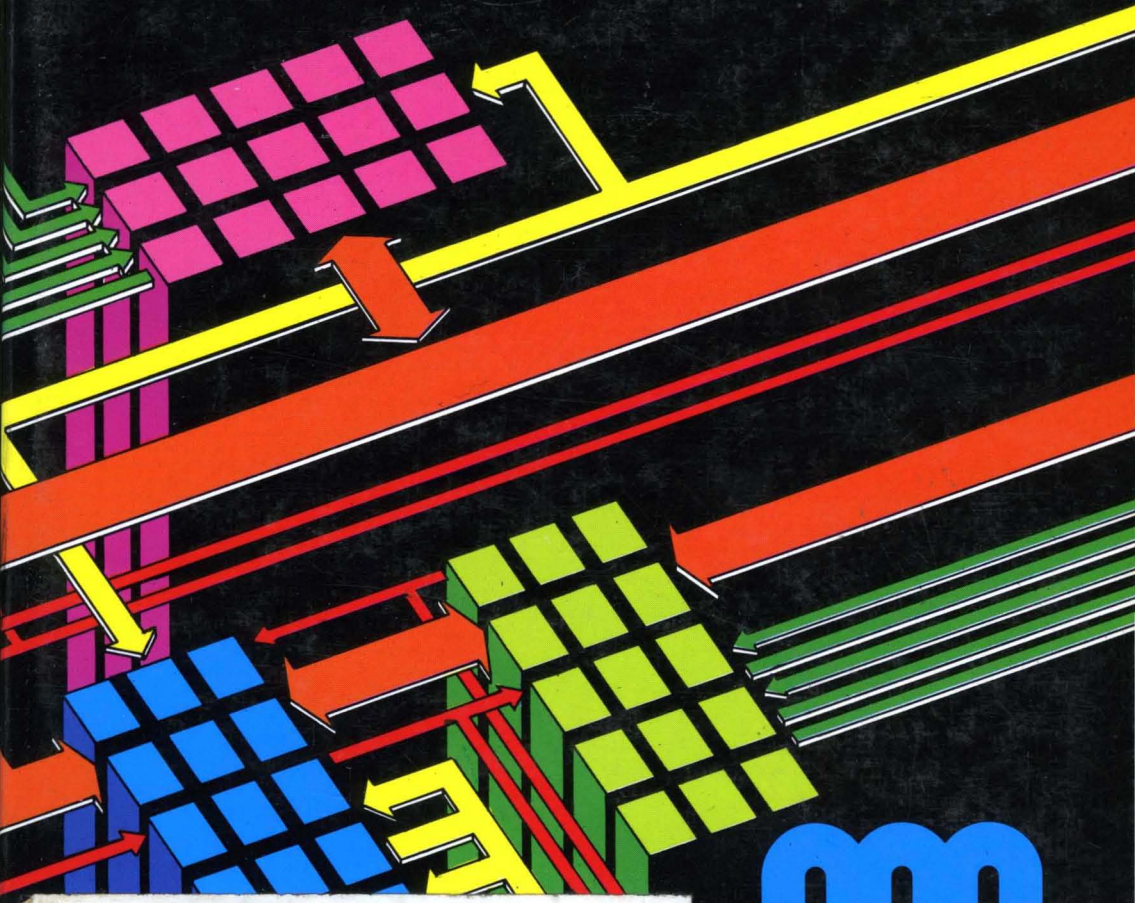
DATABOOK

Z8500 UNIVERSAL PERIPHERALS

1st EDITION



Z8500 UNIVERSAL PERIPHERALS

1ST EDITION

ISSUED JANUARY 1983

CONTENTS

Z8500
Universal Peripherals

SGS' Universal Peripheral Components Family is more than a group of simple I/O circuits—they are intelligent, fully programmable devices capable of performing complicated tasks independently. Their capabilities unburden the master CPU, reduce bus traffic, increase system throughput, and greatly simplify overall system hardware design requirements.

Z8500 series Universal Peripheral components are capable of interfacing with conventional non-multiplexed buses. All of them are extensively programmable to permit each to be tailored to its own applications. Also available is a series of functionally similar devices, identified by the number Z80xx, which interface with the Z-BUS™ or both the Z-BUS and conventional multiplexed buses.

Data communications problems are neatly handled by the **Z8530 SCC Serial Communications Controller.** This device is a serial, dual-channel, multi-protocol controller which supports all popular communications formats. The SCC support virtually all serial data transfer applications.

Counting, timing, and parallel I/O transfer problems are easily solved using the **Z8536 CIO Counter/Timer and I/O Unit.** This components has three 16-bit counter/timers, three I/O ports, and can double as a programmable priority-interrupt controller.

Interface problems with the interconnections of major components within an asynchronous, parallel processor system can be solved using the SCC **Z8538 FIO FIFO I/O Interface Unit.** This general-purpose interface unit provides expandable, bidirectional buffering between asynchronous CPUs in a parallel processing network, or between a CPU and peripheral circuits and/or devices. The FIO can be used with system having either multiplexed or nonmultiplexed buses.

General-purpose control and data manipulation problems are easily handled by the **Z8590 UCP Universal Peripheral Controller.** The UPC is a complete microcomputer designed for offline applications. This microcomputer executes the same friendly capable instruction set as SGS' Z8™ microcomputer; it has three I/O ports, six levels of priority-interrupt, and 2K bytes of memory on chip. The UPC is intended for applications that require an intelligent peripheral controller which can assume many of the tasks normally required of the master CPU.

Serial Communications Controller

Z8530 SCC

Features

- Two independent, 0 to 1M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZI, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1s or 0s.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes.

General Description

The Z8530 SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with conventional non-multiplexed buses. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, Digital Phase-Locked Loops, and crystal oscillators that dramatically reduce the need for external logic.

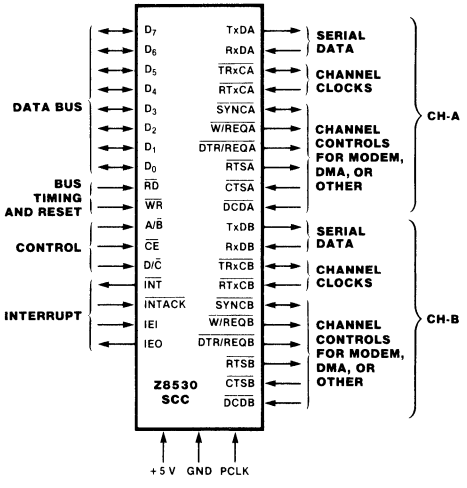


Figure 1. Logic Functions

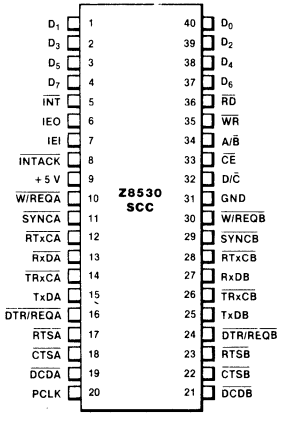


Figure 2. Pin Configuration

General Description (Continued)

The SCC handles asynchronous formats, Synchronous byte-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drives, etc.).

The device can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The Z-BUS daisy-chain interrupt hierarchy is also supported — as is standard for SGS peripheral components.

The Z8530 SCC is packaged in a 40-pin ceramic DIP and uses a single +5 V power supply.

Pin Description

The following section describes the pin functions of the SCC. Figures 1 and 2 detail the respective pin functions and pin assignments.

A/$\overline{B}$. *Channel A/Channel B Select* (input). 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 a read or write operation.

$\overline{CTSA}$, $\overline{CTSB}$. *Clear To Send* (inputs, active Low). If these pins are programmed as Auto Enables, a Low on the 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 SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

D/$\overline{C}$. *Data/Control Select* (input). This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command.

$\overline{DCDA}$, $\overline{DCDB}$. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accomodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

D_0-D_7. *Data Bus* (bidirectional, 3-state). These lines carry data and commands to and from the SCC.

$\overline{DTR}$/$\overline{REQA}$, $\overline{DTR}$/$\overline{REQB}$. *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 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 the CPU is not servicing an SCC interrupt or the SCC 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.

$\overline{INT}$. *Interrupt Request* (output, open-drain, active Low). This signal is activated when the SCC requests an interrupt.

$\overline{INTACK}$. *Interrupt Acknowledge* (input, active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the

Pin Description (Continued)

SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ is latched by the rising edge of PCLK.

PCLK. *Clock* (input). This is the master SCC clock used to synchronize internal signals PCLK is a TTL level signal.

$\overline{RD}$. *Read* (input, active Low). This signal indicates a read operation and when the SCC is selected, enables the SCC's 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.

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

$\overline{RTxCA}$, $\overline{RTxCB}$. *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed in several different modes of operation. In each channel, $\overline{RTxC}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective $\overline{SYNC}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in Asynchronous modes.

$\overline{RTSA}$, $\overline{RTSB}$. *Request To Send* (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 11) is set, the $\overline{RTS}$ 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 pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

$\overline{SYNCA}$, $\overline{SYNCB}$. *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{CTS}$ and $\overline{DCD}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 (Figure 10) but have no other function.

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

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 synchronous characters are recognized. The synchronous condition is not latched, so these outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

TxDA, TxDB. *Transmit Data* (outputs, active High). These output signals transmit serial data at standard TTL levels.

$\overline{TRxCA}$, $\overline{TRxCB}$. *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ 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.

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

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

Functional Description

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, the SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

Data Communications Capabilities. The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous data-communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes. Transmission and reception can be accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in Figure 1). If the Low does not persist (as in the case of a transient), the character assembly process does not start.

Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing or error conditions using dedicated routines. Furthermore, a built-in checking process avoids the interpretation of a framing error as a new start bit: a framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals—a feature allowing use of the wide variety of clock sources. The transmitter and receiver can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the $\overline{\text{SYNC}}$ pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes. The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes,

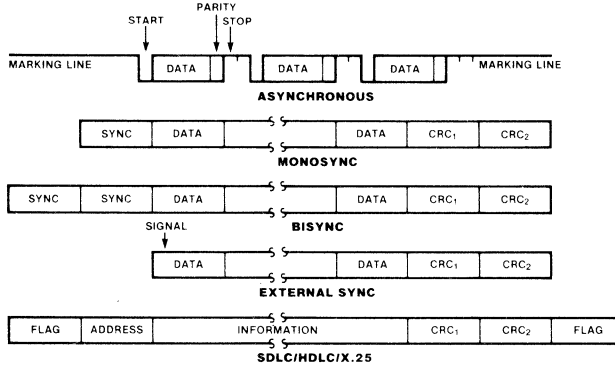


Figure 3. Some SCC Protocols

Functional Description (Continued)

allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronization pattern (Bisync), or with an external synchronous signal. Leading sync characters can be removed without interrupting the CPU.

Five- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as shown in Figure 4.

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This permits the implementation of protocols such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit synchronous characters, regardless of the programmed character length.

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change so that an abort may be issued. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{\text{SYNC}}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1s or all 0s. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

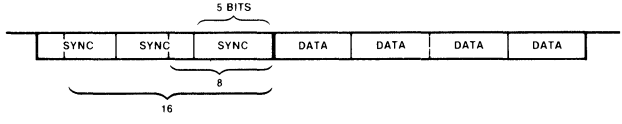


Figure 4. Detecting 5 - or 7 - Bit Synchronous Characters

Functional Description (Continued)

NRZ, NRZI or FM coding may be used in any 1x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

The SCC can be conveniently used under DMA control to provide high speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode can act as a controller (Figure 5).

