the address mark and up to the CRC characters. The CRC register is preset to ones prior to data being shifted through the circuit.

Arithmetic/Logic Unit (ALU) — The ALU is a serial comparator, incrementer, and decrementer and is used for register modification and comparisons with the disk recorded ID field.

Timing and Control — All computer and Floppy Disk interface controls are generated through this logic. The internal device timing is generated from an external crystal clock.

AM Detector — The address mark detector detects ID, data and index address marks during read and write operations.

Write Precompensation — enables write precompensation to be performed on the Write Data output.



WD279X BLOCK DIAGRAM

Data Separator — a high performance Phase-Lock-Loop Data Separator with on-chip VCO and phase comparator allows adjustable frequency range for 5¼" or 8" Floppy Disk interfacing.

PROCESSOR INTERFACE

The interface to the processor is accomplished through the eight Data Access Lines (DAL) and associated control signals. The DAL are used to transfer Data, Status, and Control words out of, or into the WD279X. The DAL are three state buffers that are enabled as output drivers when Chip Select ($\overline{CS}$) and Read Enable ($\overline{RE}$) are active (low logic state) or act as input receivers when $\overline{CS}$ and Write Enable ($\overline{WE}$) are active.

When transfer of data with the Floppy Disk Controller is required by the host processor, the device address is decoded and $\overline{CS}$ is made low. The address bits A1 and A0, combined with the signals $\overline{RE}$ during a Read operation or $\overline{WE}$ during a Write operation are interpreted as selecting the following registers:

A1	A0	READ ($\overline{RE}$)	WRITE ($\overline{WE}$)
0	0	Status Register	Command Register
0	1	Track Register	Track Register
1	0	Sector Register	Sector Register
1	1	Data Register	Data Register

During Direct Memory Access (DMA) types of data transfers between the Data Register of the WD279X and the processor, the Data Request (DRQ) output is used in Data Transfer control. This signal also appears as status bit 1 during Read and Write operations.

On Disk Read operations the Data Request is activated (set high) when an assembled serial input byte is transferred in parallel to the Data Register. This bit is cleared when the Data Register is read by the processor. If the Data Register is read after one or more characters are lost, by having new data transferred into the register prior to processor readout, the Lost Data bit is set in the Status Register. The Read operation continues until the end of sector is reached.

On Disk Write operations the data Request is activated when the Data Register transfers its contents to the Data Shift Register, and requires a new data byte. It is reset when the Data Register is loaded with new data by the processor. If new data is not loaded at the time the next serial byte is required by the Floppy Disk, a byte of zeroes is written on the diskette and the Lost Data bit is set in the Status Register.

At the completion of every command an INTRQ is generated. INTRQ is reset by either reading the status register or by loading the command register with a new command. In addition, INTRQ is generated if a Force Interrupt command condition is met.

The 279X has two modes of operation according to the state of $\overline{DDEN}$ (Pin 37). When $\overline{DDEN}$ = 1, Single Density (FM) is selected. When $\overline{DDEN}$ = 0, Double Density (MFM) is selected. In either case, the CLK input (Pin 24) is set at 2 MHz for 8" drives or 1 MHz for 5¼" drives.

On the 2791/2793, the $\overline{ENMF}$ input (Pin 25) can be used for controlling both 5¼" and 8" drives with a single 2 MHz clock. When $\overline{ENMF}$ = 0, an internal ÷ 2 of the CLK is performed. When $\overline{ENMF}$ = 1, no divide takes place. This allows the use of a 2 MHz clock for both 5¼" and 8" configurations.

The internal VCO frequency must also be set to the proper value. The $\overline{5/8}$ input (Pin 17) is used to select data separator operation by internally dividing the Read Clock. When $\overline{5/8}$ = 0, 5¼" data separation is selected; when 5/8 = 1, 8" drive data separation is selected.

CLOCK (24)	$\overline{ENMF}$ (25)	$\overline{5/8}$ (17)	DRIVE
2 MHz	1	1	8"
2 MHz	0	0	5¼"
1 MHz	1	0	5¼"

FUNCTIONAL DESCRIPTION

The WD279X-02 is software compatible with the FD179X-02 series of Floppy Disk Controllers. Commands, status, and data transfers are performed in the same way. Software generated for the 179X can be transferred to a 279X system without modification.

In addition to the 179X, the 279X contains an internal Data Separator and Write precompensation circuit. The $\overline{TEST}$ (Pin 22) line is used to adjust both data separator and precompensation. When $\overline{TEST}$ = 0, the WD (Pin 31) line is internally connected to the output of the write precomp one-shot. Adjustment of the WPW (Pin 33) line can then be accomplished. A second one-shot tracks the precomp setting at approximately 3:1 to insure adequate Write Data pulse widths to meet drive specifications.

Similarly, Data separation is also adjusted with $\overline{TEST}$ = 0. The TG43 (Pin 29) line is internally connected to the output of the read data one-shot, which is adjusted via the RPW (Pin 18) line. The DIRC (Pin 16) line contains the Read Clock output (.5 MHz for 8" drives). The VCO Trimming capacitor (Pin 26) is adjusted for center frequency.

Internal timing signals are used to generate pulses during the adjustment mode so that these adjustments can be made while the device is in-circuit. The $\overline{TEST}$ line also contains a pull-up resistor, so adjustments can be performed simply by grounding the $\overline{TEST}$ pin, overriding the pull-up. The $\overline{TEST}$ pin cannot be used to disable stepping rates during operation as its function is quite different from the 179X.

Other pins on the device also include pull-up resistors and may be left open to satisfy a Logic 1 condition. These are: ENP, $\overline{5/8}$, $\overline{ENMF}$, $\overline{WPRT}$, $\overline{DDEN}$, HLT, TEST, and $\overline{MR}$.

GENERAL DISK READ OPERATIONS

Sector lengths of 128, 256, 512 or 1024 are obtainable in either FM or MFM formats. For FM, $\overline{DDEN}$ should be placed to logical "1." For MFM formats, $\overline{DDEN}$ should be placed to a logical "0." Sector lengths are determined at format time by the fourth byte in the "ID" field.

Sector Length Table*	
Sector Length Field (hex)	Number of Bytes in Sector (decimal)
00	128
01	256
02	512
03	1024

* 2795/97 may vary — see command summary.

The WD279X recognizes tracks and sectors numbered 00-FFX. However, due to programming restrictions, only tracks and sectors 00 thru F4 can be formatted.

GENERAL DISK WRITE OPERATION

When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the 279X before the Write Gate signal can be activated.

Writing is inhibited when the $\overline{Write\ Protect}$ input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set.

For write operations, the 279X provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of pulses set to a width approximately three times greater than the precomp adjustment. Write Data provides the unique address marks in both formats.

READY

Whenever a Read or Write command (Type II or III) is received the 279X samples the Ready input. If this input is logic low the command is not executed and an interrupt is generated. All Type I commands are performed regardless of the state of the Ready input. Also, whenever a Type II or III command is received, the TG43 signal output is updated. TG43 may be tied to ENP to enable write precompensation on tracks 44-76.

COMMAND DESCRIPTION

The WD279X will accept eleven commands. Command words should only be loaded in the Command Register when the Busy status bit is off (Status bit 0). The one exception is the Force interrupt command. Whenever a command is being executed, the Busy status bit is set. When a command is completed, an interrupt is generated and the Busy status bit is reset. The Status Register indicates whether the completed command encountered an error or was fault free. For ease of discussion, commands are divided into four types. Commands and types are summarized in Table 1.

TABLE 1. COMMAND SUMMARY

A. Commands for Models: 2791, 2793

Type	Command	Bits 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	T	h	V	r_1	r_0
I	Step-in	0	1	0	T	h	V	r_1	r_0
I	Step-out	0	1	1	T	h	V	r_1	r_0
II	Read Sector	1	0	0	m	S	E	C	0
II	Write Sector	1	0	1	m	S	E	C	a_0
III	Read Address	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrupt	1	1	0	1	I_3	I_2	I_1	I_0

B. Commands for Models: 2795, 2797

Type	Command	Bits 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	T	h	V	r_1	r_0
I	Step-in	0	1	0	T	h	V	r_1	r_0
I	Step-out	0	1	1	T	h	V	r_1	r_0
II	Read Sector	1	0	0	m	L	E	U	0
II	Write Sector	1	0	1	m	L	E	U	a_0
III	Read Address	1	1	0	0	0	E	U	0
III	Read Track	1	1	1	0	0	E	U	0
III	Write Track	1	1	1	1	0	E	U	0
IV	Force Interrupt	1	1	0	1	I_3	I_2	I_1	I_0

FLAG SUMMARY

TABLE 2. FLAG SUMMARY

Command Type	Bit No(s)		Description
I	0, 1	r_1 r_0 = Stepping Motor Rate See Table 3 for Rate Summary	
I	2	V = Track Number Verify Flag	V = 0, No verify V = 1, Verify on destination track
I	3	h = Head Load Flag	h = 0, Unload head at beginning h = 1, Load head at beginning
I	4	T = Track Update Flag	T = 0, No update T = 1, Update track register
II & III	0	a_0 = Data Address Mark	a_0 = 0, FB (DAM) a_0 = 1, F8 (deleted DAM)
II	1	C = Side Compare Flag	C = 0, Disable side compare C = 1, Enable side compare
II & III	1	U = Update SSO	U = 0, Update SSO to 0 U = 1, Update SSO to 1
II & III	2	E = 15 MS Delay	E = 0, No. 15 MS delay E = 1, 15 MS delay (30 MS for 1 MHz)
II	3	S = Side Compare Flag	S = 0, Compare for side 0 S = 1, Compare for side 1
II	3	L = Sector Length Flag	LSB's Sector Length in ID Field: 00 / 01 / 10 / 11 L = 0: 256 / 512 / 1024 / 128 L = 1: 128 / 256 / 512 / 1024
II	4	m = Multiple Record Flag	m = 0, Single record m = 1, Multiple records
IV	0-3	I_x = Interrupt Condition Flags I_0 = 1 Not Ready To Ready Transition I_1 = 1 Ready To Not Ready Transition I_2 = 1 Index Pulse I_3 = 1 Immediate Interrupt, Requires A Reset* I_3-I_0 = 0 Terminate With No Interrupt (INTRQ)	

*NOTE: See Type IV Command Description for further information.

Write Precompensation

When operating in Double Density mode ($\overline{DDEN}$ = 0), the 279X has the capability of providing a user-defined precompensation value for Write Data. An external potentiometer (10K) tied to the WPW signal (Pin 33) allows a setting of 100 to 300 ns from nominal.

Setting the Write precomp value is accomplished by forcing the $\overline{TEST}$ line (Pin 22) to a Logic 0. A stream of pulses can then be seen on the Write Data (Pin 31) line. Adjust the WPW Potentiometer for the desired pulse width. This adjustment may be performed in-circuit since Write Gate (Pin 30) is inactive while $\overline{TEST}$ = 0.

Data Separation

The 279X can operate with either an external data separator or its own internal recovery circuits. The condition of the $\overline{TEST}$ line (Pin 22) in conjunction with $\overline{MR}$ (Pin 19) will select internal or external mode.

To program the 279X for external VCO, a $\overline{MR}$ pulse must be applied while $\overline{TEST}$ = 0. A clock equivalent to eight times the data rate (e.g., 4.0 MHz for 8" Double Density) is applied to the VCO input (Pin 26). The feedback reference voltage is available on the Pump output (Pin 23) for external integration to control the VCO. $\overline{TEST}$ is returned to a Logic 1 for normal operation. Note: To maintain this mode, $\overline{TEST}$ must be held low whenever $\overline{MR}$ is applied.

For internal VCO operation, the $\overline{TEST}$ line must be high during the $\overline{MR}$ pulse, then set to a Logic 0 for the adjustment procedure.

A 50K Potentiometer tied to the RPW input (Pin 18) is used to set the internal Read Data pulse for proper phasing. With a scope on Pin 29 (TG43), adjust the RPW pulse for 1/8 of the data rate (250 ns for 8" Double Density). An external variable capacitor of 5-60 pf is tied to the VCO input (Pin 26) for adjusting center frequency. With a frequency counter on Pin 16 (DIRC) adjust the trimmer cap to yield the appropriate Data Rate (500 KHz for 8" Double Density). The $\overline{DDEN}$ line must be low while the 5/8 line is held high or the adjustment times above will be doubled.

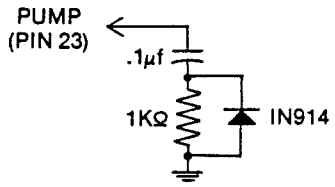
After adjustments have been made, the $\overline{TEST}$ pin is returned to a Logic 1 and the device is ready for operation. Adjustments may be made in-circuit since the DIRC and TG43 lines may toggle without affecting the drive.

The PUMP output (Pin 23) consists of positive and negative pulses, which their duration is equivalent to the phase difference of incoming Data vs. VCO frequency. This signal is internally connected to the VCO input, but a Filter is needed to connect these pulses to a slow moving DC voltage.

The internal phase-detector is unsymmetrical for a random distribution of data pulses by a factor of two, in favor of a PUMP UP condition. Therefore, it is desirable to have a PUMP DOWN twice as responsive to prevent run-away during a lock attempt.

A first order lag-lead filter can be used at the PUMP output (Pin 23). This filter controls the instantaneous response of the VCO to bit-shifted data (jitter) as well as the response to normal frequency shift, i.e., the lock-up time. A balance must be accomplished between the two conditions to inhibit over-responsiveness to jitter and to prevent an extremely wide lock-up response, leading to PUMP run-away. The filter affects these two reactions in mutually opposite directions.

The following Filter Circuit is recommended for 8" FM/MFM:



Since 5¼" Drives operate at exactly one-half the data rate (250 Kb/sec) the above capacitor should be doubled to .2 or .22μf.

TYPE I COMMANDS

The Type I Commands include the Restore, Seek, Step, Step-in, and Step-Out commands. Each of the Type I Commands contains a rate field (r_0 r_1), which determines the stepping motor rate as defined in Table 3.

A 2μs (MFM) or 4 μs (FM) pulse is provided as an output to the drive. For every step pulse issued, the drive moves one track location in a direction determined by the direction output. The chip will step the drive in the same direction it last stepped unless the command changes the direction.

The Direction signal is active high when stepping in and low when stepping out. The Direction signal is valid before the first stepping pulse is generated.

The rates (shown in Table 3) can be applied to a Step-Direction Motor through the device interface.

TABLE 3. STEPPING RATES

CLK		2 MHz	1 MHz
R1	R0	$\overline{TEST}$ = 1	$\overline{TEST}$ = 1
0	0	3 ms	6 ms
0	1	6 ms	12 ms
1	0	10 ms	20 ms
1	1	15 ms	30 ms

After the last directional step an additional 15 milliseconds of head settling time takes place if the Verify flag is set in Type I commands. Note that this time doubles to 30 ms for a 1 MHz clock. There is also a 15 ms head settling time if the E flag is set in any Type II or III command.

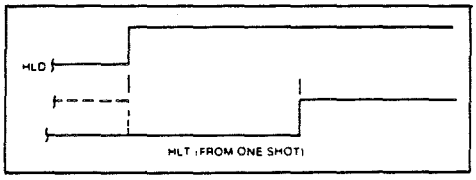
When a Seek, Step or Restore command is executed an optional verification of Read-Write head position can be performed by setting bit 2 (V = 1) in the command word to a logic 1. The verification operation begins at the end of the 15 millisecond settling time after the head is loaded against the media. The track number from the first encountered ID

Field is compared against the contents of the Track Register. If the track numbers compare and the ID Field Cyclic Redundancy Check (CRC) is correct, the verify operation is complete and an INTRQ is generated with no errors. If there is a match but not a valid CRC, the CRC error status bit is set (Status bit 3), and the next encountered ID field is read from the disk for the verification operation.

The WD279X must find an ID field with correct track number and correct CRC within 5 revolutions of the media; otherwise the seek error is set and an INTRQ is generated. If V = 0, no verification is performed.

The Head Load (HLD) output controls the movement of the read/write head against the media. HLD is activated at the beginning of a Type I command if the h flag is set (h = 1), at the end of the Type I command if the verify flag (V = 1), or upon receipt of any Type II or III command. Once HLD is active it remains active until either a Type I command is received with (h = 0 and V = 0); or if the 279X is in an idle state (non-busy) and 15 index pulses have occurred.

Head Load timing (HLT) is an input to the 279X which is used for the head engage time. When HLT = 1, the 279X assumes the head is completely engaged. The head engage time is typically 30 to 100 ms depending on drive. The low to high transition on HLD is typically used to fire a one shot. The output of the one shot is then used for HLT and supplied as an input to the 279X.



HEAD LOAD TIMING

When both HLD and HLT are true, the 279X will then read from or write to the media. The "and" of HLD and HLT appears as status Bit 5 in Type I status.

In summary for the Type I commands: if h = 0 and V = 0, HLD is reset. If h = 1 and V = 0, HLD is set at the beginning of the command and HLT is not sampled nor is there an internal 15 ms delay. If h = 0 and V = 1, HLD is set near the end of the command, an internal 15 ms occurs, and the 279X waits for HLT to be true. If h = 1 and V = 1, HLD is set at the beginning of the command. Near the end of the command, after all the steps have been issued, an internal 15 ms delay occurs and the 279X then waits for HLT to occur.

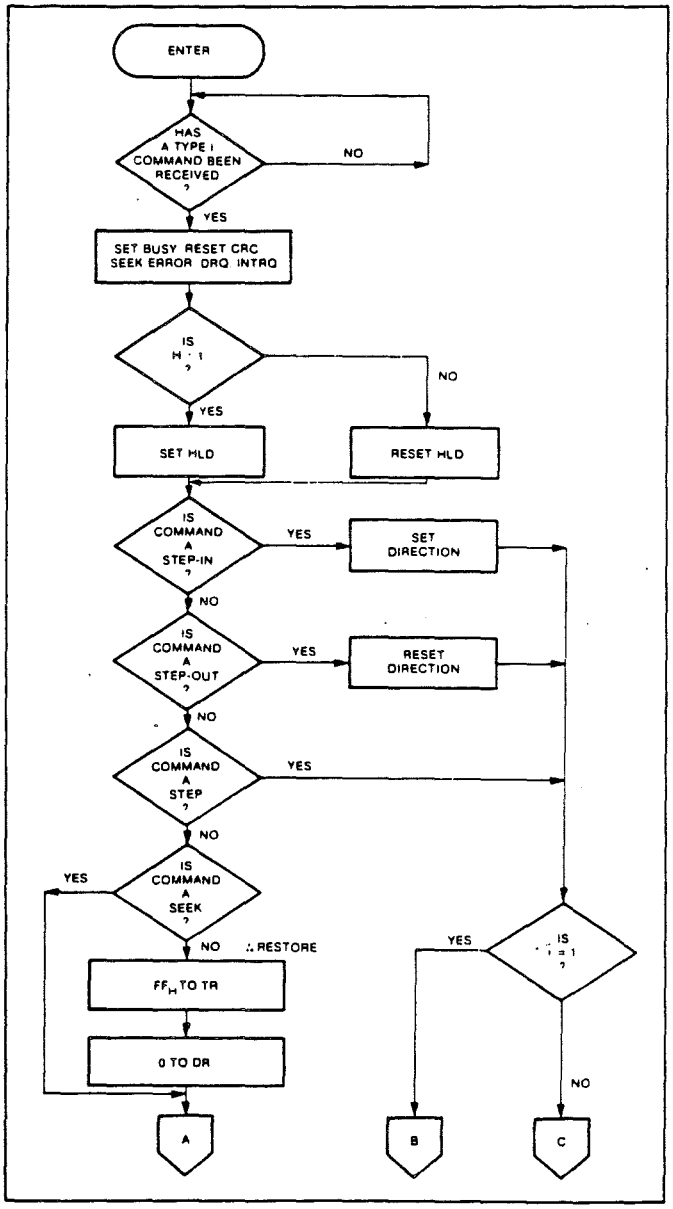
For Type II and III commands with E flag off, HLD is made active and HLT is sampled until true. With E flag on, HLD is made active, an internal 15 ms delay occurs and then HLT is sampled until true.

RESTORE (SEEK TRACK 0)

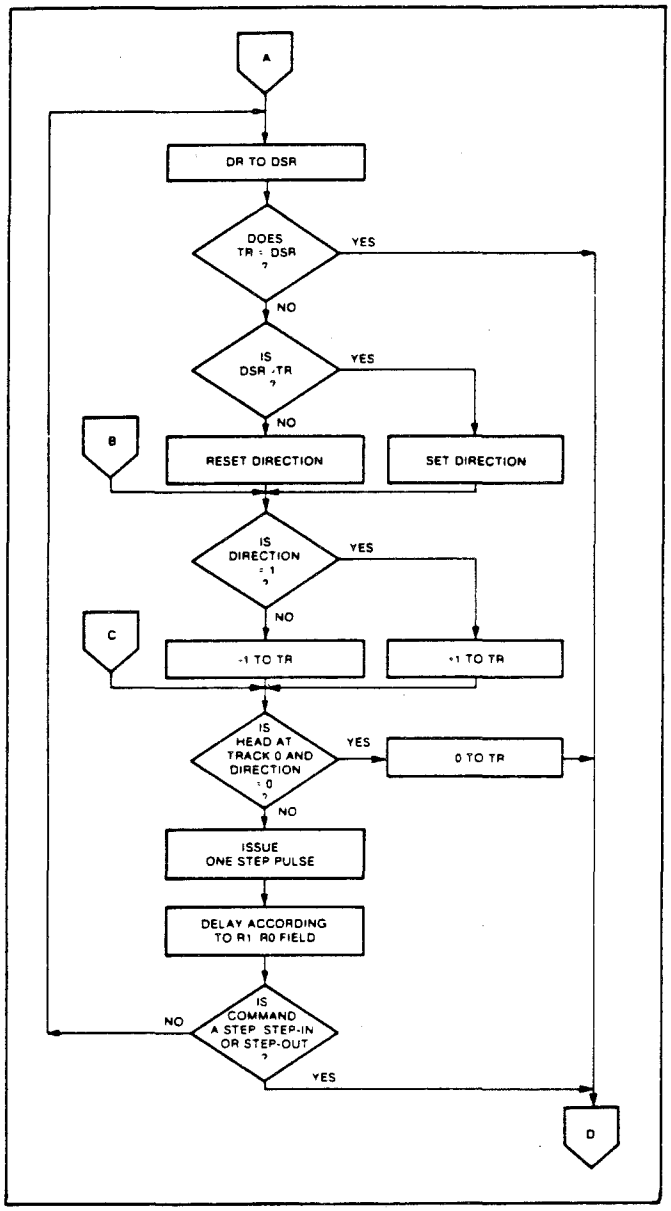
Upon receipt of this command the Track 00 ($\overline{TR00}$) input is sampled. If $\overline{TR00}$ is active low indicating the Read-Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TR00}$ is not active low, stepping pulses at a rate specified by the r1r0 field are issued until the $\overline{TR00}$ input is activated. At this time the Track Register is loaded with zeroes and an interrupt is generated. If the $\overline{TR00}$ input does not go active low after 255 stepping pulses, the 279X terminates operation, interrupts, and sets the Seek error status bit. A verification operation takes place if the V flag is set. The h bit allows the head to be loaded at the start of command. Note that the Restore command is executed when $\overline{MR}$ goes from an active to an inactive state.

SEEK

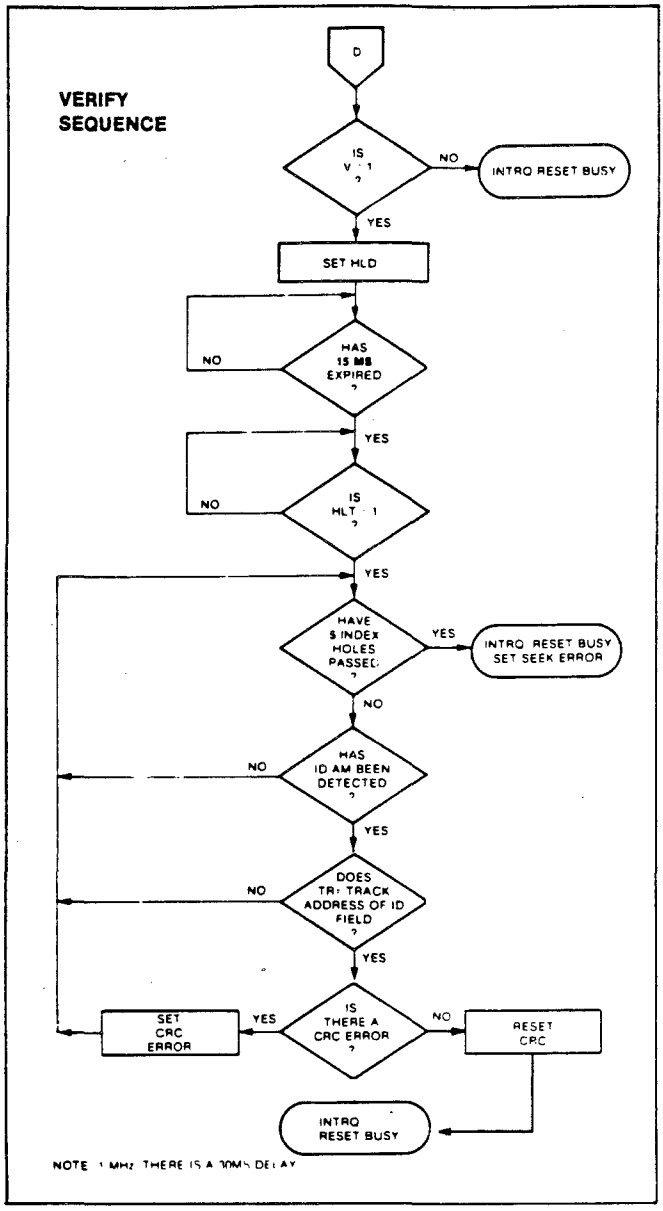
This command assumes that the Track Register contains the track number of the current position of the Read-Write head and the Data Register contains the desired track number. The WD279X will update the Track register and issue stepping pulses in the appropriate direction until the



TYPE I COMMAND FLOW



TYPE I COMMAND FLOW



TYPE I COMMAND FLOW

contents of the Track register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command. Note: When using multiple drives, the track register must be updated for the drive selected before seeks are issued.

STEP

Upon receipt of this command, the 279X issues one stepping pulse to the disk drive. The stepping motor direction is the same as in the previous step command. After a delay determined by the r1r0 field, a verification takes place if the V flag is on. If the T flag is on, the Track Register is updated. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-IN

Upon receipt of this command, the 279X issues one stepping pulse in the direction towards track 76. If the T flag is on, the Track Register is incremented by one. After a delay determined by the r1r0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-OUT

Upon receipt of this command, the 279X issues one stepping pulse in the direction towards track 0. If the T flag is on, the Track Register is decremented by one. After a delay determined by the r1r0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

EXCEPTIONS

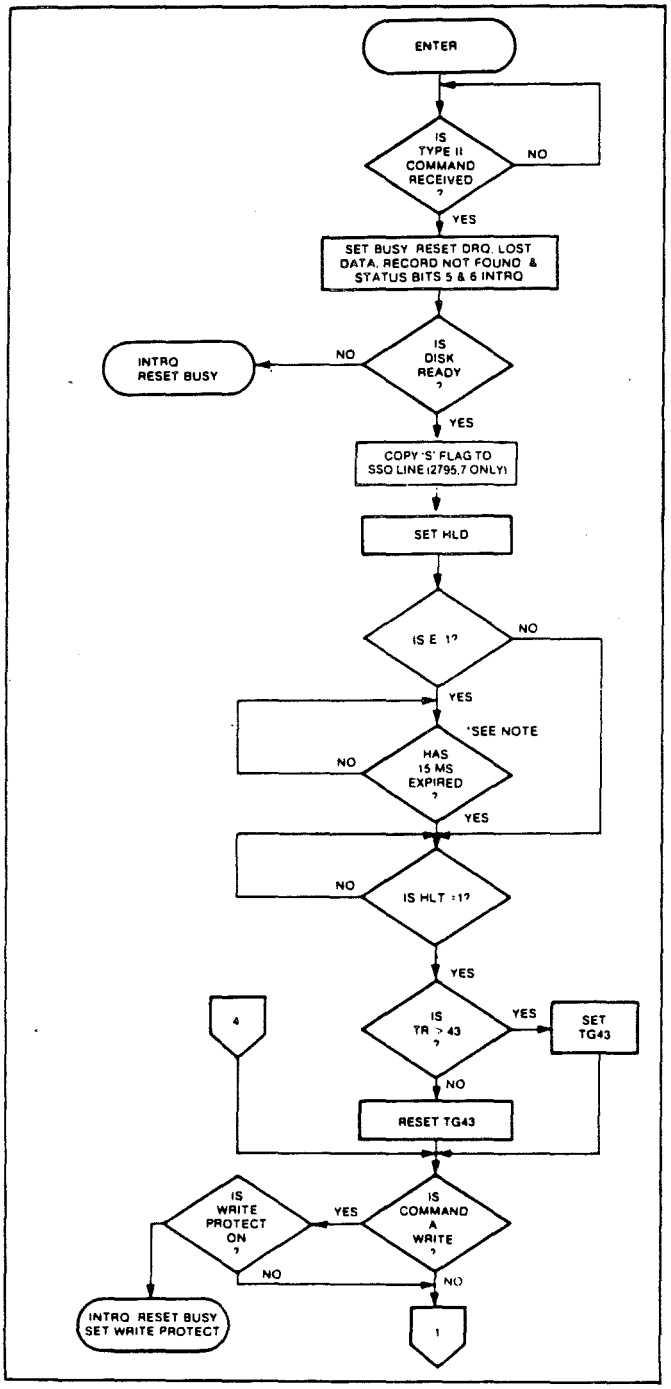
On the 2795/7 devices, the SSO output is not affected during Type I commands, and an internal side compare does not take place when the (V) Verify Flag is on.

TYPE II COMMANDS

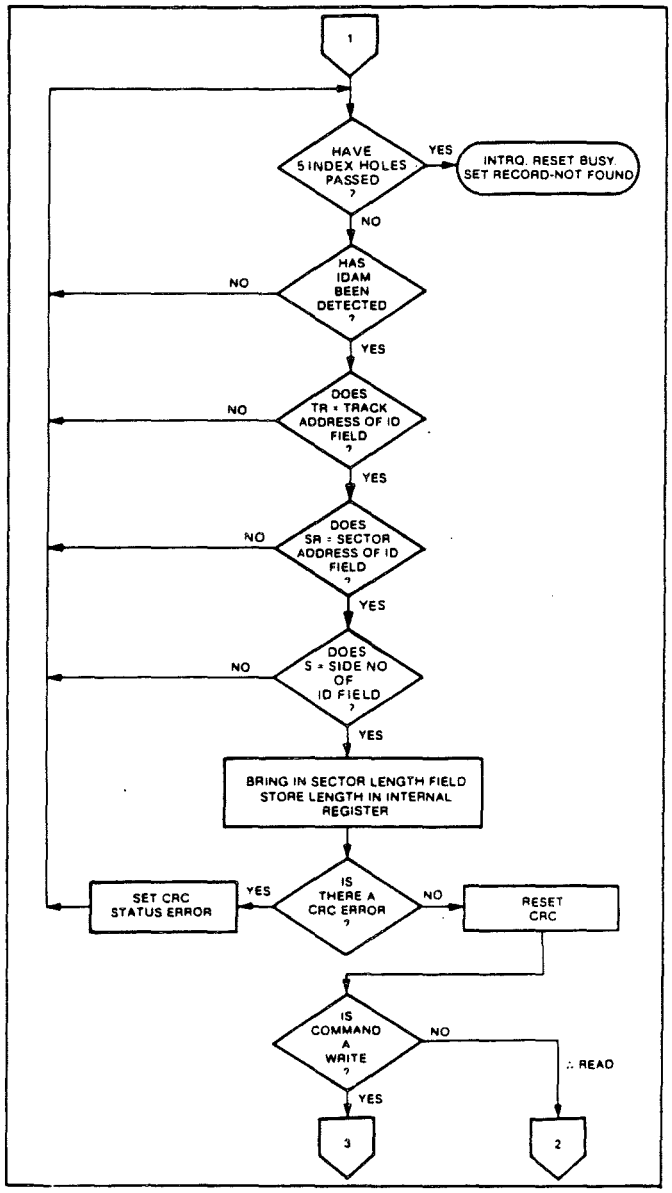
The Type II Commands are the Read Sector and Write Sector commands. Prior to loading the Type II Command into the Command Register, the computer must load the

Sector Register with the desired sector number. Upon receipt of the Type II command, the busy status Bit is set. If the E flag = 1 (this is the normal case) HLD is made active and HLT is sampled after a 15 msec delay. If the E flag is 0, the head is loaded and HLT sampled with no 15 msec delay.

When an ID field is located on the disk, the 279X compares the Track Number on the ID field with the Track Register. If there is not a match, the next encountered ID field is read and a comparison is again made. If there was a match, the Sector Number of the ID field is compared with the Sector Register. If there is not a Sector match, the next encountered ID field is read off the disk and comparisons again made. If the ID field CRC is correct, the data field is then located and will be either written into, or read from



TYPE II COMMAND

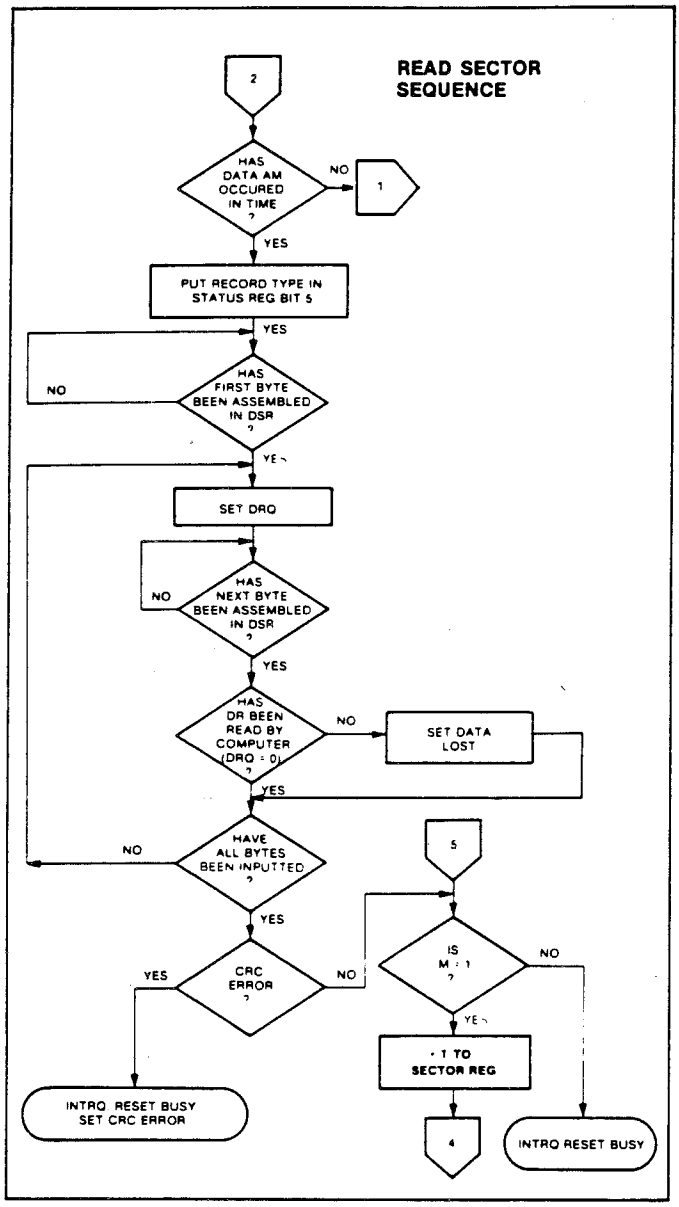


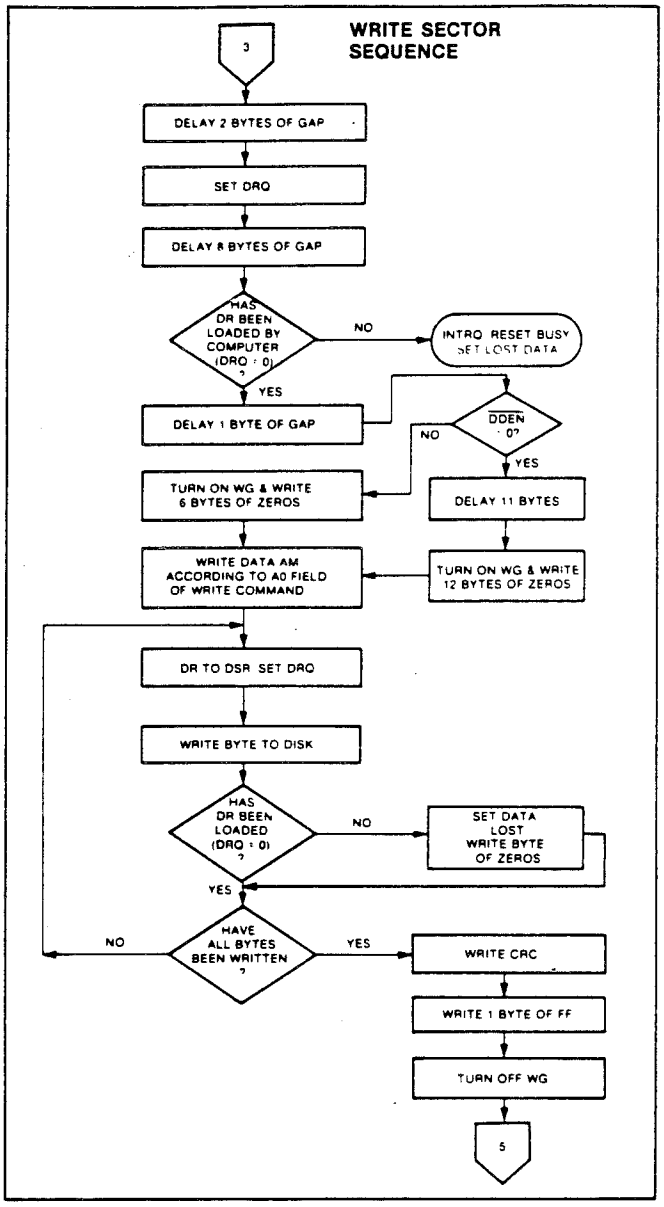
TYPE II COMMAND

depending upon the command. The 279X must find an ID field with a Track number, Sector number, side number, and CRC within 5 revolutions of the disk; otherwise, the Record not found status bit is set (Status bit 4) and the command is terminated with an interrupt.

Each of the Type II Commands contains an (m) flag which determines if multiple records (sectors) are to be read or written, depending upon the command. If m = 0, a single sector is read or written and an interrupt is generated at the completion of the command. If m = 1, multiple records are read or written with the sector register internally updated so that an address verification can occur on the next record. The 279X will continue to read or write multiple records and update the sector register in numerical ascending sequence until the sector register exceeds the number of sectors on the track or until the Force Interrupt command is loaded into the Command Register, which terminates the command and generates an interrupt.

For example: If the 279X is instructed to read sector 27 and there are only 26 on the track, the sector register exceeds





TYPE II COMMAND

the number available. The 279X will search for 5 disk revolutions, interrupt out, reset busy, and set the record not found status bit.

The Type II commands for 2791-93 also contain side select compare flags. When C = 0 (Bit 1) no side comparison is made. When C = 1, the LSB of the side number is read off the ID Field of the disk and compared with the contents of the (S) flag (Bit 3). If the S flag compares with the side number recorded in the ID field, the 279X continues with the ID search. If a comparison is not made within 5 index pulses, the interrupt line is made active and the Record-Not-Found status bit is set.

The Type II and III commands for the 2795-97 contain a side select flag (Bit 1). When U = 0, SSO is updated to 0. Similarly, U = 1 updates SSO to 1. The chip compares the SSO to the ID field. If they do not compare within 5 revolutions the interrupt line is made active and the RNF status bit is set.

The 2795/7 READ SECTOR and WRITE SECTOR commands include a 'L' flag. The 'L' flag, in conjunction with the sector length byte of the ID Field, allows different byte lengths to be implemented in each sector. For IBM compatibility, the 'L' flag should be set to a one.

READ SECTOR

Upon receipt of the Read Sector command, the head is loaded, the Busy status bit set, and when an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, the data field is presented to the computer. The Data Address Mark of the data field must be found within 30 bytes in single density and 43 bytes in double density of the last ID field CRC byte; if not, the ID field search is repeated.

When the first character or byte of the data field has been shifted through the DSR, it is transferred to the DR, and DRQ is generated. When the next byte is accumulated in the DSR, it is transferred to the DR and another DRQ is generated. If the Computer has not read the previous contents of the DR before a new character is transferred

that character is lost and the Lost Data Status bit is set. This sequence continues until the complete data field has been inputted to the computer. If there is a CRC error at the end of the data field, the CRC error status bit is set, and the command is terminated (even if it is a multiple sector command).

At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Bit 5) as shown:

STATUS BIT 5	
1	Deleted Data Mark
0	Data Mark

WRITE SECTOR

Upon receipt of the Write Sector command, the head is loaded (HLD active) and the Busy status bit is set. When an ID field is encountered that has the correct track number, correct sector number, correct side number, and correct CRC, a DRQ is generated. The 279X counts off 11 bytes in single density and 22 bytes in double density from the CRC field and the Write Gate (WG) output is made active if the DRQ is serviced (i.e., the DR has been loaded by the computer). If DRQ has not been serviced, the command is terminated and the Lost Data status bit is set. If the DRQ has been serviced, the WG is made active and six bytes of zeroes in single density and 12 bytes in double density are then written on the disk. At this time the Data Address Mark is then written on the disk as determined by the a_0 field of the command as shown below:

a_0	Data Address Mark (Bit 0)
1	Deleted Data Mark
0	Data Mark

The 279X then writes the data field and generates DRQ's to the computer. If the DRQ is not serviced in time for continuous writing the Lost Data Status Bit is set and a byte of zeroes is written on the disk. The command is not terminated. After the last data byte has been written on the disk, the two-byte CRC is computed internally and written on the disk followed by one bye of FE in FM or in MFM. The WG output is then deactivated. For a 2 MHz clock the INTRQ will set 8 to 12 μsec after the last CRC byte is written. For partial sector writing, the proper method is to write the data and fill the balance with zeroes. By letting the chip fill the zeroes, errors may be masked by the lost data status and improper CRC Bytes.

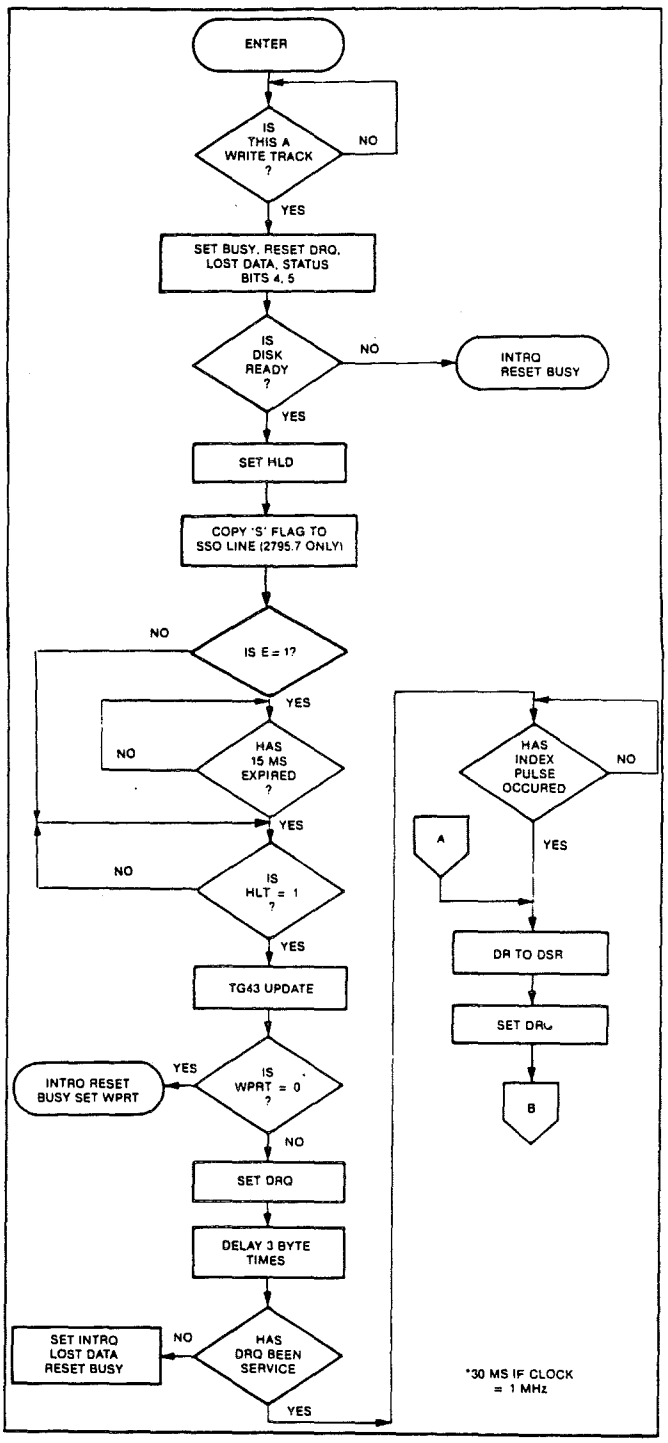
TYPES III COMMANDS

READ ADDRESS

Upon receipt of the Read Address command, the head is loaded and the Busy Status Bit is set. The next encountered ID field is then read in from the disk, and the six data bytes of the ID field are assembled and transferred to the DR, and a DRQ is generated for each byte. The six bytes of the ID field are shown below:

TRACK ADDR	SIDE NUMBER	SECTOR ADDRESS	SECTOR LENGTH	CRC 1	CRC 2
1	2	3	4	5	6

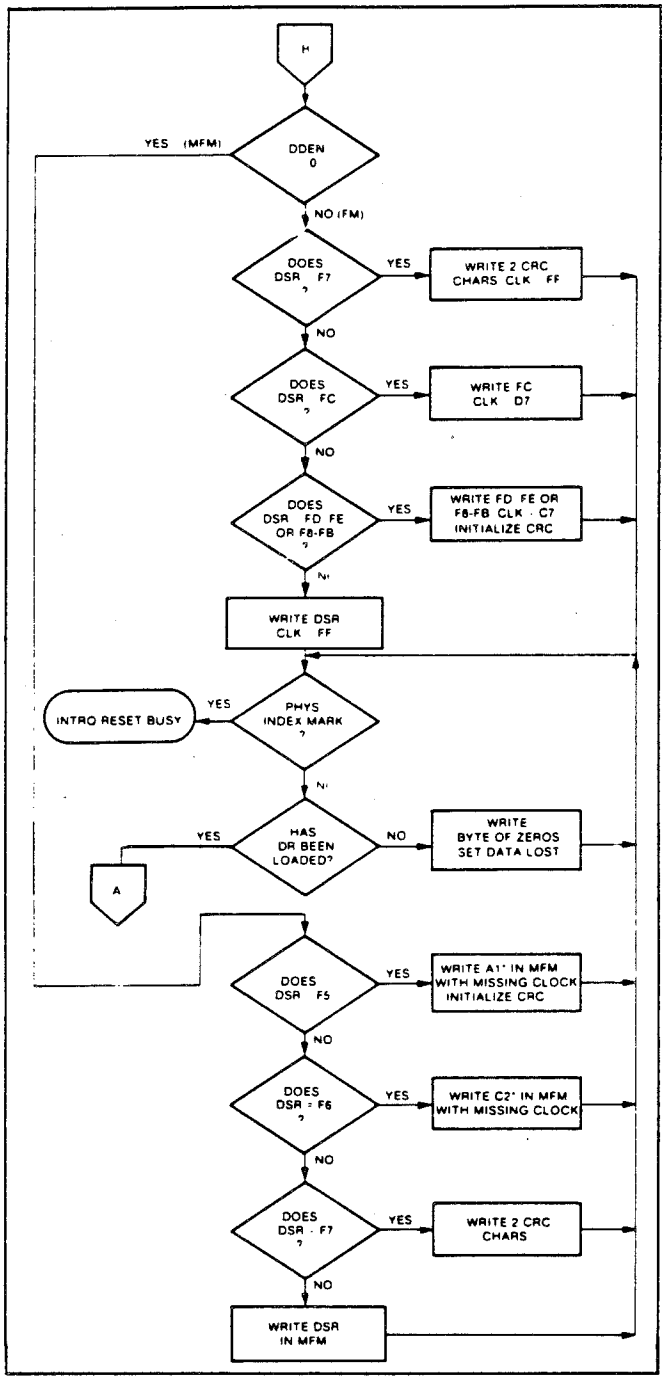
Although the CRC characters are transferred to the computer, the 279X checks for validity and the CRC error status bit is set if there is a CRC error. The Track Address of the ID field is written into the sector register so that a comparison can be made by the host. At the end of the operation an interrupt is generated and the Busy Status is reset.



TYPE III COMMAND WRITE TRACK

READ TRACK

Upon receipt of the READ track command, the head is loaded, and the Busy Status bit is set. Reading starts with the leading edge of the first encountered index pulse and continues until the next index pulse. All Gap, Header, and data bytes are assembled and transferred to the data register and DRQ's are generated for each byte. The ac-



TYPE III COMMAND WRITE TRACK

cumulation of bytes is synchronized to each address mark encountered. An interrupt is generated at the completion of the command.

This command has several characteristics which make it suitable for diagnostic purposes. They are: no CRC checking is performed; gap information is included in the data stream; the internal side compare is not performed; and the address mark detector is on for the duration of the command. Because the A.M. detector is always on, write splices or noise may cause the chip to look for an A.M. If an address mark does not appear on schedule with the Lost Data status flag being set.

The ID A.M., ID field, ID CRC bytes, DAM, Data and Data CRC Bytes for each sector will be correct. The Gap Bytes may be read incorrectly during write-splice time because of synchronization.

WRITE TRACK FORMATTING THE DISK

(Refer to section on Type III commands for flow diagrams.)

Formatting the disk is a relatively simple task when operating programmed I/O or when operating under DMA with a large amount of memory. Data and gap information must be provided at the computer interface. Formatting the disk is accomplished by positioning the R/W head over the desired track number and issuing the Write Track command.

Upon receipt of the Write Track command, the head is loaded and the Busy Status bit is set. Writing starts with the leading edge of the first encountered index pulse and continues until the next index pulse, at which time the interrupt is activated. The Data Request is activated immediately upon receiving the command, but writing will not start until after the first byte has been loaded into the Data Register. If the DR has not been loaded by the time the index pulse is encountered the operation is terminated making the device Not Busy, the Lost Data Status Bit is set, and the interrupt is activated. If a byte is not present in the DR when needed, a byte of zeroes is substituted.

This sequence continues from one index mark to the next index mark. Normally, whatever data pattern appears in the data register is written on the disk with a normal clock pattern. However, if the 279X detects a data pattern of F5 thru FE in the data register, this is interpreted as data address marks with missing clocks or CRC generation.

The CRC generator is initialized when any data byte from F8 to FE is about to be transferred from the DR to the DSR

CONTROL BYTES FOR INITIALIZATION

DATA PATTERN IN DR (HEX)	WD279X INTERPRETATION IN FM ($\overline{DDEN}$ = 1)	WD279X INTERPRETATION IN MFM ($\overline{DDEN}$ = 0)
00 thru F4	Write 00 thru F4 with CLK = FF	Write 00 thru F4, in MFM
F5	Not Allowed	Write A1* in MFM, Preset CRC
F6	Not Allowed	Write C2** in MFM
F7	Generate 2 CRC bytes	Generate 2 CRC bytes
F8 thru FB	Write F8 thru FB, Clk = C7, Preset CRC	Write F8 thru FB, in MFM
FC	Write FC with Clk = D7	Write FC in MFM
FD	Write FD with Clk = FF	Write FD in MFM
FE	Write FE, Clk = C7, Preset CRC	Write FE in MFM
FF	Write FF with Clk = FF	Write FF in MFM

* Missing clock transition between bits 4 and 5

** Missing clock transition between bits 3 and 4

or by receipt of F5 in MFM. An F7 pattern will generate two CRC characters in FM or MFM. As a consequence, the patterns F5 thru FE must not appear in the gaps, data fields, or ID fields. Also, CRC's must be generated by an F7 pattern.

Disks may be formatted in IBM 3740 or System 34 formats with sector lengths of 128, 256, 512, or 1024 bytes.

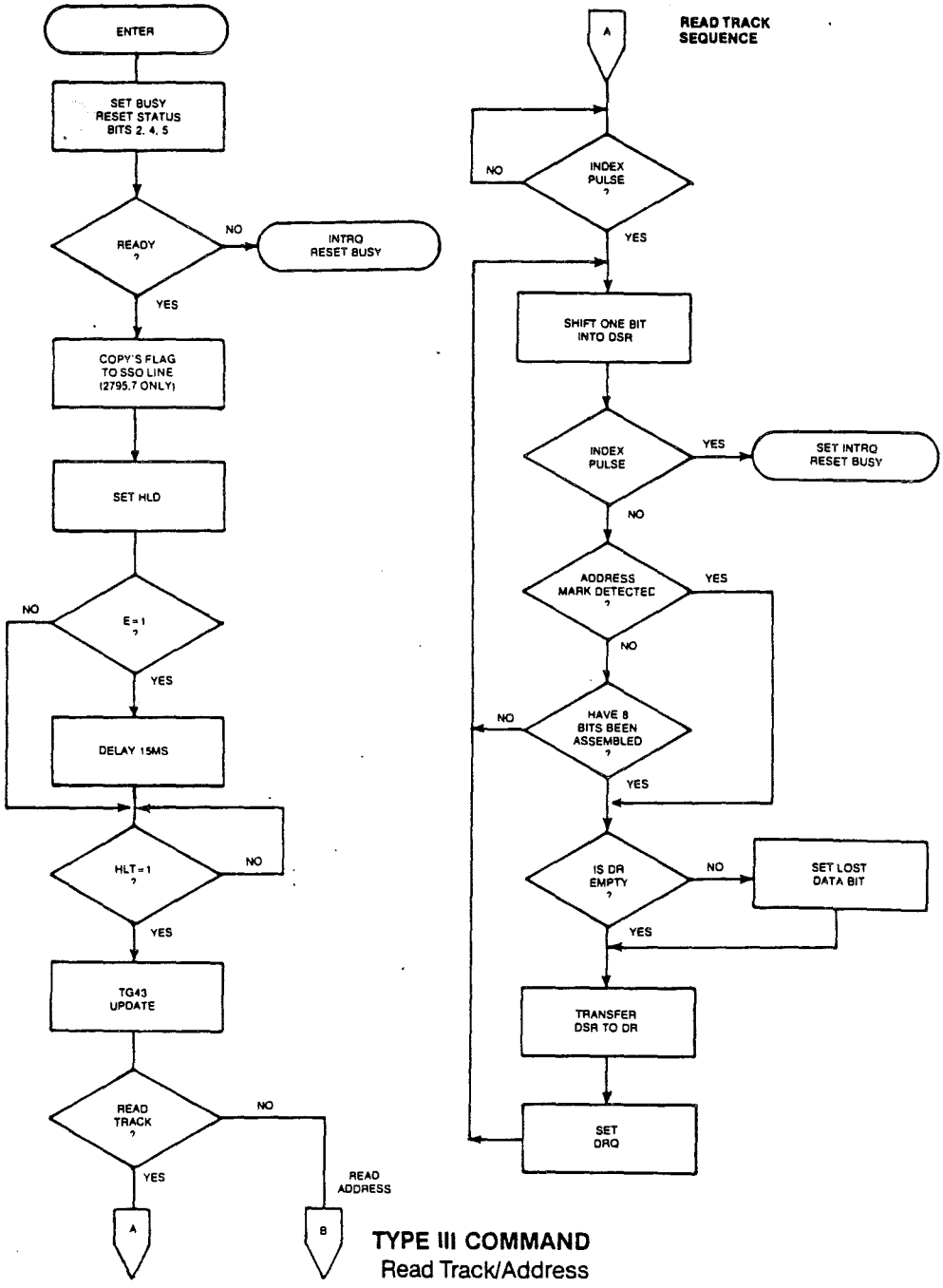
TYPE IV COMMANDS

The Forced Interrupt command is generally used to terminate a multiple sector read or write command or to insure Type I status in the status register. This command can be loaded into the command register at any time. If there is a current command under execution (busy status bit set) the command will be terminated and the busy status bit reset.

The lower four bits of the command determine the conditional interrupt as follows:

I_0 = Not-Ready to Ready Transition
I_1 = Ready to Not-Ready Transition
I_2 = Every Index Pulse
I_3 = Immediate Interrupt

The conditional interrupt is enabled when the corresponding bit positions of the command (I_3 - I_0) are set to a 1. Then, when the condition for interrupt is met, the INTRQ line will go high signifying that the condition specified has occurred. If I_3 - I_0 are all set to zero (HEX D0), no interrupt will occur but any command presently under execution will be immediately terminated. When using the immediate



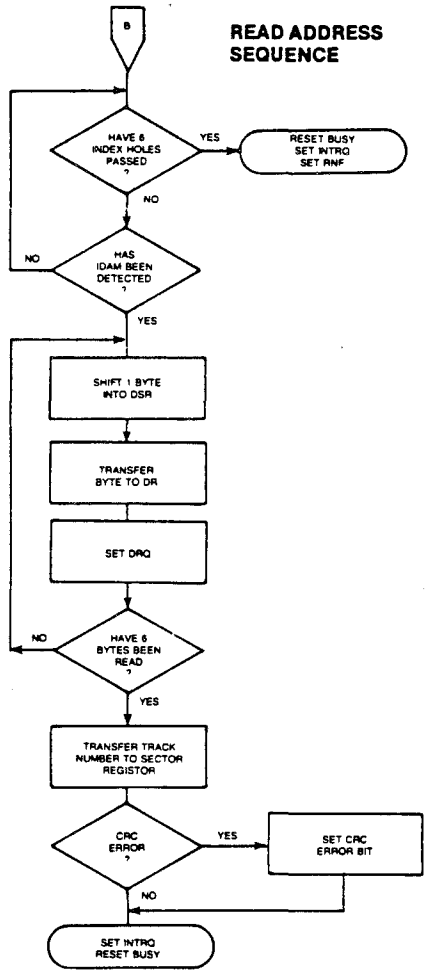
TYPE III COMMAND
Read Track/Address

interrupt condition I3 = 1), an interrupt will be immediately generated and the current command terminated. Reading the status or writing to the command register will not automatically clear the interrupt. The HEX D0 is the only command that will enable the immediate interrupt (HEX D8) to clear on a subsequent load command register or read status register operation. Follow a HEX D8 with D0 command.

Wait 8 micro sec (double density) or 16 micro sec (single density) before issuing a new command after issuing a forced interrupt (times double when clock = 1 MHz). Loading a new command sooner than this will nullify the forced interrupt.

Forced interrupt stops any command at the end of an internal micro-instruction and generates INTRQ when the specified condition is met. Forced interrupt will wait until ALU operations in progress are complete (CRC calculations, compares, etc.)

More than one condition may be set at a time. If for example, the READY TO NOT-READY condition (I1 = 1) and the Every Index Pulse (I2 = 1) are both set, the resultant command would be HEX "DA." The "OR" function is performed so that either a READY TO NOT-READY or the next Index Pulse will cause an interrupt condition.



TYPE III COMMAND
Read Track/Address

STATUS REGISTER

Upon receipt of any command, except the Force Interrupt command, the Busy Status bit is set and the rest of the status bits are updated or cleared for the new command. If the Force Interrupt Command is received when there is a current command under execution, the Busy status bit is reset, and the rest of the status bits are unchanged. If the Force Interrupt command is received when there is not a current command under execution, the Busy Status bit is reset and the rest of the status bits are updated or cleared. In this case, Status reflects the Type I commands.

The user has the option of reading the status register through program control or using the DRQ line with DMA or interrupt methods. When the Data register is read the DRQ bit in the status register and the DRQ line are automatically reset. A write to the Data register also causes both DRQ's to reset.

The busy bit in the status may be monitored with a user program to determine when a command is complete, in lieu of using the INTRQ line. When using the INTRQ, a busy status check is not recommended because a read of the status register to determine the condition of busy will reset the INTRQ line.

The format of the Status Register is shown below:

(BITS)							
7	6	5	4	3	2	1	0
S7	S6	S5	S4	S3	S2	S1	S0

Status varies according to the type of command executed as shown in Table 4.

Because of internal sync cycles, certain time delays must be observed when operating under programmed I/O. They are: (times double when clock = 1 MHz)

Operation	Next Operation	Delay Req'd. FM	MFM
Write to Command Reg.	Read Busy Bit (Status Bit 0)	12μs	6μs
Write to Command Reg.	Read Status Bits 1-7	28μs	14μs
Write Any Register	Read From Diff. Register	0	0

IBM 3740 FORMAT — 128 BYTES/SECTOR

Shown below is the IBM single-density format with 128 bytes/sector. In order to format a diskette, the user must issue the Write Track command, and load the data register with the following values. For every byte to be written, there is one Data Request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
40	FF (or 00)[3]
6	00
1	FC (Index Mark)
1 26	FF (or 00)
6	00
1	FE (ID Address Mark)
1	Track Number
1	Side Number (00 or 01)
1	Sector Number (1 thru 1A)
1	00 (Sector Length)
1	F7 (2 CRC's written)
11	FF (or 00)
6	00
1	FB (Data Address Mark)
128	Data (IBM uses E5)
1	F7 (2 CRC's written)
27	FF (or 00)
247[2]	FF (or 00)

1. Write bracketed field 26 times
2. Continue writing until 279X interrupts out. Approx. 247 bytes.
3. A '00' option is allowed.

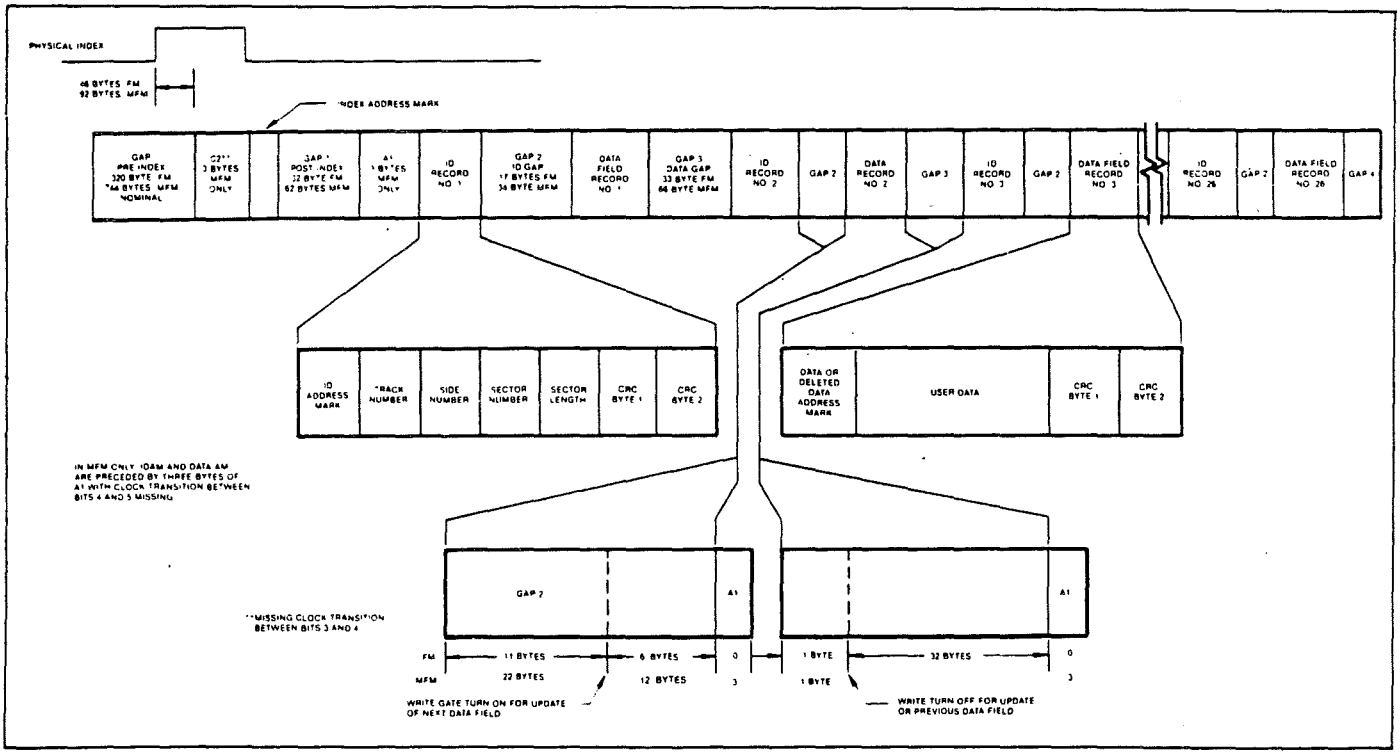
IBM SYSTEM 34 FORMAT-256 BYTES/SECTOR

Shown below is the IBM dual-density format with 256 bytes/sector. In order for format a diskette the user must issue the Write Track command and load the data register with the following values. For every byte to be written, there is one data request.

NUMBER OF BYTES	HEX VALUE OF BYTE WRITTEN
80	4E
12	00
3	F6 (Writes C2)
1	FC (Index Mark)
* 50	4E
12	00
3	F5 (Writes A1)
1	FE (ID Address Mark)
1	Track Number (0 thru 4C)
1	Side Number (0 or 1)
1	Sector Number (1 thru 1A)
1	01 (Sector Length)
1	F7 (2 CRCs written)
22	4E
12	00
3	F5 (Writes A1)
1	FB (Data Address Mark)
256	DATA
1	F7 (2 CRCs written)
54	4E
598**	4E

*Write bracketed field 26 times

**Continue writing until 279X interrupts out. Approx. 598 bytes.



IBM TRACK FORMAT

1. NON-IBM FORMATS

Variations in the IBM formats are possible to a limited extent if the following requirements are met:

1) Sector size must be 128, 256, 512 of 1024 bytes.
2) Gap 2 cannot be varied from the IBM format.
3) 3 bytes of A1 must be used in MFM.

In addition, the Index Address Mark is not required for operation by the 279X. Gap 1, 3, and 4 lengths can be as short as 2 bytes for 279X operation, however PLL lock up time, motor speed variation, write splice area, etc. will add more bytes to each gap to achieve proper operation. It is recommended that the IBM format be used for highest system reliability.

	FM	MFM
Gap I	16 bytes FF	32 bytes 4E
Gap II	11 bytes FF	22 bytes 4E
*	6 bytes 00	12 bytes 00
*		3 bytes A1
Gap III**	10 bytes FF	24 bytes 4E
	4 bytes 00	8 bytes 00
		3 bytes A1
Gap IV	16 bytes FF	16 bytes 4E

* Byte counts must be exact.
** Byte counts are minimum, except exactly 3 bytes of A1 must be written.

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Voltage to any input with
respect to V_{SS} = +7 to −0.5V
Operating temperature = 0°C to 70°C
Storage temperature = −55°C to +125°C

NOTE: Maximum limits indicate where permanent device damage occurs. Continuous operation at these limits is not intended and should be limited to those conditions specified in the DC Electrical characteristics.

OPERATING CHARACTERISTICS (DC)

T_A = 0°C to 70°C, V_{SS} = 0V, V_{CC} = + 5V ± .25V

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
I_{IL}	Input Leakage			10	μA	$V_{IN} = V_{CC}$
I_{OL}	Output Leakage			10	μA	$V_{OUT} = V_{CC}$
V_{IH}	Input High Voltage	2.0			V	
V_{IL}	Input Low Voltage			0.8	V	
V_{OH}	Output High Voltage	2.4			V	I_O = −100μA
V_{OL}	Output Low Voltage			0.45	V	I_O = 1.6 mA
V_{OHP}	Output High PUMP	2.2			V	I_{OP} = −1.0 mA
V_{OLP}	Output Low PUMP			0.2	V	I_{OP} = +1.0 mA
P_D	Power Dissipation			.75	W	All Outputs Open
R_{PU}	Internal Pull-up*	100		1700	μA	V_{IN} = 0V
I_{CC}	Supply Current		70	150	mA	All Outputs Open

* Internal Pull-up resistors on PINS 1, 17, 19, 22, 36, 37 and 40. Also pin 25 on 2791 and 3.

TIMING CHARACTERISTICS

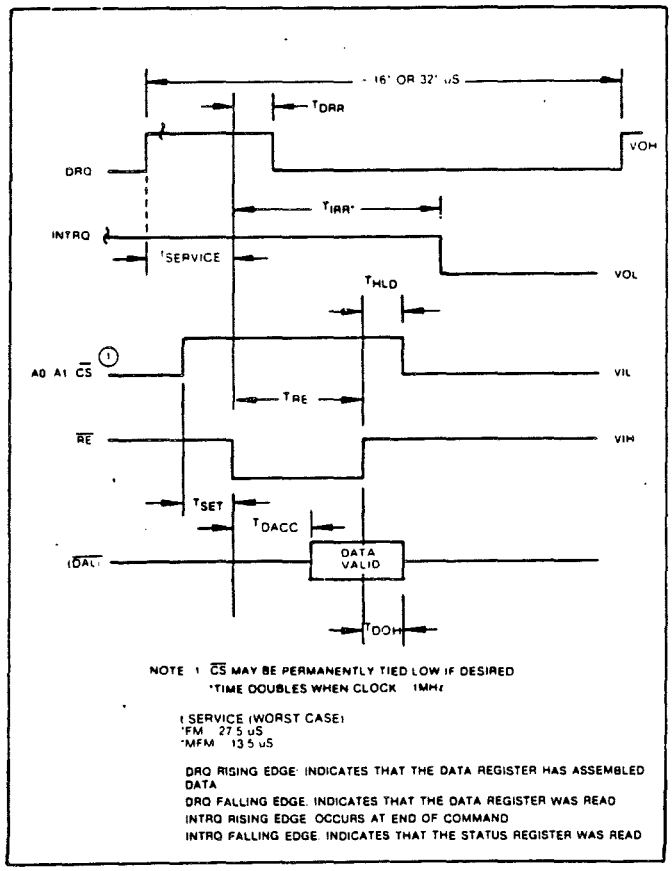
T_A = 0°C to 70°C, V_{SS} = 0V, V_{CC} = +5V ± .25V

READ ENABLE TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{SET}	Setup ADDR & CS to $\overline{RE}$	50			nsec	
T_{HLD}	Hold ADDR & CS from $\overline{RE}$	10			nsec	
T_{RE}	$\overline{RE}$ Pulse Width	200			nsec	C_L = 50 pf
T_{DRR}	DRQ Reset from $\overline{RE}$		100	200	nsec	
T_{IRR}	INTRQ Reset from $\overline{RE}$		500	3000	nsec	See Note
T_{DACC}	Data Valid from $\overline{RE}$		100	200	nsec	C_L = 50 pf
T_{DOH}	Data Hold From $\overline{RE}$	20		150	nsec	C_L = 50 pf

WRITE ENABLE TIMING

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{SET}	Setup ADDR & CS to $\overline{WE}$	50			nsec	
T_{HLD}	Hold ADDR & CS from $\overline{WE}$	10			nsec	
T_{WE}	$\overline{WE}$ Pulse Width	200			nsec	
T_{DRR}	DRQ Reset from $\overline{WE}$		100	200	nsec	
T_{IRR}	INTRQ Reset from $\overline{WE}$		500	3000	nsec	See Note
T_{DS}	Data Setup to $\overline{WE}$	150			nsec	
T_{DH}	Data Hold from $\overline{WE}$	50			nsec	

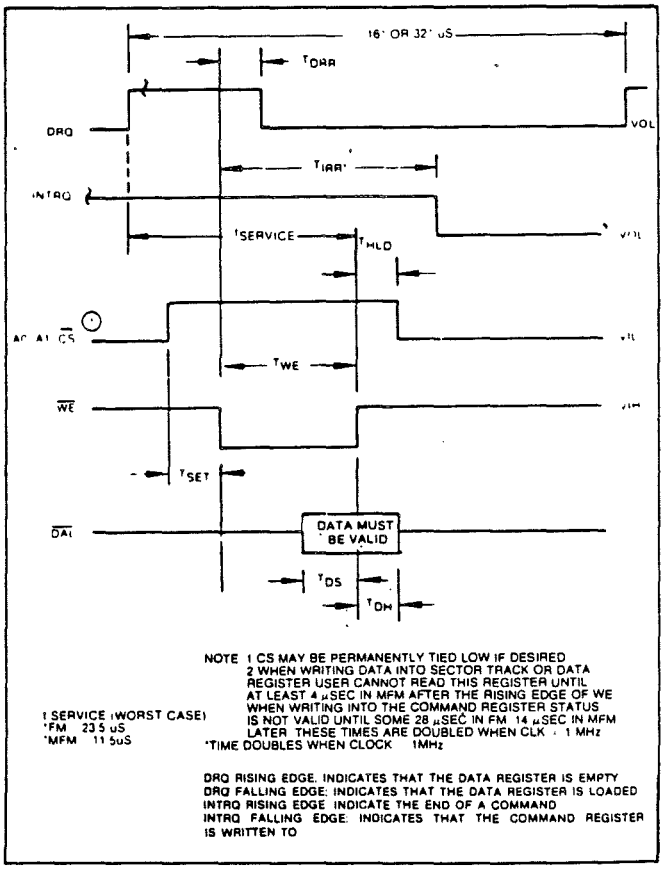


NOTE 1 $\overline{CS}$ MAY BE PERMANENTLY TIED LOW IF DESIRED
*TIME DOUBLES WHEN CLOCK 1MHz

t SERVICE (WORST CASE)
*FM 27.5 uS
*MFM 13.5 uS

DRQ RISING EDGE: INDICATES THAT THE DATA REGISTER HAS ASSEMBLED DATA
DRQ FALLING EDGE: INDICATES THAT THE DATA REGISTER WAS READ
INTRQ RISING EDGE: OCCURS AT END OF COMMAND
INTRQ FALLING EDGE: INDICATES THAT THE STATUS REGISTER WAS READ

READ ENABLE TIMING



NOTE 1 CS MAY BE PERMANENTLY TIED LOW IF DESIRED
2 WHEN WRITING DATA INTO SECTOR TRACK OR DATA REGISTER USER CANNOT READ THIS REGISTER UNTIL AT LEAST 4 μSEC IN MFM AFTER THE RISING EDGE OF WE WHEN WRITING INTO THE COMMAND REGISTER STATUS IS NOT VALID UNTIL SOME 28 μSEC IN FM 14 μSEC IN MFM LATER THESE TIMES ARE DOUBLED WHEN CLK = 1 MHz
*TIME DOUBLES WHEN CLOCK 1MHz

t SERVICE (WORST CASE)
*FM 23.5 uS
*MFM 11.5uS

DRQ RISING EDGE: INDICATES THAT THE DATA REGISTER IS EMPTY
DRQ FALLING EDGE: INDICATES THAT THE DATA REGISTER IS LOADED
INTRQ RISING EDGE: INDICATE THE END OF A COMMAND
INTRQ FALLING EDGE: INDICATES THAT THE COMMAND REGISTER IS WRITTEN TO

WRITE ENABLE TIMING

INPUT DATA TIMING

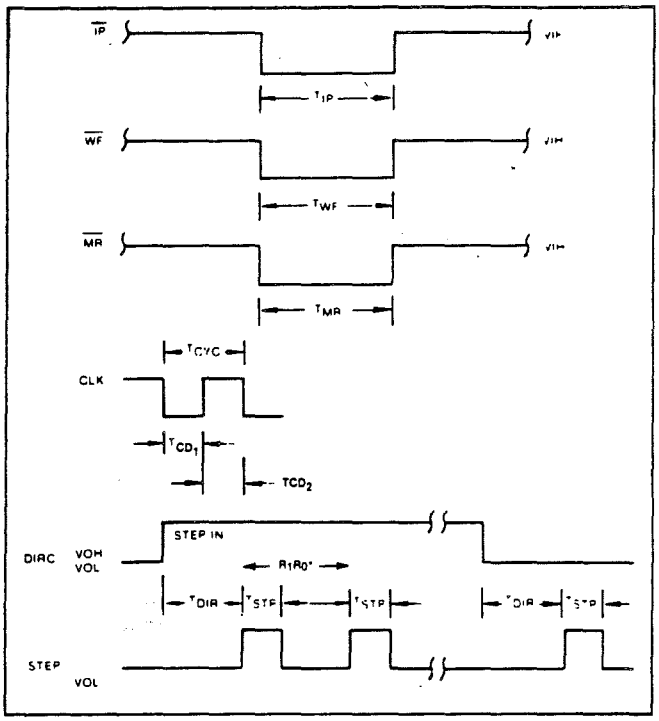
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{PW}	$\overline{\text{Raw Read}}$ Pulse Width	100	200		nsec	
T_{BC}	$\overline{\text{Raw Read}}$ Cycle Time	1500	2000		nsec	

WRITE DATA TIMING: (ALL TIMES DOUBLE WHEN CLK = 1 MHz) (NO WRITE PRECOMPENSATION)

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{WP}	Write Data Pulse Width	400	500	600	nsec	FM
		200	250	300	nsec	MFM
T_{WG}	Write Gate to Write Data		2		μsec	FM
			1		μsec	MFM
T_{WF}	Write Gate off from WD		2		μsec	FM
			1		μsec	MFM

MISCELLANEOUS TIMING:

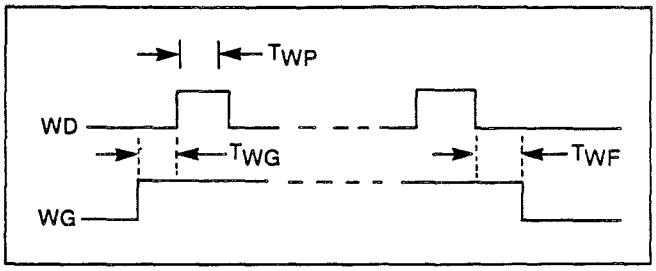
SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
T_{CD_1}	Clock Duty (low)	230	250	20000	nsec	
T_{CD_2}	Clock Duty (high)	230	250	20000	nsec	
T_{STP}	Step Pulse Output	2 or 4			μsec	See Note
T_{DIR}	Dir Setup to Step		12		μsec	± CLK ERROR
T_{MR}	Master Reset Pulse Width	50			μsec	
T_{IP}	Index Pulse Width	10			μsec	See Note
RPW	Read Window Pulse Width					Input 0-5V
		120		700	nsec	MFM
		240		1400	nsec	FM ± 15%
	Precomp Adjust.	100		300	nsec	MFM
WPW	Write Data Pulse Width					Precomp = 100 nsec
		200	300	400	nsec	MFM
WPW	Write Data Pulse Width					Precomp = 300 nsec
		600	900	1200	nsec	MFM
VCO	Free Run Voltage Controlled	6.0			MHz	Cext = 0
	Oscillator. Adjustable by ext. capacitor on Pin 26		4.0		MHz	Cext = 35 pf
	Pump Up + 25%	5.0			MHz	PU = 2.2V Cext = 35 pf
VCO	Pump Down − 25%			3.0	MHz	$\overline{PD}$ = 0.2V Cext = 35 pf
VCO	5% Change V_{CC}	3.8		4.2	MHz	Cext = 35 pf
	T_A = 75°C	3.5			MHz	Cext = 35 pf
Cext	Adjustable external capacitor	20	45	100	pf	VCO = 4.0MHz nom
RCLK	Derived read clock = VCO ÷ 8, 16, 32					VCO = 4.0MHz
			500		KHz	$\overline{DDEN}$ = 0 $\overline{5/8}$ = 1
			250		KHz	$\overline{DDEN}$ = 0 $\overline{5/8}$ = 0
			250		KHz	$\overline{DDEN}$ = 1 $\overline{5/8}$ = 1
			125		KHz	$\overline{DDEN}$ = 1 $\overline{5/8}$ = 0
PU/DON	PU/$\overline{PD}$ time on (pulse width)			250	ns	MFM
				500	ns	FM



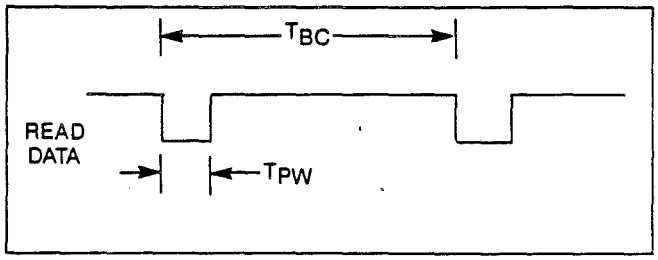
MISCELLANEOUS TIMING

NOTES:

1. Times double when clock = 1 MHz.
2. Output timing readings are at V_{OL} = 0.8v and V_{OH} = 2.0v.



WRITE DATA TIMING



READ DATA TIMING

* FROM STEP RATE TABLE

Table 4. STATUS REGISTER SUMMARY

BIT	ALL TYPE I COMMANDS	READ ADDRESS	READ SECTOR	READ TRACK	WRITE SECTOR	WRITE TRACK
S7	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY
S6	WRITE PROTECT	0	0	0	WRITE PROTECT	WRITE PROTECT
S5	HEAD LOADED	0	RECORD TYPE	0	0	0
S4	SEEK ERROR	RNF	RNF	0	RNF	0
S3	CRC ERROR	CRC ERROR	CRC ERROR	0	CRC ERROR	0
S2	TRACK 0	LOST DATA	LOST DATA	LOST DATA	LOST DATA	LOST DATA
S1	INDEX PULSE	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

STATUS FOR TYPE I COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset it indicates that the drive is ready. This bit is an inverted copy of the Ready input and logically 'ored' with MR.
S6 PROTECTED	When set, indicates Write Protect is activated. This bit is an inverted copy of $\overline{WRPT}$ input.
S5 HEAD LOADED	When set, it indicates the head is loaded and engaged. This bit is a logical "and" of HLD and HLT signals.
S4 SEEK ERROR	When set, the desired track was not verified. This bit is reset to 0 when updated.
S3 CRC ERROR	CRC encountered in ID field.
S2 TRACK 00	When set, indicates Read/Write head is positioned to Track 0. This bit is an inverted copy of the $\overline{TROO}$ input.
S1 INDEX	When set, indicates index mark detected from drive. This bit is an inverted copy of the $\overline{IP}$ input.
S0 BUSY	When set command is in progress. When reset no command is in progress.