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only 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 easily recognized.

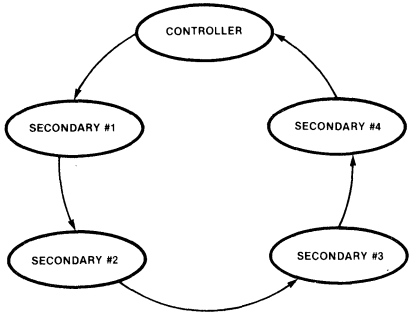


Figure 5. An SDLC Loop

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the 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 upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

Baud Rate Generator. Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On startup, the flip-flop on the output is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the baud rate generator toggles upon reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the baud rate generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-Locked Loop (see next section).

If the receive clock or transmit clock is not programmed to come from the $\overline{\text{TRxC}}$ pin, the

Functional Description (Continued)

output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate (the baud rate is in bits/second and the BR clock period is in seconds):

$$\text{baud rate} = \frac{1}{2\ (\text{time constant} + 2) \times (\text{BR clock period})}$$

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

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the incoming data stream for edges (either 1 to 0 or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL still counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream should occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.

The 32x clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

Data Encoding. The SCC may be programmed to encode and decode the serial data in four different ways (Figure 6). 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, bi-phase 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 no additional transition at the center of the bit cell. In FM0 (bi-phase 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 no additional transition at the center of the bit cell. In addition to these four methods, the SCC can be used to decode

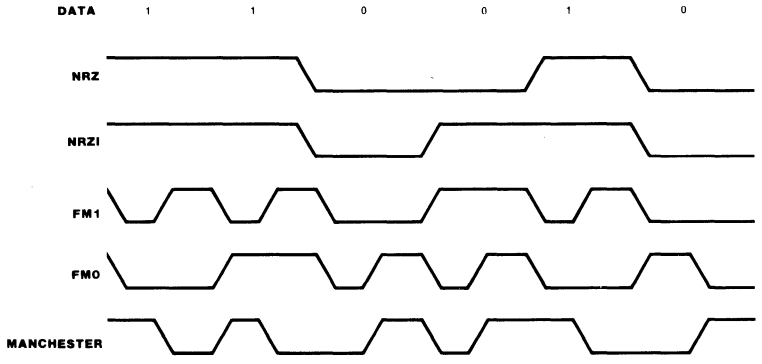


Figure 6. Data Encoding Methods

Functional Description (Continued)

Manchester (bi-phase 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 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

Auto Echo and Local Loopback. The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. 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 is responsible for disabling transmitter interrupts and $\overline{WAIT/REQUEST}$ on transmit.

The SCC is also capable of local loopback. In this mode TxD is RxD, just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The $\overline{CTS}$ and $\overline{DCD}$ inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

I/O Interface Capabilities. 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. The Block Transfer mode can be implemented under CPU or DMA control.

Polling. All interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. The idea behind polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register needs to be read; depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts. When an SCC responds to an Interrupt Acknowledge signal ($\overline{INTACK}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 10 and 11).

To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included; if it is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels) has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation of the IE bit is straightforward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts may be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 7). As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the data bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the $\overline{INT}$ output is pulled Low, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source can never be set. The IP bits are readable in RR3A.

Functional Description (Continued)

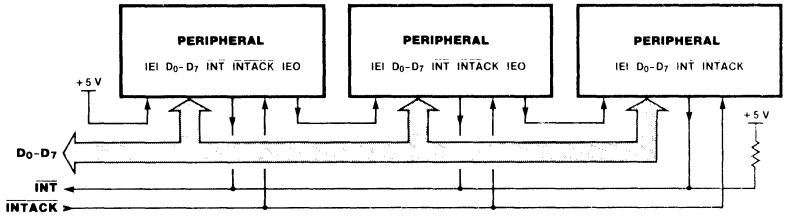


Figure 7. Interrupt Schedule

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the SCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive Condition.
- Interrupt on All Receive Characters or Special Receive Condition.
- Interrupt on Special Receive Condition Only.

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A Special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{\text{CTS}}$, $\overline{\text{DCD}}$, and $\overline{\text{SYNC}}$ 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.

CPU/DMA Block Transfer. The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers.

Functional Description (Continued)

The Block Transfer mode uses the $\overline{\text{WAIT}}$/$\overline{\text{REQUEST}}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{\text{WAIT}}$/$\overline{\text{REQUEST}}$ output can be defined under software control as a $\overline{\text{WAIT}}$ line in the CPU Block Transfer mode or as a $\overline{\text{REQUEST}}$ line in the DMA Block Transfer mode.
To a DMA controller, the SCC $\overline{\text{REQUEST}}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{\text{WAIT}}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{\text{DTR}}$/$\overline{\text{REQUEST}}$ line allows full-duplex operation under DMA control.

Architecture

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a nonmultiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 8).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two sync-character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a

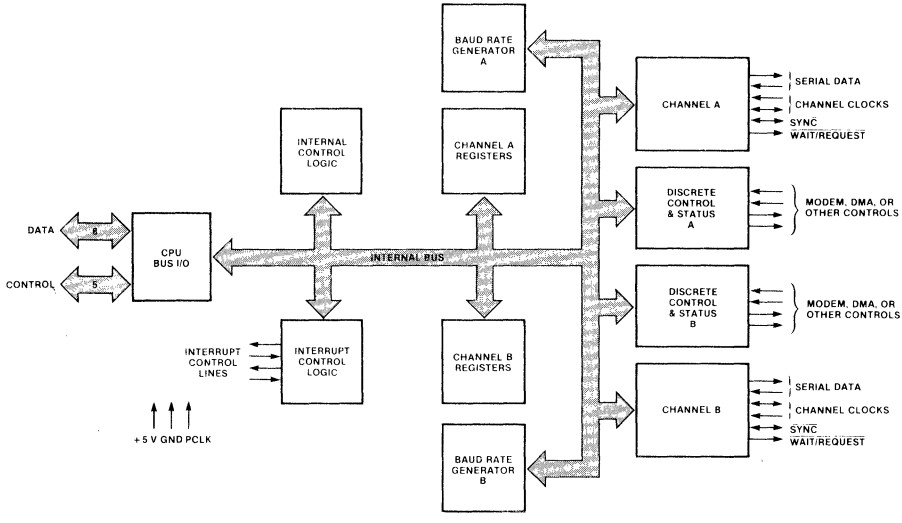


Figure 8. Block Diagram of SCC Architecture

Architecture (Continued)

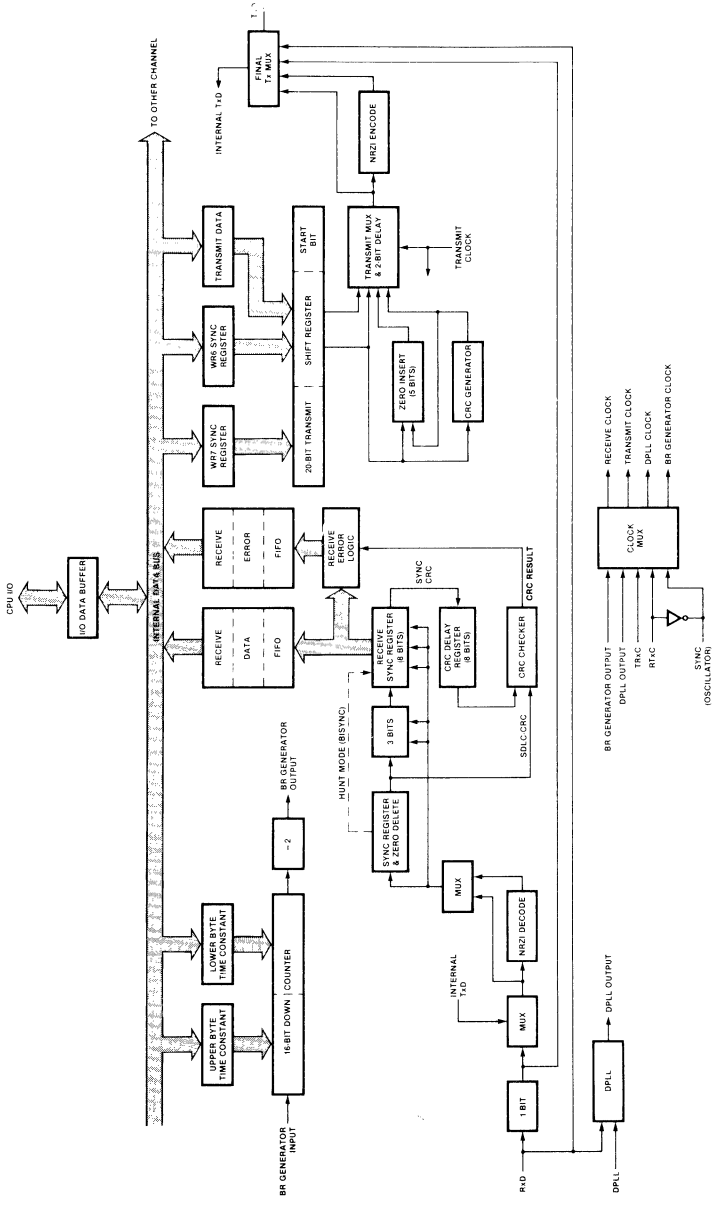


Figure 9. Data Path

Architecture (Continued)

write only Master Interrupt Control register and three read registers: one containing the vector with status infomation (Channel B only), one containing the vector without status (Channel A only), and one containing the Interrupt Pending bits (Channel A only).

The registers for each channel are designated as follows:

WR0-WR15 — Write Registers 0 through 15.

RR0-RR3, RR10, RR12, RR13, RR15 — Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

Data Path. The transmit and receive data path illustrated in Figure 9 is identical for both channels. The receiver has three 8-bit buffer registers in an FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in Asynchronous modes also determines the data path).

The transmitter has an 8-bit Transmit Data buffer register loaded from the internal data bus and a 20-bit Transmit Shift register that can be loaded either from the synchronous character registers or from the Transmit Data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD)

Read Register Functions	
RR0	Transmit/Receive buffer status and External status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive buffer
RR10	Miscellaneous status
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External/Status interrupt information
Write Register Functions	
WR0	CRC initialize, initialization commands for the various modes, Register Pointers
WR1	Transmit/Receive interrupt and data transfer mode definition
WR2	Interrupt vector (accessed through either channel)
WR3	Receive parameters and control
WR4	Transmit/Receive miscellaneous parameters and modes
WR5	Transmit parameters and controls
WR6	Sync characters or SDLC address field
WR7	Sync character or SDLC flag
WR8	Transmit buffer
WR9	Master interrupt control and reset (accessed through either channel)
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

Table 1. Read and Write Register Functions

Programming

The SCC contains 13 write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

In the SCC, register addressing is direct for the data registers only, which are selected by a High on the D/$\overline{C}$ pin. In all other cases (with the exception of WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected

Programming (Continued)

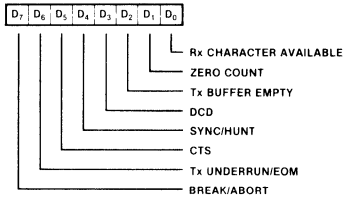
register, and if the second operation is read, the selected read register is accessed. All of the registers in the SCC, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, the Asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

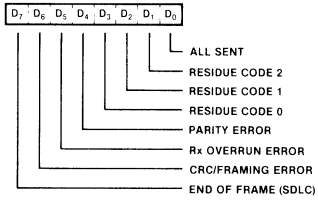
Read Registers. The SCC contains eight read registers (actually nine, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) may be read to learn the baud rate generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A). Figure 10 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).