STATUS FOR TYPE II AND III COMMANDS

BIT NAME	MEANING
S7 NOT READY	This bit when set indicates the drive is not ready. When reset, it indicates that the drive is ready. This bit is an inverted copy of the Ready input and 'ored' with MR. The Type II and III Commands will not execute unless the drive is ready.
S6 WRITE PROTECT	On Read Record: Not Used. On Read Track: Not Used. On any Write: It indicates a Write Protect. This bit is reset when updated.
S5 RECORD TYPE	On Read Record: It indicates the record-type code from data field address mark. 1 = Deleted Data Mark. 0 = Data Mark. On any Write: Forced to a Zero.
S4 RECORD NOT FOUND (RNF)	When set, it indicates that the desired track, sector, or side were not found. This bit is reset when updated.
S3 CRC ERROR	If S4 is set, an error is found in one or more ID fields; otherwise it indicates error in data field. This bit is reset when updated.
S2 LOST DATA	When set, it indicates the computer did not respond to DRQ in one byte time. This bit is reset to zero when updated.
S1 DATA REQUEST	This bit is a copy of the DRQ output. When set, it indicates the DR is full on a Read Operation or the DR is empty on a Write operation. This bit is reset to zero when updated.
S0 BUSY	When set, command is under execution. When reset, no command is under execution.

SUMMARY OF ADJUSTMENT PROCEDURE

WRITE PRECOMPENSATION

1) Set $\overline{\text{TEST}}$ (Pin 22) to a logic high.
2) Strobe $\overline{\text{MR}}$ (Pin 19).
3) Set $\overline{\text{TEST}}$ (Pin 22) to a logic low.
4) Observe pulse width on WD (Pin 31).
5) Adjust WPW (Pin 33) for desired pulse width (Precomp Value).
6) Set $\overline{\text{TEST}}$ (Pin 22) to a logic high.

DATA SEPARATOR

1) Set $\overline{\text{TEST}}$ (Pin 22) to a logic high.
2) Strobe $\overline{\text{MR}}$ (Pin 19). Insure that $\overline{5}$/8, and $\overline{\text{DDEN}}$ are set properly.
3) Set $\overline{\text{TEST}}$ (Pin 22) to a logic low.
4 Observe Pulse Width on TG43 (Pin 29).
5) Adjust RPW (Pin 18) for 1/8 of the read clock (250ns for 8" DD, 500ns for 5¼" DD, etc.).
6) Observe Frequency on DIRC (Pin 16).
7) Adjust variable capacitor on VCO pin for Data Rate (500 KHz for 8" DD, 250 KHz for 5¼" DD, etc.).
8) Set $\overline{\text{TEST}}$ (Pin 22) to a logic high.

NOTE: To maintain internal VCO operation, insure that $\overline{\text{TEST}}$ = 1 whenever a master reset pulse is applied.

WD279X-02 Floppy Disk Formatter/Controller Family Application Notes

INTRODUCTION

In an effort to simplify Floppy Diskette interfacing, Western Digital has been constantly improving the LSI Controller/Formatter, the most recent of which is the WD279X Family of LSI controller devices, incorporating advanced technology to include controller, Write Precompensation and Analog Phase Lock Loop in a single 40 pin dual-in-line package. With this package we can now offer the designer the simplest ever interfacing option.

The family consists of four members, WD2791, WD2793, WD2795 and WD2797. WD2791 and WD2793 offer internal clock divide in true and inverted data bus. The WD2795 and WD2797 offer internal side select. The family supports both 5¼" and 8" Diskette Drives and both single and double density.

HOST INTERFACING

The LSI Diskette Controller has been developed to ease the interfacing of Processor to Disk Device. The Host interfacing with WD279X Family is accomplished with minimum external devices via an 8 bit Bi-directional bus, read/write controls, register select lines and optional control line for chip select, 5¼" or 8" select, enable mini floppy, double density enable. The basic operation at the controller is accomplished by selecting the device via (CS) chip select line, enabling selection of one of the five internal registers (Figure 1).

A1	A0	READ ($\overline{RE}$)	WRITE ($\overline{WE}$)
0	0	Status Register	Command Register
0	1	Track Register	Track Register
1	0	Sector Register	Sector Register
1	1	Data Register	Data Register

Figure 1.

Each time a command is issued to the WD279X, the busy bit is set and INTRQ (Interrupt Request) line is reset. The user has the option of testing for the busy bit or polling INTRQ to determine if command has been completed.

The busy bit will be reset whenever the WD279X is idle and awaiting a new command. The INTRQ line once set, can only be reset by reading of the status register or issuing a new command.

The A_0, A_1 Lines used for register selections can be configured at the CPU in a variety of ways. These lines may actually tie to CPU addressed like RAM. They may also be used under Program Control by tying to a port device such as the 8255, 6250, etc. As a diagnostic tool when checking out the CPU interface, the Track and Sector registers should respond like "RAM" when the WD279X is idle (Busy = INTRQ = 0).

Because of internal synchronization cycles, certain time delays must be introduced when operating under Programmed I/O. The worst case delays are:

OPERATION	NEXT OPERATION	DELAY REQ'D. FM	DELAY REQ'D. MFM
Write to Command Reg.	Read Busy Bit (Status Bit 0)	12μs	6μs
Write to Command Reg.	Read Status Bits 1-7	28μs	14μs
Write Any Register	Read From Diff. Register	0	0

Other CPU interface lines are CLK, $\overline{MR}$ and $\overline{DDEN}$. The CLK line should be 2 MHz (8" drive) or 1 MHz (5¼" drive) with 50% duty cycle. Accuracy should be +1% (crystal source) since all internal timing, including stepping rates, are based upon this clock, or a single 2 MHz CLK on WD2791 and WD2793 since $\overline{ENMF}$ line will internally divide CLK.

The MR or Master Reset Line should be strobed a minimum of 50 microseconds upon each power-on condition. This line clears and initalizes all internal registers and issues a restore command (Hex '03') on the rising edge. A quicker stepping rate can be written to the command register after a $\overline{MR}$, in which case the remaining steps will occur at the faster programmed rate. The WD179X will issue a maximum of 255 stepping pulses in an attempt to expect the $\overline{TR00}$ line to go active low. This line should be connected to the drive's $\overline{TR00}$ sensor.

The DDEN line causes selection of either single density (DDEN = 1) or double density operation. DDEN should not be switched during a read or write operation.

The 5/8 Line selects interal VCO frequency to be used with 5¼" or 8" drives.

FLOPPY DISK INTERFACE

The Floppy Disk Interface can be divided into three sections: Motor Control, Write Signals and Read Signals. All of these lines are capable of driving one TTL load and not compatible for direct connection to the

drive. Most drives require an open-collector TTL interface with high current drive capability. This must be done on all outputs from the WD279X. Inputs to the WD279X may be buffered or tied to the Drives outputs, providing the appropriate resistor termination networks are used. Undershoot should not exceed −0.3 volts, while integrity of V_{IH} and V_{OH} levels should be kept within spec.

MOTOR CONTROL

Motor Control is accomplished by the STEP and DIRC Lines. The STEP Line issues stepping pulses with period defined by the rate field in all Type I commands. The DIRC Line defines the direction of steps (DIRC = 1 STEP IN/DIRC = 0 STEP OUT).

Other Control Lines include the $\overline{IP}$ or Index Pulse. This Line is tied to the drives' Index L.E.D. sensor that informs the WD279X that the stepper motor is at its furthest position, over Track 00. The READY Line can be used for a number of functions, such as sensing "door open," Drive motor on, etc. Most drives provide a programmable READY Signal selected by option jumpers on the drive. The WD279X will look at the ready signal prior to executing READ/WRITE commands. READY is not inspected during any Type 1 commands. All type 1 commands will execute regardless of the Logic Level on this Line.

GENERAL DISK WRITE OPERATION

When writing is to take place on the diskette the Write Gate (WG) output is activated, allowing current to flow into the Read/Write head. As a precaution to erroneous writing the first data byte must be loaded into the Data Register in response to a Data Request from the WD279X before the Write Gate signal can be activated.

Writing is inhibited when the $\overline{\text{Write Protect}}$ input is a logic low, in which case any Write command is immediately terminated, an interrupt is generated and the Write Protect status bit is set.

RESTORE (SEEK TRACK 0)

Upon receipt of this command the Track 00 ($\overline{TR00}$) input is sampled. If TR00 is active low indicating the Read-Write head is positioned over track 0, the Track Register is loaded with zeroes and an interrupt is generated. If $\overline{TR00}$ is not active low, stepping pulses (pins 15 to 16) at a rate specified by the r_1r_0 field are issued until the $\overline{TR00}$ input is activated. At this time the Track Register is loaded with zeroes and an interrupt is generated. If the TR00 input does not go active low after 255 stepping pulses, the WD279X terminated operations, interrupts, and sets the Seek error status bit. A verification operation takes place if the V flag is set. The h bit allows the head to be loaded at the start of command. Note that the Restore command is executed when MR goes from an active to an inactive state.

SEEK

This command assumes that the Track Register contains the track number of the current position of the Read-Write head and the Data Register contains the desired track number. The WD279X will update the Track register and issue stepping pulses in the appropriate direction until the contents of the Track register are equal to the contents of the Data Register (the desired track location). A verification operation takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command. Note: When using multiple drives, the track register must be updated for the drive selected before seeks are issued.

STEP

Upon receipt of this command, the WD279X issues one stepping pulse to the disk drive. The stepping motor direction is the same as in the previous step command. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. If the T flag is on, the Track Register is updated. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

STEP-IN

Upon receipt of this command, the WD279X issues one stepping pulse in the direction towards track 0. If the T flag is on, the Track Register is decremented by one. After a delay determined by the r_1r_0 field, a verification takes place if the V flag is on. The h bit allows the head to be loaded at the start of the command. An interrupt is generated at the completion of the command.

EXCEPTIONS

On the WD2795/7 devices, the SSO output is not affected during Type I commands, and an internal side compare does not take place when the (V) Verify Flag is on.

For write operations, the WD279X provides Write Gate (Pin 30) and Write Data (Pin 31) outputs. Write data consists of a series of pulses set to a width approximately three times greater than the precomp adjustment. Write Data provides the unique address marks in both formats.

READY

Whenever a Read or Write command (Type II or III) is received the WD279X samples the Ready input. If this input is logic low the command is not executed and an interrupt is generated. All Type I commands are performed regardless of the state of the Ready input. Also, whenever a Type II or III command is received, the TG43 signal output is updated. TG43 may be tied to ENP to enable write precompensation on tracks 44-76.

COMMAND DESCRIPTION

The WD279X will accept eleven commands. Command words should only be loaded in the Command Register when the Busy status bit is off (Status bit 0). The one exception is the Force interrupt command. Whenever a command is being executed, the Busy status bit is set. When a command is completed, and interrupt is generated and the Busy status bit is reset. The Status Register indicates whether the completed command encountered an error or was fault free. For ease of discussion, commands are divided into four types. Command types are summarized in Table 1 and Table 2.

Table 1. COMMAND SUMMARY

A. Commands for Models: 2791, 2793

Type	Command	Bits 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	T	h	V	r_1	r_0
I	Step-in	0	1	0	T	h	V	r_1	r_0
I	Step-out	0	1	1	T	h	V	r_1	r_0
II	Read Sector	1	0	0	m	S	E	C	0
II	Write Sector	1	0	1	m	S	E	C	a_0
III	Read Address	1	1	0	0	0	E	0	0
III	Read Track	1	1	1	0	0	E	0	0
III	Write Track	1	1	1	1	0	E	0	0
IV	Force Interrupt	1	1	0	I_1	I_3	I_2	I_1	I_0

B. Commands for Models: 2795, 2797

Type	Command	Bits 7	6	5	4	3	2	1	0
I	Restore	0	0	0	0	h	V	r_1	r_0
I	Seek	0	0	0	1	h	V	r_1	r_0
I	Step	0	0	1	T	h	V	r_1	r_0
I	Step-in	0	1	0	T	h	V	r_1	r_0
I	Step-out	0	1	1	T	h	V	r_1	r_0
II	Read Sector	1	0	0	m	L	E	U	0
II	Write Sector	1	0	1	m	L	E	U	a_0
III	Read Address	1	1	0	0	0	E	U	0
III	Read Track	1	1	1	0	0	E	U	0
III	Write Track	1	1	1	1	0	E	U	0
IV	Force Interrupt	1	1	0	1	I_3	I_2	I_1	I_0

Table 2. FLAG SUMMARY

Command Type	Bit No(s)		Description
I	0, 1	r_1 r_0 = Stepping Motor Rate See Table 3 for Rate Summary	
I	2	V = Track Number Verify Flag	V = 0, No verify V = 1, Verify on destination track
I	3	h = Head Load Flag	h = 0, Load head at beginning h = 2, Unload head at beginning
I	4	T = Track Update Flag	T = 0, No update T = 1, Update track register
II	0	a_0 = Data Address Mark	a_0 = 0, FB (DAM) a_0 = 1, F8 (deleted DAM)
II & III	1	C = Side Compare Flag	C = 0, Disable side compare C = 1, Enable side compare
II & III	1	U = Update SSO	U = 0, Update SSO to 0 U = 1, Update SSO to 1
II & III	2	E = 15 MS Delay	E = 0, No 15 MS delay E = 1, 15 MS delay (30 MS for 1 MHz)
II	3	S = Side Compare Flag	S = 0, Compare for side 0 S = 1, Compare for side 1
II	3	L = Sector Length Flag	LSB's Sector Length in ID Field 00: L = 0: 256; L = 1: 128 01: L = 0: 512; L = 1: 256 10: L = 0: 1024; L = 1: 512 11: L = 0: 128; L = 1: 1024
II	4	m = Multiple Record Flag	m = 0, Single record m = 1, Multiple records
IV	0-3	I_x = Interrupt Condition Flags I_0 = 1 Not Ready To Ready Transition I_1 = 1 Ready To Not Ready Transition I_2 = 1 Index Pulse I_3 = 1 Immediate Interrupt, Requires A Reset* I_3-I_0 = 0 Terminate With No Interrupt (INTRQ)	

*NOTE: See Type IV Command Description for further information.

Printed in U.S.A.

8259A/8259A-2/8259A-8
PROGRAMMABLE INTERRUPT CONTROLLER

- iAPX 86, iAPX 88 Compatible
- MCS-80®, MCS-85® Compatible
- Eight-Level Priority Controller
- Expandable to 64 Levels
- Programmable Interrupt Modes
- Individual Request Mask Capability
- Single +5V Supply (No Clocks)
- 28-Pin Dual-In-Line Package
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range

The Intel® 8259A Programmable Interrupt Controller handles up to eight vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts without additional circuitry. It is packaged in a 28-pin DIP, uses NMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259A is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

The 8259A is fully upward compatible with the Intel® 8259. Software originally written for the 8259 will operate the 8259A in all 8259 equivalent modes (MCS-80/85, Non-Buffered, Edge Triggered).

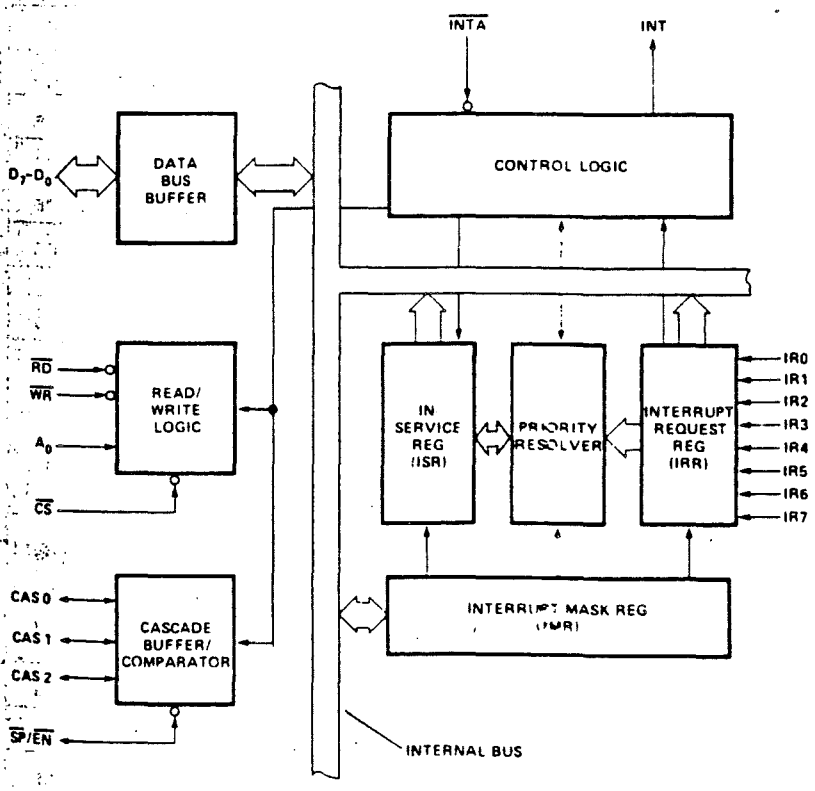


Figure 1. Block Diagram

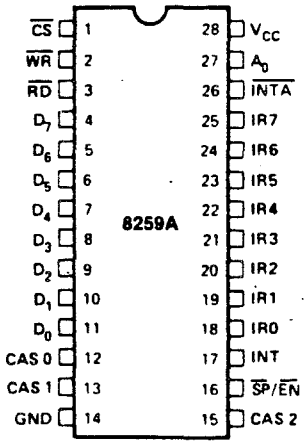


Figure 2. Pin Configuration

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}	28	I	**Supply:** +5V Supply.
GND	14	I	**Ground.**
$\overline{CS}$	1	I	**Chip Select:** A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 8259A. INTA functions are independent of CS.
$\overline{WR}$	2	I	**Write:** A low on this pin when CS is low enables the 8259A to accept command words from the CPU.
$\overline{RD}$	3	I	**Read:** A low on this pin when CS is low enables the 8259A to release status onto the data bus for the CPU.
D_7–D_0	4–11	I/O	**Bidirectional Data Bus:** Control, status and interrupt-vector information is transferred via this bus.
CAS_0–CAS_2	12, 13, 15	I/O	**Cascade Lines:** The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A and inputs for a slave 8259A.
$\overline{SP}/\overline{EN}$	16	I/O	**Slave Program/Enable Buffer:** This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	**Interrupt:** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR_0–IR_7	18–25	I	**Interrupt Requests:** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
$\overline{INTA}$	26	I	**Interrupt Acknowledge:** This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	27	I	**A0 Address Line:** This pin acts in conjunction with the $\overline{CS}$, $\overline{WR}$, and $\overline{RD}$ pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for iAPX 86, 88).

FUNCTIONAL DESCRIPTION

Interrupts in Microcomputer Systems

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the *Polled* approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system throughput, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called *Interrupt*. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

The 8259A

The 8259A is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the 8259A can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

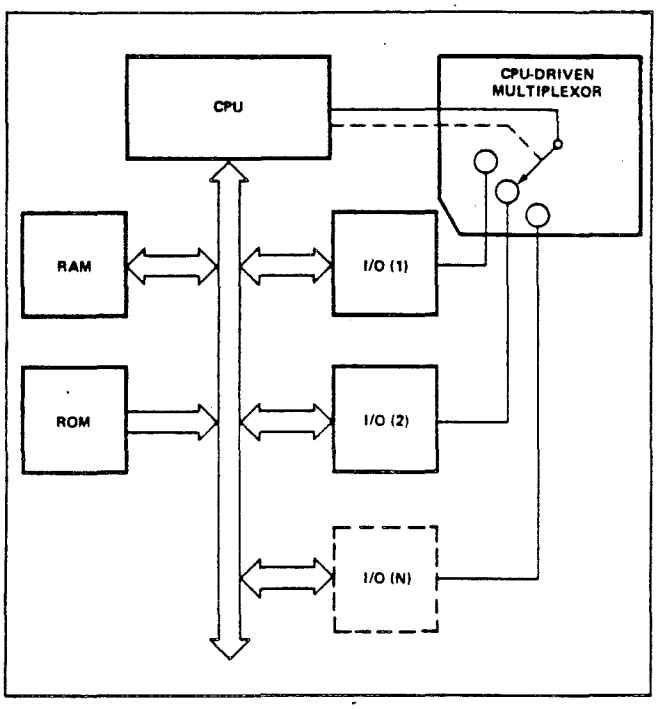


Figure 3a. Polled Method

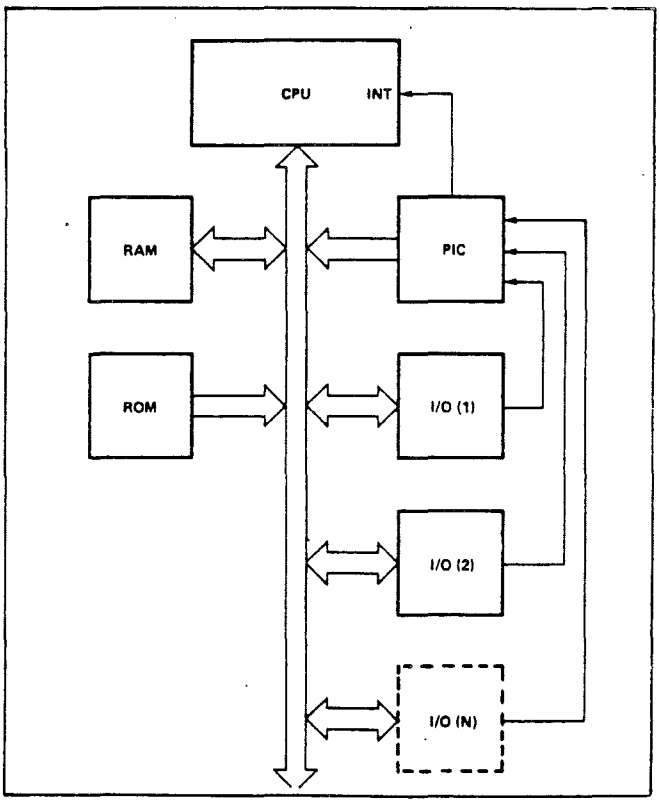


Figure 3b. Interrupt Method

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INTERRUPT MASK REGISTER (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

INT (INTERRUPT)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

$\overline{INTA}$ (INTERRUPT ACKNOWLEDGE)

$\overline{INTA}$ pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (μPM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

$\overline{CS}$ (CHIP SELECT)

A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

$\overline{RD}$ (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.

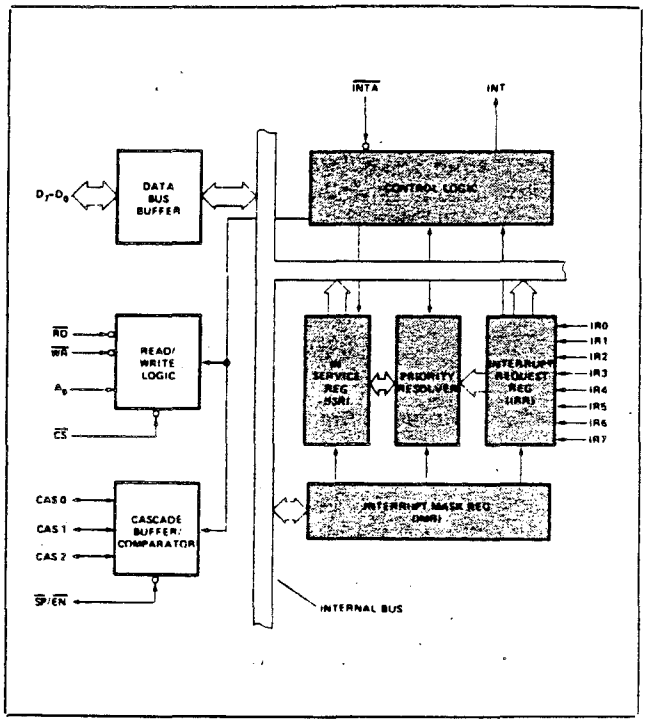


Figure 4a. 8259A Block Diagram

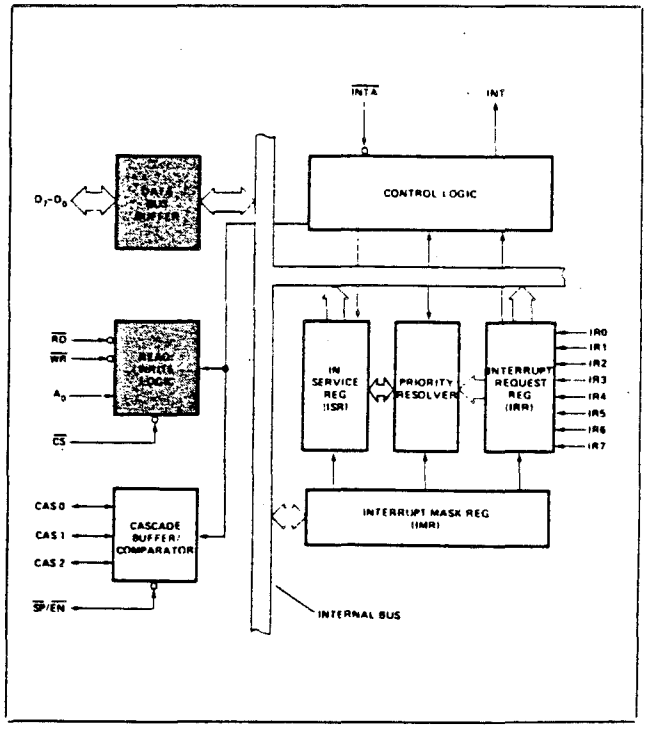


Figure 4b. 8259A Block Diagram

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

THE CASCADE BUFFER/COMPARATOR

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 8259A".)

INTERRUPT SEQUENCE

The powerful features of the 8259A in a microcomputer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specific interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7-0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 8259A from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOI mode the ISR bit is reset at the end of the third $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occurring in an iAPX 86 system are the same until step 4.

4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The iAPX 86/10 will initiate a second $\overline{INTA}$ pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second $\overline{INTA}$ pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt request is present at step 4 of either sequence (i.e., the request was too short in duration) the 8259A will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

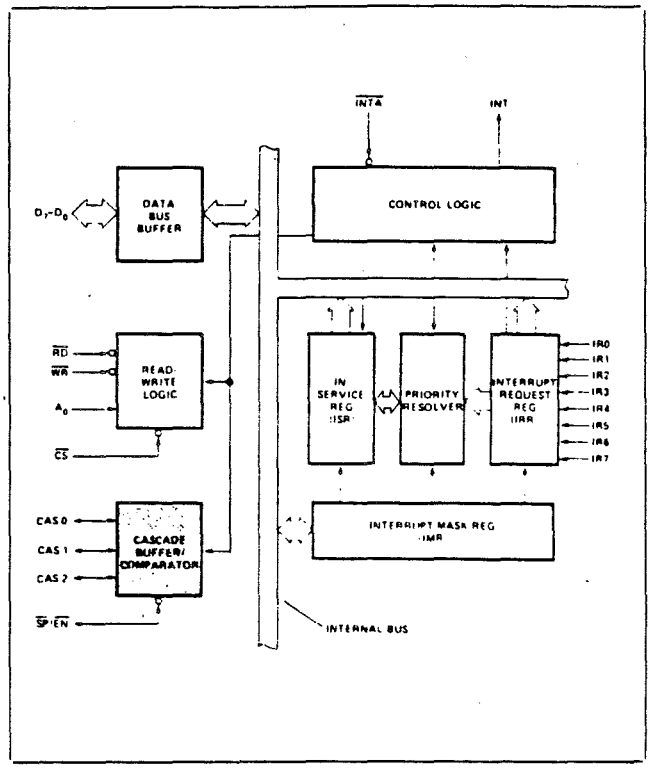


Figure 4c. 8259A Block Diagram

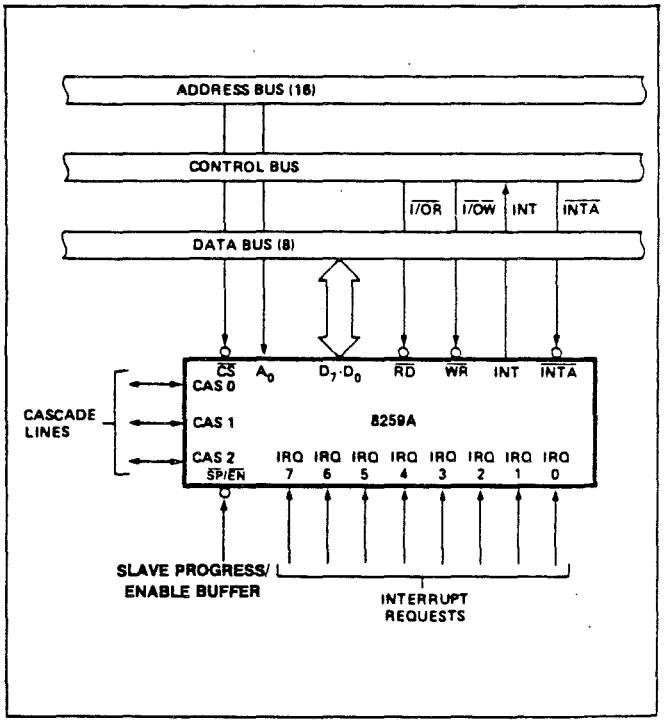


Figure 5. 8259A Interface to Standard System Bus

INTERRUPT SEQUENCE OUTPUTS

MCS-80®, MCS-85®

This sequence is timed by three $\overline{INTA}$ pulses. During the first $\overline{INTA}$ pulse the CALL opcode is enabled onto the data bus.

Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second $\overline{INTA}$ pulse the lower address of the appropriate service routine is enabled onto the data bus. When Interval = 4 bits A_5-A_7 are programmed, while A_0-A_4 are automatically inserted by the 8259A. When Interval = 8 only A_6 and A_7 are programmed, while A_0-A_5 are automatically inserted.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third INTA pulse the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence (A_8 – A_{15}), is enabled onto the bus.

Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

iAPX 86, iAPX 88

iAPX 86 mode is similar to MCS-80 mode except that only two Interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80, 85 systems in that the 8259A uses it to internally freeze the state of the interrupts for priority resolution and as a master it issues the interrupt code on the cascade lines at the end of the INTA pulse. On this first cycle it does not issue any data to the processor and leaves its data bus buffers disabled. On the second interrupt acknowledge cycle in iAPX 86 mode the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and A_5–A_{11} are unused in iAPX 86 mode):

Content of Interrupt Vector Byte for iAPX 86 System Mode

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

PROGRAMMING THE 8259A

The 8259A accepts two types of command words generated by the CPU:

1. *Initialization Command Words (ICWs):* Before normal operation can begin, each 8259A in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by $\overline{WR}$ pulses.
2. *Operation Command Words (OCWs):* These are the command words which command the 8259A to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 8259A anytime after initialization.

INITIALIZATION COMMAND WORDS (ICWS)

GENERAL

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transition to generate an interrupt.
b. The Interrupt Mask Register is cleared.
c. IR7 input is assigned priority 7.
d. The slave mode address is set to 7.
e. Special Mask Mode is cleared and Status Read is set to IRR.
f. If IC4=0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no Auto-EOI, MCS-80, 85 system).

*Note: Master/Slave in ICW4 is only used in the buffered mode.

INITIALIZATION COMMAND WORDS 1 AND 2 (ICW1, ICW2)

A_5-A_{15}: *Page starting address of service routines.* In an MCS 80/85 system, the 8 request levels will generate CALLs to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long (A_0-A_{15}). When the routine interval is 4, A_0-A_4 are automatically inserted by the 8259A, while A_5-A_{15} are programmed externally. When the routine interval is 8, A_0-A_5 are automatically inserted by the 8259A, while A_6-A_{15} are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

In an iAPX 86 system A_{15}-A_{11} are inserted in the five most significant bits of the vectoring byte and the 8259A sets the three least significant bits according to the interrupt level. A_{10}-A_5 are ignored and ADI (Address interval) has no effect.

LTIM: If LTIM = 1, then the 8259A will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

ADI: CALL address interval. ADI = 1 then interval = 4; ADI = 0 then interval = 8.

SNGL: Single. Means that this is the only 8259A in the system. If SNGL = 1 no ICW3 will be issued.

IC4: If this bit is set — ICW4 has to be read. If ICW4 is not needed, set IC4 = 0.

INITIALIZATION COMMAND WORD 3 (ICW3)

This word is read only when there is more than one 8259A in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4) a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS-80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for iAPX 86 only byte 2) through the cascade lines.

b. In the slave mode (either when $\overline{SP}$ = 0, or if BUF = 1 and M/S = 0 in ICW4) bits 2-0 identify the slave. The slave compares its cascade input with these bits and, if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for iAPX 86 are released by it on the Data Bus.

INITIALIZATION COMMAND WORD 4 (ICW4)

SFNM: If SFNM = 1 the special fully nested mode is programmed.

BUF: If BUF = 1 the buffered mode is programmed. In buffered mode $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.

M/S: If buffered mode is selected: M/S = 1 means the 8259A is programmed to be a master, M/S = 0 means the 8259A is programmed to be a slave. If BUF = 0, M/S has no function.

AEOI: If AEOI = 1 the automatic end of interrupt mode is programmed.

μPM: Microprocessor mode: μPM = 0 sets the 8259A for MCS-80, 85 system operation, μPM = 1 sets the 8259A for iAPX 86 system operation.

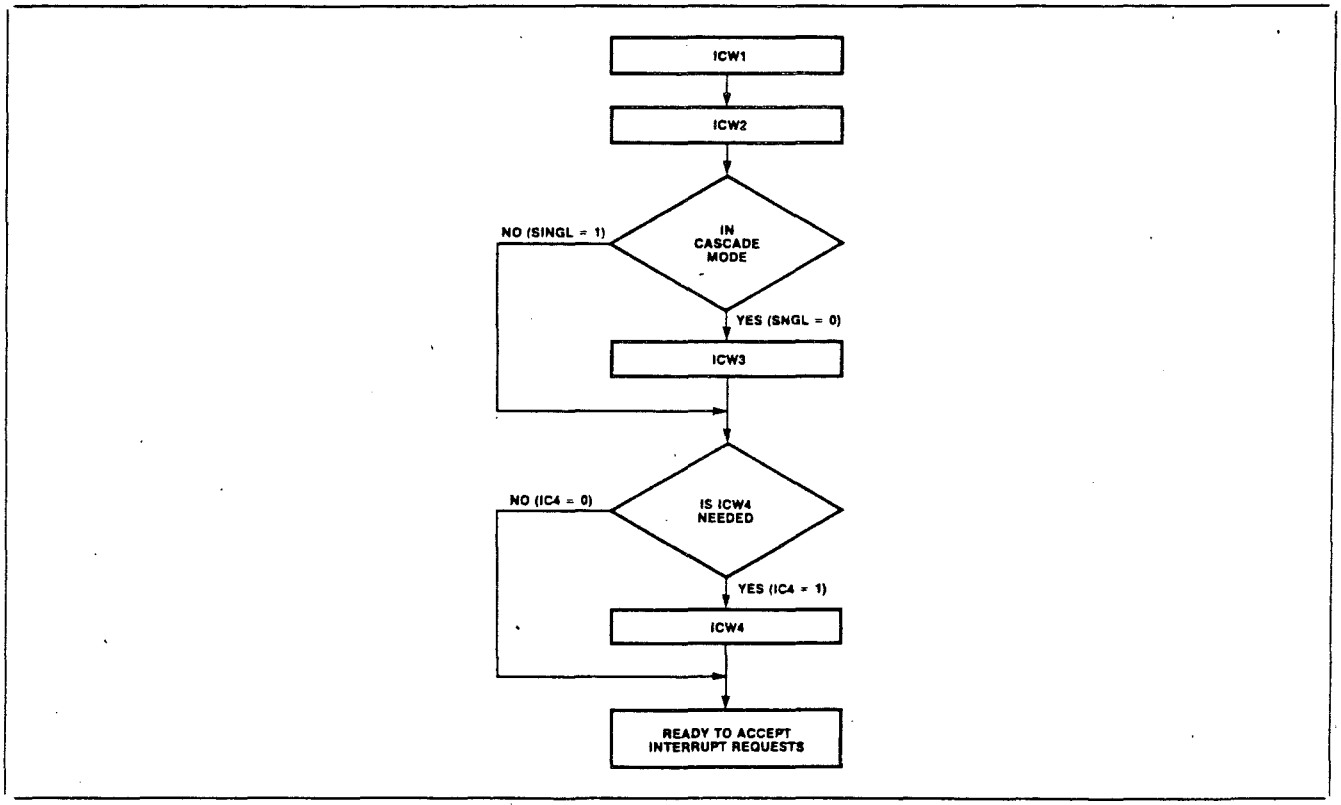
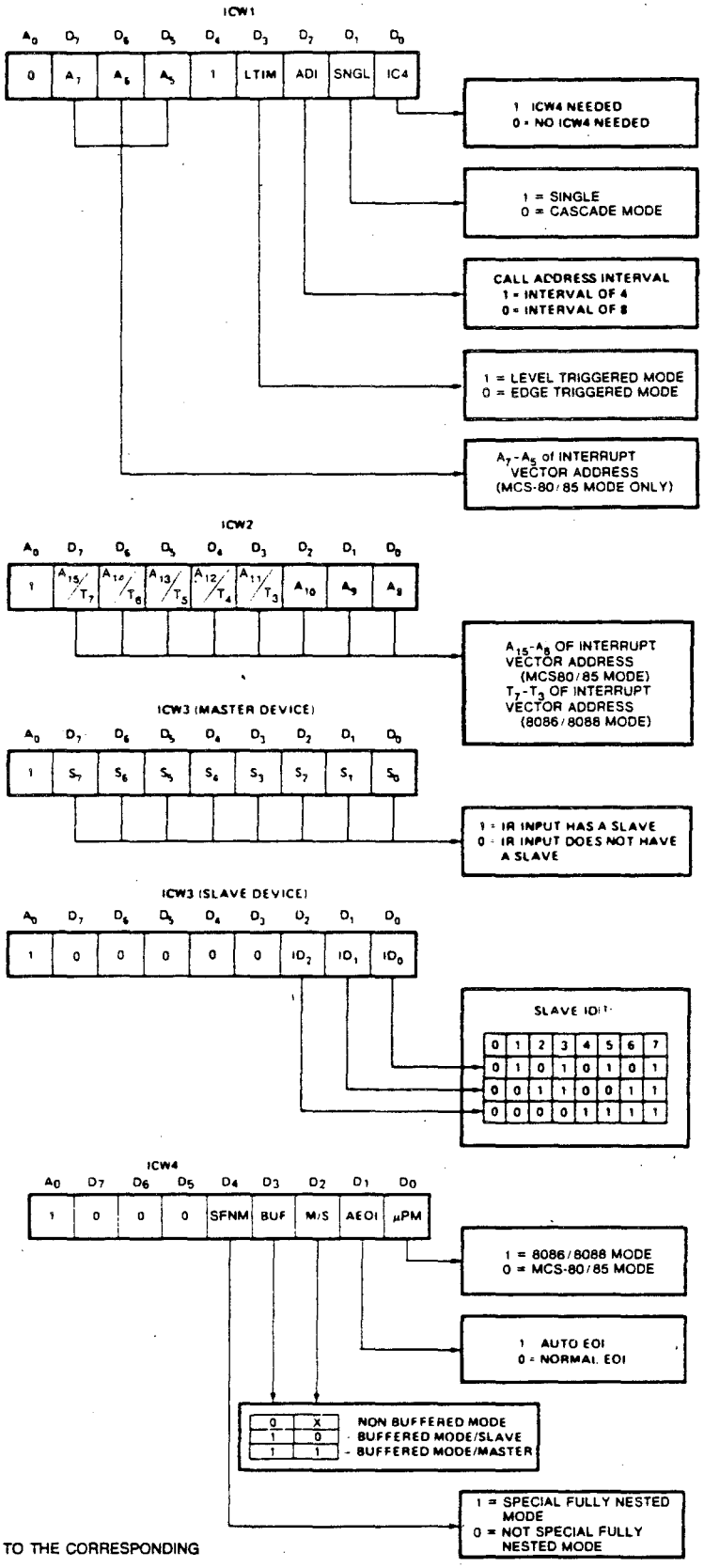


Figure 6. Initialization Sequence



NOTE 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.

Figure 7. Initialization Command Word Format

OPERATION COMMAND WORDS (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines. However, during the 8259A operation, a selection of algorithms can command the 8259A to operate in various modes through the Operation Command Words (OCWs).

OPERATION CONTROL WORDS (OCWs)

OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

OCW2

0	R	SL	EOI	0	0	L2	L1	L0

OCW3

0	0	ESMM	SMM	0	1	P	RR	RIS

OPERATION CONTROL WORD 1 (OCW1)

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). $M_7 - M_0$ represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

OPERATION CONTROL WORD 2 (OCW2)

R, SL, EOI — These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L_2, L_1, L_0—These bits determine the interrupt level acted upon when the SL bit is active.

OPERATION CONTROL WORD 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0 the SMM bit becomes a "don't care".

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1 the 8259A will enter Special Mask Mode. If ESMM = 1 and SMM = 0 the 8259A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

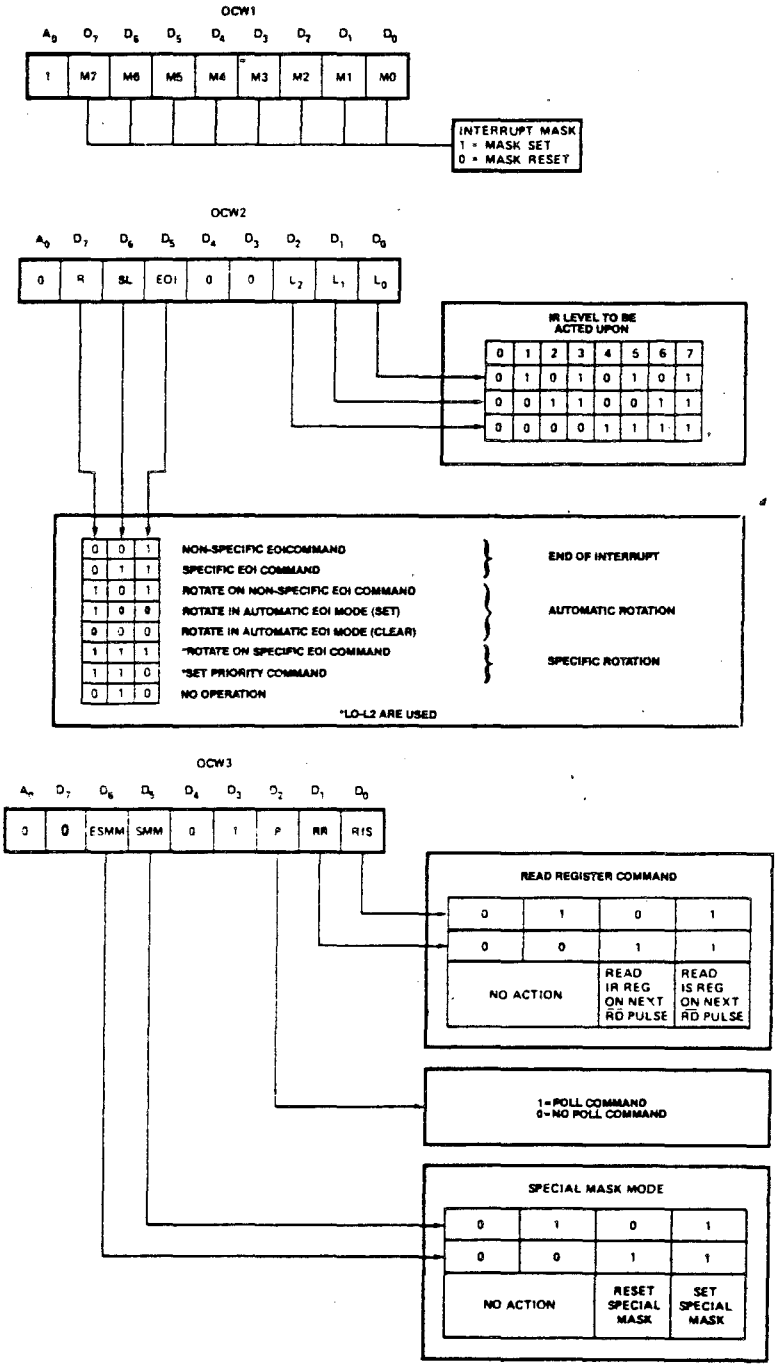


Figure 8. Operation Command Word Format

FULLY NESTED MODE

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority form 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (ISO-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal Interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

END OF INTERRUPT (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence $\overline{\text{INTA}}$ pulse (when AEOI bit in ICW1 is set) or by a command word that must be issued to the 8259A before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the 8259A is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the 8259A will automatically reset the highest IS bit of those that are set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the 8259A may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and LO-L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the 8259A is in the Special Mask Mode.

AUTOMATIC END OF INTERRUPT (AEOI) MODE

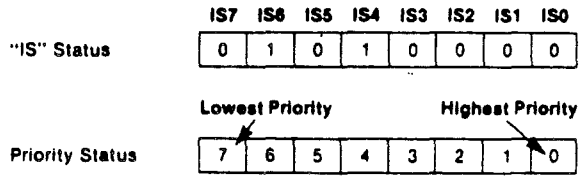
If AEOI = 1 in ICW4, then the 8259A will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the 8259A will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in iAPX 86). Note that from a system standpoint, this mode should be used only when a nested multilevel interrupt structure is not required within a single 8259A.

The AEOI mode can only be used in a master 8259A and not a slave.

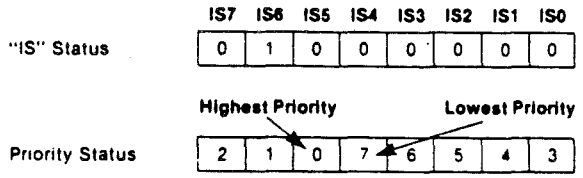
AUTOMATIC ROTATION (Equal Priority Devices)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most *once*. For example, if the priority and "in service" status is:

Before Rotate (IR4 the highest priority requiring service)



After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R = 1, SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

SPECIFIC ROTATION (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1; LO–L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and LO–L2 = IR level to receive bottom priority).

INTERRUPT MASKS

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

SPECIAL MASK MODE

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the 8259A would have inhibited all lower priority requests with no easy way for the routine to enable them

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level *and enables* interrupts from *all other* levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectively enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

POLL COMMAND

In this mode the INT output is not used or the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The 8259A treats the next $\overline{RD}$ pulse to the 8259A (i.e., $\overline{RD} = 0$, $\overline{CS} = 0$) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from $\overline{WR}$ to $\overline{RD}$.

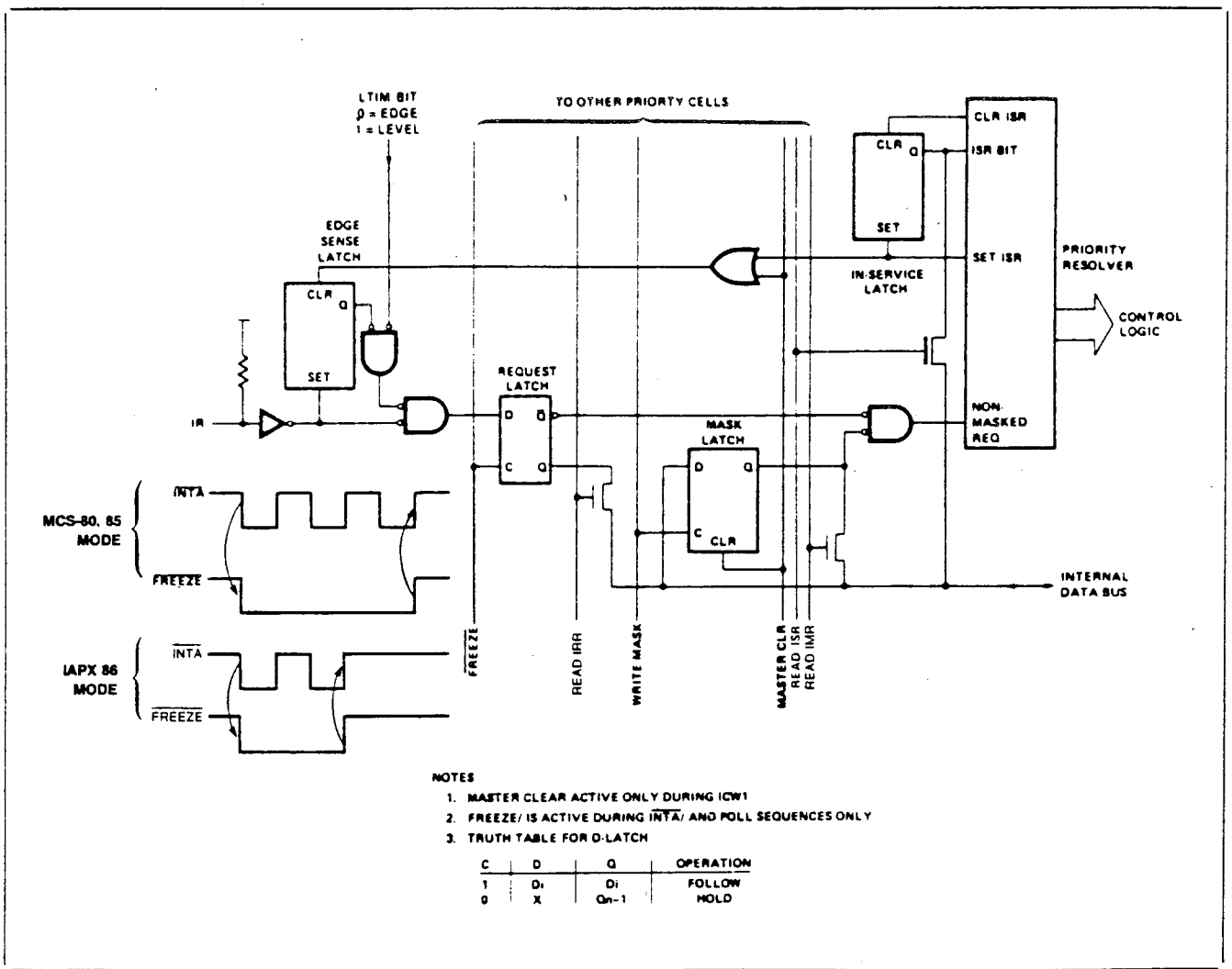
The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	—	—	—	—	W2	W1	W0

W0-W2: Binary code of the highest priority level requesting service.

I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the $\overline{INTA}$ sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.



NOTES

1. MASTER CLEAR ACTIVE ONLY DURING ICW1
2. FREEZE/ IS ACTIVE DURING $\overline{INTA}$/ AND POLL SEQUENCES ONLY
3. TRUTH TABLE FOR D-LATCH

C	D	Q	OPERATION
1	D_i	D_i	FOLLOW
0	X	Q_{n-1}	HOLD

Figure 9. Priority Cell—Simplified Logic Diagram

READING THE 8259A STATUS

The input status of several internal registers can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 [IMR]).

Interrupt Request Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR.)

In-Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 0.)

The ISR can be read when, prior to the RD pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the 8259A "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used.

After initialization the 8259A is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever $\overline{RD}$ is active and AO = 1 (OCW1).

Polling overrides status read when P = 1, RR = 1 in OCW3.

EDGE AND LEVEL TRIGGERED MODES

This mode is programmed using bit 3 in ICW1.

If LTIM = '0', an interrupt request will be recognized by a low to high transition on an IR input. The IR input can remain high without generating another interrupt.

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU interrupt is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the 8259A. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes the IR inputs must remain high until after the falling edge of the first INTA. If the IR input goes low before this time a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature the IR7 routine is used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit, a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs it is a default.

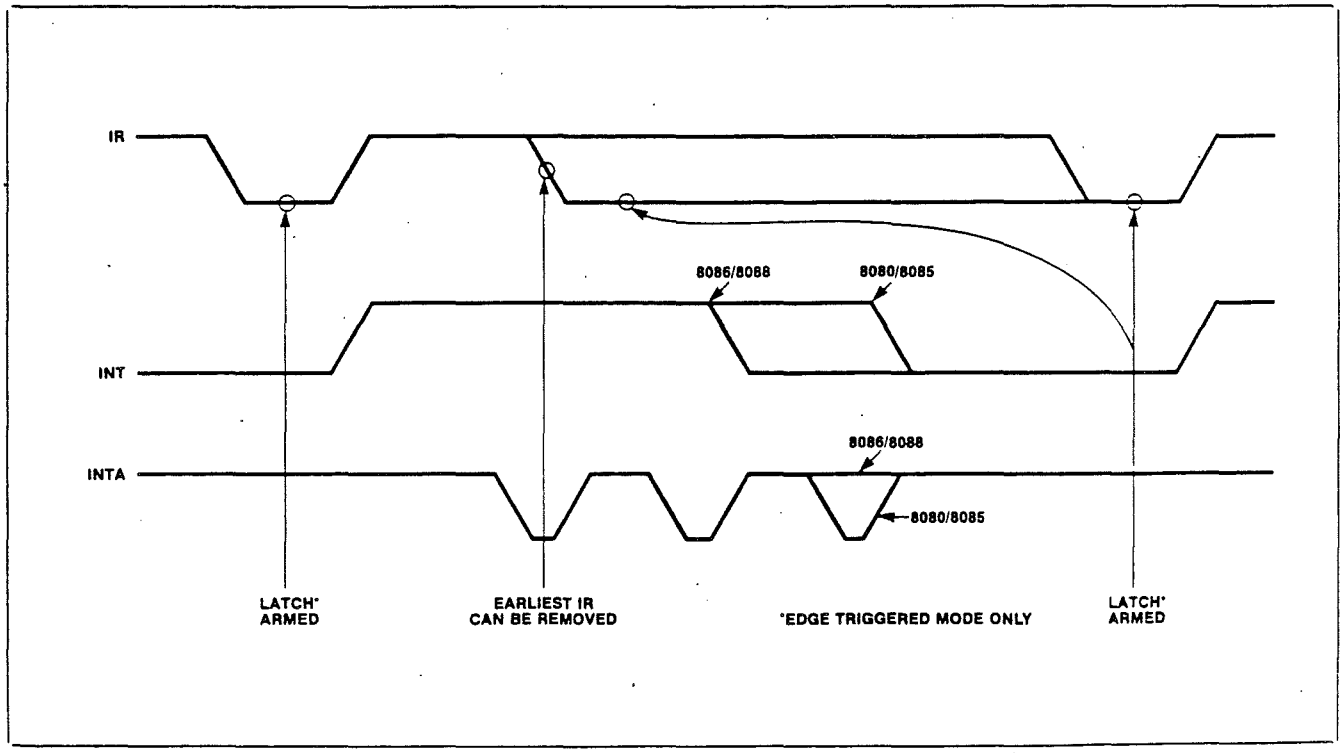


Figure 10. IR Triggering Timing Requirements

THE SPECIAL FULLY NESTED MODE

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

a. When an interrupt request from a certain slave is in service this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

b. When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specific EOI can be sent to the master too. If not, no EOI should be sent.

BUFFERED MODE

When the 8259A is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the 8259A to send an enable signal on $\overline{SP}/\overline{EN}$ to enable the buffers. In this mode, whenever the 8259A's data bus outputs are enabled, the $\overline{SP}/\overline{EN}$ output becomes active.

This modification forces the use of software programming to determine whether the 8259A is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave.

CASCADE MODE

The 8259A can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the $\overline{INTA}$ sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of INTA. (Byte 2 only for 8086/8088).

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first INTA pulse to the trailing edge of the third pulse. Each 8259A in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select (CS) input of each 8259A.

The cascade lines of the Master 8259A are activated only for slave inputs, non slave inputs leave the cascade line inactive (low).

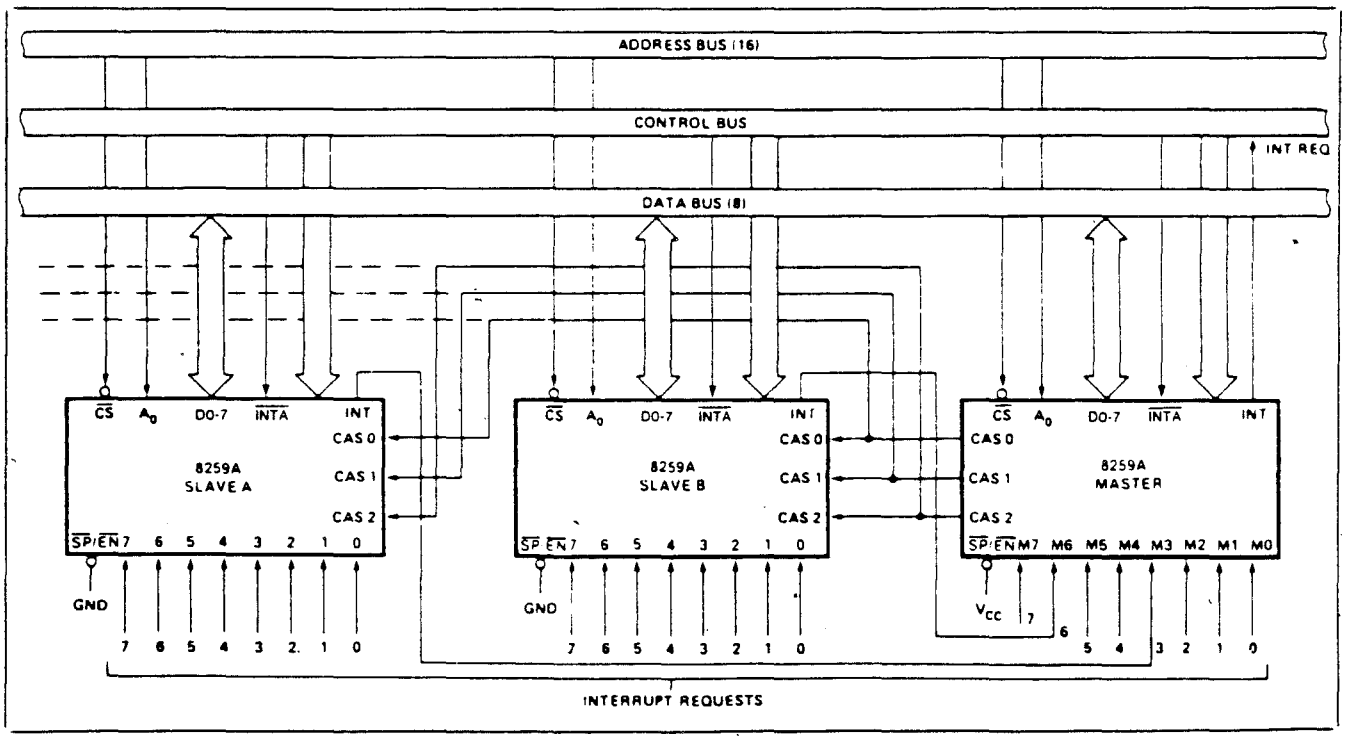


Figure 11. Cascading the 8259A

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin
with Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1 Watt

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

D.C. CHARACTERISTICS [T_A = 0°C to 70°C, V_{CC} = 5V ±5% (8259A-8), V_{CC} = 5V ±10% (8259A, 8259A-2)]

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0*	V_{CC} +0.5V	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400μA
$V_{OH(INT)}$	Interrupt Output High Voltage	3.5		V	I_{OH} = −100μA
		2.4		V	I_{OH} = −400μA
I_{LI}	Input Load Current	−10	+10	μA	$0V \leq V_{IN} \leq V_{CC}$
I_{LOL}	Output Leakage Current	−10	+10	μA	$0.45V \leq V_{OUT} \leq V_{CC}$
I_{CC}	V_{CC} Supply Current		85	mA	
I_{LIR}	IR Input Load Current		−300	μA	V_{IN} = 0
			10	μA	$V_{IN} = V_{CC}$

*Note: For Extended Temperature EXPRESS V_{IH} = 2.3V.

CAPACITANCE (T_A = 25°C; V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHZ
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

A.C. CHARACTERISTICS [T_A = 0°C to 70°C, V_{CC} = 5V ±5% (8259A-8), V_{CC} = 5V ± 10% (8259A, 8259A-2)]

TIMING REQUIREMENTS

Symbol	Parameter	8259A-8		8259A		8259A-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
TAHRL	AO/$\overline{CS}$ Setup to $\overline{RD}$/$\overline{INTA}$↓	50		0		0		ns	
TRHAX	AO/$\overline{CS}$ Hold after $\overline{RD}$/$\overline{INTA}$↑	5		0		0		ns	
TRLRH	$\overline{RD}$ Pulse Width	420		235		160		ns	
TAHWL	AO/$\overline{CS}$ Setup to $\overline{WR}$↓	50		0		0		ns	
TWHAX	AO/$\overline{CS}$ Hold after $\overline{WR}$↑	20		0		0		ns	
TWLWH	$\overline{WR}$ Pulse Width	400		290		190		ns	
TDVWH	Data Setup to $\overline{WR}$↑	300		240		160		ns	
TWHDX	Data Hold after $\overline{WR}$↑	40		0		0		ns	
TJLJH	Interrupt Request Width (Low)	100		100		100		ns	See Note 1
TCVIAL	Cascade Setup to Second or Third $\overline{INTA}$↓ (Slave Only)	55		55		40		ns	
TRHRL	End of $\overline{RD}$ to next $\overline{RD}$ End of $\overline{INTA}$ to next $\overline{INTA}$ within an $\overline{INTA}$ sequence only	160		160		160		ns	
TWHWL	End of $\overline{WR}$ to next $\overline{WR}$	190		190		190		ns	

A.C. CHARACTERISTICS (Continued)

Symbol	Parameter	8259A-8		8259A		8259A-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
*TCHCL	End of Command to next Command (Not same command type)	500		500		500		ns	
	End of $\overline{INTA}$ sequence to next $\overline{INTA}$ sequence.								

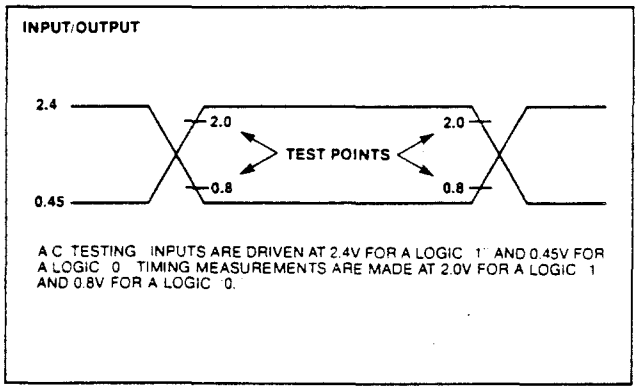
*Worst case timing for TCHCL in an actual microprocessor system is typically much greater than 500 ns (i.e. 8085A = 1.6μs, 8085A-2 = 1μs, 8086 = 1μs, 8086-2 = 625 ns)

NOTE: This is the low time required to clear the input latch in the edge triggered mode.

TIMING RESPONSES

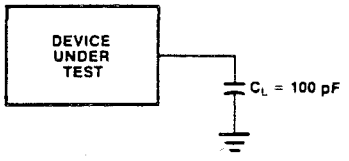
Symbol	Parameter	8259A-8		8259A		8259A-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
TRLDV	Data Valid from $\overline{RD}$/$\overline{INTA}\downarrow$		300		200		120	ns	C of Data Bus = 100 pF
TRHDZ	Data Float after $\overline{RD}$/$\overline{INTA}\uparrow$	10	200	10	100	10	85	ns	C of Data Bus Max text C = 100 pF Min. test C = 15 pF
TJHIH	Interrupt Output Delay		400		350		300	ns	
TIALCV	Cascade Valid from First $\overline{INTA}\downarrow$ (Master Only)		565		565		360	ns	C_{INT} = 100 pF
TRLEL	Enable Active from $\overline{RD}\downarrow$ or $\overline{INTA}\downarrow$		160		125		100	ns	$C_{CASCADE}$ = 100 pF
TRHEH	Enable Inactive from $\overline{RD}\uparrow$ or $\overline{INTA}\uparrow$		325		150		150	ns	
TAHDV	Data Valid from Stable Address		350		200		200	ns	
TCVDV	Cascade Valid to Valid Data		300		300		200	ns	

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

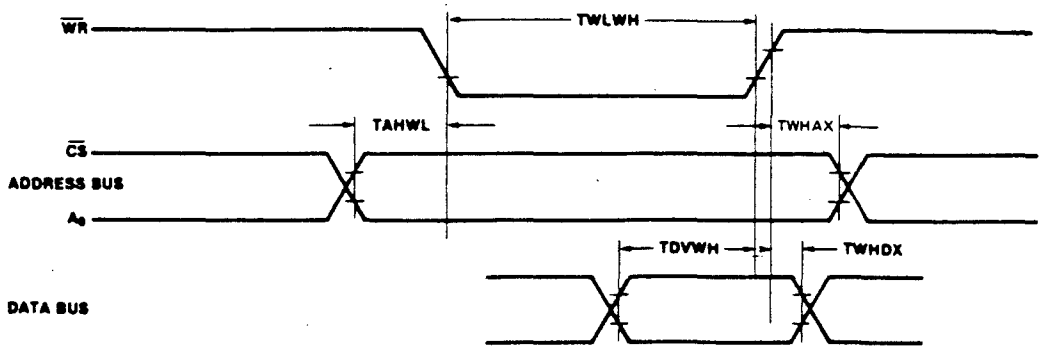
A.C. TESTING LOAD CIRCUIT



C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

WAVEFORMS

WRITE



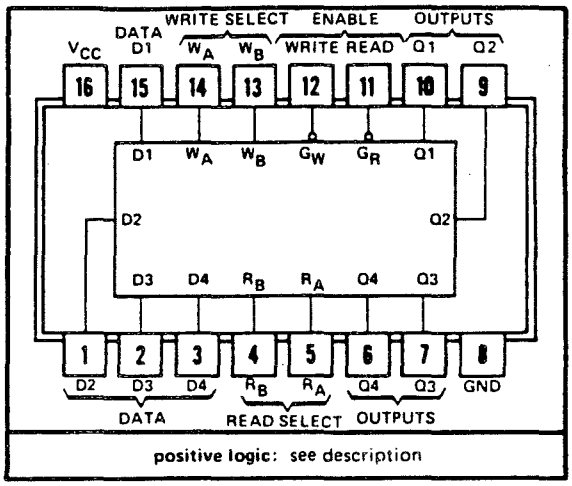
MMU

TYPES SN54LS670, SN74LS670
4-BY-4 REGISTER FILES WITH 3-STATE OUTPUTS

BULLETIN NO. DL-S 7612122, MARCH 1974—REVISED OCTOBER 1976

- Separate Read/Write Addressing Permits Simultaneous Reading and Writing
- Fast Access Times . . . Typically 20 ns
- Organized as 4 Words of 4 Bits
- Expandable to 512 Words of n-Bits
- For Use as:
  - Scratch-Pad Memory
  - Buffer Storage between Processors
  - Bit Storage in Fast Multiplication Designs
- 3-State Outputs
- SN54LS170 and SN74LS170 Are Similar But Have Open-Collector Outputs

SN54LS670 . . . J OR W PACKAGE
SN74LS670 . . . J OR N PACKAGE
(TOP VIEW)



description

The SN54LS670 and SN74LS670 MSI 16-bit TTL register files incorporate the equivalent of 98 gates. The register file is organized as 4 words of 4 bits each and separate on-chip decoding is provided for addressing the four word locations to either write-in or retrieve data. This permits simultaneous writing into one location and reading from another word location.

Four data inputs are available which are used to supply the 4-bit word to be stored. Location of the word is determined by the write-address inputs A and B in conjunction with a write-enable signal. Data applied at the inputs should be in its true form. That is, if a high-level signal is desired from the output, a high-level is applied at the data input for that particular bit location. The latch inputs are arranged so that new data will be accepted only if both internal address gate inputs are high. When this condition exists, data at the D input is transferred to the latch output. When the write-enable input, G_W, is high, the data inputs are inhibited and their levels can cause no change in the information stored in the internal latches. When the read-enable input, G_R, is high, the data outputs are inhibited and go into the high-impedance state.

The individual address lines permit direct acquisition of data stored in any four of the latches. Four individual decoding gates are used to complete the address for reading a word. When the read address is made in conjunction with the read-enable signal, the word appears at the four outputs.

This arrangement—data-entry addressing separate from data-read addressing and individual sense line—eliminates recovery times, permits simultaneous reading and writing, and is limited in speed only by the write time (27 nanoseconds typical) and the read time (24 nanoseconds typical). The register file has a nondestructive readout in that data is not lost when addressed.

All inputs except read enable and write enable are buffered to lower the drive requirements to one Series 54LS/74LS standard load, and input-clamping diodes minimize switching transients to simplify system design. High-speed, double-ended AND-OR-INVERT gates are employed for the read-address function and have high-sink-current, three-state outputs. Up to 120 of these outputs may be wire-AND connected for increasing the capacity up to 512 words. Any number of these registers may be paralleled to provide n-bit word length.

The SN54LS670 characterized for operation over the full military temperature range of −55°C to 125°C; the SN74LS670 is characterized for operation from 0°C to 70°C.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

REVISED OCTOBER 1976

logic

WRITE FUNCTION TABLE (SEE NOTES A, B, AND C)

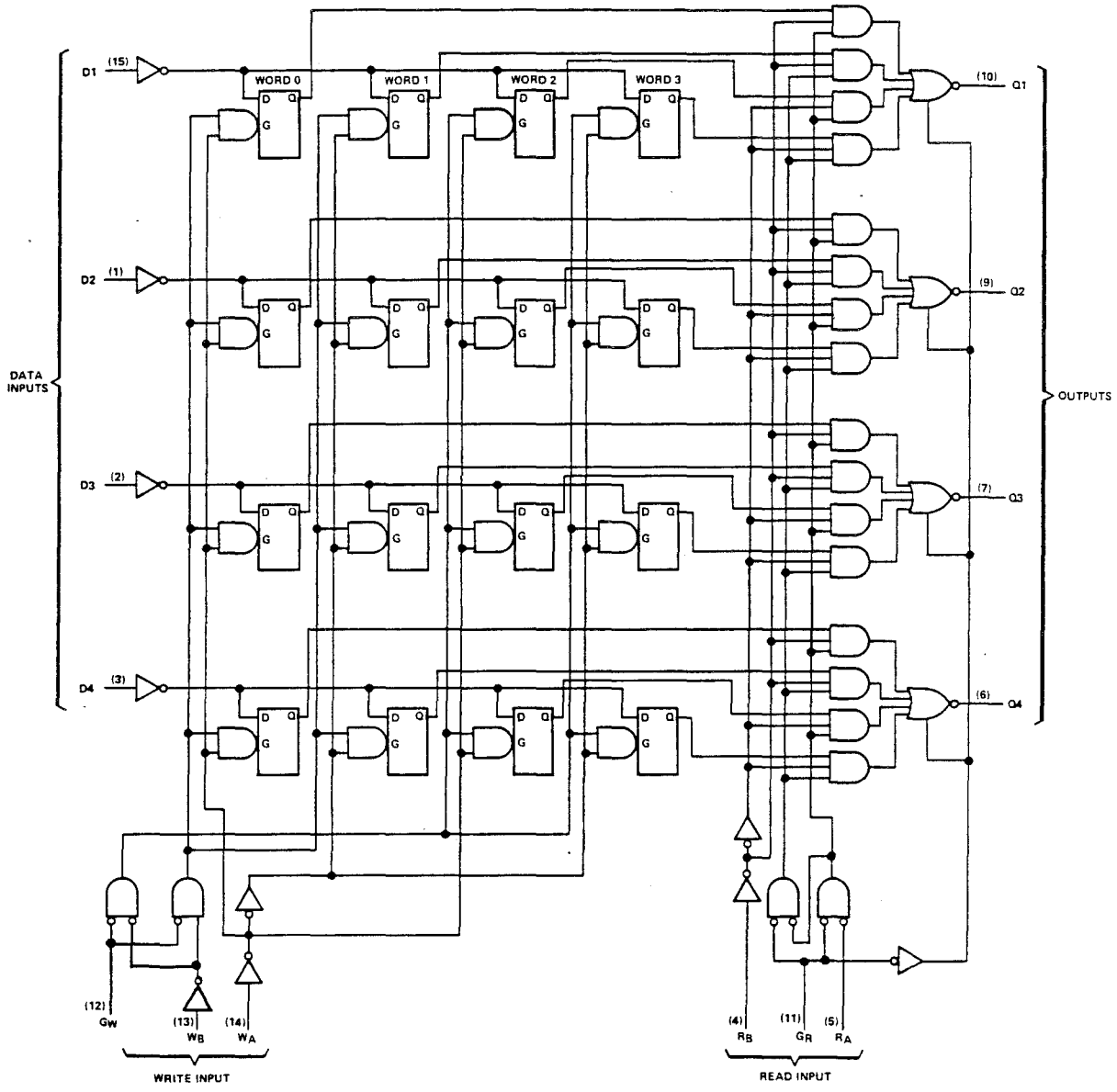
WRITE INPUTS			WORD			
W_B	W_A	G_W	0	1	2	3
L	L	L	Q = D	Q_0	Q_0	Q_0
L	H	L	Q_0	Q = D	Q_0	Q_0
H	L	L	Q_0	Q_0	Q = D	Q_0
H	H	L	Q_0	Q_0	Q_0	Q = D
X	X	H	Q_0	Q_0	Q_0	Q_0

READ FUNCTION TABLE (SEE NOTES A AND D)

READ INPUTS			OUTPUTS			
R_B	R_A	G_R	Q1	Q2	Q3	Q4
L	L	L	W0B1	W0B2	W0B3	W0B4
L	H	L	W1B1	W1B2	W1B3	W1B4
H	L	L	W2B1	W2B2	W2B3	W2B4
H	H	L	W3B1	W3B2	W3B3	W3B4
X	X	H	Z	Z	Z	Z

NOTES: A. H = high level, L = low level, X = irrelevant, Z = high impedance (off)
B. (Q = D) = The four selected internal flip-flop outputs will assume the states applied to the four external data inputs.
C. Q_0 = the level of Q before the indicated input conditions were established.
D. W0B1 = The first bit of word 0, etc.