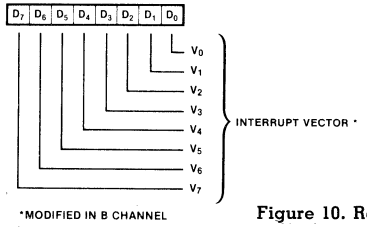
Read Register 0



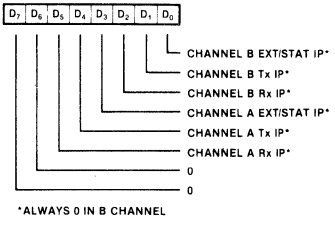
Read Register 1



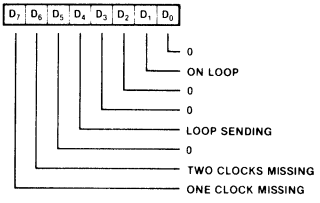
Read Register 2



Read Register 3



Read Register 10



Read Register 12

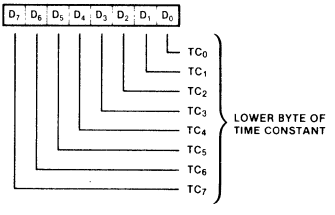
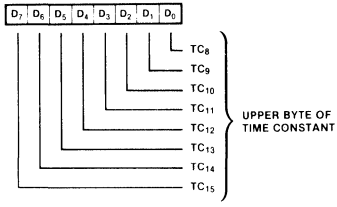


Figure 10. Read Register Bit Functions

Programming (Continued)

Read Register 13



Read Register 15

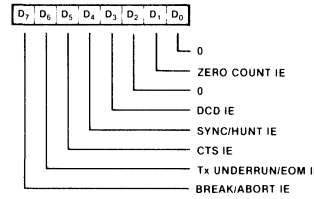
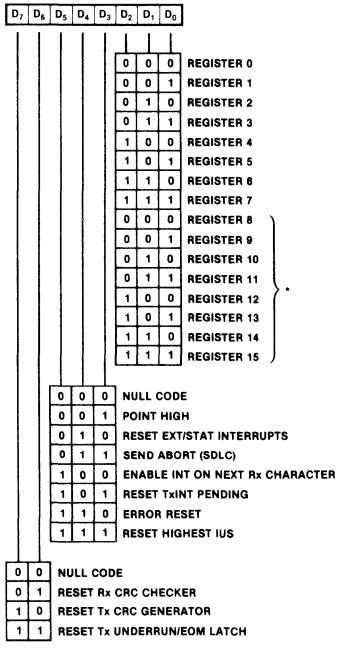


Figure 10. Read Register Bit Functions (Continued)

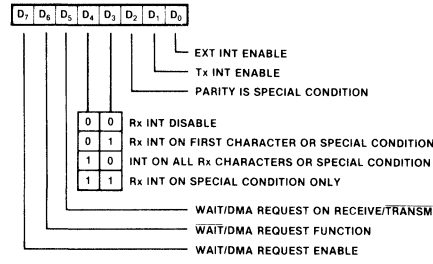
Write Registers. The SCC contains 13 write registers (14 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 11 shows the format of each write register.

Write Register 0



*WITH POINT HIGH COMMAND

Write Register 1



Write Register 2

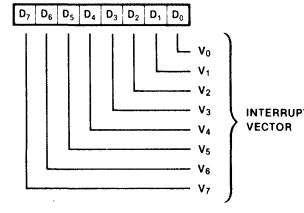
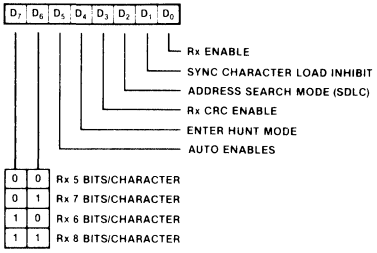


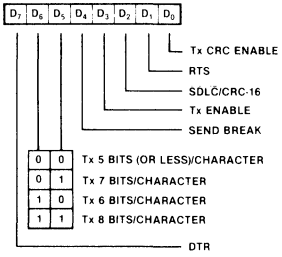
Figure 11. Write Register Bit Functions

Programming (Continued)

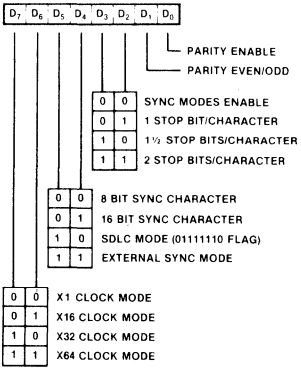
Write Register 3



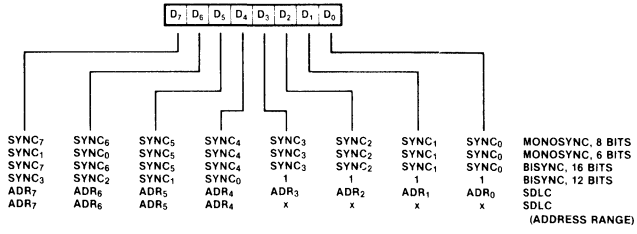
Write Register 5



Write Register 4



Write Register 6



Write Register 7

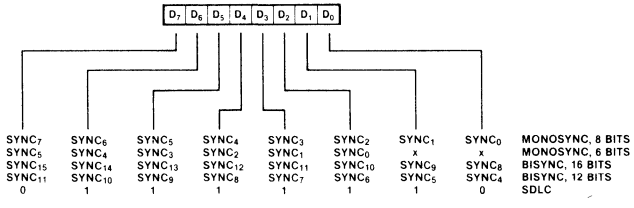
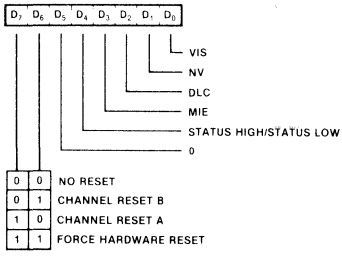


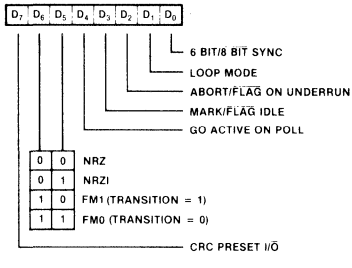
Figure 11. Write Register Bit Functions (Continued)

Programming (Continued)

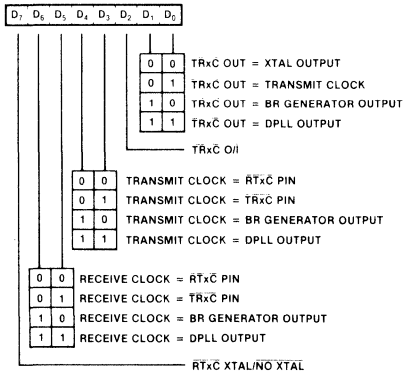
Write Register 9



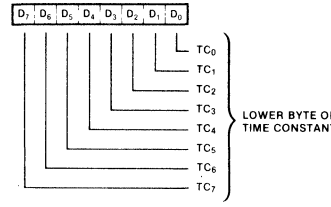
Write Register 10



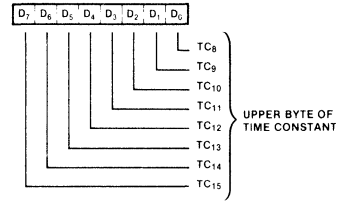
Write Register 11



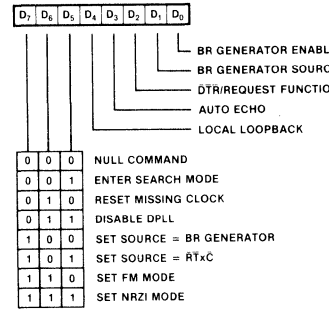
Write Register 12



Write Register 13



Write Register 14



Write Register 15

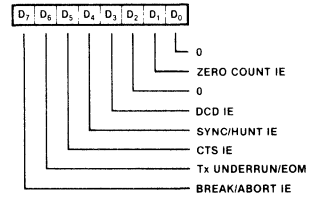


Figure 11. Write Register Bit Functions (Continued)

Timing

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the rising edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the SCC to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the SCC. This time must be at least 6 PCLK cycles plus 200 ns.

Read Cycle Timing. Figure 12 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

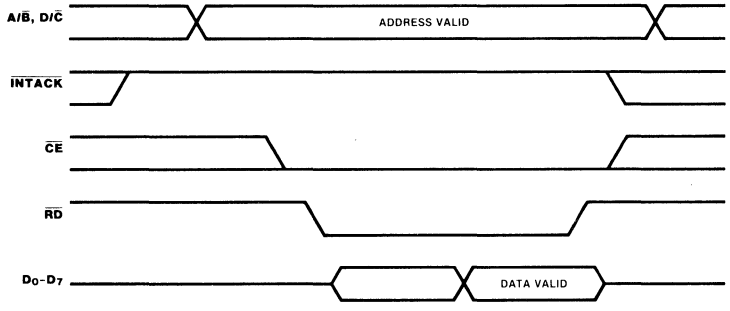


Figure 12. Read Cycle Timing

Timing (Continued)

Write Cycle Timing. Figure 13 illustrates Write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened.

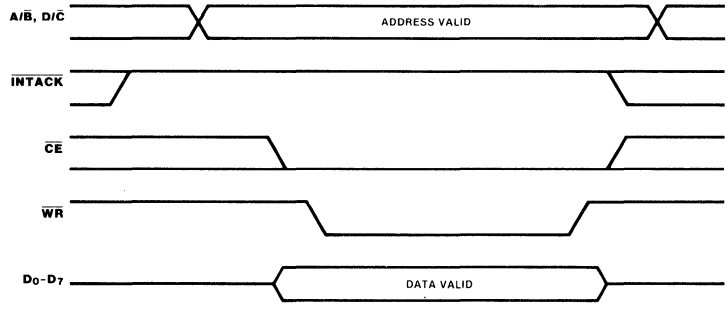


Figure 13. Write Cycle Timing

Interrupt Acknowledge Cycle Timing. Figure 14 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes Low and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{RD}$ Low by placing its interrupt vector on D_0-D_7 and it then sets the appropriate Interrupt-Under-Service latch internally.

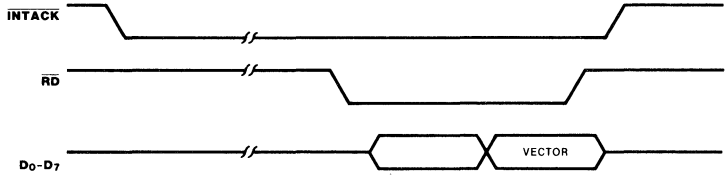


Figure 14. Interrupt Acknowledge Cycle Timing

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 ≤ 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.

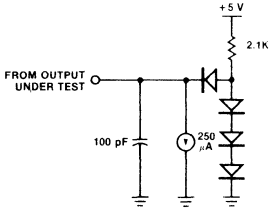


Figure 15. Standard Test Load

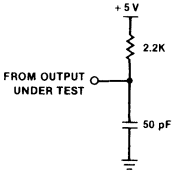


Figure 16. Open-Drain Test Load

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 µA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = +2.0 mA
I_{IL}	Input Leakage		±10.0	µA	0.4 ≤ V_{IN} ≤ +2.4V
I_{OL}	Output Leakage		±10.0	µA	0.4 ≤ V_{OUT} ≤ +2.4V
I_{CC}	V_{CC} Supply Current		250	mA	

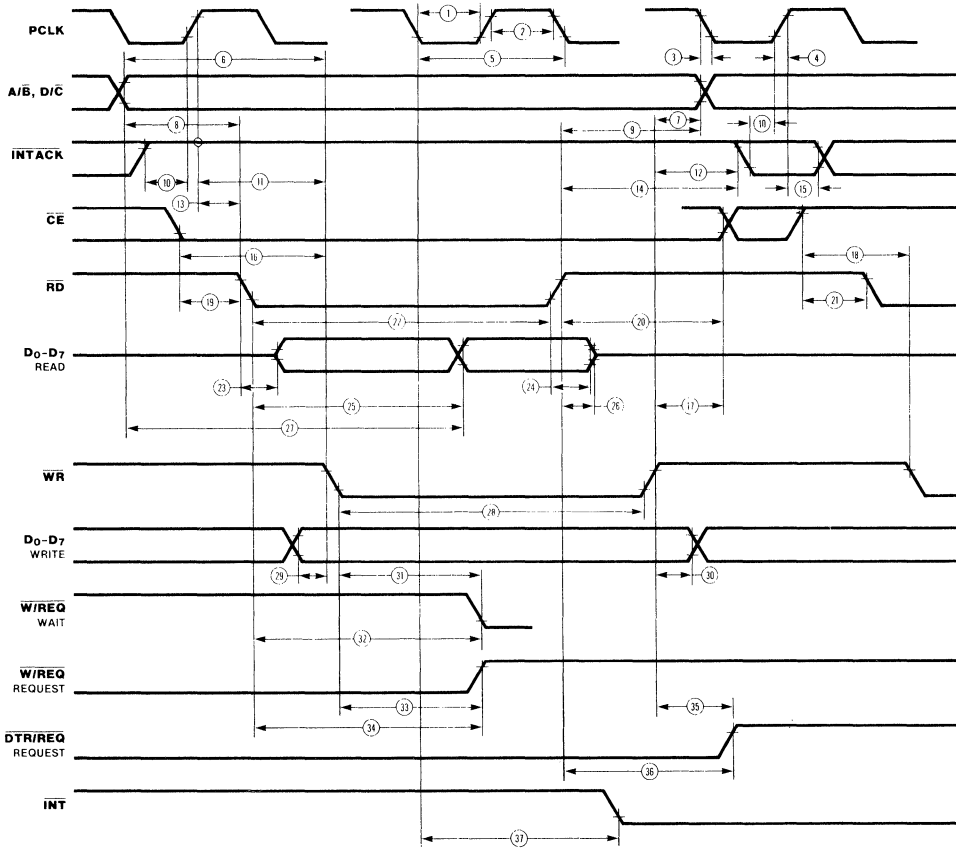
V_{CC} = 5 V ± 5% unless otherwise specified, over specified temperature range.

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.

Read and Write Timing



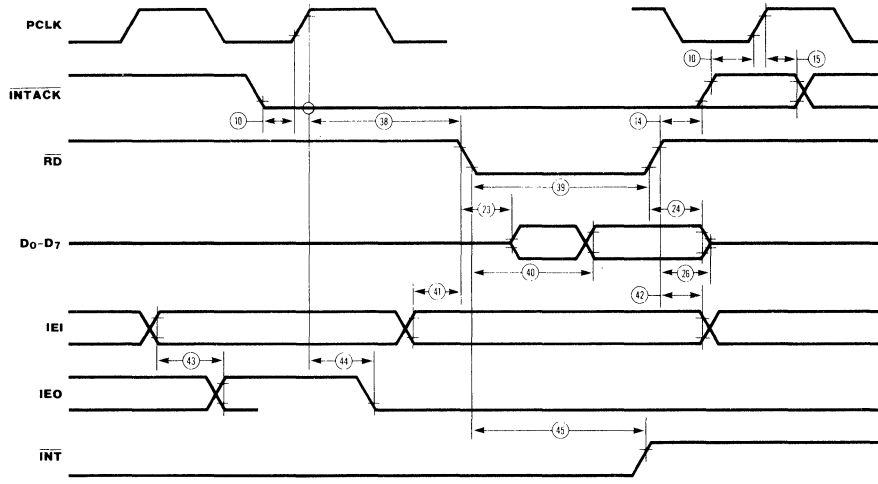
Read and Write Timing (Continued)

Number	Symbol	Parameters	Min(ns)	Max(ns)	Notes*
1	TwPC1	PCLK Low Width	105	2000	
2	TwPCh	PCLK High Width	105	2000	
3	TfPC	PCLK Fall Time		20	
4	TrPC	PCLK Rise Time		20	
5	TcPC	PCLK Cycle Time	250	4000	
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	0		
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time	200		1
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time	200		1
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		
16	TsCE1(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		
19	TsCE1(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time	0		1
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time	0		1
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time	100		1
22	TwRD1	$\overline{RD}$ Low Width	390		1
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		255	
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay		70	2

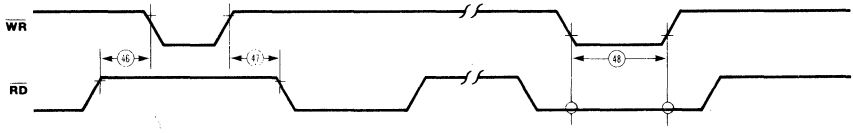
NOTES:

1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ±0.5 V change in the output with a maximum dc load and minimum ac load.

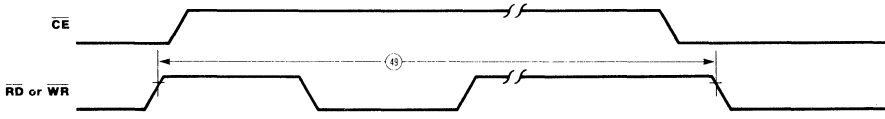
Interrupt Acknowledge Timing



Reset Timing



Cycle Timing



Number	Symbol	Parameters	Min(ns)	Max(ns)	Notes*
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		590	
28	TwWR1	$\overline{WR}$ Low Width	390		
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	0		
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay		240	4
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay		240	4
33	TdWRf(REQ)	$\overline{WR}$ ↓ to W/$\overline{REQ}$ Not Valid Delay		240	
34	TdRDf(REQ)	$\overline{RD}$ ↓ to W/$\overline{REQ}$ Not Valid Delay		240	
35	TdWRr(REQ)	$\overline{WR}$ ↑ to $\overline{DTR}$/$\overline{REQ}$ Not Valid Delay		5TcPC + 300	
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR}$/$\overline{REQ}$ Not Valid Delay		5TcPC + 300	
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay		500	4
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay			5
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	285		
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		190	
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	120		
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		
43	TdIEI(IEO)	IEI to IEO Delay Time		120	
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250	
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay		500	4
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		
49	Trc	Valid Access Recovery Time	6TcPC + 200		3

NOTES:

3. Parameter applies only between transactions involving the SCC.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.

*Timings are preliminary and subject to change.

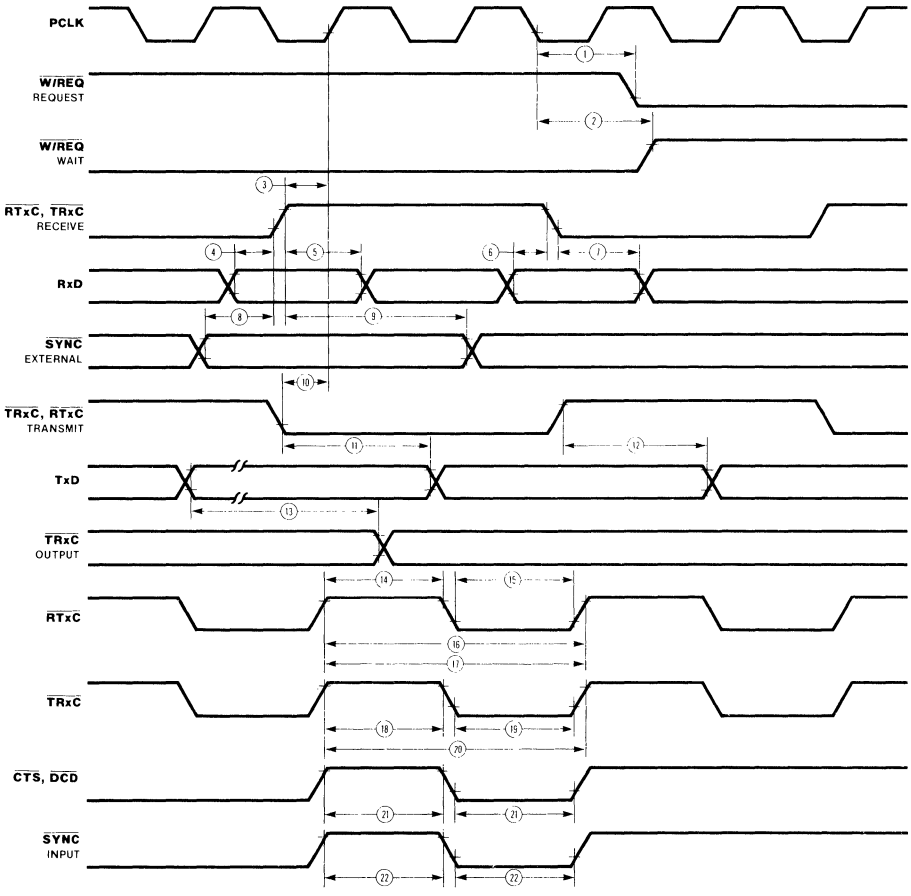
General Timing

Number	Symbol	Parameters	Min(ns)	Max(ns)	Notes*
1	TdPC(REQ)	PCLK ↓ to W/REQ Valid Delay		250	
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350	
3	TsRXC(PC)	RxC ↑ to PCLK ↑ Setup Time	50		1,4
4	TsRXD(RXCr)	RxD to RxC ↑ Setup Time (X1 Mode)	0		1
5	ThRXD(RXCr)	RxD to RxC ↑ Hold Time (X1 Mode)	150		1
6	TsRXD(RXCf)	RxD to RxC ↓ Setup Time (X1 Mode)	0		1,5
7	ThRXD(RXCf)	RxD to RxC ↓ Hold Time (X1 Mode)	150		1,5
8	TsSY(RXC)	SYNC to RxC ↑ Setup Time	-200		1
9	ThSY(RXC)	SYNC to RXC↑ Hold Time	3TcPC +200		1
10	TsTXC(PC)	TxC ↓ to PCLK ↑ Setup Time	0		2,4
11	TdTXCf(TXD)	TxC ↓ to TxD Delay (X1 Mode)		300	2
12	TdTXCr(TXD)	TxC ↑ to TxD Delay (X1 Mode)		300	2,5
13	TdTXD(TRX)	TxD to TRxC Delay (Send Clock Echo)			
14	TwRTXh	RTxC High Width	180		
15	TwRTXl	RTxC Low Width	180		
16	TcRTX	RTxC Cycle Time	400		
17	TcRTXX	Crystal Oscillator Period	250	1000	3
18	TwTRXh	TRxC High Width	180		
19	TwTRXl	TRxC Low Width	180		
20	TcTRX	TRxC Cycle Time	400		
21	TwEXT	DCD or CTS Pulse Width	200		
22	TwSY	SYNC Pulse Width	200		