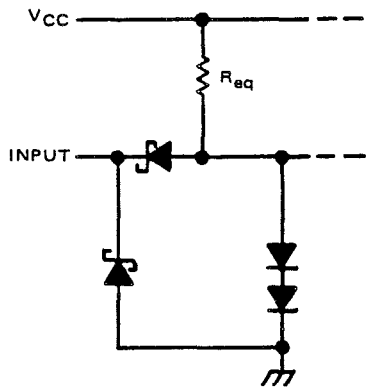
functional block diagram



7

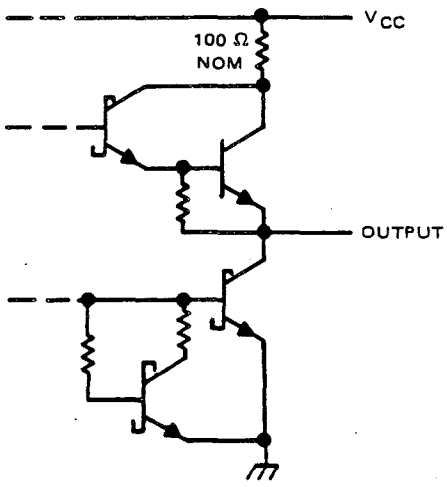
schematics of inputs and outputs

EQUIVALENT OF EACH INPUT



Any D, R, or W: R_{eq} = 20 kΩ NOM
G_R: R_{eq} = 6.67 kΩ NOM
G_W: R_{eq} = 10 kΩ NOM

TYPICAL OF ALL OUTPUTS



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

Supply voltage, V_{CC} (see Note 1) . . . . . 7 V
Input voltage . . . . . 7 V
Off-state output voltage . . . . . 5.5 V
Operating free-air temperature range: SN54LS670 . . . . . −55°C to 125°C
SN74LS670 . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

recommended operating conditions

		SN54LS670			SN74LS670			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}				−1			−2.6	mA
Low-level output current, I_{OL}				4			8	mA
Width of write-enable or read-enable pulse, t_w		25			25			ns
Setup times, high- or low-level data (see Figure 2)	Data input with respect to write enable, $t_{su(D)}$	10			10			ns
	Write select with respect to write enable, $t_{su(W)}$	15			15			ns
Hold times, high- or low-level data (see Note 2 and Figure 2)	Data input with respect to write enable, $t_{h(W)}$	15			15			ns
	Write select with respect to write enable, $t_{h(D)}$	5			5			ns
Latch time for new data, t_{latch} (see Note 3)		25			25			ns
Operating free-air temperature range, T_A		−55		125	0		70	°C

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

2. Write-select setup time will protect the data written into the previous address. If protection of data in the previous address is not required, $t_{su(W)}$ can be ignored as any address selection sustained for the final 30 ns of the write-enable pulse and during $t_{h(W)}$ will result in data being written into that location. Depending on the duration of the input conditions, one or a number of previous addresses may have been written into.

3. Latch time is the time allowed for the internal output of the latch to assume the state of new data. See Figure 2. This is important only when attempting to read from a location immediately after that location has received new data.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

REVISED OCTOBER 1976

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

PARAMETER		TEST CONDITIONS†			SN54LS670			SN74LS670			UNIT
					MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V_{IH}	High-level input voltage				2			2			V
V_{IL}	Low-level input voltage						0.7			0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN,	I_I = −18 mA				−1.5			−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IL} = V_{IL} max	V_{IH} = 2 V,	I_{OH} = −1 mA	2.4	3.4					V
				I_{OH} = −2.6 mA				2.4	3.1		
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IL} = V_{IL} max	V_{IH} = 2 V,	I_{OL} = 4 mA		0.25	0.4		0.25	0.4	V
				I_{OL} = 8 mA					0.35	0.5	
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX,	V_{IH} = 2 V,	V_O = 2.7 V			20			20	μA
I_{OZL}	Off-state output current, low-level voltage applied	V_{CC} = MAX,	V_{IH} = 2 V,	V_O = 0.4 V			−20			−20	μA
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7 V	Any D, R, or W				0.1			0.1	mA
			G_W				0.2			0.2	
			G_R				0.3			0.3	
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7 V	Any D, R, or W				20			20	μA
			G_W				40			40	
			G_R				60			60	
I_{IL}	Low-level input current	V_{CC} = MAX	Any D, R, or W				−0.4			−0.4	mA
			G_W				−0.8			−0.8	
			G_R				−1.2			−1.2	
I_{OS}	Short-circuit output current§	V_{CC} = MAX			−30		−130	−30		−130	mA
I_{CC}	Supply current	V_{CC} = MAX,	See Note 4			30	50		30	50	mA

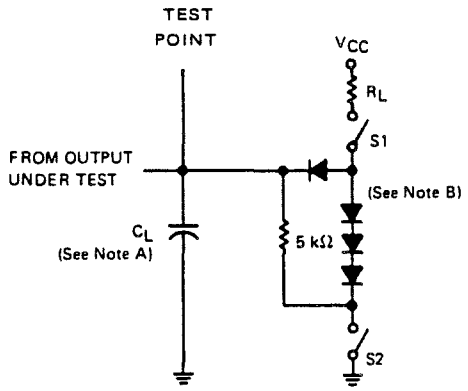
†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at V_{CC} = 5 V, T_A = 25°C.
§Not more than one output should be shorted at a time, and duration of the short-circuit should not exceed one second.
NOTE 4: Maximum I_{CC} is guaranteed for the following worst-case conditions: 4.5 V is applied to all data inputs and both enable inputs, all address inputs are grounded and all outputs are open.

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

PARAMETER¶	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Read select	Any Q	C_L = 15 pF, R_L = 2 kΩ, See Figures 1 and 2		23	40	ns
t_{PHL}					25	45	
t_{PLH}	Write enable	Any Q	C_L = 15 pF, R_L = 2 kΩ, See Figures 1 and 3		26	45	ns
t_{PHL}					28	50	
t_{PLH}	Data	Any Q			25	45	ns
t_{PHL}					23	40	
t_{ZH}	Read enable	Any Q	C_L = 5 pF, R_L = 2 kΩ, See Figures 1 and 4		15	35	ns
t_{ZL}					22	40	
t_{HZ}					30	50	ns
t_{LZ}					16	35	

¶t_{PLH} ≡ propagation delay time, low-to-high-level output
t_{PHL} ≡ propagation delay time, high-to-low-level output
t_{ZH} ≡ output enable time to high level
t_{ZL} ≡ output enable time to low level
t_{HZ} ≡ output disable time from high level
t_{LZ} ≡ output disable time from low level

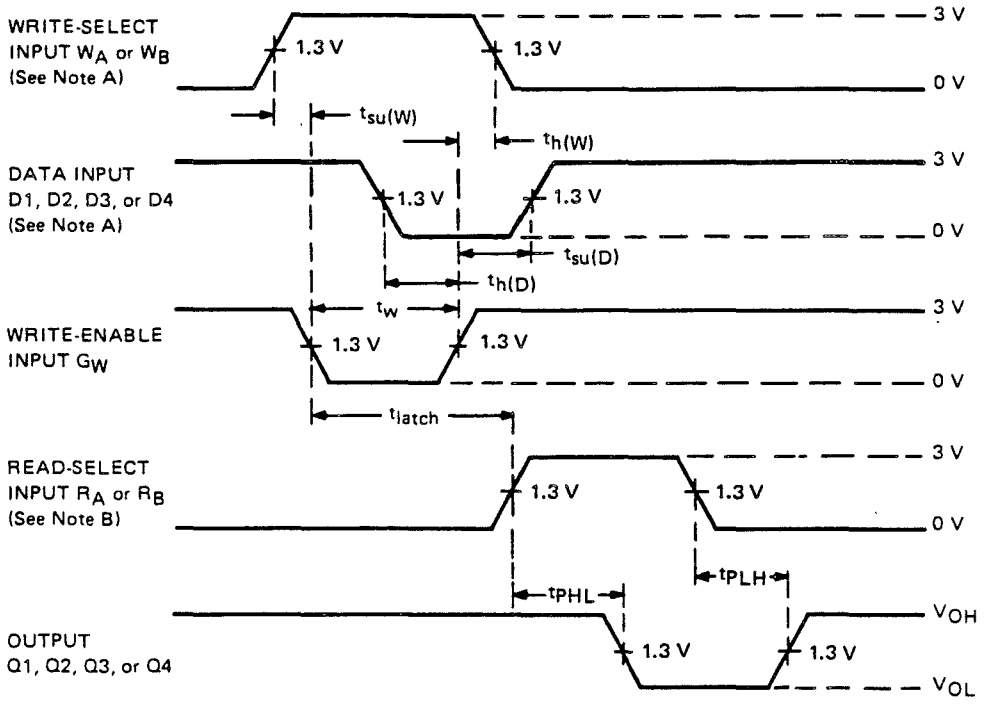
PARAMETER MEASUREMENT INFORMATION



NOTES: A. C_L includes probe and jig capacitance.
B. All diodes are 1N916 or 1N3064.

LOAD CIRCUIT

FIGURE 1



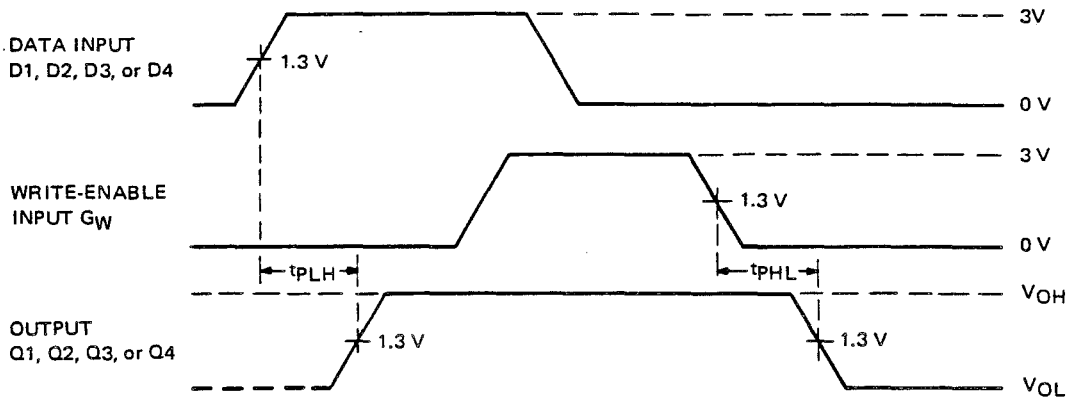
VOLTAGE WAVEFORMS (S1 AND S2 ARE CLOSED)

NOTES: A. High-level input pulses at the select and data inputs are illustrated; however, times associated with low-level pulses are measured from the same reference points.
B. When measuring delay times from a read-select input, the read-enable input is low.
C. Input waveforms are supplied by generators having the following characteristics: PRR ≤ 2 MHz, $Z_{out} \approx 50\ \Omega$, duty cycle ≤ 50%, $t_r \leq 15$ ns, $t_f \leq 6$ ns.

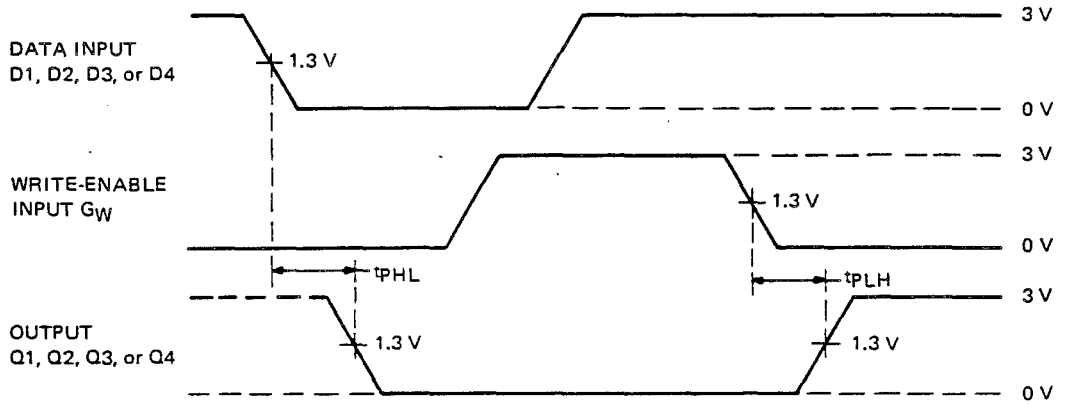
FIGURE 2

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 5012 • DALLAS, TEXAS 75222

PARAMETER MEASUREMENT INFORMATION



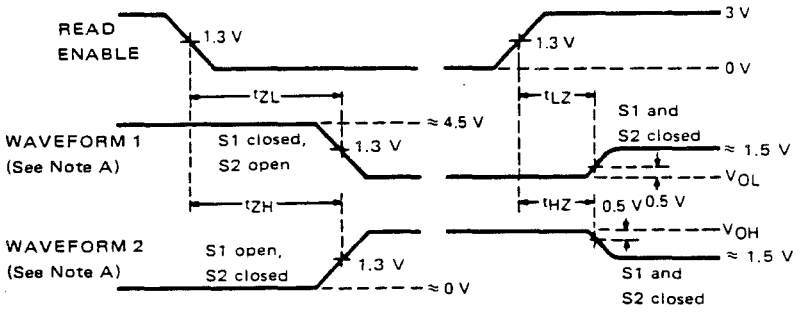
VOLTAGE WAVEFORM 1 (S1 AND S2 ARE CLOSED)



VOLTAGE WAVEFORM 2 (S1 AND S2 ARE CLOSED)

NOTES: A. Each select address is tested. Prior to the start of each of the above tests both write and read address inputs are stabilized with $W_A = R_A$ and $W_B = R_B$. During the test G_R is low.
B. Input waveforms are supplied by generators having the following characteristics: PRR ≤ 1 MHz, $Z_{out} \approx 50\ \Omega$, duty cycle < 50%, $t_r \leq 15$ ns, $t_f \leq 6$ ns.

FIGURE 3



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, THREE-STATE OUTPUTS

NOTES: A. Waveforms 1 is for an output with internal conditions such that the output is low except when disabled by the read-enable input. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the read-enable input.
B. When measuring delay times from the read-enable input, both read-select inputs have been established at steady states.
C. Input waveforms are supplied by generators having the following characteristics: PRR ≤ 1 MHz, $Z_{out} \approx 50\ \Omega$, duty cycle < 50%, $t_r \leq 15$ ns, $t_f \leq 6$ ns.

FIGURE 4

iAPX 80186

High Integration 16-Bit Microprocessor

DISTINCTIVE CHARACTERISTICS

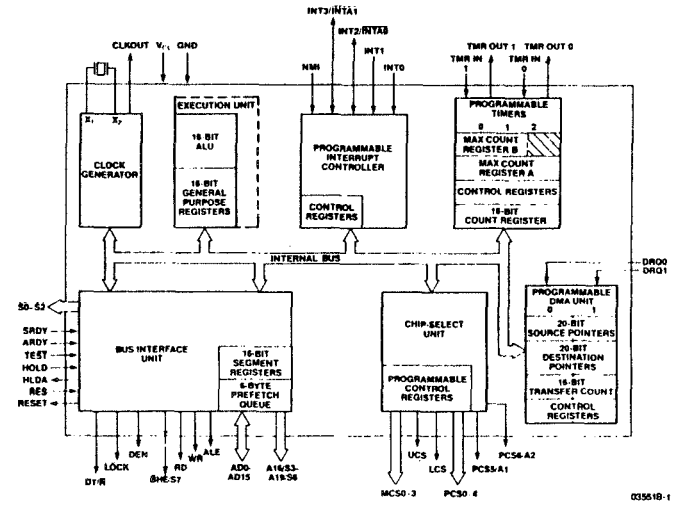
- Integrated feature set
  - Enhanced 8086-2 CPU
  - Clock generator
  - Two independent, high-speed DMA channels
  - Programmable interrupt controller
  - Three programmable 16-bit timers
  - Programmable memory and peripheral chip-select logic
  - Programmable wait state generator
  - Local bus controller
- Available in 8 MHz (80186-3) and cost effective 6 MHz (80186-6) versions
- High performance processor
  - Two times the performance of the standard iAPX 86
  - 4M byte/sec bus bandwidth interface
- Direct addressing capability to 1M byte of memory
- Completely object code compatible with all existing iAPX 86, 88 software
  - Ten new instruction types
- Compatible with 8282/83/86/87, 8288, 8289 bus support components
- Optional numeric processor extension
  - iAPX 186 with a high-performance 80-bit numeric data processor the 8087

GENERAL DESCRIPTION

The iAPX 186 (80186 part number) is a highly integrated 16-bit microprocessor. It effectively combines 15 20 of the most common iAPX 86 system components onto one. The 80186 provides two times greater throughput than the standard 5MHz iAPX 86. The iAPX 186 is upward compatible with iAPX 86 and 88 software and adds 10 new instruction types to the existing set.

The iAPX 186 comes in a 68-pin package and requires a single +5V power supply.

Figure 1. iAPX*186 Block Diagram



Order # 03551B

Advanced Micro Devices

September 1984

Figure 2. 80186 Pinout Diagram

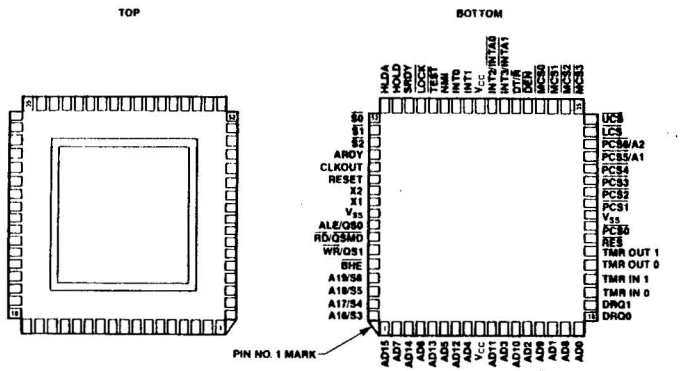


Table 1. 80186 Pin Description

Symbol	Pin No.	Type	Name and Function
V_{CC}, V_{CC}	9,43	I	System Power: +5 volt power supply.
V_{SS}, V_{SS}	26,60	I	System Ground.
RESET	57	O	Reset Output indicates that the 80186 CPU is being reset, and can be used as a system reset. It is active HIGH, synchronized with the processor clock, and lasts an integer number of clock periods corresponding to the length of the $\overline{RES}$ signal.
X1, X2	59,58	I	Crystal Inputs, X1 and X2, provide an external connection for a fundamental mode parallel resonant crystal for the internal crystal oscillator. X1 can interface to an external clock instead of a crystal. The input or oscillator frequency is internally divided by two to generate the clock signal (CLKOUT).
CLKOUT	56	O	Clock Output provides the system with a 50% duty cycle waveform. All device pin timings are specified relative to CLKOUT. CLKOUT has sufficient MOS drive capabilities for the 8087 Numeric Processor Extension.
$\overline{RES}$	24	I	System Reset causes the 80186 to immediately terminate its present activity, clear the internal logic, and enter a dormant state. This signal may be asynchronous to the 80186 clock. The 80186 begins fetching instructions approximately 7 clock cycles after $\overline{RES}$ is returned HIGH. $\overline{RES}$ is required to be LOW for greater than 4 clock cycles and is internally synchronized. For proper initialization, the LOW-to-HIGH transition of $\overline{RES}$ must occur no sooner than 50 microseconds after power up. This input is provided with a Schmitt-trigger to facilitate power-on $\overline{RES}$ generation via an RC network. When $\overline{RES}$ occurs, the 80186 will drive the status lines to an inactive level for one clock, and then tri-state them.

Table 1. 80186 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
$\overline{TEST}$	47	I	$\overline{TEST}$ is examined by the WAIT instruction. If the $\overline{TEST}$ input is HIGH when "WAIT" execution begins, instruction execution will suspend. $\overline{TEST}$ will be resampled until it goes LOW, at which time execution will resume. If interrupts are enabled while the 80186 is waiting for $\overline{TEST}$, interrupts will be serviced. This input is synchronized internally.
TMR IN 0, TMR IN1	20 21	I I	Timer Inputs are used either as clock or control signals, depending upon the programmed timer mode. These inputs are active HIGH (or LOW-to-HIGH transitions are counted) and internally synchronized.
TMR OUT 0, TMR OUT 1	22 23	O O	Timer outputs are used to provide single pulse or continuous waveform generation, depending upon the timer mode selected.
DRQ0 DRQ1	18 19	I I	DMA Request is driven HIGH by an external device when it desires that a DMA channel (Channel 0 or 1) perform a transfer. These signals are active HIGH, level-triggered, and internally synchronized.
NMI	46	I	Non-Maskable Interrupt is an edge-triggered input which causes a type 2 interrupt. NMI is not maskable internally. A transition from a LOW to HIGH initiates the interrupt at the next instruction boundary. NMI is latched internally. An NMI duration of one clock or more will guarantee service. This input is internally synchronized.
INT0, INT1, INT2/$\overline{INTA0}$ INT3/$\overline{INTA1}$	45,44 42 41	I I/O I/O	Maskable Interrupt Requests can be requested by strobing one of these pins. When configured as inputs, these pins are active HIGH. Interrupt Requests are synchronized internally. INT2 and INT3 may be configured via software to provide active-LOW interrupt-acknowledge output signals. All interrupt inputs may be configured via software to be either edge- or level-triggered. To ensure recognition, all interrupt requests must remain active until the interrupt is acknowleged. When iRMX mode is selected, the function of these pins changes (see Interrupt Controller section of this data sheet).
A19/S6, A18/S5, A17/S4, A16/S3	65–68	O O O O	Address Bus Outputs (16–19) and Bus Cycle Status (3–6) reflect the four most significant address bits during T_1. These signals are active HIGH. During T_2, T_3, T_W, and T_4, status information is available on these lines as encoded below: S6: Low = Processor Cycle; High = DMA Cycle S3,S4, and S5 are defined as LOW during T_2–T_4.
AD15–AD0	10–17 1–8	I/O	Address/Data Bus (0–15) signals constitute the time mutiplexed memory or I/O address (T_1) and data (T_2, T_3, T_W, and T_4) bus. The bus is active HIGH. A_0 is analogous to $\overline{BHE}$ for the lower byte of the data bus, pins D_7 through D_0. It is LOW during T_1 when a byte is to be transferred onto the lower portion of the bus in memory or I/O operations.
$\overline{BHE}$/S7	64	O	During T_1 the Bus High Enable signal should be used to determine if data is to be enabled onto the most significant half of the data bus, pins D_{15}–D_8. $\overline{BHE}$ is LOW during T_1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the higher half of the bus. The S_7 status information is available during T_2, T_3, and T_4. S_7 is logically equivalent to $\overline{BHE}$. The signal is active LOW, and is tristated OFF during bus HOLD. $\overline{BHE}$ and A0 Encodings: $\overline{BHE}$ Value / A0 Value / Function 0 / 0 / Word Transfer 0 / 1 / Byte Transfer on upper half of data bus (D15–D8) 1 / 0 / Byte Transfer on lower half of data bus (D_7–D_0) 1 / 1 / Reserved

Table 1. 80186 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
ALE/QS0	61	O	Address Latch Enable/Queue Status 0 is provided by the 80186 to latch the address into the 8282/8283 address latches. ALE is active HIGH. Addresses are guaranteed to be valid on the trailing edge of ALE. The ALE rising edge is generated off the rising edge of the CLKOUT immediately preceding T_1 of the associated bus cycle, effectively one-half clock cycle earlier than in the standard 8086. The trailing edge is generated off the CLKOUT rising edge in T_1 as in the 8086. Note that ALE is never floated.
$\overline{WR}$/QS1	63	O	Write Strobe/Queue Status 1 indicates that the data on the bus is to be written into a memory or an I/O device. $\overline{WR}$ is active for T_2, T_3, and T_W of any write cycle. It is active LOW, and floats during "HOLD." It is driven HIGH for one clock during Reset, and then floated. When the 80186 is in queue status mode, the ALE/QS0 and $\overline{WR}$/QS1 pins provide information about processor/instruction queue interaction. QS1 / QS0: 0 0 — No queue operation; 0 1 — First opcode byte fetched from the queue; 1 1 — Subsequent byte fetched from the queue; 1 0 — Empty the queue
$\overline{RD}$/$\overline{QSMD}$	62	O	Read Strobe indicates that the 80186 is performing a memory or I/O read cycle. $\overline{RD}$ is active LOW for T_2, T_3, and T_W of any read cycle. It is guaranteed not to go LOW in T_2 until after the Address Bus is floated. $\overline{RD}$ is active LOW, and floats during "HOLD." $\overline{RD}$ is driven HIGH for one clock during Reset, and then the output driver is floated. A weak internal pull-up mechanism on the $\overline{RD}$ line holds it HIGH when the line is not driven. During RESET the pin is sampled to determine whether the 80186 should provide ALE, $\overline{WR}$, and $\overline{RD}$, or if the Queue-Status should be provided. $\overline{RD}$ should be connected to GND to provide Queue-Status data.
ARDY	55	I	Asynchronous Ready informs the 80186 that the addressed memory space or I/O device will complete a data transfer. The ARDY input pin will accept an asynchronous input, and is active HIGH. Only the rising edge is internally synchronized by the 80186. This means that the falling edge of ARDY must be synchronized to the 80186 clock. If connected to V_{CC}, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active to terminate a bus cycle.
SRDY	49	I	Synchronous Ready must be synchronized externally to the 80186. The use of SRDY provides a relaxed system-timing specification on the Ready input. This is accomplished by eliminating the one-half clock cycle which is required for internally resolving the signal level when using the ARDY input. This line is active HIGH. If this line is connected to V_{CC}, no WAIT states are inserted. Asynchronous ready (ARDY) or synchronous ready (SRDY) must be active before a bus cycle is terminated.
$\overline{LOCK}$	48	O	$\overline{LOCK}$ output indicates that other system bus masters are not to gain control of the system bus while $\overline{LOCK}$ is active LOW. The $\overline{LOCK}$ signal is requested by the LOCK prefix instruction and is activated at the beginning of the first data cycle associated with the instruction following the LOCK prefix. It remains active until the completion of the instruction following the LOCK prefix. No prefetches will occur while $\overline{LOCK}$ is asserted. $\overline{LOCK}$ is active LOW, is driven HIGH for one clock during RESET, and then floated.

Table 1. 80186 Pin Description (Continued)

Symbol	Pin No.	Type	Name and Function
$\overline{S0}$,$\overline{S1}$,$\overline{S2}$	52–54	O	Bus cycle status $\overline{S0}$–$\overline{S2}$ are encoded to provide bus-transaction information: 80186 Bus Cycle Status Information $\overline{S2}$ / $\overline{S1}$ / $\overline{S0}$ / Bus Cycle Initiated 0 0 0 Interrupt Acknowledge 0 0 1 Read I/O 0 1 0 Write I/O 0 1 1 Halt 1 0 0 Instruction Fetch 1 0 1 Read Data from Memory 1 1 0 Write Data to Memory 1 1 1 Passive (no bus cycle) The status pins float during "HOLD." $\overline{S2}$ may be used as a logical M/$\overline{IO}$ indicator, and $\overline{S1}$ as a DT/$\overline{R}$ indicator. The status lines are driven HIGH for one clock during Reset, and then floated until a bus cycle begins.
HOLD (input) HLDA (output)	50 51	I O	HOLD indicates that another bus master is requesting the local bus. The HOLD input is active HIGH. HOLD may be asynchronous with respect to the 80186 clock. The 80186 will issue a HLDA in response to a HOLD request at the end of T_4 or T_I. Simultaneous with the issuance of HLDA, the 80186 will float the local bus and control lines. After HOLD is detected as being LOW, the 80186 will lower HLDA. When the 80186 needs to run another bus cycle, it will again drive the local bus and control lines.
$\overline{UCS}$	34	O	Upper Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion (1K–256K block) of memory. This line is not floated during bus HOLD. The address range activating $\overline{UCS}$ is software programmable.
$\overline{LCS}$	33	O	Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K–256K) of memory. This line is not floated during bus HOLD. The address range activating $\overline{LCS}$ is software programmable.
$\overline{MCS0}$–3	38,37,36,35	O	Mid-Range Memory Chip Select signals are active LOW when a memory reference is made to the defined mid-range portion of memory (8K–512K). These lines are not floated during bus HOLD. The address ranges activating $\overline{MCS0}$–3 are software programmable.
$\overline{PCS0}$–4	25,27–30	O	Peripheral Chip Select signals 0–4 are active LOW when a reference is made to the defined peripheral area (64K byte I/O space). These lines are not floated during bus HOLD. The address ranges activating $\overline{PCS0}$–4 are software programmable.
$\overline{PCS5}$/A1	31	O	Peripheral Chip Select 5 or Latched A1 may be programmed to provide a sixth peripheral chip select, or to provide an internally latched A1 signal. The address range activating $\overline{PCS5}$ is software programmable. When programmed to provide latched A1, rather than $\overline{PCS5}$, this pin will retain the previously latched value of A1 during a bus HOLD. A1 is active HIGH.
$\overline{PCS6}$/A2	32	O	Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating $\overline{PCS6}$ is software programmable. When programmed to provide latched A2, rather than $\overline{PCS6}$, this pin will retain the previously latched value of A2 during a bus HOLD. A2 is active HIGH.
DT/$\overline{R}$	40	O	Data Transmit/Receive controls the direction of data flow through the external 8286/8287 data bus transceiver. When LOW, data is transferred to the 80186. When HIGH the 80186 places write data on the data bus.
$\overline{DEN}$	39	O	Data Enable is provided as an 8286/8287 data bus transceiver output enable. $\overline{DEN}$ is active LOW during each memory and I/O access. $\overline{DEN}$ is HIGH whenever DT/$\overline{R}$ changes state.

FUNCTIONAL DESCRIPTION

Introduction

The following Functional Description describes the base architecture of the iAPX 186. This architecture is common to the iAPX 86, 88, and 286 microprocessor families as well. The iAPX 186 is a very high integration 16-bit microprocessor. It combines 15–20 of the most common microprocessor system components onto one chip while providing twice the performance of the standard iAPX 86. The 80186 is object code compatible with the iAPX 86, 88 microprocessors and adds 10 new instruction types to the existing iAPX 86, 88 instruction set.

iAPX 186 BASE ARCHITECTURE

The iAPX 86, 88, 186, and 286 family all contain the same basic set of registers, instructions, and addressing modes. The 80186 processor is upward compatible with the 8086, 8088, and 80286 CPUs.

Register Set

The 80186 base architecture has fourteen registers as shown in Figures 3a and 3b. These registers are grouped into the following categories.

General Registers
Eight 16-bit general purpose registers used to contain arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers.

Segment Registers
Four 16-bit special purpose registers select, at any given time, the segments of memory that are immediately addressable for code, stack, and data. (For usage, refer to Memory Organization.)

Base and Index Registers
Four of the general purpose registers may also be used to determine offset addresses of operands in memory. These registers may contain base addresses or indexes to particular locations within a segment. The addressing mode selects the specific registers for operand and address calculations.

Status and Control Registers
Two 16-bit special purpose registers record or alter certain aspects of the 80186 processor state. These are the Instruction Pointer Register, which contains the offset address of the next sequential instruction to be executed, and the Status Word Register, which contains status and control flag bits (see Figures 3a and 3b).

Status Word Description

The Status Word records specific characteristics of the result of logical and arithmetic instructions (bits 0, 2, 4, 6, 7, and 11) and controls the operation of the 80186 within a given operating mode (bits 8, 9, and 10). The Status Word Register is 16-bits wide. The function of the Status Word bits is shown in Table 2.

Figure 3a. 80186 General Purpose Register Set

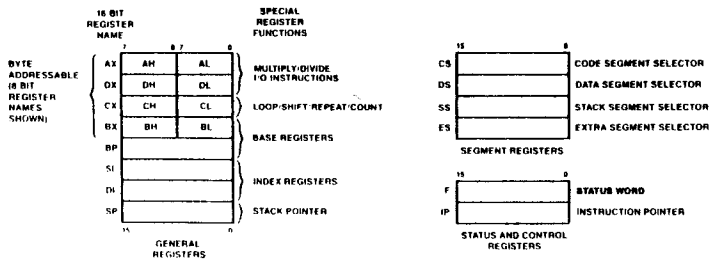


Figure 3b. Status Word Format

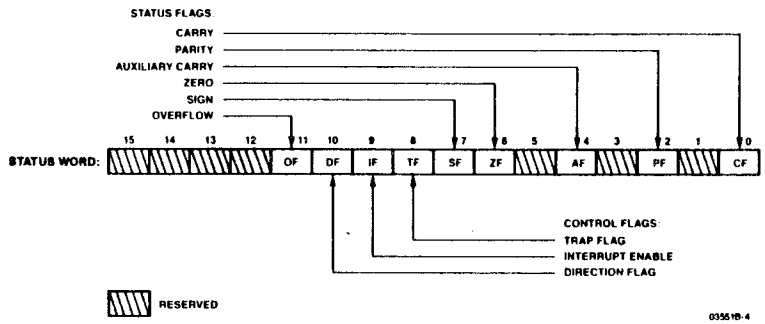


Table 2. Status Word Bit Functions

Bit Position	Name	Function
0	CF	Carry Flag—Set on high-order bit carry or borrow; cleared otherwise
2	PF	Parity Flag—Set if low-order 8 bits of result contain an even number of 1-bits; cleared otherwise
4	AF	Set on carry from or borrow to the low order four bits of AL; cleared otherwise
6	ZF	Zero Flag–Set if result is zero; cleared otherwise
7	SF	Sign Flag–Set equal to high-order bit of result (0 if positive, 1 if negative)
8	TF	Single Step Flag—Once set, a single step interrupt occurs after the next instruction executes. TF is cleared by the single step interrupt
9	IF	Interrupt-enable Flag—When set, maskable interrupts will cause the CPU to transfer control to an interrupt vector specified location.
10	DF	Direction Flag—Causes string instructions to auto decrement the appropriate index register when set. Clearing DF causes auto increment.
11	OF	Overflow Flag—Set if the signed result cannot be expressed within the number of bits in the destination operand, cleared otherwise

Instruction Set

The instruction set is divided into seven categories: data transfer, arithmetic, shift/rotate/logical, string manipulation, control transfer, high-level instructions, and processor control. These categories are summarized in Figure 4.

An 80186 instruction can reference anywhere from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory. Specific operand addressing modes are discussed later in this data sheet.

Memory Organization

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of up to 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address (a pointer) that consists of a 16-bit base segment and a 16-bit offset. The 16-bit base values are contained in one of four internal segment registers (code, data, stack, extra). The physical address is calculated by shifting the base value LEFT by four bits and adding the 16-bit offset value to yield a 20-bit physical address (see Figure 5). This allows for a 1 MByte physical address size.

All instructions that address operands in memory must specify the base segment and the 16-bit offset value. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Table 3). These rules follow the way programs are written (see Figure 6) as independent modules that require areas for code and data, a stack, and access to external data areas.

Special segment override instruction prefixes allow the implicit segment register selection rules to be overridden for special cases. The stack, data, and extra segments may coincide for simple programs.

Figure 4. iAPX 186 Instruction Set

GENERAL PURPOSE	
MOV	Move byte or word
PUSH	Push word onto stack
POP	Pop word off stack
PUSHA	Push all registers on stack
POPA	Pop all registers from stack
XCHG	Exchange byte or word
XLAT	Translate byte
INPUT/OUTPUT	
IN	Input byte or word
OUT	Output byte or word
ADDRESS OBJECT	
LEA	Load effective address
LDS	Load pointer using DS
LES	Load pointer using ES
FLAG TRANSFER	
LAHF	Load AH register from flags
SAHF	Store AH register in flags
PUSHF	Push flags onto stack
POPF	Pop flags off stack

ADDITION	
ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition
SUBTRACTION	
SUB	Subtract byte or word
SBB	Subtract byte or word with borrow
DEC	Decrement byte or word by 1
NEG	Negate byte or word
CMP	Compare byte or word
AAS	ASCII adjust for subtraction
DAS	Decimal adjust for subtraction
MULTIPLICATION	
MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII adjust for multiply
DIVISION	
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to doubleword

MOVS	Move byte or word string
INS	Input bytes or word string
OUTS	Output bytes or word string
CMPS	Compare byte or word string
SCAS	Scan byte or word string
LODS	Load byte or word string
STOS	Store byte or word string
REP	Repeat
REPE/REPZ	Repeat while equal/zero
REPNE/REPNZ	Repeat while not equal/not zero

LOGICALS	
NOT	"Not" byte or word
AND	"And" byte or word
OR	"Inclusive or" byte or word
XOR	"Exclusive or" byte or word
TEST	"Test" byte or word
SHIFTS	
SHL/SAL	Shift logical/arithmetic left byte or word
SHR	Shift logical right byte or word
SAR	Shift arithmetic right byte or word
ROTATES	
ROL	Rotate left byte or word
ROR	Rotate right byte or word
RCL	Rotate through carry left byte or word
RCR	Rotate through carry right byte or word

FLAG OPERATIONS	
STC	Set carry flag
CLC	Clear carry flag
CMC	Complement carry flag
STD	Set direction flag
CLD	Clear direction flag
STI	Set interrupt enable flag
CLI	Clear interrupt enable flag
EXTERNAL SYNCHRONIZATION	
HLT	Halt until interrupt or reset
WAIT	Wait for $\overline{TEST}$ pin active
ESC	Escape to extension processor
LOCK	Lock bus during next instruction
NO OPERATION	
NOP	No operation
HIGH LEVEL INSTRUCTIONS	
ENTER	Format stack for procedure entry
LEAVE	Restore stack for procedure exit
BOUND	Detects values outside prescribed range

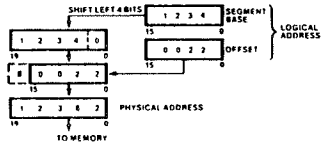
Figure 4. iAPX 186 Instruction Set (continued)

CONDITIONAL TRANSFERS	
JA/JNBE	Jump if above/not below nor equal
JAE/JNB	Jump if above or equal/not below
JB/JNAE	Jump if below/not above nor equal
JBE/JNA	Jump if below or equal/not above
JC	Jump if carry
JE/JZ	Jump if equal/zero
JG/JNLE	Jump if greater/not less nor equal
JGE/JNL	Jump if greater or equal/not less
JL/JNGE	Jump if less/not greater nor equal
JLE/JNG	Jump if less or equal/not greater
JNC	Jump if not carry
JNE/JNZ	Jump if not equal/not zero
JNO	Jump if not overflow
JNP/JPO	Jump if not parity/parity odd
JNS	Jump if not sign
JO	Jump if overflow
JP/JPE	Jump if parity/parity even
JS	Jump if sign

UNCONDITIONAL TRANSFERS	
CALL	Call procedure
RET	Return from procedure
JMP	Jump
ITERATION CONTROLS	
LOOP	Loop
LOOPE/LOOPZ	Loop if equal/zero
LOOPNE/LOOPNZ	Loop if not equal/not zero
JCXZ	Jump if register CX = 0
INTERRUPTS	
INT	Interrupt
INTO	Interrupt if overflow
IRET	Interrupt return

To access operands that do not reside in one of the four immediately available segments, a full 32-bit pointer can be used to reload both the base (segment) and offset values.

Figure 5. Two Component Address

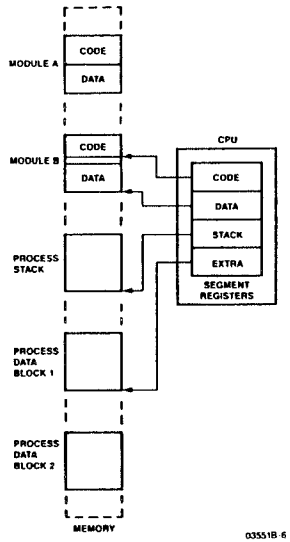


03551B-5

Table 3. Segment Register Selection Rules

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Instructions	Code (CS)	Instruction prefetch and immediate data.
Stack	Stack (SS)	All stack pushes and pops; any memory references which use BP Register as a base register.
External Data (Global)	Extra (ES)	All string instruction references which use the DI register as an index.
Local Data	Data (DS)	All other data references.

Figure 6. Segmented Memory Helps Structure Software



03551B-6

Addressing Modes

The 80186 provides eight categories of addressing modes to specify operands. Two addressing modes are provided for instructions that operate on register or immediate operands:

- *Register Operand Mode:* The operand is located in one of the 8- or 16-bit general registers.
- *Immediate Operand Mode:* The operand is included in the instruction.