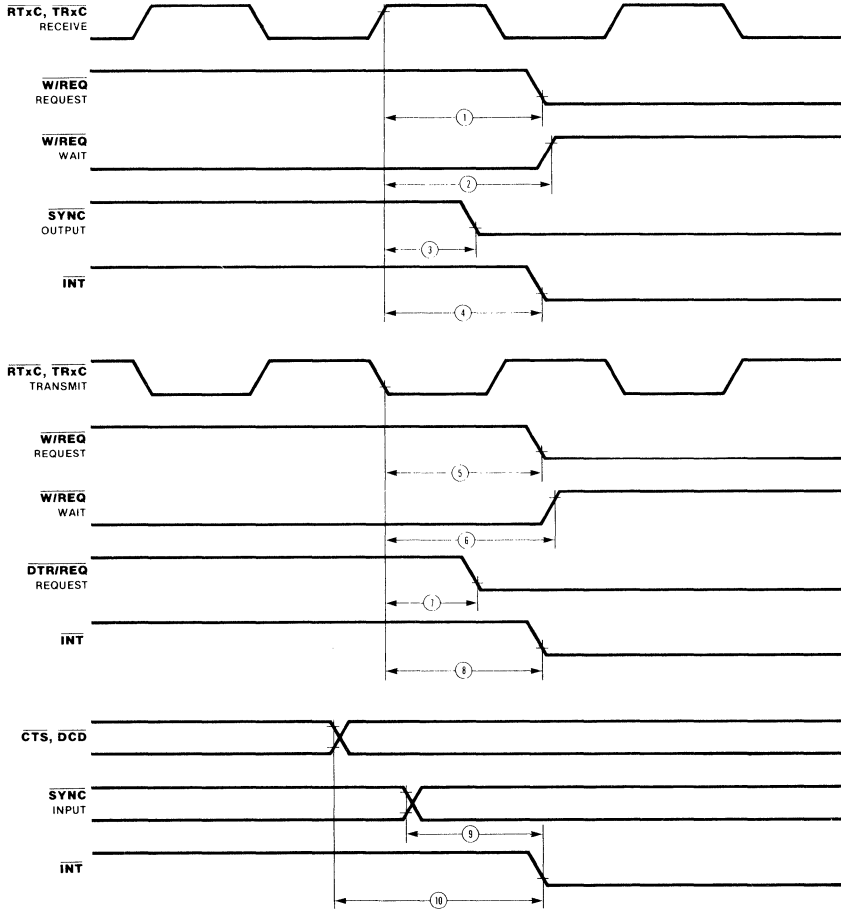
NOTES:
1. RxC is RTxC or TRxC, whichever is supplying the receive clock.
2. TxC is TRxC or RTxC, whichever is supplying the transmit clock.
3. Both RTxC and SYNC have 30 pF capacitors to ground connected to them.
4. Parameter applies *only* if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between RxC and PCLK or TxC and PCLK is required.
5. Parameter applies only to FM encoding/decoding.

*Timings are preliminary and subject to change.

General Timing (Continued)



System Timing



System Timing (Continued)

Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TdRXC (REQ)	$\overline{RxC}$ ↑ to $\overline{W/REQ}$ Valid Delay	8	12	TcPC	2
2	TdRXC(W)	$\overline{RxC}$ ↑ to Wait Inactive Delay	8	12	TcPC	1,2
3	TdRXC(SY)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay	4	7	TcPC	2
4	TdRXC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay	10	16	TcPC	1,2
5	TdTXC(REQ)	$\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay	5	8	TcPC	3
6	TdTXC(W)	$\overline{TxC}$ ↓ to Wait Inactive Delay	5	8	TcPC	1,3
7	TdTXC(DRQ)	$\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay	4	7	TcPC	3
8	TdTXC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay	6	10	TcPC	1,3
9	TdSY(INT)	$\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay	2	6	TcPC	1
10	TdEXT(INT)	$\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay	2	6	TcPC	1

NOTES:
1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.

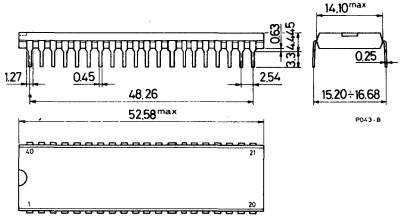
*Timings are preliminary and subject to change.

Ordering Information

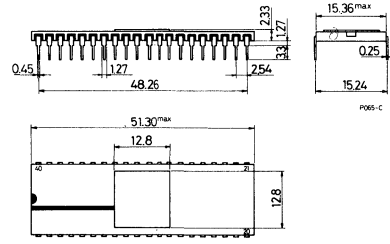
Type	Package	Temp	Clock	Description
Z8530 B1	Plastic 40 pin	0/+70°C	4MHz	Z8530 Serial Communications Controller
B6	Plastic 40 pin	-40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D2	Ceramic 40 pin	-5S/+125°C		
D6	Ceramic 40 pin	-40/+85°C		
Z8530A B1	Plastic 40 pin	0/+70°C	6MHz	
B6	Plastic 40 pin	-40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D6	Ceramic 40 pin	-40/+85°C		

Packages

Plastic



Ceramic



Counter/Timer and Parallel I/O Unit

Features

- Two independent 8-bit, double-buffered, bidirectional I/O ports plus a 4-bit special-purpose I/O port. I/O ports feature programmable polarity, programmable direction (Bit mode), "pulse 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 with up to four external access lines per counter/timer (count input, output, gate, and trigger), and three output duty cycles (pulsed, one-shot, and square-wave), programmable as retriggerable or nonretriggerable.
- Easy to use since all registers are read/write.

General Description

The Z8536 CIO Counter/Timer and Parallel I/O element is a general-purpose peripheral circuit, satisfying most counter/timer and parallel I/O needs encountered in system designs. This versatile device contains three I/O ports and three counter/timers. Many programmable options tailor its configuration to specific applications. The use of the device is simplified by making all internal registers (command, status, and data) readable and (except for status bits) writable. In addition, each register is given its own unique internal address, so that any register can be accessed in two operations. All data registers can be directly accessed in a single operation. The CIO is easily interfaced to all popular microprocessors.

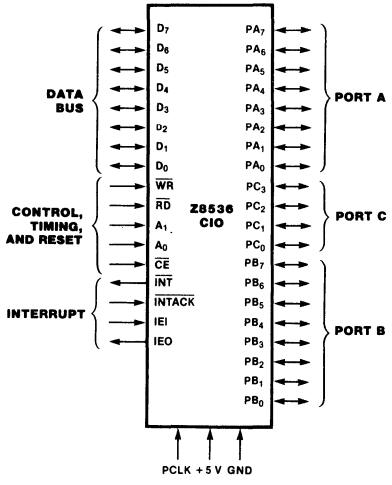


Figure 1. Logic Functions

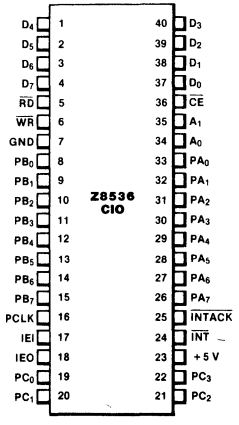


Figure 2. Pin Configurations

Pin Description

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 CIO 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 the CPU is not servicing an interrupt from the requesting CIO or 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.

$\overline{INT}$. *Interrupt Request* (output, open-drain, active Low). This signal is pulled Low when the CIO requests an interrupt.

$\overline{INTACK}$. *Interrupt Acknowledge* (input, active Low). This input indicates to the CIO 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 CIO's Port B and external devices. 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 or to provide external access to Counter/Timer 3 or access to the CIO'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 CIO is the highest priority device requesting an interrupt.

$\overline{WR}$*. *Write* (input, active Low). This signal indicates a CPU write to the CIO.

*When $\overline{RD}$ and $\overline{WR}$ are detected Low at the same time (normally an illegal condition), the CIO is reset.

Architecture

The CIO Counter/Timer and Parallel I/O element (Figure 3) 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 the internal-control logic block. An extensive number of programmable options allow the user to tailor the configuration to best suit the specific application.

The two general-purpose 8-bit I/O ports (Figure 4) are identical, except that Port B can be specified to provide external access to Counter/Timers 1 and 2. Either port can be programmed to be a handshake-driven, double-buffered port (input, output, or bidirectional) or a control-type port with the direction of each bit individually programmable. Each port includes pattern-recognition logic, allowing interrupt generation when a specific pattern is detected. The pattern-recognition logic

Architecture (Continued)

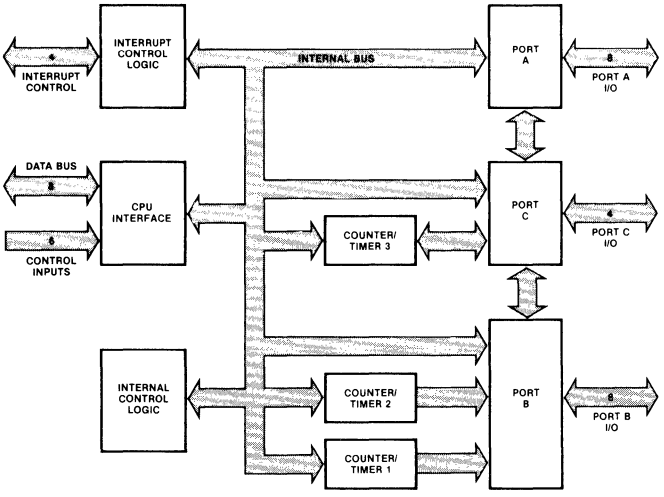


Figure 3. CIO Block Diagram

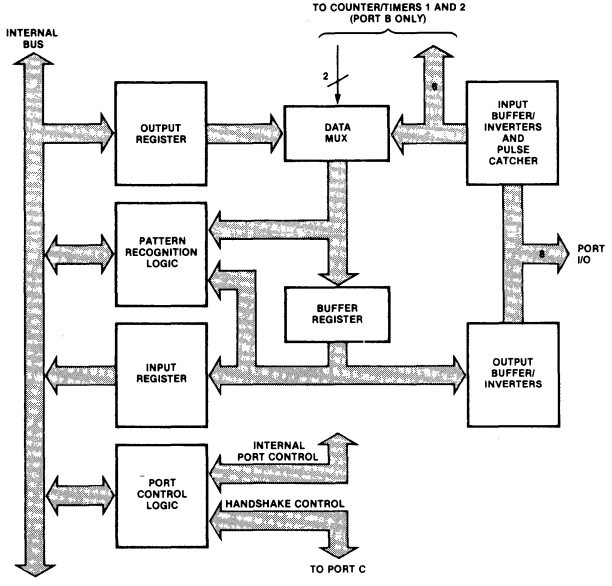


Figure 4. Ports A and B Block Diagram

Architecture (Continued)

can be programmed so the port functions like a priority-interrupt controller. Ports A and B can also be linked to form a 16-bit I/O port.

To control these capabilities, both ports contain 12 registers. Three of these registers, the Input, Output, and Buffer registers, comprise the data path registers. Two registers, the Mode Specification and Handshake Specification registers, are used to define the mode of the port and to specify which handshake, if any, is to be used. The reference pattern for the pattern-recognition logic is defined via three registers: the Pattern Polarity, Pattern Transition, and Pattern Mask registers. The detailed characteristics of each bit path (for example, the direction of data flow or whether a path is inverting or noninverting) are programmed using the Data Path Polarity, Data Direction, and Special I/O Control registers.

The primary control and status bits are grouped in a single register, the Command and Status register, so that after the port is initially configured, only this register must 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.

The function of the special-purpose 4-bit port, Port C (Figure 5), depends upon the roles of Ports A and B. Port C provides the required handshake lines. Any bits of Port C not used as handshake lines can be used as I/O lines or to provide external access for the third counter/timer.

Since Port C's function is defined primarily by Ports A and B, only three registers (besides the Data Input and Output registers) are needed. These registers specify the details of each bit path: the Data Path Polarity, Data Direction, and Special I/O Control registers.

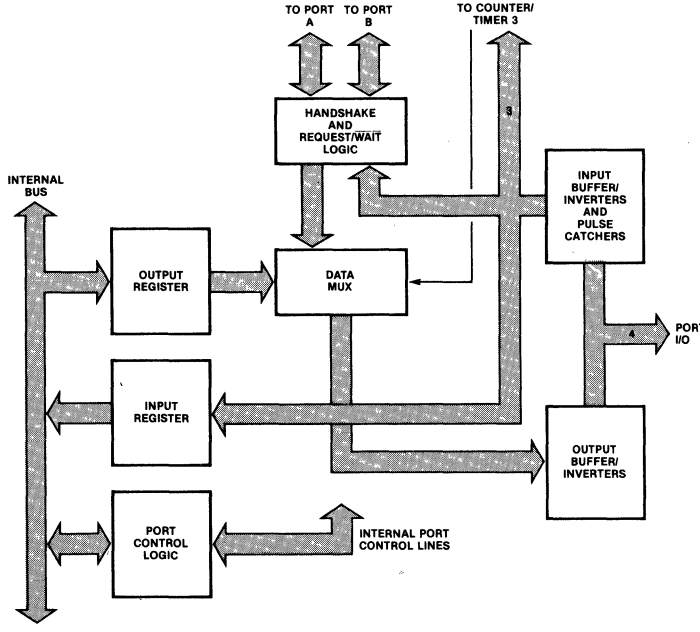


Figure 5. Port C Block Diagram

Architecture (Continued)

The three counter/timers (Figure 6) are all identical. Each is comprised 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 Command and Status registers).

The capabilities of the counter/timer are numerous. Up to four port I/O lines can be dedicated as external access lines for each counter/timer: counter input, gate input, trigger input, and counter/timer output. Three different counter/timer output duty cycles are available: pulse, one-shot, or square-wave. The operation of the counter/timer can be programmed as either retriggerable or nonretriggerable. With these and other options, most counter/timer applications are covered.

There are five registers (Master Interrupt Control register, three Interrupt Vector registers, and the Current Vector register) associated with the interrupt logic. In addition, the ports' Command and Status registers and the counter/timers' Command and Status registers include bits associated with the interrupt logic. Each of these registers contains three bits for interrupt control and status: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE).

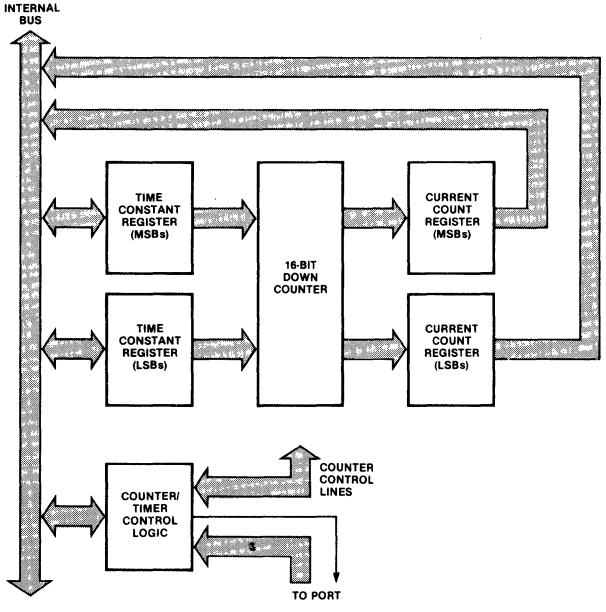


Figure 6. Counter/Timer Block Diagram

Functional Description

The following describes the functions of the ports, pattern-recognition logic, counter/timers, and interrupt logic.

I/O Port Operations. Of the CIO's three I/O ports, two (Ports A and B) are general-purpose, and the third (Port C) is a special-purpose 4-bit port. Ports A and B can be configured as input, output, or bidirectional ports with handshake. (Four different handshakes are available.) They can also be linked to form a single 16-bit port. If they are not used as ports with handshake, they provide 16 input or output bits with the data direction programmable on a bit-by-bit basis. Port B also provides access for Counter/Timers 1 and 2. In all configurations, Ports A and B can be programmed to recognize specific data patterns and to generate interrupts when the pattern is encountered.

The four bits of Port C provide the handshake lines for Ports A and B when required. A REQUEST/$\overline{\text{WAIT}}$ line can also be provided so that CIO transfers can be synchronized with DMAs or CPUs. Any Port C bits not used for handshake or REQUEST/$\overline{\text{WAIT}}$ can be used as input or output bits (individually data-direction programmable) or external access lines for Counter/Timer 3. Port C does not contain any pattern-recognition logic. It is, however, capable of bit-addressable writes. With this feature, any combination of bits can be set and/or cleared while the other bits remain undisturbed without first reading the register.

Bit Port Operations. In bit port operations, the port's Data Direction register specifies the direction of data flow for each bit. A 1 specifies an input bit, and a 0 specifies an output bit. If bits are used as I/O bits for a counter/timer, they should be set as input or output, as required.

The Data Path Polarity register provides the capability of inverting the data path. A 1 specifies inverting, and a 0 specifies noninverting. All discussions of the port operations assume that the path is noninverting.

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 set to 1 until it is cleared. The 1's catcher is cleared by writing a 0 to the bit. In all other cases, attempted writes to input bits are ignored.

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 pin. 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). If the write protect bit is written with a 1, the state of the corresponding output bit is not changed.

Ports with Handshake 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 and out of the port and interrupt generation is under control of the handshake logic. Port C provides the handshake lines as shown in Table 1. Any Port C lines not used for handshake can be used as simple I/O lines or as access lines for Counter/Timer 3.

When Ports A and B are configured as ports with handshake, they are double-buffered. This 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. Normally, the Interrupt Pending (IP) bit is set and an interrupt is generated when data is shifted into the Input register (input port) or out of the Output register (output port). For input and output ports, the IP is automatically cleared when the data is read or written. In bidirectional ports, IP is cleared only by command.

Functional Description (Continued)

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 a minimum of 16 bits 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.

When the Single Buffer (SB) bit is set to 1, the port acts as if it is only single-buffered. This is useful if the handshake line must be stopped on a byte-by-byte basis.

Ports A and B can be linked to form a 16-bit port by programming a 1 in the Port Link Control (PLC) bit. In this mode, only Port A's Handshake Specification and Command 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.

When a port is specified as a port with handshake, the type of port it is (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 (Table 1) In all cases, the contents of the Data Direction register are ignored. The contents of the Special I/O Control register apply only to output bits (3-state or open-drain). Inputs may not have 1's catchers; therefore, those bits in the Special I/O Control register are ignored. Port C lines used for handshake should be programmed as inputs. The handshake specification overrides Port C's Data Direction register for bits that must be outputs. The contents of Port C's Data Path Polarity register still apply.

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. Figure 7 shows timing for Interlocked Handshake. 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 the data is no longer available, thereby acknowledging the input port's acceptance of the last byte. This allows the CIO to interface directly to the port of a Z8 microcomputer, a UPC, an FIO, an FIFO, or to another CIO port with no external logic.

A 4-bit deskew timer can be inserted in the Data Available ($\overline{DAV}$) output for output ports. As data is transferred to the Buffer register, the deskew timer is triggered. After the number of PCLK cycles specified by the

Port A/B Configuration	PC_3	PC_2	PC_1	PC_0
Ports A and 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 and B can be specified input or output with Interlocked, Strobed, or Pulsed Handshake at the same time if neither uses REQUEST/$\overline{WAIT}$.

Table 1. Port C Bit Utilization

Functional Description (Continued)

deskew timer time constant plus one, $\overline{DAV}$ is allowed to go Low. 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.

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. Figure 7 shows timing for the Strobed Handshake. In contrast to the Interlocked handshake, the signal indicating the port is ready for another data transfer operates independently of the $\overline{ACKIN}$ input. It is up to the external logic to ensure that data overflows or underflows do not occur.

3-Wire Handshake. The 3-Wire Handshake is designed for the situation 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 (Figure 8), the rising edge of one status line indicates that the port is ready for data, and the rising edge of another status line indicates that the data has been accepted. With the 3-Wire Handshake, the output lines of many input ports can be bussed together with open-drain drivers; the output port knows when all the ports have accepted the data and are ready. This is the same handshake as is 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. Because the port's direction can be changed under software control, however, bidirectional IEEE-488-type transfers can be performed.

Pulsed Handshake. The Pulsed Handshake (Figure 9) is designed to interface to mechanical-type devices that 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 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 is used as the Interlocked 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.

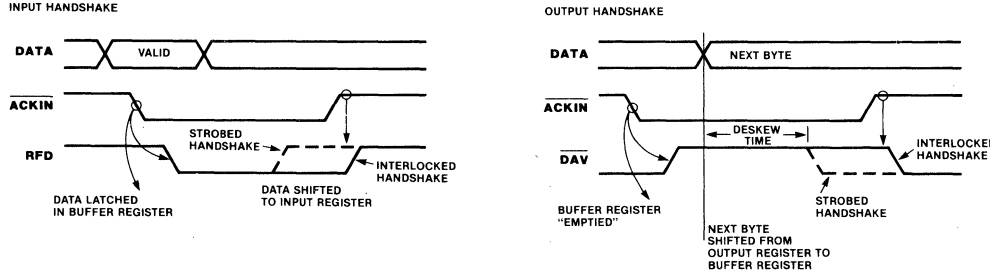


Figure 7. Interlocked and Strobed Handshakes

Functional Description (Continued)

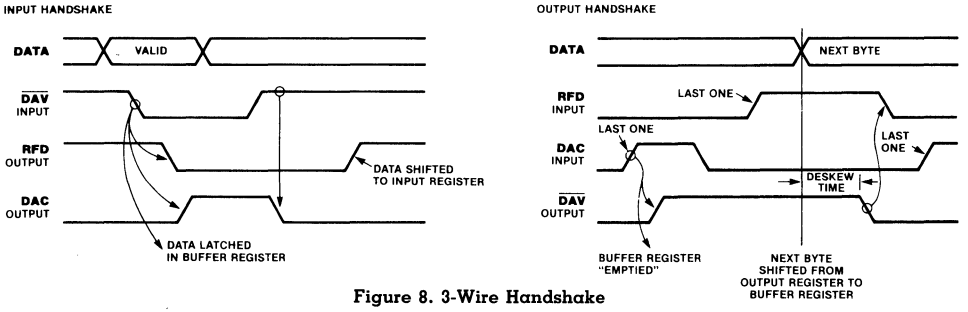


Figure 8. 3-Wire Handshake

REQUEST/$\overline{\text{WAIT}}$ Line Operation. Port C can be programmed to provide a status signal output in addition to the normal handshake lines for either Port A or B when used as a port with handshake. The additional signal is either a REQUEST or $\overline{\text{WAIT}}$ signal. The REQUEST signal indicates when a port is ready to perform a data transfer via the CPU interface. It is intended for use with a DMA-type device. The $\overline{\text{WAIT}}$ signal provides synchronization for transfers with a CPU. Three bits in the Port Handshake Specification register provide controls for the REQUEST/$\overline{\text{WAIT}}$ logic. Because the extra Port C line is used, only one port can be specified as a port with a handshake and a REQUEST/$\overline{\text{WAIT}}$ line. The other port must be a bit port.