Six modes are provided to specify the location of an operand in a memory segment. A memory operand address consists of two 16-bit components: a segment base and an offset. The segment base is supplied by a 16-bit segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix. The offset, also called the effective address, is calculated by summing any combination of the following three address elements:

- the *displacement* (an 8- or 16-bit immediate value contained in the instruction);
- the *base* (contents of either the BX or BP base registers); and
- the *index* (contents of either the SI or DI index registers).

Any carry out from the 16-bit addition is ignored. Eight-bit displacements are sign extended to 16-bit values.

Combinations of these three address elements define the six memory addressing modes, described below.

- *Direct Mode:* The operand's offset is contained in the instruction as an 8- or 16-bit displacement element.
- *Register Indirect Mode:* The operand's offset is in one of the registers SI, DI, BX, or BP.
- *Based Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
- *Indexed Mode:* The operand's offset is the sum of an 8- or 16-bit displacement and the contents of an index register (SI or DI).
- *Based Indexed Mode:* The operand's offset is the sum of the contents of a base register and an index register.
- *Based Indexed Mode with Displacement:* The operand's offset is the sum of a base register's contents, an index register's contents, and an 8- or 16-bit displacement.

Data Types

The 80186 directly supports the following data types:

- *Integer:* A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a 2's complement representation. Signed 32 and 64 bit integers are supported using the 8087 Numeric Data Processor.
- *Ordinal:* An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.
- *Pointer:* A 16- or 32-bit quantity, composed of a 16-bit offset component or a 16-bit segment base component in addition to a 16-bit offset component.
- *String:* A contiguous sequence of bytes or words. A string may contain from 1K to 64K bytes.
- *ASCII:* A byte representation of alphanumeric and control characters using the ASCII standard of character representation.
- *BCD:* A byte (unpacked) representation of the decimal digits 0–9.
- *Packed BCD:* A byte (packed) representation of two decimal digits (0–9). One digit is stored in each nibble (4-bits) of the byte.
- *Floating Point:* A signed 32-, 64-, or 80-bit real number representation. (Floating point operands are supported using the iAPX 186/8087 Numeric Data Processor configuration.)

In general, individual data elements must fit within defined segment limits. Figure 7 graphically represents the data types supported by the iAPX 186.

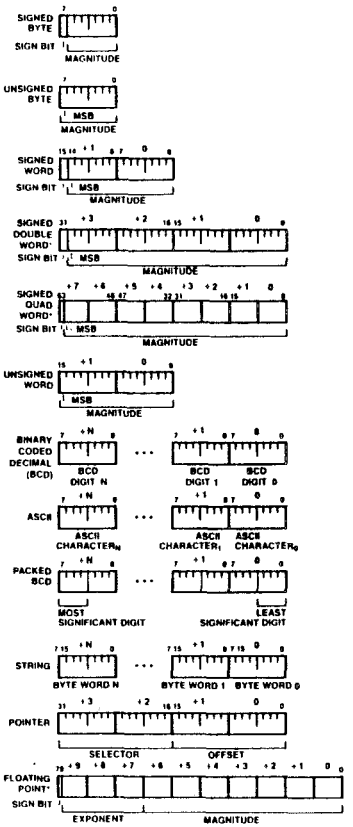
I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. Separate instructions address the I/O space with either an 8-bit port address, specified in the instruction, or a 16-bit port address in the DX register. 8-bit port addresses are zero extended such that A_{15}-A_8 are LOW. I/O port addresses 00F8(H) through 00FF(H) are reserved.

Interrupts

An interrupt transfers execution to a new program location. The old program address (CS:IP) and machine state (Status Word) are saved on the stack to allow resumption of the interrupted program. Interrupts fall into three classes: hardware initiated, INT instructions, and instruction exceptions. Hardware initiated interrupts occur in response to an external input and are classified as non-maskable or maskable.

Figure 7. iAPX 186 Supported Data Types



0355IB-7

NOTE:
*SUPPORTED BY iAPX 186 WITH A NUMERIC DATA PROCESSOR

Programs may cause an interrupt with an INT instruction. Instruction exceptions occur when an unusual condition, which prevents further instruction processing, is detected while attempting to execute an instruction. If the exception was caused by executing an ESC instruction with the ESC trap bit set in the relocation register, the return instruction will point to the ESC instruction, or to the segment override prefix immediately preceding the ESC instruction if the prefix was present. In all other cases, the return address from an exception will point at the instruction immediately following the instruction causing the exception.

A table containing up to 256 pointers defines the proper interrupt service routine for each interrupt. Interrupts 0–31, some of which are used for instruction exceptions, are reserved. Table 4 shows the 80186 predefined types and default priority levels. For each interrupt, an 8-bit vector must be supplied to the 80186 which identifies the appropriate table entry. Exceptions supply the interrupt vector internally. In addition, internal peripherals and non-cascaded external interrupts will generate their own vectors through the internal interrupt controller. INT instructions contain or imply the vector and allow access to all 256 interrupts. Maskable hardware initiated interrupts supply the 8-bit vector to the CPU during an interrupt acknowledge bus sequence. Non-maskable hardware interrupts use a predefined internally supplied vector.

Interrupt Sources

The 80186 can service interrupts generated by software or hardware. The software interrupts are generated by specific instructions (INT, ESC, unused OP, etc.) or the results of conditions specified by instructions (array bounds check, INT0, DIV, IDIV, etc.). All interrupt sources are serviced by an indirect call through an element of a vector table. This vector table is indexed by using the interrupt vector type (Table 4), multiplied by four. All hardware-generated interrupts are sampled at the end of each instruction. Thus, the software interrupts will begin service first. Once the service routine is entered and interrupts are enabled, any hardware source of sufficient priority can interrupt the service routine in progress.

The software generated 80186 interrupts are described below.

DIVIDE ERROR EXCEPTION (TYPE 0)
Generated when a DIV or IDIV instruction quotient cannot be expressed in the number of bits in the destination.

Table 4. 80186 Interrupt Vectors

Interrupt Name	Vector Type	Default Priority	Related Instructions
Divide Error Exception	0	*1	DIV, IDIV
Single Step Interrupt	1	12**2	All
NMI	2	1	All
Breakpoint Interrupt	3	*1	INT
INT0 Detected Overflow Exception	4	*1	INT0
Array Bounds Exception	5	*1	BOUND
Unused-Opcode Exception	6	*1	Undefined Opcodes
ESC Opcode Exception	7	*1***	ESC Opcodes
Timer 0 Interrupt	8	2A****	
Timer 1 Interrupt	16	2B****	
Timer 2 Interrupt	17	2C****	
Reserved	9	3	
DMA 0 Interrupt	10	4	
DMA 1 Interrupt	11	5	
INT0 Interrupt	12	6	
INT1 Interrupt	13	7	
INT2 Interrupt	14	8	
INT3 Interrupt	15	9	

NOTES:
- *1 These are generated as the result of an instruction execution
- **2 This is handled as in the 8086
- ***3 All three timers constitute one source of request to the interrupt controller The Timer interrupts all have the same default priority level with respect to all other interrupt sources However, they have a defined priority ordering amongst themselves (Priority 2A is higher priority than 2B) Each Timer interrupt has a separate vector type number
- 4 Default priorities for the interrupt sources are used only if the user does not program each source into a unique priority level
- ***5. An escape opcode will cause a trap only if the proper bit is set in the peripheral control block relocation register

SINGLE-STEP INTERRUPT (TYPE 1)
Generated after most instructions if the TF flag is set. Interrupts will not be generated after prefix instructions (e.g., REP), instructions which modify segment registers (e.g., POP DS), or the WAIT instruction.

NON-MASKABLE INTERRUPT—NMI (TYPE 2)
An external interrupt source which cannot be masked.

BREAKPOINT INTERRUPT (TYPE 3)
A one-byte version of the INT instruction. It uses 12 as an index into the service routine address table (because it is a type 3 interrupt).

INT0 DETECTED OVERFLOW EXCEPTION (TYPE 4)
Generated during an INT0 instruction if the 0F bit is set.

ARRAY BOUNDS EXCEPTION (TYPE 5)
Generated during a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

UNUSED OPCODE EXCEPTION (TYPE 6)
Generated if execution is attempted on undefined opcodes.

ESCAPE OPCODE EXCEPTION (TYPE 7)
Generated if execution is attempted of ESC opcodes (D8H–DFH). This exception will only be generated if a bit in the relocation register is set. The return address of this exception will point to the ESC instruction causing the exception. If a segment override prefix preceded the ESC instruction, the return address will point to the segment override prefix.

Hardware-generated interrupts are divided into two groups: maskable interrupts and non-maskable interrupts. The 80186 provides maskable hardware interrupt request pins INT0–INT3. In addition, maskable interrupts may be generated by the 80186 integrated DMA controller and the integrated timer unit. The vector types for these interrupts is shown in Table 4. Software enables these inputs by setting the interrupt flag bit (IF) in the Status Word. The interrupt controller is discussed in the peripheral section of this data sheet.

Further maskable interrupts are disabled while servicing an interrupt because the IF bit is reset as part of the response to an interrupt or exception. The saved Status Word will reflect the enable status of the processor prior to the interrupt. The interrupt flag will remain zero unless specifically set. The interrupt return instruction restores the Status Word, thereby restoring the original status of IF bit. If the interrupt return re-enables interrupts, and another interrupt is pending, the 80186 will immediately service the highest-priority interrupt pending, i.e., no instructions of the main line program will be executed.

Non-Maskable Interrupt Request (NMI)

A non-maskable interrupt (NMI) is also provided. This interrupt is serviced regardless of the state of the IF bit. A typical use of NMI would be to activate a power failure routine. The activation of this input

causes an interrupt with an internally supplied vector value of 2. No external interrupt acknowledge sequence is performed. The IF bit is cleared at the beginning of an NMI interrupt to prevent maskable interrupts from being serviced.

Single-Step Interrupt

The 80186 has an internal interrupt that allows programs to execute one instruction at a time. It is called the single-step interrupt and is controlled by the single-step flag bit (TF) in the Status Word. Once this bit is set, an internal single-step interrupt will occur after the next instruction has been executed. The interrupt clears the TF bit and uses an internally supplied vector of 1. The IRET instruction is used to set the TF bit and transfer control to the next instruction to be single-stepped.

Initialization and Processor Reset

Processor initialization or startup is accomplished by driving the $\overline{RES}$ input pin LOW. $\overline{RES}$ forces the 80186 to terminate all execution and local bus activity. No instruction or bus activity will occur as long as $\overline{RES}$ is active. After $\overline{RES}$ becomes inactive and an internal processing interval elapses, the 80186 begins execution with the instruction at physical location FFFF0(H). $\overline{RES}$ also sets some registers to predefined values as shown in Table 5.

Table 5. 80186 Initial Register State after RESET

Status Word	F002(H)
Instruction Pointer	0000(H)
Code Segment	FFFF(H)
Data Segment	0000(H)
Extra Segment	0000(H)
Stack Segment	0000(H)
Relocation Register	20FF(H)
UMCS	FFFB(H)

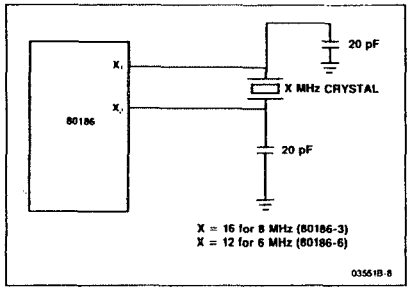
iAPX 186 CLOCK GENERATOR

The iAPX 186 provides an on-chip clock generator for both internal and external clock generation. The clock generator features a crystal oscillator, a divide-by-two counter, synchronous and asynchronous ready inputs, and reset circuitry.

Oscillator

The oscillator circuit of the iAPX 186 is designed to be used with a parallel resonant fundamental mode crystal. This is used as the time base for the iAPX 186. The crystal frequency selected will be double the CPU clock frequency. Use of an LC or RC circuit is not recommended with this oscillator. If an external oscillator is used, it can be connected directly to input pin X1 in lieu of a crystal. The output of the oscillator is not directly available outside the iAPX 186. The recommended crystal configuration is shown in Figure 8.

Figure 8. Recommended iAPX 186 Crystal Configuration



Clock Generator

The iAPX 186 clock generator provides the 50% duty cycle processor clock for the iAPX 186. It does this by dividing the oscillator output by 2 forming the symmetrical clock. If an external oscillator is used, the state of the clock generator will change on the falling edge of the oscillator signal. The CLKOUT pin provides the processor clock signal for use outside the iAPX 186. This may be used to drive other system components. All timings are referenced to the output clock.

READY Synchronization

The iAPX 186 provides both synchronous and asynchronous ready inputs. Asynchronous ready synchronization is accomplished by circuitry which samples ARDY in the middle of T_2, the end of T_2, and again in the middle of each T_W until ARDY is sampled HIGH. One-half CLKOUT cycle of resolution time is used. Full synchronization is performed only on the rising edge of ARDY, i.e., the falling edge of ARDY must be synchronized to the CLKOUT signal if it will occur during T_2 or T_W. HIGH-to-LOW transitions of ARDY must be performed synchronously to the CPU clock.

A second ready input (SRDY) is provided to interface with externally synchronized ready signals. This input is sampled at the end of T_2 and again at the end of each T_W until it is sampled HIGH. By using this input rather than the asynchronous ready input, the half-clock cycle resolution time penalty is eliminated.

This input must satisfy set-up and hold times to guarantee proper operation of the circuit.

In addition, the iAPX 186, as part of the integrated chip-select logic, has the capability to program WAIT states for memory and peripheral blocks. This is discussed in the Chip Select/Ready Logic description.

RESET Logic

The iAPX 186 provides both a $\overline{RES}$ input pin and a synchronized RESET pin for use with other system components. The $\overline{RES}$ input pin on the iAPX 186 is provided with hysteresis in order to facilitate power-on Reset generation via an RC network. RESET is guaranteed to remain active for at least five clocks given a $\overline{RES}$ input of at least six clocks. RESET may be delayed up to two and one-half clocks behind $\overline{RES}$.

Multiple iAPX 186 processors may be synchronized through the $\overline{RES}$ input pin, since this input resets both the processor and divide-by-two internal counter in the clock generator. In order to insure that the divide-by-two counters all begin counting at the same time, the active going edge of $\overline{RES}$ must satisfy a 25 ns setup time before the falling edge of the 80186 clock input. In addition, in order to insure that all CPUs begin executing in the same clock cycle, the reset must satisfy a 25 ns setup time before the rising edge of the CLKOUT signal of all the processors.

LOCAL BUS CONTROLLER

The iAPX 186 provides a local bus controller to generate the local bus control signals. In addition, it employs a HOLD/HLDA protocol for relinquishing the local bus to other bus masters. It also provides control lines that can be used to enable external buffers and to direct the flow of data on and off the local bus.

Memory/Peripheral Control

The iAPX 186 provides ALE, $\overline{RD}$, and $\overline{WR}$ bus control signals. The $\overline{RD}$ and $\overline{WR}$ signals are used to strobe data from memory to the iAPX 186 or to strobe data from the iAPX 186 to memory. The ALE line provides a strobe to address latches for the multiplexed address/data bus. The iAPX 186 local bus controller does not provide a memory/$\overline{I/O}$ signal. If this is required, the user will have to use the $\overline{S2}$ signal (which will require external latching), make the memory and I/O spaces nonoverlapping, or use only the integrated chip-select circuitry.

Transceiver Control

The iAPX 186 generates two control signals to be connected to 8286/8287 transceiver chips. This capability allows the addition of transceivers for extra buffering without adding external logic. These control lines, DT/$\overline{R}$ and $\overline{DEN}$, are generated to control the flow of data through the transceivers. The operation of these signals is shown in Table 6.

Table 6. Transceiver Control Signals Description

Pin Name	Function
$\overline{DEN}$ (Data Enable)	Enables the output drivers of the transceivers. It is active LOW during memory, I/O, or INTA cycles.
DT/$\overline{R}$ (Data Transmit/ Receive)	Determines the direction of travel through the transceivers. A HIGH level directs data away from the processor during write operations, while a LOW level directs data toward the processor during a read operation.

Local Bus Arbitration

The iAPX 186 uses a HOLD/HLDA system of local bus exchange. This provides an asynchronous bus exchange mechanism. This means multiple masters utilizing the same bus can operate at separate clock frequencies. The iAPX 186 provides a single HOLD/HLDA pair through which all other bus masters may gain control of the local bus. This requires external circuitry to arbitrate which external device will gain control of the bus from the iAPX 186 when there is more than one alternate local bus master. When the iAPX 186 relinquishes control of the local bus, it floats $\overline{DEN}$, $\overline{RD}$, $\overline{WR}$, $\overline{S0}$–$\overline{S2}$, $\overline{LOCK}$, AD0–AD15, A16–A19, $\overline{BHE}$, and DT/$\overline{R}$ to allow another master to drive these lines directly.

The iAPX 186 HOLD latency time, i.e., the time between HOLD request and HOLD acknowledge, is a function of the activity occurring in the processor when the HOLD request is received. A HOLD request is the highest-priority activity request which the processor may receive: higher than instruction fetching or internal DMA cycles. However, if a DMA cycle is in progress, the iAPX 186 will complete the transfer before relinquishing the bus. This implies that if a HOLD request is received just as a DMA transfer begins, the HOLD latency time can be as great as 4 bus cycles. This will occur if a DMA word transfer operation is taking place from an odd address to an odd address. This is a total of 16 clocks or more, if WAIT states are required. In addition, if locked transfers are performed, the HOLD latency time will be increased by the length of the locked transfer.

Local Bus Controller and Reset

Upon receipt of a RESET pulse from the $\overline{RES}$ input, the local bus controller will perform the following actions:

- Drive $\overline{DEN}$, $\overline{RD}$, and $\overline{WR}$ HIGH for one clock cycle, then float.

NOTE: $\overline{RD}$ is also provided with an internal pull-up device to prevent the processor from inadvertently entering Queue Status mode during reset.

- Drive $\overline{S0}$–$\overline{S2}$ to the passive state (all HIGH) and then float.
- Drive $\overline{LOCK}$ HIGH and then float.
- Tristate AD0–15, A16–19, $\overline{BHE}$, DT/$\overline{R}$.
- Drive ALE LOW (ALE is never floated).
- Drive HLDA LOW.

INTERNAL PERIPHERAL INTERFACE

All the iAPX 186 integrated peripherals are controlled via 16-bit registers contained within an internal 256-byte control block. This control block may be mapped into either memory or I/O space. Internal logic will recognize the address and respond to the bus cycle. During bus cycles to internal registers, the bus controller will signal the operation externally (i.e., the $\overline{RD}$, $\overline{WR}$, status, address, data, etc., lines will be driven as in a normal bus cycle), but D_{15-0}, SRDY, and ARDY will be ignored. The base address of the control block must be on an even 256-byte boundary (i.e., the lower 8 bits of the base address are all zeros). All of the defined registers within this control block may be read or written by the 80186 CPU at any time. The location of any register contained within the 256-byte control block is determined by the current base address of the control block.

The control block base address is programmed via a 16-bit relocation register contained within the control block at offset FEH from the base address of the control block (see Figure 9). It provides the upper 12 bits of the base address of the control block. Note that mapping the control register block into an address range corresponding to a chip-select range is not recommended (the chip select circuitry is discussed later in this data sheet). In addition, bit 12 of this register determines whether the control block will be mapped into I/O or memory space. If this bit is 1, the control block will be located in memory space, whereas if the bit is 0, the control block will be located in I/O space. If the control register block is mapped into I/O space, the upper 4 bits of the base address must be programmed as 0 (since I/O addresses are only 16 bits wide).

In addition to providing relocation information for the control block, the relocation register contains bits which place the interrupt controller into iRMX mode, and cause the CPU to interrupt upon encountering ESC instructions. At RESET, the relocation register is set to 20FFH. This causes the control block to start at FF00H in I/O space. An offset map of the 256-byte control register block is shown in Figure 10.

The integrated iAPX 186 peripherals operate semi-autonomously from the CPU. Access to them for the most part is via software read/write of the control and data locations in the control block. Most of these registers can be both read and written. A few dedicated lines, such as interrupts and DMA request provide real-time communication between the CPU and peripherals as in a more conventional system utilizing discrete peripheral blocks. The overall interaction and function of the peripheral blocks has not substantially changed.

CHIP-SELECT/READY GENERATION LOGIC

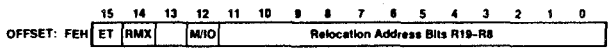
The iAPX 186 contains logic which provides programmable chip-select generation for both memories and peripherals. In addition, it can be programmed to provide READY (or WAIT state) generation. It can also provide latched address bits A1 and A2. The chip-select lines are active for all memory and I/O cycles in their programmed areas, whether they be generated by the CPU or by the integrated DMA unit.

Memory Chip Selects

The iAPX 186 provides 6 memory chip select outputs for 3 address areas: upper memory, lower memory, and midrange memory. One each is provided for upper memory and lower memory, while four are provided for midrange memory.

The range for each chip select is user-programmable and can be set to 2K, 4K, 8K, 16K, 32K, 64K, 128K (plus 1K and 256K for upper and lower chip selects). In addition, the beginning or base address of the midrange memory chip select may also be selected. Only one chip select may be programmed to be active for any memory location at a time. All chip select sizes are in bytes, whereas iAPX 186 memory is arranged in words. This means that if, for example, 16 64K x 1 memories are used, the memory block size will be 128K, not 64K.

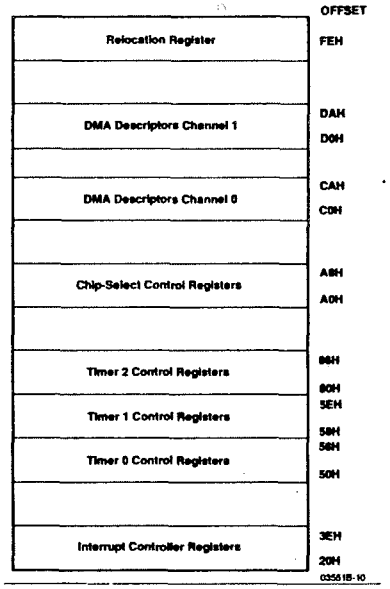
Figure 9. Relocation Register



ET ESC Trap / No ESC Trap (1/0)
M/IO Register block located in Memory / I/O Space (1/0)
RMX Normal Interrupt Controller mode / iRMX compatible Interrupt Controller mode (0/1)

035518-9

Figure 10. Internal Register Map



035518-10

Table 7. UMCS Programming Values

Starting Address (Base Address)	Memory Block Size	UMCS Value (Assuming R0=R1=R2=0)
FFC00	1K	FFF8H
FF800	2K	FFB8H
FF000	4K	FF38H
FE000	8K	FE38H
FC000	16K	FC38H
F8000	32K	F838H
F0000	64K	F038H
E0000	128K	E038H
C0000	256K	C038H

Upper Memory $\overline{CS}$

The iAPX 186 provides a chip select, called $\overline{UCS}$, for the top of memory. The top of memory is usually used as the system memory because after reset the iAPX 186 begins executing at memory location FFFF0H.

The upper limit of memory defined by this chip select is always FFFFFH, while the lower limit is programmable. By programming the lower limit, the size of the select block is also defined. Table 7 shows the relationship between the base address selected and the size of the memory block obtained.

The lower limit of this memory block is defined in the UMCS register (see Figure 11). This register is at offset A0H in the internal control block. The legal values for bits 6–13 and the resulting starting address and memory block sizes are given in Table 7. Any combination of bits 6–13 not shown in Table 7 will result in undefined operation. After reset, the UMCS register is programmed for a 1K area. It must be reprogrammed if a larger upper memory area is desired.

Any internally generated 20-bit address whose upper 16 bits are greater than or equal to UMCS (with bits 0–5 "0") will cause UCS to be activated. UMCS bits R2–R0 are used to specify READY mode for the area of memory defined by this chip-select register, as explained below.

Lower Memory $\overline{CS}$

The iAPX 186 provides a chip select for low memory called $\overline{LCS}$. The bottom of memory contains the interrupt vector table, starting at location 00000H.

The lower limit of memory defined by this chip select is always 0H, while the upper limit is programmable. By programming the upper limit, the size of the memory block is also defined. Table 8 shows the relationship between the upper address selected and the size of the memory block obtained.

Table 8. LMCS Programming Values

Upper Address	Memory Block Size	LMCS Value (Assuming R0=R1=R2=0)
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

The upper limit of this memory block is defined in the LMCS register (see Figure 12). This register is at offset A2H in the internal control block. The legal values for bits 6–15 and the resulting upper address and memory block sizes are given in Table 8. Any combination of bits 6–15 not shown in Table 8 will result in undefined operation. After reset, the LMCS register value is undefined. However, the $\overline{LCS}$ chip-select line will not become active until the LMCS register is accessed.

Any internally generated 20-bit address whose upper 16 bits are less than or equal to LMCS (with bits 0–5 "1") will cause $\overline{LCS}$ to be active. LMCS register bits R2–R0 are used to specify the READY mode for the area of memory defined by this chip-select register.

Mid-Range Memory $\overline{CS}$

The iAPX 186 provides four $\overline{MCS}$ lines which are active within a user-locatable memory block. This block can be located anywhere within the iAPX 186 1M byte memory address space exclusive of the areas defined by $\overline{UCS}$ and $\overline{LCS}$. Both the base address and size of this memory block are programmable.

The size of the memory block defined by the mid-range select lines, as shown in Table 9, is determined by bits 8–14 of the MPCS register (see Figure 13). This register is at location A8H in the internal control block. One and only one of bits 8–14 must be set at a time. Unpredictable operation of the $\overline{MCS}$ lines will otherwise occur. Each of the four chip-select lines is active for one of the four equal contiguous divisions of the mid-range block. Thus, if the total block size is 32K, each chip select is active for 8K of memory with $\overline{MCS0}$ being active for the first range and $\overline{MCS3}$ being active for the last range.

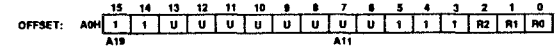
The EX and MS in MPCS relate to peripheral functionality as described in a later section

Table 9. MMCS Programming Values

Total Block Size	Individual Select Size	MMCS Bits 14–8
8K	2K	0000001B
16K	4K	0000010B
32K	8K	0000100B
64K	16K	0001000B
128K	32K	0010000B
256K	64K	0100000B
512K	128K	1000000B

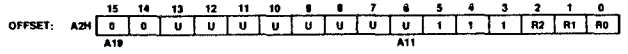
The base address of the mid-range memory block is defined by bits 15–9 of the MMCS register (see Figure 14). This register is at offset A6H in the internal control block. These bits correspond to bits A19–A13 of the 20-bit memory address. Bits A12–A0 of the base address are always 0. The base address may be set at any integer multiple of the size of the total memory block selected. For example, if the mid-range block size is 32K (or the size of the block for which each $\overline{MCS}$ line is active is 8K), the block could be located at 10000H or 18000H, but not at 14000H, since the first few integer multiples of a 32K memory block are 0H, 8000H, 10000H, 18000H, etc. After reset, the contents of both of these registers is undefined. However, none of the $\overline{MCS}$ lines will be active until both the MMCS and MPCS registers are accessed.

Figure 11. UMCS Register



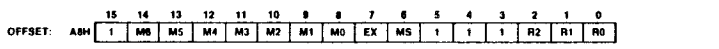
035518-11

Figure 12. LMCS Register



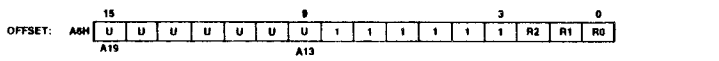
035518-12

Figure 13. MPCS Register



03551B-13

Figure 14. MMCS Register



03551B-14

MMCS bits R2–R0 specify READY mode of operation for all mid-range chip selects. All devices in mid-range memory must use the same number of WAIT states.

The 512K block size for the mid-range memory chip selects is a special case. When using 512K, the base address would have to be at either locations 00000H or 80000H. If it were to be programmed at 00000H when the $\overline{LCS}$ line was programmed, there would be an internal conflict between the $\overline{LCS}$ ready generation logic and the $\overline{MCS}$ ready generation logic. Likewise, if the base address were programmed at 80000H, there would be a conflict with the $\overline{UCS}$ ready generation logic. Since the $\overline{LCS}$ chip-select line does not become active until programmed, while the $\overline{UCS}$ line is active at reset, the memory base can be set only at 00000H. If this base address is selected, however, the $\overline{LCS}$ range must not be programmed.

Peripheral Chip Selects

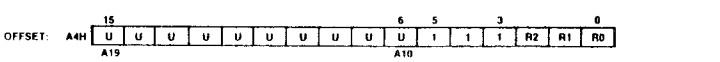
The iAPX 186 can generate chip selects for up to seven peripheral devices. These chip selects are active for seven contiguous blocks of 128 bytes above a programmable base address. This base address may be located in either memory or I/O space.

Seven $\overline{CS}$ lines called $\overline{PCS0}$–6 are generated by the iAPX 186. The base address is user-programmable; however it can only be a multiple of 1K bytes, i.e., the least significant 10 bits of the starting address are always 0.

$\overline{PCS5}$ and $\overline{PCS6}$ can also be programmed to provide latched address bits A1, A2. If so programmed, they cannot be used as peripheral selects. These outputs can be connected directly to the A0, A1 pins used for selecting internal registers of 8-bit peripheral chips. This scheme simplifies the hardware interface because the 8-bit registers of peripherals are simply treated as 16-bit registers located on even boundaries in I/O space or memory space where only the lower 8-bits of the register are significant: the upper 8-bits are "don't cares."

The starting address of the peripheral chip-select block is defined by the PACS register (see Figure 15). This register is located at offset A4H in the internal control block. Bits 15–6 of this register correspond to bits 19–10 of the 20-bit Programmable Base Address (PBA) of the peripheral chip-select block. Bits 9–0 of the PBA of the peripheral chip-select block are all zeros. If the chip-select block is located in I/O space, bits 12–15 must be programmed zero, since the I/O address is only 16 bits wide. Table 10 shows the address range of each peripheral chip select with respect to the PBA contained in PACS register.

Figure 15. PACS Register



03551B-15

The user should program bits 15–6 to correspond to the desired peripheral base location. PACS bits 0–2 are used to specify READY mode for $\overline{PCS0}$–$\overline{PCS3}$.

Table 10. PCS Address Ranges

PCS Line	Active between Locations
PCS0	PBA —PBA+127
PCS1	PBA+128 —PBA+255
PCS2	PBA+256 —PBA+383
PCS3	PBA+384 —PBA+511
PCS4	PBA+512 —PBA+639
PCS5	PBA+640 —PBA+767
PCS6	PBA+768 —PBA+895

The mode of operation of the peripheral chip selects is defined by the MPCS register (which is also used to set the size of the mid-range memory chip-select block, see Figure 16). This register is located at offset A8H in the internal control block. Bit 7 is used to select the function of $\overline{PCS5}$ and $\overline{PCS6}$, while bit 6 is used to select whether the peripheral chip selects are mapped into memory or I/O space. Table 11 describes the programming of these bits. After reset, the contents of both the MPCS and the PACS registers are undefined, however none of the PCS lines will be active until both of the MPCS and PACS registers are accessed.

Table 11. MS, EX Programming Values

Bit	Description
MS	1 = Peripherals mapped into memory space. 0 = Peripherals mapped into I/O space.
EX	0 = 5 $\overline{PCS}$ lines. A1, A2 provided. 1 = 7 $\overline{PCS}$ lines. A1, A2 are not provided.

MPCS bits 0–2 are used to specify READY mode for $\overline{PCS4}$–$\overline{PCS6}$ as outlined below.

READY Generation Logic

The iAPX 186 can generate a "READY" signal internally for each of the memory or peripheral $\overline{CS}$ lines. The number of WAIT states to be inserted for each peripheral or memory is programmable to provide 0–3 wait states for all accesses to the area for which the chip select is active. In addition, the iAPX 186 may be programmed to either ignore external READY for each chip-select range individually or to factor external READY with the integrated ready generator.

READY control consists of 3 bits for each $\overline{CS}$ line or group of lines generated by the iAPX 186. The interpretation of the ready bits is shown in Table 12.

Table 12. READY Bits Programming

R2	R1	R0	Number of WAIT States Generated
0	0	0	0 wait states, external RDY also used.
0	0	1	1 wait state inserted, external RDY also used.
0	1	0	2 wait states inserted, external RDY also used.
0	1	1	3 wait states inserted, external RDY also used.
1	0	0	0 wait states, external RDY ignored.
1	0	1	1 wait state inserted, external RDY ignored.
1	1	0	2 wait states inserted, external RDY ignored.
1	1	1	3 wait states inserted, external RDY ignored.

The internal ready generator operates in parallel with external READY, not in series if the external READY is used (R2 = 0). This means, for example, if the internal generator is set to insert two wait states, but activity on the external READY lines will insert four wait states, the processor will only insert four wait states, not six. This is because the two wait states generated by the internal generator overlapped the first two wait states generated by the external ready signal. Note that the external ARDY and SRDY lines are always ignored during cycles accessing internal peripherals.

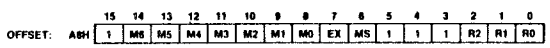
R2–R0 of each control word specifies the READY mode for the corresponding block, with the exception of the peripheral chip selects: R2–R0 of PACS set the $\overline{PCS0}$–3 READY mode, R2–R0 of MPCS set the $\overline{PCS4}$–6 READY mode.

Chip Select/Ready Logic and Reset

Upon reset, the Chip-Select/Ready Logic will perform the following actions:

- All chip-select outputs will be driven HIGH.
- Upon leaving RESET, the $\overline{UCS}$ line will be programmed to provide chip selects to a 1K block with the accompanying READY control bits set at 011 to

Figure 16. MPCS Register



03551B-16

allow the maximum number of internal wait states in conjunction with external Ready consideration (i.e., UMCS resets to FFFBH).

- No other chip select or READY control registers have any predefined values after RESET. They will not become active until the CPU accesses their control registers. Both the PACS and MPCS registers must be accessed before the $\overline{PCS}$ lines will become active.

DMA CHANNELS

The 80186 DMA controller provides two independent high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., Memory to I/O) or within the same space (e.g., Memory to Memory or I/O to I/O). Data can be transferred either in bytes (8 bits) or in words (16 bits) to or from even or odd addresses. Each DMA channel maintains both a 20-bit source and destination pointer which can be optionally incremented or decremented after each data transfer (by one or two depending on byte or word transfers). Each data transfer consumes 2 bus cycles (a minimum of 8 clocks), one cycle to fetch data and the other to store data. This provides a maximum data transfer rate of one Mword/sec or 2 MBytes/sec.

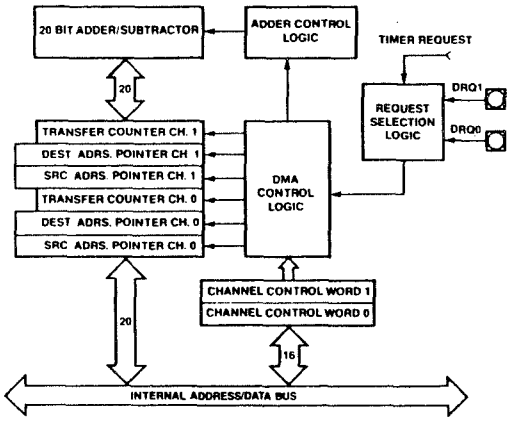
DMA Operation

Each channel has six registers in the control block which define each channel's specific operation. The control registers consist of a 20-bit Source pointer (2 words), a 20-bit Destination pointer (2 words), a 16-bit Transfer Counter, and a 16-bit Control Word. The format of the DMA Control Blocks is shown in Table 13. The Transfer Count Register (TC) specifies the number of DMA transfers to be performed. Up to 64K byte or word transfers can be performed with automatic termination. The Control Word defines the channel's operation (see Figure 18). All registers may be modified or altered during any DMA activity. Any changes made to these registers will be reflected immediately in DMA operation.

Table 13. DMA Control Block Format

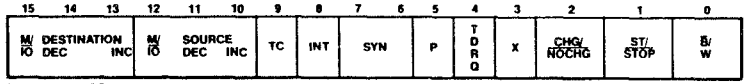
Register Name	Register Address	
	Ch. 0	Ch. 1
Control Word	CAH	DAH
Transfer Count	C8H	D8H
Destination Pointer (upper 4 bits)	C6H	D6H
Destination Pointer	C4H	D4H
Source Pointer (upper 4 bits)	C2H	D2H
Source Pointer	C0H	D0H

Figure 17. DMA Unit Block Diagram



035518-17

Figure 18. DMA Control Register



X = DON'T CARE.

035518-18

DMA Channel Control Word Register

Each DMA Channel Control Word determines the mode of operation for the particular 80186 DMA channel. This register specifies:

- the mode of synchronization;
- whether bytes or words will be transferred;
- whether interrupts will be generated after the last transfer;
- whether DMA activity will cease after a programmed number of DMA cycles;
- the relative priority of the DMA channel with respect to the other DMA channel;
- whether the source pointer will be incremented, decremented, or maintained constant after each transfer;
- whether the source pointer addresses memory or I/O space;
- whether the destination pointer will be incremented, decremented, or maintained constant after each transfer; and
- whether the destination pointer will address memory or I/O space.

The DMA channel control registers may be changed while the channel is operating. However, any changes made during operation will affect the current DMA transfer.

DMA Control Word Bit Descriptions

$\overline{B}$/W: Byte/Word (0/1) Transfers.

ST/$\overline{STOP}$: Start/stop (1/0) Channel.

CHG/$\overline{NOCHG}$: Change/Do not change (1/0) ST/$\overline{STOP}$ bit. If this bit is set when writing to the control word, the ST/$\overline{STOP}$ bit will be programmed by the write to the control word. If this bit is cleared when writing the control word, the ST/$\overline{STOP}$ bit will not be altered. This bit is not stored; it will always be a 0 on read.

INT: Enable Interrupts to CPU on byte count termination.

TC: If set, DMA will terminate when the contents of the Transfer Count register reach zero. The ST/$\overline{STOP}$ bit will also be reset at this point if TC is set. If this bit is cleared, the DMA unit will decrement the transfer count register for each DMA cycle, but the DMA transfer will not stop when the contents of the TC register reach zero.

SYN: (2 bits)
- 00 No synchronization.
  NOTE: The ST bit will be cleared automatically when the contents of the TC register reach zero regardless of the state of the TC bit.
- 01 Source synchronization.
- 10 Destination synchronization.
- 11 Unused.

SOURCE:INC Increment source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Source pointer is in M/IO space (1/0).

DEC Decrement source pointer by 1 or 2 (depends on $\overline{B}$/W) after each transfer.

DEST: INC Increment destination pointer by 1 or 2 ($\overline{B}$/W) after each transfer.

M/$\overline{IO}$ Destination pointer is in M/IO space (1/0).

DEC Decrement destination pointer by 1 or 2 (depending on $\overline{B}$/W) after each transfer.

P Channel priority—relative to other channel.
- 0 low priority.
- 1 high priority.

Channels will alternate cycles if both set at same priority level.

TDRQ — 0: Disable DMA requests from timer 2.
1: Enable DMA requests from timer 2.

Bit 3 — Bit 3 is not used.

If both INC and DEC are specified for the same pointer, the pointer will remain constant after each cycle.