Operation of the REQUEST line is modified by the state of the port's Interrupt on Two Bytes (ITB) control bit. When ITB is 0, the REQUEST line goes active as soon as the CIO is ready for a data transfer. If ITB is 1, REQUEST does not go active until two bytes can be transferred. REQUEST stays active as long as a byte is available to be read or written.

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/$\overline{\text{OUT}}$ line is High, the REQUEST line is High when the Output register is empty. If IN/$\overline{\text{OUT}}$ is Low, the REQUEST line is High when the Input register is full.

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, rising edge, falling edge, or any transition. Individual bits may be masked off. 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 of the OR or OR-Priority Encoded Vector modes.

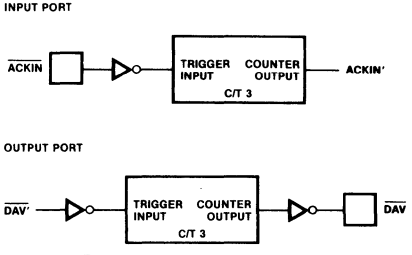


Figure 9. Pulsed Handshake

The pattern specified in the Pattern Definition register assumes that the data path is programmed to be noninverting. 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/noninvert logic.

Functional Description (Continued)

Bit Port Pattern-Recognition Operations. During bit port operations, pattern-recognition may be performed on all bits, including those used as I/O for the counter/timers. The input to the pattern-recognition logic follows the value at the pins (through the invert/noninvert logic) in all cases except for simple inputs with 1's catchers. In this case, the output of the 1's catcher is used. When operating in the AND or OR mode, it is the transition from a no-match to a match state that causes the interrupt. In the "OR" mode, if a second match occurs before the first match goes away, it does not cause an interrupt. Since a match condition only lasts a short time when edges are specified, care must be taken to avoid losing a match condition. Bit ports specified in the OR-Priority Encoded Vector mode generate interrupts as long as any match state exists. A transition from a no-match to a match state is not required.

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 interrupt was generated is latched in the input register and held until IP is cleared. In all cases, the PMF indicates the state of the port at the time it is read.

If a 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, it is automatically set to 1 along with the Interrupt Error (ERR) flag. Matches occurring while ERR is set are ignored. ERR is cleared when the corresponding IP is cleared.

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 and bit 0 is the lowest. 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 Acknowledge cycle is initiated, the vector is frozen until the corresponding IP is cleared. Where inputs that cause interrupts might change before the interrupt is serviced, the 1's catcher can be used to hold the value. 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 error detection is performed in this mode, and the Interrupt On Error bit should be set to 0.

Ports with Handshake Pattern-Recognition Operation. In this mode, the handshake logic normally controls the setting of IP and, therefore, the generation of interrupt requests. The pattern-match logic controls the Pattern-Match Flag (PMF). The data is compared with the match pattern when it is shifted from the Buffer register to the Input register (input port) or when it is shifted from the Output register to the Buffer register (output port). The pattern match logic can override the handshake logic in certain situations. If the port is programmed to interrupt when two bytes of data are available to be read or written, but the first byte matches the specified pattern, the pattern-recognition logic sets IP and generates an interrupt. While PMF is set, IP cannot be cleared by reading or writing the data registers. IP must be cleared by command. The input register is not emptied while IP is set, nor is the output register filled until IP is cleared.

If the Interrupt on Match Only (IMO) bit is set, IP is set only when the data matches the pattern. This is useful in DMA-type application when interrupts are required only after a block of data is transferred.

Counter/Timer Operation. The three independent 16-bit counter/timers consist of a presettable 16-bit down counter, a 16-bit Time

Functional Description (Continued)

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.

Function	C/T_1	C/T_2	C/T_3
Counter/Timer Output	PB 4	PB 0	PC 0
Counter Input	PB 5	PB 1	PC 1
Trigger Input	PB 6	PB 2	PC 2
Gate Input	PB 7	PB 3	PC 3

Table 2. Counter/Timer External Access

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 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 2) 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 noninverting; they can be read and used with the pattern-recognition logic. They can also include the 1's catcher input.

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. The only restriction is that when Counter/Timer 1 drives Counter/Timer 2's count input, Counter/Timer 2 must be programmed with its external count input disabled.

There are three duty cycles available for the timer/counter output: pulse, one-shot, and square-wave. Figure 10 shows the counter/timer waveforms. When the Pulse mode is specified, the output goes High for one clock 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-

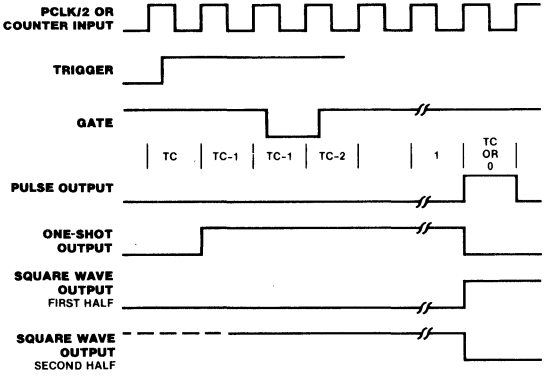


Figure 10. Counter/Timer Waveforms

Functional Description (Continued)

down sequence to begin. If 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 pulled back Low.

The Continuous/Single Cycle (C/$\overline{SC}$) bit in the Mode Specification register controls operation of the down-counter when it reaches terminal count. If C/$\overline{SC}$ is 0 when a terminal count is reached, the countdown sequence stops. If the C/$\overline{SC}$ bit is 1 each time the countdown 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.

Counter/timer operations require loading the time constant value in the Time Constant register and initiating the countdown sequence by loading the down-counter with the time constant value. The Time Constant register is accessed as two 8-bit registers. The registers are readable as well as writable, and the access order is irrelevant. A 0 in the Time Constant register specifies a time constant of 65,536. The down-counter is loaded in one of three ways: by writing a 1 to the Trigger Command Bit (TCB) of the Command and Status register, on the rising edge of the external trigger input, or, for Counter/Timer 2 only, on the rising edge of Counter/Timer 1's internal output if the counters are linked via the trigger input. The TCB is write-only, and read always returns 0.

Once the down-counter is loaded, the countdown sequence continues toward terminal count as long as all the counter/timers' hardware and software gate inputs are High. If any of the gate inputs goes Low (0), the countdown halts. It resumes when all gate inputs are 1 again.

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, retrigger causes the sequence to start over from the initial load of the time constant.

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.

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 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 1 along with the Interrupt Error (ERR) flag. Errors that occur after the internal flag is set are ignored.

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 is valid only if the counter is stopped. The down-counter can be reliably read "on the fly" by the first writing of 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.

Interrupt Logic Operation. The CIO has five potential sources of interrupts: the three counter/timers and Ports A and B. The

Functional Description (Continued)

priorities of these sources are fixed in the following order: 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 relative priorities.

The CIO interrupt priority, relative to other components within the system, is determined by an 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 following discussion assumes that the IEI line is High.

Each source of interrupt in the CIO contains three bits for the control and status of the interrupt logic: an Interrupt Pending (IP) status bit, an Interrupt Under Service (IUS) status bit, and an Interrupt Enable (IE) control bit. IP is set when an event requiring CPU intervention occurs. The setting of IP results in forcing the Interrupt ($\overline{INT}$) output Low, if the associated IE is 1.

The IUS status bit is set as a result of the Interrupt Acknowledge cycle by the CPU and is set only if its IP is of highest priority at the time the Interrupt Acknowledge commences. It can also be set directly by the CPU. Its primary function is to control the interrupt daisy chain. When set, it disables lower priority sources in the daisy chain, so that lower priority interrupt sources do not request servicing while higher priority devices are being serviced.

The IE bit provides the CPU with a means of masking off individual sources of interrupts. When IE is set to 1, interrupt is generated normally. When IE is set to 0, the IP bit is set when an event occurs that would normally require service; however, the $\overline{INT}$ output is not forced Low.

The Master Interrupt Enable (MIE) bit allows all sources of interrupts within the CIO to be disabled without having to individually set each IE to 0. If MIE is set to 0, all IPs are masked off and no interrupt can be requested or acknowledged. 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 all lower priority devices' interrupts are disabled.

As part of the Interrupt Acknowledge cycle, the CIO is capable of responding with an 8-bit interrupt vector that specifies the source of the interrupt. The CIO contains three vector registers: one for Port A, one for Port B, and one shared by the three counter/timers. The vector output is inhibited by setting the No Vector (NV) control bit to 1. The vector output can be modified to include status information to pinpoint more precisely the cause of interrupt. Whether the vector includes status or not is controlled by a Vector Includes Status (VIS) control bit. 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. In this way, all the information obtained by the vector, including status, can 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, allows use of the CIO in a polled environment. When read, the data returned is the same as the interrupt vector that would be output in an acknowledge, based on the highest priority IP set. If no unmasked IPs are set, the value FF_H is returned. The Current Vector register is read-only.

Programming

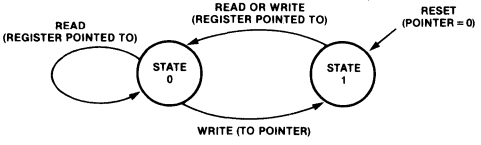
The data registers within the CIO are directly accessed by address lines A_0 and A_1 (Table 3). All other internal registers are accessed by the following two-step sequence, with the address lines specifying a control operation. First, write the address of the target register to an internal 6-bit Pointer Register; then read from or write to the target register. The Data registers can also be accessed by this method.

An internal state machine determines if accesses with A_0 and A_1 equalling 1 are to the Pointer Register or to an internal control register (Figure 11). Following any control read operation, 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 CIO is in State 1 (next control access is to the register pointed to), 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 CIO should not be left in State 1.

The CIO is reset by forcing $\overline{RD}$ and $\overline{WR}$ Low simultaneously (normally an illegal condition) or by writing a 1 to the Reset bit. Reset disables all functions except a read from or write to the Reset bit; writes to all other bits are ignored, and all reads return 01_H. In this state, all control bits are forced to 0 and may be programmed only after clearing the Reset bit (by writing a 0 to it).

A_1	A_0	Register
0	0	Port C's Data Register
0	1	Port B's Data Register
1	0	Port A's Data Register
1	1	Control Registers

Table 3. Register Selection

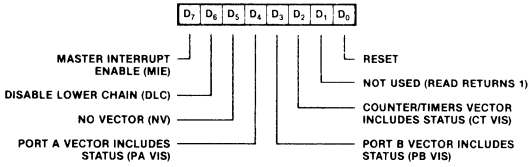


NOTE: State changes occur only when $A_0 = A_1 = 1$. No other accesses have effect.