DMA Destination and Source Pointer Registers

Each DMA channel maintains a 20-bit source and a 20-bit destination pointer. Each of these pointers takes up two full 16-bit registers in the peripheral control block. The lower four bits of the upper register contain the upper four bits of the 20-bit physical address (see Figure 18a). These pointers may be individually incremented or decremented after each transfer. If word transfers are performed the pointer is incremented or decremented by two. Each pointer may point into either memory or I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the pointer registers. Higher transfer rates can be obtained if all word transfers are performed to even addresses, since this will allow data to be accessed in a single memory access.

DMA Transfer Count Register

Each DMA channel maintains a 16-bit transfer count register (TC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control Register. If the TC bit in the DMA control word is set, however, DMA activity will terminate when the transfer count register reaches zero.

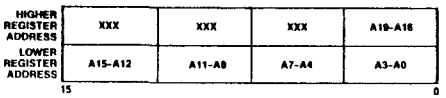
DMA Requests

Data transfers may be either source or destination synchronized, that is either the source of the data or the destination of the data may request the data transfer. In addition, DMA transfers may be unsynchronized; that is, the transfer will take place continually until the correct number of transfers has occurred. When source or unsynchronized transfers are performed, the DMA channel may begin another transfer immediately after the end of a previous DMA transfer. This allows a complete transfer to take place every 2 bus cycles or eight clock cycles (assuming no wait states). No prefetching occurs when destination synchronization is performed, however. Data will not be fetched from the source address until the destination device signals that it is ready to receive it. When destination synchronized transfers are requested, the DMA controller will relinquish control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle will begin after two processor clocks. This is done to allow the destination device time to remove its request if another transfer is not desired. Since the DMA controller will relinquish the bus, the CPU can initiate a bus cycle. As a result, a complete bus cycle will often be inserted between destination synchronized transfers. These lead to the maximum DMA transfer rates shown in Table 14.

Table 14. Maximum DMA Transfer Rates

Type of Synchronization Selected	CPU Running	CPU Halted
Unsynchronized	2MBytes/sec	2MBytes/sec
Source Synch	2MBytes/sec	2MBytes/sec
Destination Synch	1.3MBytes/sec	1.5MBytes/sec

Figure 18a. DMA Memory Pointer Register Format



0355 18 19

DMA Acknowledge

No explicit DMA acknowledge pulse is provided. Since both source and destination pointers are maintained, a read from a requesting source, or a write to a requesting destination, should be used as the DMA acknowledge signal. Since the chip-select lines can be programmed to be active for a given block of memory or I/O space, and the DMA pointers can be programmed to point to the same given block, a chip-select line could be used to indicate a DMA acknowledge.

DMA Priority

The DMA channels may be programmed such that one channel is always given priority over the other, or they may be programmed such as to alternate cycles when both have DMA requests pending. DMA cycles always have priority over internal CPU cycles except between locked memory accesses or word accesses the odd memory locations; however, an external bus hold takes priority over an internal DMA cycle. Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time will suffer during sequences of continuous DMA cycles. An NMI request, however, will cause all internal DMA activity to halt. This allows the CPU to quickly respond to the NMI request.

DMA Programming

DMA cycles will occur whenever the ST/STOP bit of the Control Register is set. If synchronized transfers are programmed, a DRQ must also have been generated. Therefore, the source and destination transfer pointers, and the transfer count register (if used) must be programmed before this bit is set.

Each DMA register may be modified while the channel is operating. If the CHG/NOCHG bit is cleared when the control register is written, the ST/STOP bit of the control register will not be modified by the write. If multiple channel registers are modified, it is recommended that a LOCKED string transfer be used to prevent a DMA transfer from occurring between updates to the channel registers.

DMA Channels and Reset

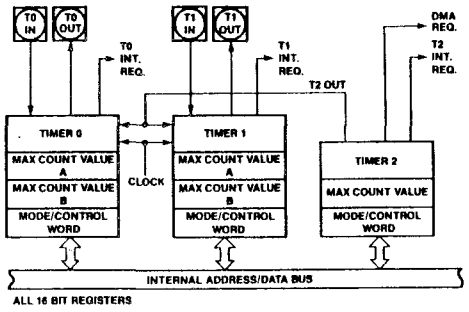
Upon RESET, the DMA channels will perform the following actions:

- The Start/Stop bit for each channel will be reset to STOP.
- Any transfer in progress is aborted.

TIMERS

The 80186 provides three internal 16-bit programmable timers (see Figure 19). Two of these are highly flexible and are connected to four external pins (2 per timer). They can be used to count external events, time external events, generate nonrepetitive waveforms, etc. The third timer is not connected to any external pins, and is useful for real-time coding and time delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source.

Figure 19. Timer Block Diagram



0355 18 20

Timer Operation

The timers are controlled by 11 16-bit registers in the internal peripheral control block. The configuration of these registers is shown in Table 15. The count register contains the current value of the timer. It can be read or written at any time independent of whether the timer is running or not. The value of this register will be incremented for each timer event. Each of the timers is equipped with a MAX COUNT register, which defines the maximum count the timer will reach. After reaching the MAX COUNT register value, the timer count value will reset to zero during that same clock, i.e., the maximum count value is never stored in the count register itself. Timers 0 and 1 are, in addition, equipped with a second MAX COUNT register, which enables the timers to alternate their count between two different MAX COUNT values programmed by the user. If a single MAX COUNT register is used, the timer output pin will switch LOW for a single clock, 2 clocks after the maximum count value has been reached. In the dual MAX COUNT register mode, the output pin will indicate which MAX COUNT register is currently in use, thus allowing nearly complete freedom in selecting waveform duty cycles. For the timers with two MAX COUNT registers, the RIU bit in the control register determines which is used for the comparison.

Each timer gets serviced every fourth CPU-clock cycle, and thus can operate at speeds up to one-quarter the internal clock frequency (one-eighth the crystal rate). External clocking of the timers may be done at up to a rate of one-quarter of the internal CPU-clock rate (2 MHz for an 8 MHz CPU clock). Due to internal synchronization and pipelining of the timer circuitry, a timer output may take up to 6 clocks to respond to any individual clock or gate input.

Since the count registers and the maximum count registers are all 16 bits wide, 16 bits of resolution are provided. Any Read or Write access to the timers will add one wait state to the minimum four-clock bus cycle. However, this is needed to synchronize and coordinate the internal data flows between the internal timers and the internal bus.

The timers have several programmable options.

- All three timers can be set to halt or continue on a terminal count.
- Timers 0 and 1 can select between internal and external clocks, alternate between MAX COUNT registers and be set to retrigger on external events.
- The timers may be programmed to cause an interrupt on terminal count.

These options are selectable via the timer mode/control word.

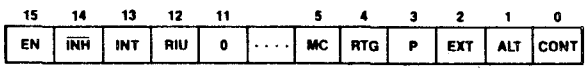
Timer Mode/Control Register

The mode/control register (see Figure 20) allows the user to program the specific mode of operation or check the current programmed status for any of the three integrated timers.

Table 15. Timer Control Block Format

Register Name	Register Offset		
	Tmr. 0	Tmr. 1	Tmr. 2
Mode/Control Word	56H	5EH	66H
Max Count B	54H	5CH	not present
Max Count A	52H	5AH	62H
Count Register	50H	58H	60H

Figure 20. Timer Mode/Control Register



ALT:
The ALT bit determines which of two MAX COUNT registers is used for count comparison. If ALT = 0, register A for that timer is always used, while if ALT = 1, the comparison will alternate between register A and register B when each maximum count is reached. This alternation allows the user to change one MAX COUNT register while the other is being used, and thus provides a method of generating non-repetitive waveforms. Square waves and pulse outputs of any duty cycle are a subset of available signals obtained by not changing the final count registers. The ALT bit also determines the function of the timer output pin. If ALT is zero, the output pin will go LOW for one clock, the clock after the maximum count is reached. If ALT is one, the output pin will reflect the current MAX COUNT register being used (0/1 for B/A).

CONT:
Setting the CONT bit causes the associated timer to run continuously, while resetting it causes the timer to halt upon maximum count. If CONT = 0 and ALT = 1, the timer will count to the MAX COUNT register A value, reset, count to the register B value, reset, and halt.

EXT:
The external bit selects between internal and external clocking for the timer. The external signal may be asynchronous with respect to the 80186 clock. If this bit is set, the timer will count LOW-to-HIGH transitions on the input pin. If cleared, it will count an internal clock while using the input pin for control. In this mode, the function of the external pin is defined by the RTG bit. The maximum input to output transition latency time may be as much as 6 clocks. However, clock inputs may be pipelined as closely together as every 4 clocks without losing clock pulses.

P:
The prescaler bit is ignored unless internal clocking has been selected (EXT = 0). If the P bit is a zero, the timer will count at one-fourth the internal CPU clock rate. If the P bit is a one, the output of timer 2 will be used as a clock for the timer. Note that the user must initialize and start timer 2 to obtain the prescaled clock.

RTG:
Retrigger bit is only active for internal clocking (EXT = 0). In this case it determines the control function provided by the input pin.

If RTG = 0, the input level gates the internal clock on and off. If the input pin is HIGH, the timer will count; if the input pin is LOW, the timer will hold its value. As indicated previously, the input signal may be asynchronous with respect to the 80186 clock.

When RTG = 1, the input pin detects LOW-to-HIGH transitions. The first such transition starts the timer running, clearing the timer value to zero on the first clock, and then incrementing thereafter. Further transitions on the input pin will again reset the timer to zero, from which it will start counting up again. If CONT = 0, when the timer has reached maximum count, the EN bit will be cleared, inhibiting further timer activity.

EN:
The enable bit provides programmer control over the timer's RUN/HALT status. When set, the timer is enabled to increment subject to the input pin constraints in the internal clock mode (discussed previously). When cleared, the timer will be inhibited from counting. All input pin transitions during the time EN is zero will be ignored. If CONT is zero, the EN bit is automatically cleared upon maximum count.

INH:
The inhibit bit allows for selective updating of the enable (EN) bit. If INH is a one during the write to the mode/control word, then the state of the EN bit will be modified by the write. If INH is a zero during the write, the EN bit will be unaffected by the operation. This bit is not stored; it will always be a 0 on a read.

INT:
When set, the INT bit enables interrupts from the timer, which will be generated on every terminal count. If the timer is configured in dual MAX COUNT register mode, an interrupt will be generated each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. If this enable bit is cleared after the interrupt request has been generated, but before a pending interrupt is serviced, the interrupt request will still be in force. (The request is latched in the Interrupt Controller.)

MC:
The Maximum Count bit is set whenever the timer reaches its final maximum count value. If the timer is configured in dual MAX COUNT register mode, this bit will be set each time the value in MAX COUNT register A is reached, and each time the value in MAX COUNT register B is reached. This bit is set regardless of the timer's interrupt-enable bit. The MC bit gives the user the ability to monitor timer status through software instead of through interrupts.

RIU:
The Register In Use bit indicates which MAX COUNT register is currently being used for comparison to the timer count value. A zero value indicates register A. The RIU bit cannot be written, i.e., its value is not affected when the control register is written. It is always cleared when the ALT bit is zero.

Not all mode bits are provided for timer 2. Certain bits are hardwired as indicated below:

ALT = 0, EXT = 0, P = 0, RTG = 0, RIU = 0

Count Registers

Each of the three timers has a 16-bit count register. The current contents of this register may be read or written by the processor at any time. If the register is written into while the timer is counting, the new value will take effect in the current count cycle.

Max Count Registers

Timers 0 and 1 have two MAX COUNT registers, while timer 2 has a single MAX COUNT register. These contain the number of events the timer will count. In timers 0 and 1, the MAX COUNT register used can alternate between the two max count values whenever the current maximum count is reached. The condition which causes a timer to reset is equivalent between the current count value and the max count being used. This means that if the count is changed to be above the max count value, or if the max count value is changed to be below the current value, the timer will not reset to zero, but rather will count to its maximum value, "wrap around" to zero, then count until the max count is reached.

Timers and Reset

Upon RESET, the Timers will perform the following actions:

- All EN (Enable) bits are reset preventing timer counting.
- All SEL (Select) bits are reset to zero. This selects MAX COUNT register A, resulting in the Timer Out pins going HIGH upon RESET.

INTERRUPT CONTROLLER

The 80186 can receive interrupts from a number of sources, both internal and external. The internal interrupt controller serves to merge these requests on a priority basis, for individual service by the CPU.

Internal interrupt sources (Timers and DMA channels) can be disabled by their own control registers or by mask bits within the interrupt controller. The 80186 interrupt controller has its own control registers that set the mode of operation for the controller.

The interrupt controller will resolve priority among requests that are pending simultaneously. Nesting is provided so interrupt service routines for lower priority interrupts may themselves be interrupted by higher priority interrupts. A block diagram of the interrupt controller is shown in Figure 21.

The interrupt controller has a special iRMX 86 compatibility mode that allows the use of the 80186 within the iRMX 86 operating system interrupt structure. The controller is set in this mode by setting bit 14 in the peripheral control block relocation register (see iRMX 86 Compatibility Mode section). In this mode, the internal 80186 interrupt controller functions as a "slave" controller to an external "master" controller. Special initialization software must be included to properly set up the 80186 interrupt controller in iRMX 86 mode.

NON-iRMX MODE OPERATION

Interrupt Controller External Interface

For external interrupt sources, five dedicated pins are provided. One of these pins is dedicated to NMI, non-maskable interrupt. This is typically used for power-fail interrupts, etc. The other four pins may function either as four interrupt input lines with internally generated interrupt vectors, as an interrupt line and an interrupt acknowledge line (called the "cascade mode") along with two other input lines with internally generated interrupt vectors, or as two interrupt input lines and two dedicated interrupt acknowledge ouput lines. When the interrupt lines are configured in cascade mode, the 80186 interrupt controller will not generate internal interrupt vectors.

External sources in the cascade mode use externally generated interrupt vectors. When an interrupt is acknowledged, two $\overline{INTA}$ cycles are initiated and the vector is read into the 80186 on the second cycle. The capability to interface to external 8259A programmable interrupt controllers is thus provided when the inputs are configured in cascade mode.

Interrupt Controller Modes of Operation

The basic modes of operation of the interrupt controller in non-iRMX mode are similar to the 8259A. The interrupt controller responds identically to internal interrupts in all three modes: the difference is only in the interpretation of function of the four external interrupt pins. The interrupt controller is set into one of these three modes by programming the correct bits in the INT0 and INT1 control registers. The modes of interrupt controller operation are as follows:

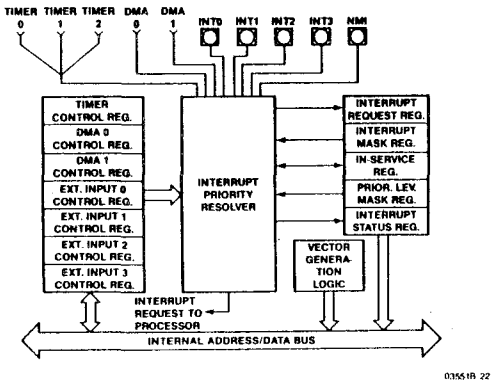
Fully Nested Mode
When in the fully nested mode four pins are used as direct interrupt requests. The vectors for these four inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set, no interrupt will be generated by the interrupt controller. In addition, if another interrupt request occurs from the same interrupt source while the inservice bit is set, no interrupt will be generated by the interrupt controller. This allows interrupt service routines to operate with interrupts enabled without being themselves interrupted by lower-priority interrupts. Since interrupts are enabled, higher-priority interrupts will be serviced.

When a service routine is completed, the proper IS bit must be reset by writing the proper pattern to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. An EOI command is issued at the end of the service routine just before the issuance of the return from interrupt instruction. If the fully nested structure has been upheld, the next highest-priority source with its IS bit set is then serviced.

Cascade Mode
The 80186 has four interrupt pins and two of them have dual functions. In the fully nested mode the four pins are used as direct interrupt inputs and the corresponding vectors are generated internally. In the cascade mode, the four pins are configured into interrupt input-dedicated acknowledge signal pairs. The interconnection is shown in Figure 22. INT0 is an interrupt input interfaced to an 8259A, while INT2/$\overline{INTA0}$ serves as the dedicated interrupt acknowledge signal to that peripheral. The same is true for INT1 and INT3/$\overline{INTA1}$. Each pair can selectively be placed in the cascade or non-cascade mode by programming the proper value into INT0 and INT1 control registers. The use of the dedicated acknowledge signals eliminates the need for the use of external logic to generate $\overline{INTA}$ and device select signals.

The primary cascade mode allows the capability to serve up to 128 external interrupt sources through the use of external master and slave 8259As. Three levels of priority are created, requiring priority resolution in the 80186 interrupt controller, the master 8259As, and the slave 8259As. If an external interrupt is serviced, one IS bit is set at each of these levels. When the interrupt service routine is completed, up to three end-of-interrupt commands must be issued by the programmer.

Figure 21. Interrupt Controller Block Diagram



Special Fully Nested Mode
This mode is entered by setting the SFNM bit in INT0 or INT1 control register. It enables complete nestability with external 8259A masters. Normally, an interrupt request from an interrupt source will not be recognized unless the in-service bit for that source is reset. If more than one interrupt source is connected to an external interrupt controller, all of the interrupts will be funneled through the same 80186 interrupt request pin. As a result, if the external interrupt controller receives a higher-priority interrupt, its interrupt will not be recognized by the 80186 controller until the 80186 in-service bit is reset. In special fully nested mode, the 80186 interrupt controller will allow interrupts from an external pin regardless of the state of the in-service bit for an interrupt source in order to allow multiple interrupts from a single pin. An in-service bit will continue to be set, however, to inhibit interrupts from other lower-priority 80186 interrupt sources.

Special procedures should be followed when resetting IS bits at the end of interrupt service routines. Software polling of the external master's IS register is required to determine if there is more than one bit set. If so, the IS bit in the 80186 remains active and the next interrupt service routine is entered.

Operation in a Polled Environment

The controller may be used in a polled mode if interrupts are undesirable. When polling, the processor disables interrupts and then polls the interrupt controller whenever it is convenient. Polling the interrupt controller is accomplished by reading the Poll Word (Figure 9). Bit 15 in the poll word indicates to the processor that an interrupt of high enough priority is requesting service. Bits 0–4 indicate to the processor the type vector of the highest-priority source requesting service. Reading the Poll Word causes the In-Service bit of the highest-priority source to be set.

It is desirable to be able to read the Poll Word information without guaranteeing service of any pending interrupt, i.e., not set the indicated in-service bit. The 80186 provides a Poll Status Word in addition to the conventional Poll Word to allow this to be done. Poll Word information is duplicated in the Poll Status Word, but reading the Poll Status Word does not set the associated in-service bit. These words are located in two adjacent memory locations in the register file.

Non-IRMX Mode Features

Programmable Priority
The user can program the interrupt sources into any of eight different priority levels. The programming is done by placing a 3-bit priority level (0–7) in the control register of each interrupt source. (A source with a priority level of 4 has higher priority over all priority levels from 5 to 7. Priority registers containing values lower than 4 have greater priority.) All interrupt sources have preprogrammed default priority levels (see Table 4).

If two requests with the same programmed priority level are pending at once, the priority ordering scheme shown in Table 4 is used. If the serviced interrupt routine reenables interrupts, it allows other requests to be serviced.

End-of-Interrupt Command
The end-of-interrupt (EOI) command is used by the programmer to reset the In-Service (IS) bit when an interrupt service routine is completed. The EOI command is issued by writing the proper pattern to the EOI register. There are two types of EOI commands, specific and nonspecific. The nonspecific command does not specify which IS bit is reset. When issued, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine. A specific EOI command requires that the programmer send the interrupt vector type to the interrupt controller indicating which source's IS bit is to be reset. This command is used when the fully nested structure has been disturbed or the highest priority IS bit that was set does not belong to the service routine in progress.

Trigger Mode
The four external interrupt pins can be programmed in either edge- or level-trigger mode. The control register for each external source has a level-trigger mode (LTM) bit. All interrupt inputs are active HIGH. In the edge sense mode or the level-trigger mode, the interrupt request must remain active (HIGH) until the interrupt request is acknowledged by the 80186 CPU. In the edge-sense mode, if the level remains high after the interrupt is acknowledged, the input is disabled and no further requests will be generated. The input level must go LOW for at least one clock cycle to reenable the input. In the level-trigger mode, no such provision is made: holding the interrupt input HIGH will cause continuous interrupt requests.

Interrupt Vectoring
The 80186 Interrupt Controller will generate interrupt vectors for the integrated DMA channels and the integrated Timers. In addition, the Interrupt Controller will generate interrupt vectors for the external interrupt lines if they are not configured in Cascade or Special Fully Nested Mode. The interrupt vectors generated are fixed and cannot be changed (see Table 4).

Interrupt Controller Registers

The Interrupt Controller register model is shown in Figure 23. It contains 15 registers. All registers can both be read or written unless specified otherwise.

In-Service Register
This register can be read from or written into. The format is shown in Figure 24. It contains the In-Service bit for each of the interrupt sources. The In-Service bit is set to indicate that a source's service routine is in progress. When an In-Service bit is set, the interrupt controller will not generate interrupts to the CPU when it receives interrupt requests from devices with a lower programmed priority level. The TMR bit is the In-Service bit for all three timers; the D0 and D1 bits are the In-Service bits for the two DMA channels; the I0–I3 are the In-Service bits for the external interrupt pins. The IS bit is set when the processor acknowledges an interrupt request either by an interrupt acknowledge or by reading the poll register. The IS bit is reset at the end of the interrupt service routine by an end-of-interrupt command issued by the CPU.

Interrupt Request Register
The internal interrupt sources have interrupt request bits inside the interrupt controller. The format of this register is shown in Figure 24. A read from this register yields the status of these bits. The TMR bit is the logical OR of all timer interrupt requests. D0 and D1 are the interrupt request bits for the DMA channels.

The state of the external interrupt input pins is also indicated. The state of the external interrupt pins is not a stored condition inside the interrupt controller, therefore the external interrupt bits cannot be written. The external interrupt request bits show exactly when an interrupt request is given to the interrupt controller, so if edge-triggered mode is selected, the bit in the register will be HIGH only after an inactive-to-active transition. For internal interrupt sources, the register bits are set when a request arrives and are reset when the processor acknowledges the requests.

Mask Register
This is a 16-bit register that contains a mask bit for each interrupt source. The format for this register is shown in Figure 24. A one in a bit position corresponding to a particular source serves to mask the source from generating interrupts. These mask bits are the exact same bits which are used in the individual control registers; programming a mask bit using the mask register will also change this bit in the individual control registers, and vice versa.

Figure 22. Cascade Mode Interrupt Connection

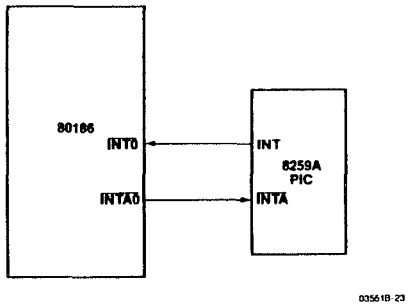


Figure 23. Interrupt Controller Registers (Non-iRMX 86 Mode)

	OFFSET
INT3 CONTROL REGISTER	3EH
INT2 CONTROL REGISTER	3CH
INT1 CONTROL REGISTER	3AH
INT0 CONTROL REGISTER	38H
DMA 1 CONTROL REGISTER	36H
DMA 0 CONTROL REGISTER	34H
TIMER CONTROL REGISTER	32H
INTERRUPT CONTROLLER STATUS REGISTER	30H
INTERRUPT REQUEST REGISTER	2EH
IN-SERVICE REGISTER	2CH
PRIORITY MASK REGISTER	2AH
MASK REGISTER	28H
POLL STATUS REGISTER	26H
POLL REGISTER	24H
EOI REGISTER	22H

03551B 24

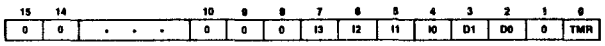
Priority Mask Register
This register is used to mask all interrupts below particular interrupt priority levels. The format of this register is shown in Figure 25. The code in the lower three bits of this register inhibits interrupts of priority lower (a higher priority number) than the code specified. For example, 100 written into this register masks interrupts of level five (101), six (110), and seven (111). The register is reset to seven (111) upon RESET so all interrupts are unmasked.

Interrupt Status Register
This register contains general interrupt controller status information. The format of this register is shown in Figure 26. The bits in the status register have the following functions:

DHLT: DMA Halt Transfer; setting this bit halts all DMA transfers. It is automatically set whenever a non-maskable interrupt occurs, and it is reset when an IRET instruction is executed. The purpose of this bit is to allow prompt service of all non-maskable interrupts. This bit may also be set by the CPU.

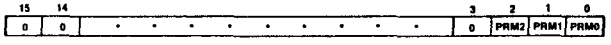
IRTx: These three bits represent the individual timer interrupt request bits. These bits are used to differentiate the timer interrupts, since the timer IR bit in the interrupt request register is the "OR" function of all timer interrupt requests. Note that setting any one of these three bits initiates an interrupt request to the interrupt controller.

Figure 24. In-Service, Interrupt Request, and Mask Register Formats



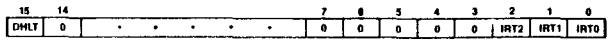
03551B 25

Figure 25. Priority Mask Register Format



03551B 26

Figure 26. Interrupt Status Register Format



03551B 27

Timer, DMA 0, 1; Control Registers
These registers are the control words for all the internal interrupt sources. The format for these registers is shown in Figure 27. The three bit positions PR0, PR1, and PR2 represent the programmable priority level of the interrupt source. The MSK bit inhibits interrupt requests from the interrupt source. The MSK bits in the individual control registers are the exact same bits as are in the Mask Register; modifying them in the individual control registers will also modify them in the Mask Register, and vice versa.

INT0–INT3 Control Registers
These registers are the control words for the four external input pins. Figure 28 shows the format of the INT0 and INT1 Control registers; Figure 29 shows the format of the INT2 and INT3 Control registers. In cascade mode or special fully nested mode, the control words for INT2 and INT3 are not used.

The bits in the various control registers are encoded as follows:

PRO-2: Priority programming information. Highest priority = 000, lowest priority = 111.

LTM: Level-trigger mode bit. 1 = level-triggered; 0 = edge-triggered. Interrupt input levels are active high. In level-triggered mode, an interrupt is generated whenever the external line is high. In edge-triggered mode, an interrupt will be generated only when this level is preceded by an inactive-to-active transition on the line. In both cases, the level must remain active until the interrupt is acknowledged.

MSK: Mask bit. 1 = mask; 0 = nonmask

C: Cascade mode bit. 1 = cascade; 0 = direct

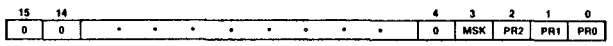
SFNM: Special fully nested mode bit. 1 = SFNM; 0 = normal nested mode

EOI Register
The end of the interrupt register is a command register which can only be written into. The format of this register is shown in Figure 30. It initiates an EOI command when written to by the 80186 CPU.

The bits in the EOI register are encoded as follows:

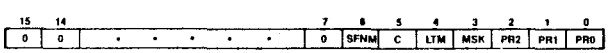
S_x: Encoded information that specifies an interrupt source vector type as shown in Table 4. For example, to reset the In-Service bit for DMA channel 0, these bits should be set to 01010, since the vector type for DMA channel 0 is 10. Note that to reset the single In-Service bit for any of the three timers, the vector type for timer 0 (8) should be written in this register.

Figure 27. Timer/DMA Control Register Formats



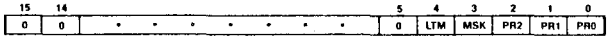
03551B 28

Figure 28. INT0/INT1 Control Register Formats



03551B 29

Figure 29. INT2/INT3 Control Register Formats



03551B 30

NSPEC/SPEC: A bit that determines the type of EOI command. Nonspecific = 1, Specific = 0.

Poll and Poll Status Registers

These registers contain polling information. The format of these registers is shown in Figure 31. They can only be read. Reading the Poll register constitutes a software poll. This will set the IS bit of the highest priority pending interrupt. Reading the poll status register will not set the IS bit of the highest priority pending interrupt; only the status of pending interrupts will be provided.

Encoding of the Poll and Poll Status register bits are as follows:

S_x: Encoded information that indicates the vector type of the highest priority interrupting source. Valid only when INTREQ = 1.

INTREQ: This bit determines if an interrupt request is present. Interrupt Request = 1; no Interrupt Request = 0.

iRMX 86 COMPATIBILITY MODE

This mode allows iRMX 86-80186 compatibility. The interrupt model of iRMX 86 requires one master and multiple slave 8259As in cascaded fashion. When iRMX mode is used, the internal 80186 interrupt controller will be used as a slave controller to an external master interrupt controller. The internal 80186 resources will be monitored through the internal interrupt controller, while the external controller functions as the system master interrupt controller.

Upon reset, the 80186 interrupt controller will be in the non-iRMX 86 mode of operation. To set the controller in the iRMX 86 mode, bit 14 of the Relocation Register should be set.

Because of pin limitations caused by the need to interface to an external 8259A master, the internal interrupt controller will no longer accept external inputs. There are however, enough 80186 interrupt controller inputs (internally) to dedicate one to each timer. In this mode, each timer interrupt source has its own mask bit, IS bit, and control word.

The iRMX 86 operating system requires peripherals to be assigned fixed priority levels. This is incompatible with the normal operation of the 80186 interrupt controller. Therefore, the initialization software must program the proper priority levels for each source. The required priority levels for the internal interrupt sources in iRMX mode are shown in Table 16.

Table 16. Internal Source Priority Level

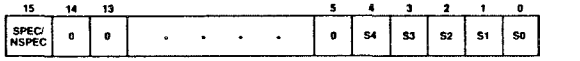
Priority Level	Interrupt Source
0	Timer 0
1	(reserved)
2	DMA 0
3	DMA 1
4	Timer 1
5	Timer 2

These level assignments must remain fixed in the iRMX 86 mode of operation.

iRMX 86 Mode External Interface

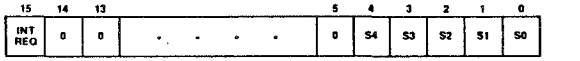
The configuration of the 80186 with respect to an external 8259A master is shown in Figure 32. The INT0 input is used as the 80186 CPU interrupt input. INT3 functions as an output to send the 80186 slave-interrupt-request to one of the 8 master-PIC-inputs.

Figure 30. EOI Register Format



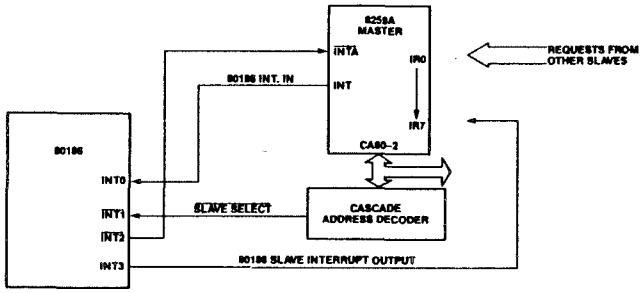
035518-31

Figure 31. Poll Register Format



035518-32

Figure 32. iRMX 86 Interrupt Controller Interconnection



035518-33

Correct master-slave interface requires decoding of the slave addresses (CAS0–2). Slave 8259As do this internally. Because of pin limitations, the 80186 slave address will have to be decoded externally. $\overline{INT1}$ is used as a slave-select input. Note that the slave vector address is transferred internally, but the READY input must be supplied externally.

$\overline{INT2}$ is used as an acknowledge output, suitable to drive the $\overline{INTA}$ input of an 8259A.

Interrupt Nesting

iRMX 86 mode operation allows nesting of interrupt requests. When an interrupt is acknowledged, the priority logic masks off all priority levels except those with equal or higher priority.

Vector Generation in the iRMX 86 Mode

Vector generation in iRMX mode is exactly like that of an 8259A slave. The interrupt controller generates an 8-bit vector which the CPU multiplies by four and uses as an address into a vector table. The significant five bits of the vector are user-programmable while the lower three bits are generated by the priority logic. These bits represent the encoding of the priority level requesting service. The significant five bits of the vector are programmed by writing to the Interrupt Vector register at offset 20H.

Specific End-of-Interrupt

In iRMX mode the specific EOI command operates to reset an in-service bit of a specific priority. The user supplies a 3-bit priority-level value that points to an in-service bit to be reset. The command is executed by writing the correct value in the Specific EOI register at offset 22H.

Interrupt Controller Registers in the iRMX 86 Mode

All control and command registers are located inside the internal peripheral control block. Figure 33 shows the offsets of these registers.

End-of-Interrupt Register

The end-of-interrupt register is a command register which can only be written. The format of this register is shown in Figure 34. It initiates an EOI command when written by the 80186 CPU.

The bits in the EOI register are encoded as follows:

L_x: Encoded value indicating the priority of the IS bit to be reset.

In-Service Register

This register can be read from or written into. It contains the in-service bit for each of the internal

interrupt sources. The format for this register is shown in Figure 35. Bit positions 2 and 3 correspond to the DMA channels; positions 0, 4, and 5 correspond to the integral timers. The source's IS bit is set when the processor acknowledges its interrupt request.

Interrupt Request Register
This register indicates which internal peripherals have interrupt requests pending. The format of this register is shown in Figure 35. The interrupt request bits are set when a request arrives from an internal source, and are reset when the processor acknowledges the request.

Mask Register
This register contains a mask bit for each interrupt source. The format for this register is shown in Figure 35. If the bit in this register corresponding to a particular interrupt source is set, any interrupts from that source will be masked. These mask bits are exactly the same bits which are used in the individual control registers, i.e., changing the state of a mask bit in this register will also change the state of the mask bit in the individual interrupt control register corresponding to the bit.

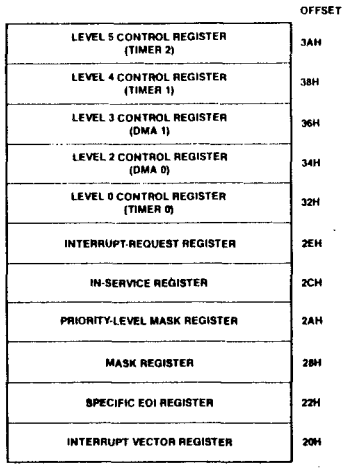
Control Registers
These registers are the control words for all the internal interrupt sources. The format of these registers is shown in Figure 36. Each of the timers and both of the DMA channels have their own Control Register.

The bits of the Control Registers are encoded as follows:

pr_x: 3-bit encoded field indicating a priority level for the source; note that each source must be programmed at specified levels.

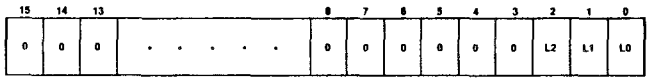
msk: mask bit for the priority level indicated by pr_x bits.

Figure 33. Interrupt Controller Registers (iRMX 86 Mode)



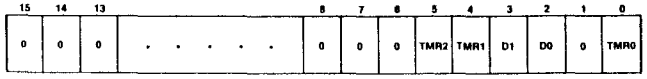
03551B-34

Figure 34. Specific EOI Register Format



03551B-35

Figure 35. In-Service, Interrupt Request, and Mask Register Format



03551B-36

Interrupt Vector Register
This register provides the upper five bits of the interrupt vector address. The format of this register is shown in Figure 37. The interrupt controller itself provides the lower three bits of the interrupt vector as determined by the priority level of the interrupt request.

The format of the bits in this register is:

t_x: 5-bit field indicating the upper five bits of the vector address.

Priority-Level Mask Register
This register indicates the lowest priority-level interrupt which will be serviced.

The encoding of the bits in this register is:

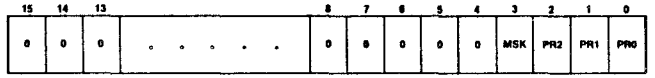
m_x: 3-bit encoded field indication priority-level value. All levels of lower priority will be masked.

Interrupt Controller and Reset

Upon RESET, the interrupt controller will perform the following actions:

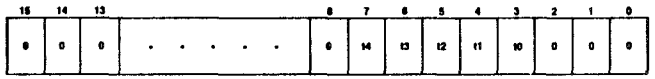
- All SFNM bits reset to 0, implying Fully Nested Mode.
- All PR bits in the various control registers set to 1. This places all sources at lowest priority (level 111).
- All LTM bits reset to 0, resulting in edge-sense mode.
- All Interrupt Service bits reset to 0.
- All Interrupt Request bits reset to 0.
- All MSK (Interrupt Mask) bits set to 1 (mask).
- All C (Cascade) bits reset to 0 (non-cascade).
- All PRM (Priority Mask) bits set to 1, implying no levels masked.
- Initialized to non-iRMX 86 mode.

Figure 36. Control Word Format



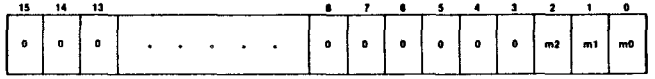
03551B-37

Figure 37. Interrupt Vector Register Format



03551B-38

Figure 38. Priority Level Mask Register



03551B-39

Figure 39. Typical iAPX 186 Computer

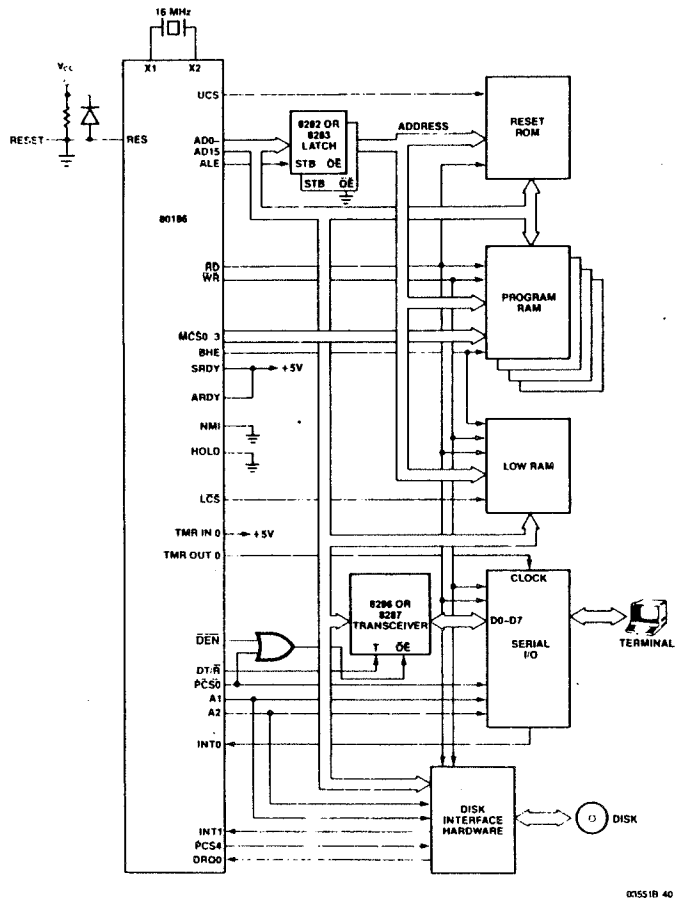
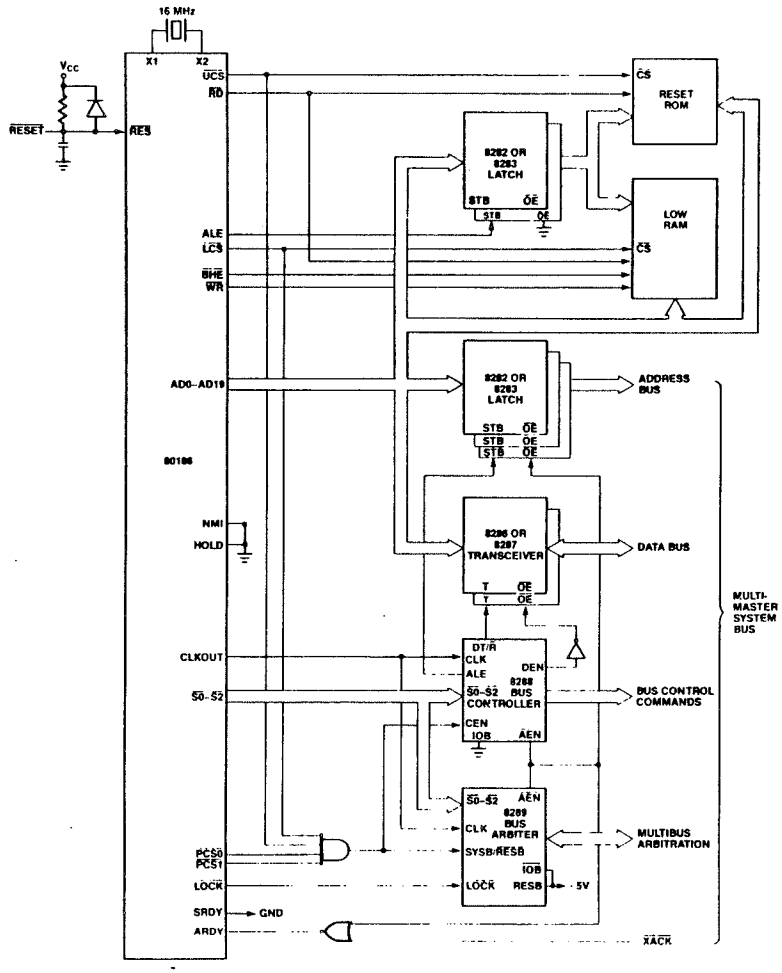


Figure 40. Typical iAPX 186 Multi-Master Bus Interface



PACKAGE

The 80186 is housed in a 68-pin, leadless JEDEC type A hermetic chip carrier. Figure 41 illustrates the package dimensions.

NOTE: The IDT 3M Textool 68-pin JEDEC Socket is required for ICE™ operation. See Figure 40 for details.

Figure 41. 80186 JEDEC Type A Package

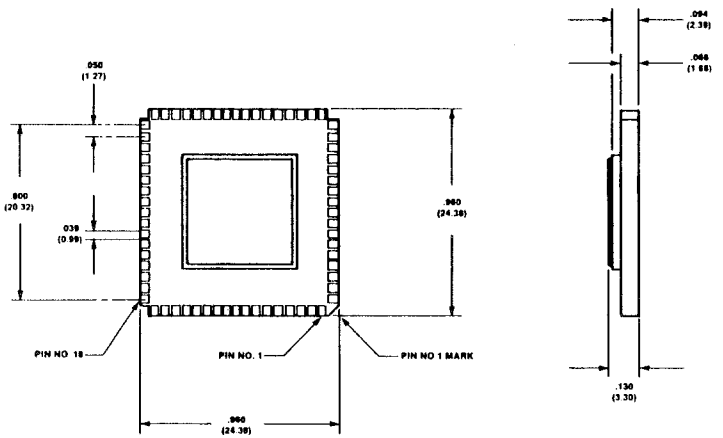
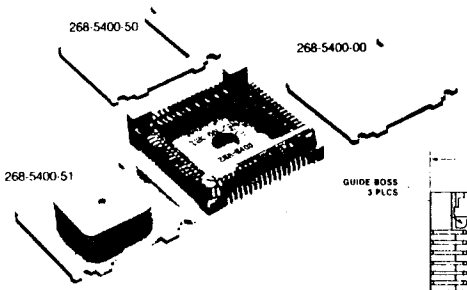
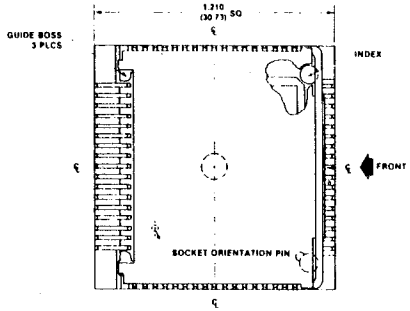
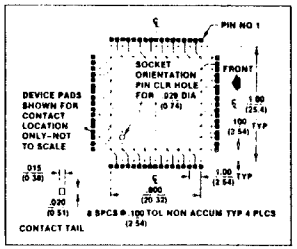


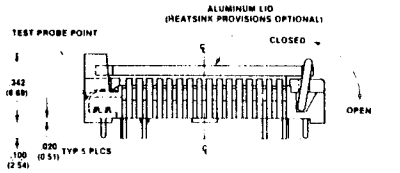
Figure 42. Textool 68 Lead Chip Carrier Socket





PC BOARD PATTERN





03551B-43

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to +150°C
Voltage on Any Pin with
Respect to Ground ..... −1.0V to +7V
Power Dissipation ..... 3 Watt

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

D.C. CHARACTERISTICS (T_A = 0°–70°C, V_{CC} = 5V ±10%)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.5	+0.8	Volts	
V_{IH}	Input High Voltage (All except X1 and $\overline{RES}$)	2.0	V_{CC} + 0.5	Volts	
V_{IH1}	Input High Voltage ($\overline{RES}$)	3.0	V_{CC} + 0.5	Volts	
V_{OL}	Output Low Voltage		0.45	Volts	I_a = 2.5 mA for $\overline{S0}$–$\overline{S2}$ I_a = 2.0 mA for all other outputs
V_{OH}	Output High Voltage		2.4	Volts	I_{oa} = −400 μA
I_{CC}	Power Supply Current		550	mA	T_A = 25°C
I_{LI}	Input Leakage Current		±10	μA	0V < V_{IN} < V_{CC}
I_{LO}	Output Leakage Current		±10	μA	0.45V < V_{OUT} < V_{CC}
V_{CLO}	Clock Output Low		0.6	Volts	I_a = 2.5 mA
V_{CHO}	Clock Output High	4.0		Volts	I_{oa} = −200 μA
V_{CLI}	Clock Input Low Voltage	−0.5	0.6	Volts	
V_{CHI}	Clock Input High Voltage	3.9	V_{CC} + 1.0	Volts	
C_{IN}	Input Capacitance		10	pF	
C_{IO}	I/O Capacitance		20	pF	

PIN TIMINGS

A.C. CHARACTERISTICS (T_A = 0°–70°C, V_{CC} = 5V ± 10%)

80186 Timing Requirements All Timings Measured At 1.5 Volts Unless Otherwise Noted.

Symbol	Parameter	Min.	Max.	Units	Test Conditions
TDVCL	Data in Setup (A/D)	20		ns	
TCLDX	Data in Hold (A/D)	10		ns	
TARYHCH	Asynchronous Ready (AREADY) active setup time*	20		ns	
TARYLCL	AREADY inactive setup time	35		ns	
TCHARYX	AREADY hold time	15		ns	
TSRYCL	Synchronous Ready (SREADY) transition setup time	35		ns	
TCLSRY	SREADY transition hold time	15		ns	
THVCL	HOLD Setup*	25		ns	
TINVCH	INTR, NMI, $\overline{TEST}$, TIMERIN, Setup*	25		ns	
TINVCL	DRQ0, DRQ1, Setup*	25		ns	

*To guarantee recognition at next clock

A.C. CHARACTERISTICS (Continued)

80186 Master Interface Timing Responses

		80186-3 (8 MHz)		80186-6 (6 MHz)			
Symbol	Parameter	Min.	Max.	Min.	Max.	Units	Test Conditions
TCLAV	Address Valid Delay	10	44	10	63	ns	C_L = 20-200 pF all outputs
TCLAX	Address Hold	10		10		ns	
TCLAZ	Address Float Delay	TCLAX	35	TCLAX	44	ns	
TCHCZ	Command Lines Float Delay		45		56	ns	
TCHCV	Command Lines Valid Delay (after float)		55		TCHCL	ns	
TLHLL	ALE Width	TCLCL-35		TCLCL-35		ns	
TCHLH	ALE Active Delay		35		44	ns	
TCHLL	ALE Inactive Delay		35		44	ns	
TLLAX	Address Hold to ALE Inactive	TCHCL-25		TCHCL-30		ns	
TCLDV	Data Valid Delay	10	44	10	55	ns	
TCLDOX	Data Hold Time	10		10		ns	
TWHDX	Data Hold after WR	TCLCL-40		TCLCL-50		ns	
TCVCTV	Control Active Delay1	10	70	10	87	ns	
TCHCTV	Control Active Delay2	10	55	10	TCHCL	ns	
TCVCTX	Control Inactive Delay	10	55	10	TCLCH	ns	
TCVDEX	$\overline{DEN}$ Inactive Delay (Non-Write Cycle)		70		87	ns	
TAZRL	Address Float to $\overline{RD}$ Active	0		0		ns	
TCLRL	$\overline{RD}$ Active Delay	10	70	10	87	ns	
TCLRH	$\overline{RD}$ Inactive Delay	10	55	10	TCLCH	ns	
TRHAV	$\overline{RD}$ Inactive to Address Active	TCLCL-40		TCLCL-50		ns	
TCLHAV	HLDA Valid Delay	10	50	10	67	ns	
TRLRH	$\overline{RD}$ Width	2TCLCL-50		2TCLCL-50		ns	
TWLWH	$\overline{WR}$ Width	2TCLCL-40		2TCLCL-40		ns	
TAVAL	Address Valid to ALE Low	TCLCH-25		TCLCH-45		ns	
TCHSV	Status Active Delay	10	55	10	TCHCL	ns	
TCLSH	Status Inactive Delay	10	55	10	TCLCH	ns	
TCLTMV	Timer Output Delay		60		75	ns	100 pF max
TCLRO	Reset Delay		60		75	ns	
TCHQSV	Queue Status Delay		35		44	ns	
TCHDX	Data Hold Time	10		10		ns	

80186 Chip-Select Timing Responses

Symbol	Parameter	Min.	Max.	Min.	Max.	Units	Test Conditions
TCLCSV	Chip-Select Active Delay		66		80	ns	
TCXCSX	Chip-Select Hold from Command Inactive	35		35		ns	
TCHCSX	Chip-Select Inactive Delay	10	35	10	47	ns	

A.C. CHARACTERISTICS (Continued)

80186 CLKIN Requirements

		80186-3 (8 MHz)		80186-6 (6 MHz)			
Symbol	**Parameter**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**	**Test Conditions**
TCKIN	CLKIN Period	62.5	250	83	250	ns	
TCKHL	CLKIN Fall Time		10		10	ns	3.5 to 1.0 volts
TCKLH	CLKIN Rise Time		10		10	ns	1.0 to 3.5 volts
TCLCK	CLKIN Low Time	25		33		ns	1.5 volts
TCHCK	CLKIN High Time	25		33		ns	1.5 volts

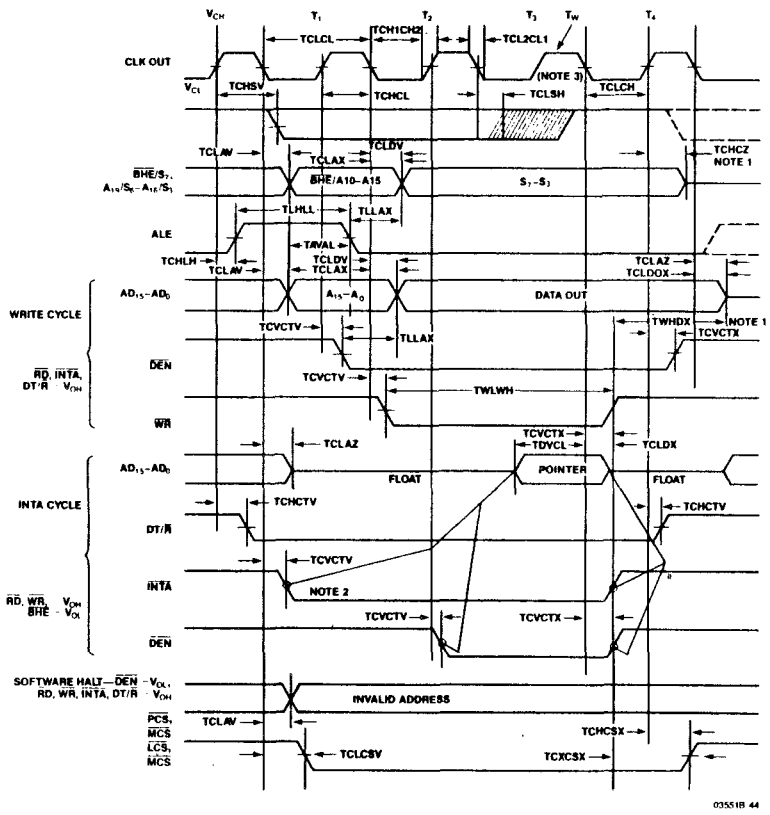
80186 CLKOUT Timing (200 pF load)

Symbol	**Parameter**	**Min.**	**Max.**	**Min.**	**Max.**	**Units**	**Test Conditions**
TCICO	CLKIN to CLKOUT Skew		50		62.5	ns	
TCLCL	CLKOUT Period	125	500	167	500	ns	
TCLCH	CLKOUT Low Time	½ TCLCL-7.5		½ TCLCL-7.5		ns	1.5 volts
TCHCL	CLKOUT High Time	½ TCLCL-7.5		½ TCLCL-7.5		ns	1.5 volts
TCH1CH2	CLKOUT Rise Time		15		15	ns	1.0 to 3.5 volts
TCL2CL1	CLKOUT Fall Time		15		15	ns	3.5 to 1 volts

All timings measured at 1.5 volts unless otherwise noted.

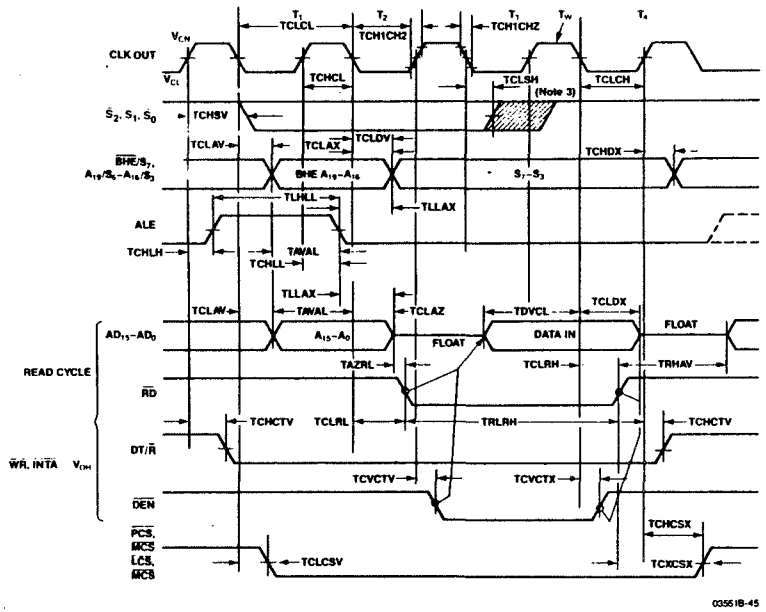
WAVEFORMS

MAJOR CYCLE TIMING



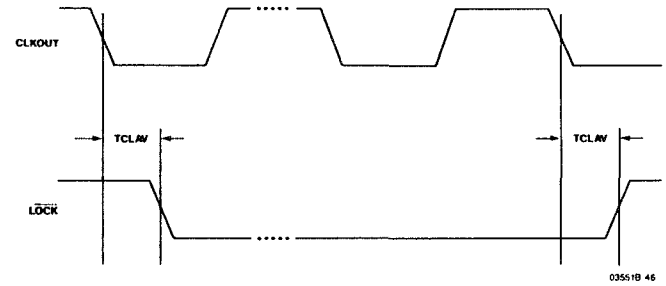
WAVEFORMS (Continued)

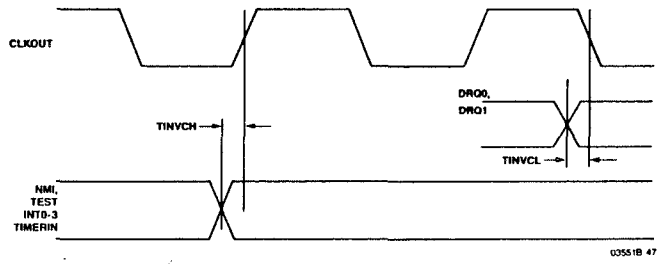
MAJOR CYCLE TIMING (Continued)

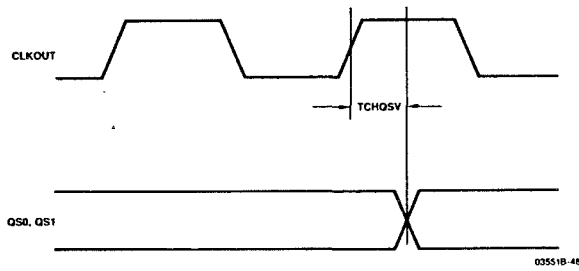


NOTES
1 Following a Write cycle, the Local Bus is floated by the 80186 only when the 80186 enters a "Hold Acknowledge" state.
2 INTA occurs one clock later in RMX-mode.
3 Status inactive just prior to T_4.

WAVEFORMS (Continued)

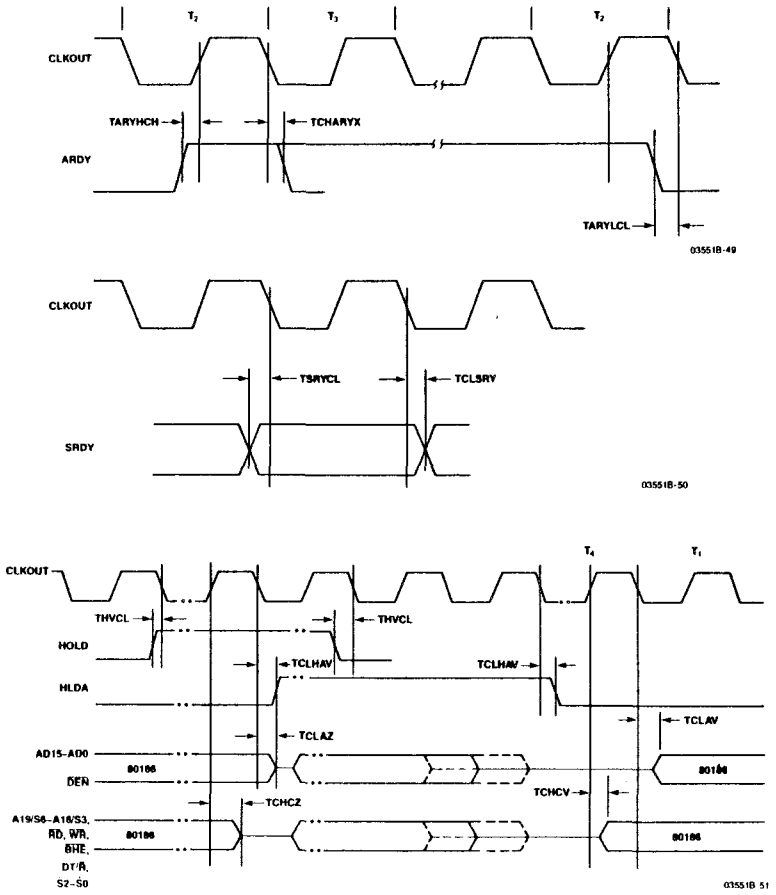






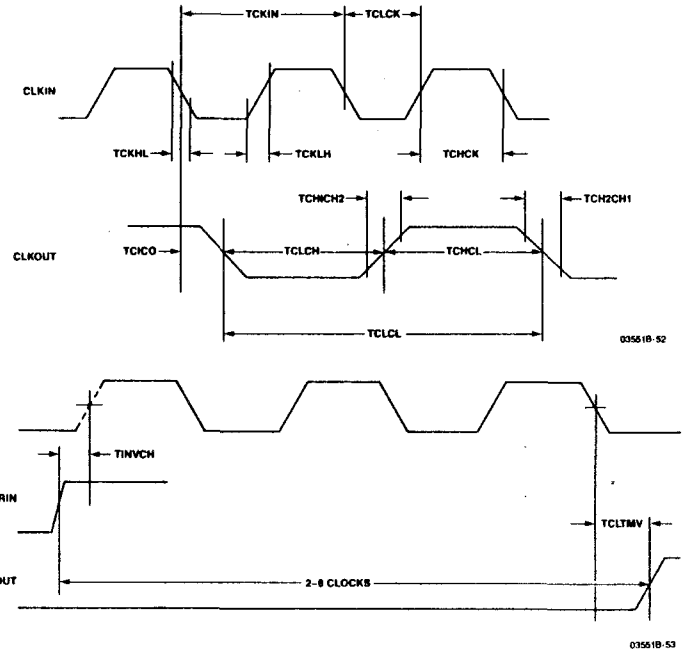
WAVEFORMS (Continued)

HOLD-HLDA TIMING



WAVEFORMS (Continued)

TIMER ON 80186



80186 INSTRUCTION TIMINGS

The following instruction timings represent the minimum execution time in clock cycles for each instruction. The timings given are based on the following assumptions:

- The opcode, along with any data or displacement required for execution of a particular instruction, has been prefetched and resides in the queue at the time it is needed.
- No wait states or bus HOLDS occur.
- All word-data is located on even-address boundaries.

All jumps and calls include the time required to fetch the opcode of the next instruction at the destination address.

All instructions which involve memory reference can require one (and in some cases, two) additional clocks above the minimum timings shown. This is due to the asynchronous nature of the handshake between the BIU and the Execution unit.

INSTRUCTION SET SUMMARY

FUNCTION	FORMAT	Clock Cycles	Comments
DATA TRANSFER			
MOV = Move:			
Register to Register/Memory	1 0 0 0 1 0 0 w | mod reg r/m	2/12	
Register memory to register	1 0 0 0 1 0 1 w | mod reg r/m	2/9	
Immediate to register/memory	1 1 0 0 0 1 1 w | mod 0 0 0 r/m | data | data if w = 1	12–13	8/16-bit
Immediate to register	1 0 1 1 w reg | data | data if w = 1	3–4	8/16-bit
Memory to accumulator	1 0 1 0 0 0 0 w | addr-low | addr-high	9	
Accumulator to memory	1 0 1 0 0 0 1 w | addr-low | addr-high	8	
Register/memory to segment register	1 0 0 0 1 1 1 0 | mod 0 reg r/m	2/9	
Segment register to register/memory	1 0 0 0 1 1 0 0 | mod 0 reg r/m	2/11	
PUSH = Push:			
Memory	1 1 1 1 1 1 1 1 | mod 1 1 0 r/m	16	
Register	0 1 0 1 0 reg	10	
Segment register	0 0 0 reg 1 1 0	9	
Immediate	0 1 1 0 1 0 s 0 | data | data if s = 0	10	
PUSHA = Push All	0 1 1 0 0 0 0 0	36	
POP = Pop:			
Memory	1 0 0 0 1 1 1 1 | mod 0 0 0 r/m	20	
Register	0 1 0 1 1 reg	10	
Segment register	0 0 0 reg 1 1 1 (reg ≠ 01)	8	
POPA = Pop All	0 1 1 0 0 0 0 1	51	
XCHG = Exchange:			
Register/memory with register	1 0 0 0 0 1 1 w | mod reg r/m	4/17	
Register with accumulator	1 0 0 1 0 reg	3	
IN = Input from:			
Fixed port	1 1 1 0 0 1 0 w | port	10	
Variable port	1 1 1 0 1 1 0 w	8	
OUT = Output to:			
Fixed port	1 1 1 0 0 1 1 w | port	9	
Variable port	1 1 1 0 1 1 1 w	7	
XLAT = Translate byte to AL	1 1 0 1 0 1 1 1	11	
LEA = Load EA to register	1 0 0 0 1 1 0 1 | mod reg r/m	6	
LDS = Load pointer to DS	1 1 0 0 0 1 0 1 | mod reg r/m (mod ≠ 11)	18	
LES = Load pointer to ES	1 1 0 0 0 1 0 0 | mod reg r/m (mod ≠ 11)	18	
LAHF = Load AH with flags	1 0 0 1 1 1 1 1	2	
SAHF = Store AH into flags	1 0 0 1 1 1 1 0	3	
PUSHF = Push flags	1 0 0 1 1 1 0 0	9	
POPF = Pop flags	1 0 0 1 1 1 0 1	8	

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
ARITHMETIC			
ADD = Add:			
Reg. memory with register to either	0 0 0 0 0 0 d w | mod reg r/m	3/10	
Immediate to register/memory	1 0 0 0 0 0 s w | mod 0 0 0 r/m | data | data if s w = 01	4/16	
Immediate to accumulator	0 0 0 0 0 1 0 w | data | data if w = 1	3/4	8/16-bit
ADC = Add with carry:			
Reg. memory with register to either	0 0 0 1 0 0 d w | mod reg r/m	3/10	
Immediate to register/memory	1 0 0 0 0 0 s w | mod 0 1 0 r/m | data | data if s w = 01	4/16	
Immediate to accumulator	0 0 0 1 0 1 0 w | data | data if w = 1	3/4	8/16-bit
INC = Increment:			
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 0 r/m	3/15	
Register	0 1 0 0 0 reg	3	
SUB = Subtract:			
Reg. memory and register to either	0 0 1 0 1 0 d w | mod reg r/m	3/10	
Immediate from register/memory	1 0 0 0 0 0 s w | mod 1 0 1 r/m | data | data if s w = 01	4/16	
Immediate from accumulator	0 0 1 0 1 1 0 w | data | data if w = 1	3/4	8/16-bit
SBB = Subtract with borrow:			
Reg. memory and register to either	0 0 0 1 1 0 d w | mod reg r/m	3/10	
Immediate from register/memory	1 0 0 0 0 0 s w | mod 0 1 1 r/m | data | data if s w = 01	4/16	
Immediate from accumulator	0 0 0 1 1 1 0 w | data | data if w = 1	3/4	8/16-bit
DEC = Decrement:			
Register/memory	1 1 1 1 1 1 1 w | mod 0 0 1 r/m	3/15	
Register	0 1 0 0 1 reg	3	
CMP = Compare:			
Register/memory with register	0 0 1 1 1 0 1 w | mod reg r/m	3/10	
Register with register/memory	0 0 1 1 1 0 0 w | mod reg r/m	3/10	
Immediate with register/memory	1 0 0 0 0 0 s w | mod 1 1 1 r/m | data | data if s w = 01	3/10	
Immediate with accumulator	0 0 1 1 1 1 0 w | data | data if w = 1	3/4	8/16-bit
NEG = Change sign	1 1 1 1 0 1 1 w | mod 0 1 1 r/m	3	
AAA = ASCII adjust for add	0 0 1 1 0 1 1 1	8	
DAA = Decimal adjust for add	0 0 1 0 0 1 1 1	4	
AAS = ASCII adjust for subtract	0 0 1 1 1 1 1 1	7	
DAS = Decimal adjust for subtract	0 0 1 0 1 1 1 1	4	
MUL = Multiply (unsigned)	1 1 1 1 0 1 1 w | mod 1 0 0 r/m		
Register-Byte		26–28	
Register-Word		35–37	
Memory-Byte		32–34	
Memory-Word		41–43	
IMUL = Integer multiply (signed)	1 1 1 1 0 1 1 w | mod 1 0 1 r/m		
Register-Byte		25–28	
Register-Word		34–37	
Memory-Byte		31–34	
Memory-Word		40–43	
IMUL = Integer immediate multiply (signed)	0 1 1 0 1 0 s 1 | mod reg r/m | data | data if s = 0	22–25/29–32	
DIV = Divide (unsigned)	1 1 1 1 0 1 1 w | mod 1 1 0 r/m		
Register-Byte		29	
Register-Word		38	
Memory-Byte		35	
Memory-Word		44	

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
ARITHMETIC (Continued):			
IDIV = Integer divide (signed)	1 1 1 1 0 1 1 w | mod 1 1 1 r/m		
Register-Byte		44–52	
Register-Word		53–61	
Memory-Byte		50–58	
Memory-Word		59–67	
AAM = ASCII adjust for multiply	1 1 0 1 0 1 0 0 | 0 0 0 0 1 0 1 0	19	
AAD = ASCII adjust for divide	1 1 0 1 0 1 0 1 | 0 0 0 0 1 0 1 0	15	
CBW = Convert byte to word	1 0 0 1 1 0 0 0	2	
CWD = Convert word to double word	1 0 0 1 1 0 0 1	4	
LOGIC			
Shift Rotate Instructions:			
Register/Memory by 1	1 1 0 1 0 0 0 w | mod TTT r/m	2/15	
Register/Memory by CL	1 1 0 1 0 0 1 w | mod TTT r/m		
Register/Memory by Count	1 1 0 0 0 0 0 w | mod TTT r/m | count ("n")	5+n/17+n	

TTT	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

AND = And:					
Reg memory and register to either	0 0 1 0 0 0 0 d w	mod reg r/m		3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 0 0 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 1 0 0 1 0 w	data	data if w = 1	3/4	8/16-bit
TEST = And function to flags, no result:					
Register memory and register	1 0 0 0 0 1 0 w	mod reg r/m		3/10	
Immediate data and register/memory	1 1 1 1 0 1 1 w	mod 0 0 0 r/m	data	data if w = 1	4/10
Immediate data and accumulator	1 0 1 0 1 0 0 w	data	data if w = 1	3/4	8/16-bit
OR = Or:					
Reg memory and register to either	0 0 0 0 1 0 d w	mod reg r/m		3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 0 0 1 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 0 0 1 1 0 w	data	data if w = 1	3/4	8/16-bit
XOR = Exclusive or:					
Reg memory and register to either	0 0 1 1 0 0 d w	mod reg r/m		3/10	
Immediate to register/memory	1 0 0 0 0 0 0 w	mod 1 1 0 r/m	data	data if w = 1	4/16
Immediate to accumulator	0 0 1 1 0 1 0 w	data	data if w = 1	3/4	8/16-bit
NOT = Invert register/memory	1 1 1 1 0 1 1 w	mod 0 1 0 r/m		3	
STRING MANIPULATION:					
MOVS = Move byte word	1 0 1 0 0 1 0 w			8+8n	
CMPS = Compare byte word	1 0 1 0 0 1 1 w			5+22n	
SCAS = Scan byte word	1 0 1 0 1 1 1 w			5+15n	
LODS = Load byte wd to AL AX	1 0 1 0 1 1 0 w			6+11n	
STOS = Stor byte wd from AL A	1 0 1 0 1 0 1 w			6+9n	
INS = Input byte wd from DX port	0 1 1 0 1 1 0 w			8	
OUTS = Output byte wd to DX port	0 1 1 0 1 1 1 w			7	

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
STRING MANIPULATION (Continued):			
Repeated by count in CX			
MOVS = Move string	1 1 1 1 0 0 1 0 | 1 0 1 0 0 1 0 w	14	
CMPS = Compare string	1 1 1 1 0 0 1 z | 1 0 1 0 0 1 1 w	22	
SCAS = Scan string	1 1 1 1 0 0 1 z | 1 0 1 0 1 1 1 w	15	
LODS = Load string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 1 0 w	12	
STOS = Store string	1 1 1 1 0 0 1 0 | 1 0 1 0 1 0 1 w	10	
INS = Input string	[illegible]	[illegible]	Not Repeated
OUTS = Output string	1 1 1 0 1 1 0 | 0 1 1 0 1 1 1 w	8+8n/14	Repeated Not Repeated
CONTROL TRANSFER			
CALL = Call:			
Direct within segment	1 1 1 0 1 0 0 0 | disp-low | disp-high	14	
Register/memory indirect within segment	1 1 1 1 1 1 1 1 | mod 0 1 0 r/m	13/19	
Direct intersegment	1 0 0 1 1 0 1 0 | segment offset | segment selector	23	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 0 1 1 r/m (mod ≠ 11)	38	
JMP = Unconditional jump:			
Short long	1 1 1 0 1 0 1 1 | disp-low	13	
Direct within segment	1 1 1 0 1 0 0 1 | disp-low | disp-high	13	
Register/memory indirect within segment	1 1 1 1 1 1 1 1 | mod 1 0 0 r/m	11/17	
Direct intersegment	1 1 1 0 1 0 1 0 | segment offset | segment selector	13	
Indirect intersegment	1 1 1 1 1 1 1 1 | mod 1 0 1 r/m (mod ≠ 11)	26	
RET = Return from CALL:			
Within segment	1 1 0 0 0 0 1 1	16	
Within seg adding immed to SP	1 1 0 0 0 0 1 0 | data-low | data-high	18	
Intersegment	1 1 0 0 1 0 1 1	22	
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0 | data-low | data-high	25	

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
CONTROL TRANSFER (Continued)			
JE/JZ = Jump on equal zero	0 1 1 1 0 1 0 0 | disp	4/13	13 if JMP taken 4 if JMP not taken
JL/JNGE = Jump on less, not greater or equal	0 1 1 1 1 1 0 0 | disp	4/13	
JLE/JNG = Jump on less or equal, not greater	0 1 1 1 1 1 1 0 | disp	4/13	
JB/JNAE = Jump on below, not above or equal	0 1 1 1 0 0 1 0 | disp	4/13	
JBE/JNA = Jump on below or equal, not above	0 1 1 1 0 1 1 0 | disp	4/13	
JP/JPE = Jump on parity, parity even	0 1 1 1 1 0 1 0 | disp	4/13	
JO = Jump on overflow	0 1 1 1 0 0 0 0 | disp	4/13	
JS = Jump on sign	0 1 1 1 1 0 0 0 | disp	4/13	
JNE/JNZ = Jump on not equal, not zero	0 1 1 1 0 1 0 1 | disp	4/13	
JNL/JGE = Jump on not less, greater or equal	0 1 1 1 1 0 1 | disp	4/13	
JNLE/JG = Jump on not less or equal, greater	0 1 1 1 1 1 1 1 | disp	4/13	
JNB/JAE = Jump on not below, above or equal	0 1 1 1 0 0 1 1 | disp	4/13	
JNBE/JA = Jump on not below or equal, above	0 1 1 1 0 1 1 1 | disp	4/13	
JNP/JPO = Jump on not par, par odd	0 1 1 1 1 0 1 1 | disp	4/13	
JNO = Jump on not overflow	0 1 1 1 0 0 0 1 | disp	4/13	
JNS = Jump on not sign	0 1 1 1 1 0 0 1 | disp	4/13	
LOOP = Loop CX times	1 1 1 0 0 0 1 0 | disp	5/15	
LOOPZ/LOOPE = Loop while zero equal	1 1 1 0 0 0 0 1 | disp	6/16	
LOOPNZ/LOOPNE = Loop while not zero equal	1 1 1 0 0 0 0 0 | disp	6/16	
JCXZ = Jump on CX zero	1 1 1 0 0 0 1 1 | disp	16 5	JMP taken/ JMP not taken
ENTER = Enter Procedure	1 1 0 0 1 0 0 0 | data-low | data-high | L		
L = 0		15	
L = 1		25	
L > 1		22+16(n−1)	
LEAVE = Leave Procedure	1 1 0 0 1 0 0 1	8	
INT = Interrupt:			
Type specified	1 1 0 0 1 1 0 1 | type	47	
Type 3	1 1 0 0 1 1 0 0	45	if INT. taken/ if INT. not taken
INTO = Interrupt on overflow	1 1 0 0 1 1 1 0	48/4	
IRET = Interrupt return	1 1 0 0 1 1 1 1	28	
BOUND = Detect value out of range	0 1 1 0 0 0 1 0 | mod reg r/m	33–35	

Shaded areas indicate instructions not available in iAPX 86, 88 microsystems.

INSTRUCTION SET SUMMARY (Continued)

FUNCTION	FORMAT	Clock Cycles	Comments
PROCESSOR CONTROL			
CLC = Clear carry	1 1 1 1 1 0 0 0	2	
CMC = Complement carry	1 1 1 1 0 1 0 1	2	
STC = Set carry	1 1 1 1 1 0 0 1	2	
CLD = Clear direction	1 1 1 1 1 1 0 0	2	
STD = Set direction	1 1 1 1 1 1 0 1	2	
CLI = Clear interrupt	1 1 1 1 1 0 1 0	2	
STI = Set interrupt	1 1 1 1 1 0 1 1	2	
HLT = Halt	1 1 1 1 0 1 0 0	2	
WAIT = Wait	1 0 0 1 1 0 1 1	6	if $\overline{\text{test}}$ = 0
LOCK = Bus lock prefix	1 1 1 1 0 0 0 0	2	
ESC = Processor Extension Escape	1 1 0 1 1 T T T | mod LLL r/m (TTT LLL are opcode to processor extension)	6	

FOOTNOTES

The effective Address (EA) of the memory operand is computed according to the mod and r/m fields:

if mod = 11 then r/m is treated as a REG field

if mod = 00 then DISP = 0*, disp-low and disp-high are absent

if mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent

if mod = 10 then DISP = disp-high: disp-low

if r/m = 000 then EA = (BX) + (SI) + DISP

if r/m = 001 then EA = (BX) + (DI) + DISP

if r/m = 010 then EA = (BP) + (SI) + DISP

if r/m = 011 then EA = (BP) + (DI) + DISP

if r/m = 100 then EA = (SI) + DISP

if r/m = 101 then EA = (DI) + DISP

if r/m = 110 then EA = (BP) + DISP*

if r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

*except if mod = 00 and r/m = 110 then EA = disp-high: disp-low.

SEGMENT OVERRIDE PREFIX

0 0 1 reg 1 1 0

reg is assigned according to the following:

reg	Segment Register
00	ES
01	CS
10	SS
11	DS

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)
000 AX	000 AL
001 CX	001 CL
010 DX	010 DL
011 BX	011 BL
100 SP	100 AH
101 BP	101 CH
110 SI	110 DH
111 DI	111 BH

The physical addresses of all operands addressed by the BP register are computed using the SS segment register. The physical addresses of the destination operands of the string primitive operations (those addressed by the DI register) are computed using the ES segment, which may not be overridden.

Intercontinental Micro
RPB-200 Application Note
12-22-86

1) RPB-200 used as a modem:

A) Move 2x5 Dip shunt to side marked modem (near edge of card).

B) Move 2 pin shunt from second left position to first left position (nearest resistor):

```
o o o o o o
|
o o o o o o
```

C) On the DB-25 connector, pins are defined as follows:

2 - Transmit Data (to modem)
3 - Receive Data (from modem)
4 - Request to send (to modem)
5 - Clear to send (from modem)
6 - Data Set ready (from modem)
7 - Ground
8 - Data Carrier detect (from modem)
15- Transmit clock (from modem; jumper required)
17- Receive Clock (from modem; jumper required)
20- Data Terminal Ready (to modem)
22- Ring (from modem)

2) RPB-200 handshaking configuration:

At default, the RPB-200 handshaking is disabled to be compatable with terminals. When using the Clear to Send option on printers, the following pin options are available:

```
                o   o   o   o   o   o
                    |
                o   o   o   o   o   o

DB-25 pin -->   5   d   20  4   11  19
                    i
                    s
                    a
                    b
                    l
                    e
```

2 3 7 20 optional

10/12/87 Jay Fo says default config should support printer just fine.

TurboLAN Networking

InterContinental Microsystems
4020 Leaverton Ct., Anaheim, CA 92807
Phone: (714) 630-0964
Telex: 821375 SUPPORT UD
FAX: (714) 630-4811
Easylink 62562040 BBX (714)632-8750