Figure 11. State Machine Operation

Registers

Master Interrupt Control Register
Address: 000000
(Read/Write)



Master Configuration Control Register
Address: 000001
(Read/Write)

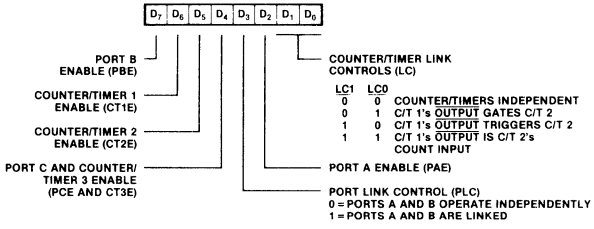
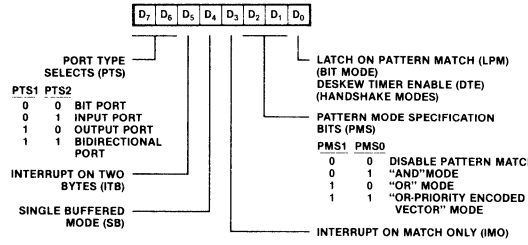


Figure 12. Master Control Registers

Registers (Continued)

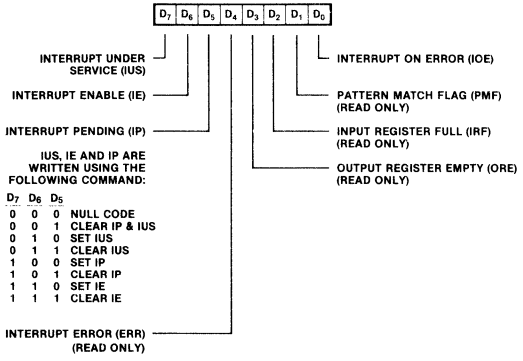
Port Mode Specification Registers

Addresses: 001000 Port A
001001 Port B
(Read/Partial Write)



Port Command and Status Registers

Addresses: 100001 Port A
101001 Port B
(Read/Write)



Port Handshake Specification Registers

Addresses: 100000 Port A
101000 Port B
(Read/Write)

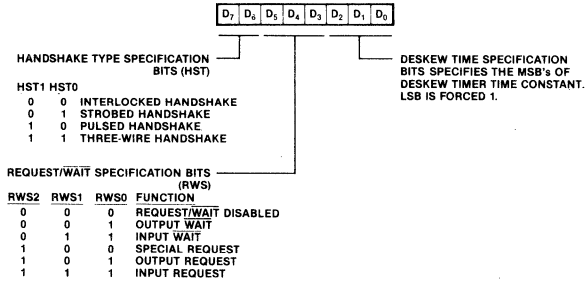
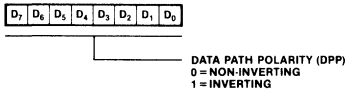


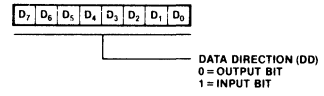
Figure 13. Port Specification Registers

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)

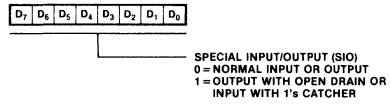
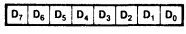


Figure 14. Bit Path Definition Registers

Port Data Registers
Addresses: 001101 Port A*
001110 Port B*
(Read/Write)



*These registers can be addressed directly.

Port C Data Register
Address: 001111*
(Read/Write)

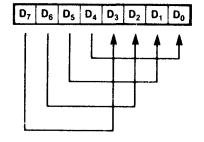


Figure 15. Port Data Registers

Registers (Continued)

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)

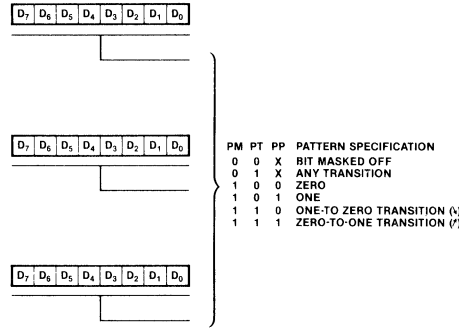


Figure 16. Pattern Definition Registers

Counter/Timer Command and Status Registers
Addresses: 011100 Counter/Timer 1
011101 Counter/Timer 2
011110 Counter/Timer 3
(Read/Write)

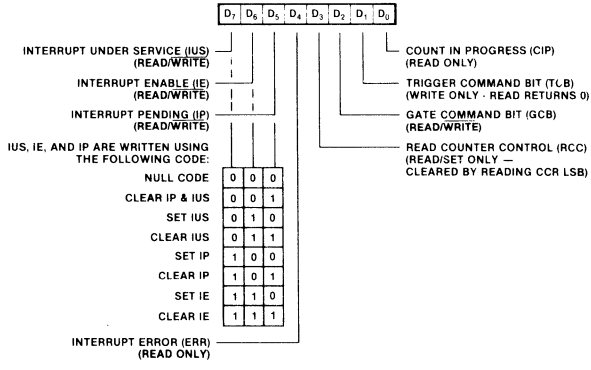
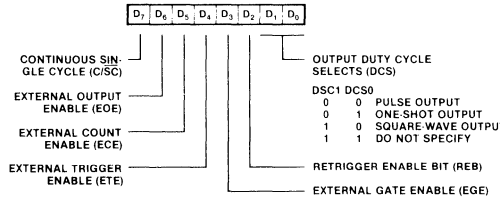


Figure 17. Counter/Timer Registers

Registers (Continued)

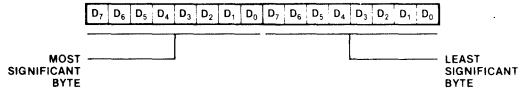
Counter/Timer Mode Specification Registers

Addresses: 001010 Counter/Timer 1
001011 Counter/Timer 2
001100 Counter/Timer 3
(Read/Partial 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)

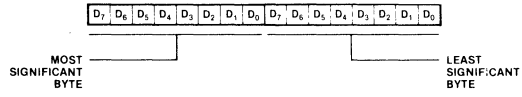
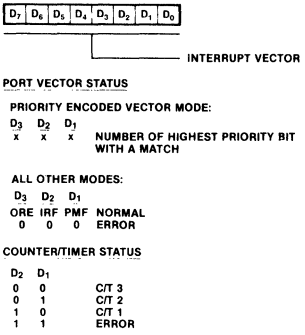


Figure 17. Counter/Timer Registers (Continued)

Registers (Continued)

Interrupt Vector Register
Addresses: 000010 Port A
000011 Port B
000100 Counter/Timers
(Read/Write)



Current Vector Register
Address: 011111
(Read only)

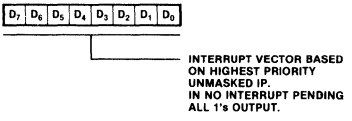


Figure 18. Interrupt Vector Registers

Register Address Summary

Main Control Registers

Address (AD_7-AD_0)	Register Name
000000XX	Master Interrupt Control
000001XX	Master Configuration Control
000010XX	Port A's Interrupt Vector
000011XX	Port B's Interrupt Vector
000100XX	Counter/Timer's Interrupt Vector
000101XX	Port C's Data Path Polarity
000110XX	Port C's Data Direction
000111XX	Port C's Special I/O Control

Most Often Accessed Registers

Address (AD_7-AD_0)	Register Name
001000XX	Port A's Command and Status
001001XX	Port B's Command and Status
001010XX	Counter/Timer 1's Control
001011XX	Counter/Timer 2's Control
001100XX	Counter/Timer 3's Control
001101XX	Port A's Data (can be accessed directly)
001110XX	Port B's Data (can be accessed directly)
001111XX	Port C's Data (can be accessed directly)

Counter/Timer Related Registers

Address (AD_7-AD_0)	Register Name
010000XX	Counter/Timer 1's Current Count-MSBs
010001XX	Counter/Timer 1's Current Count-LSBs
010010XX	Counter/Timer 2's Current Count-MSBs
010011XX	Counter/Timer 2's Current Count-LSBs
010100XX	Counter/Timer 3's Current Count-MSBs
010101XX	Counter/Timer 3's Current Count-LSBs
010110XX	Counter/Timer 1's Time Constant-MSBs
010111XX	Counter/Timer 1's Time Constant-LSBs

Counter/Timer Related Registers (Continued)

Address (AD_7-AD_0)	Register Name
011000XX	Counter/Timer 2's Time Constant-MSBs
011001XX	Counter/Timer 2's Time Constant-LSBs
011010XX	Counter/Timer 3's Time Constant-MSBs
011011XX	Counter/Timer 3's Time Constant-LSBs
011100XX	Counter/Timer 1's Mode Specification
011101XX	Counter/Timer 2's Mode Specification
011110XX	Counter/Timer 3's Mode Specification
011111XX	Current Vector

Port A Specification Registers

Address (AD_7-AD_0)	Register Name
100000XX	Port A's Mode Specification
100001XX	Port A's Handshake Specification
100010XX	Port A's Data Path Polarity
100011XX	Port A's Data Direction
100100XX	Port A's Special I/O Control
100101XX	Port A's Pattern Polarity
100110XX	Port A's Pattern Transition
100111XX	Port A's Pattern Mask

Port B Specification Registers

Address (AD_7-AD_0)	Register Name
101000XX	Port B's Mode Specification
101001XX	Port B's Handshake Specification
101010XX	Port B's Data Path Polarity
101011XX	Port B's Data Direction
101100XX	Port B's Special I/O Control
101101XX	Port B's Pattern Polarity
101110XX	Port B's Pattern Transition
101111XX	Port B's Pattern Mask

Timing

Read Cycle. At the beginning of a read cycle, the CPU places an address on the address bus. Bits A_0 and A_1 specify a CIO register; the remaining address bits and status information are combined and decoded to generate a Chip Enable ($\overline{CE}$) signal that selects the CIO. When Read ($\overline{RD}$) goes Low, data from the specified register is gated onto the data bus.

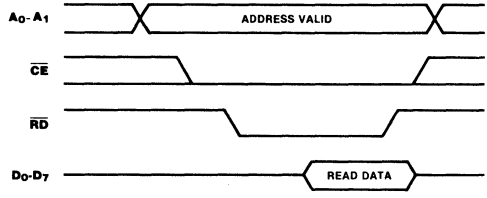


Figure 19. Read Cycle Timing

Write Cycle. At the beginning of a write cycle, the CPU places an address on the data bus. Bits A_0 and A_1 specify a CIO register; the remaining address bits and status information are combined and decoded to generate a Chip Enable ($\overline{CE}$) signal that selects the CIO. When $\overline{WR}$ goes Low, data placed on the bus by the CPU is strobed into the specified CIO register.

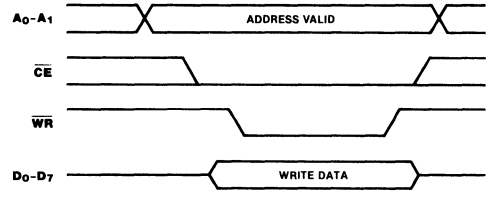


Figure 20. Write Cycle Timing

Interrupt Acknowledge. The CIO 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 Acknowledge cycle. When Interrupt Acknowledge ($\overline{INTACK}$) goes true and the IP is set, the CIO 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.

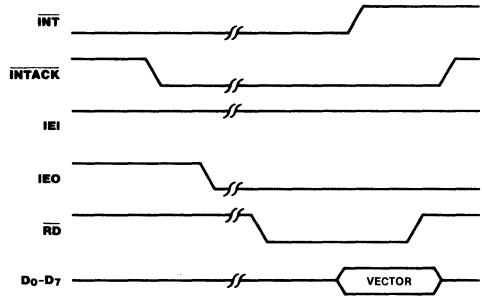


Figure 21. Interrupt Acknowledge Timing

Absolute Maximum Rating

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 ≤ 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.

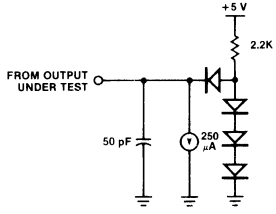


Figure 22. Standard Test Load

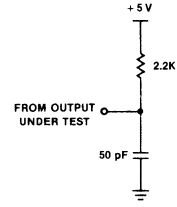


Figure 23. Open-Drain Test Load

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	

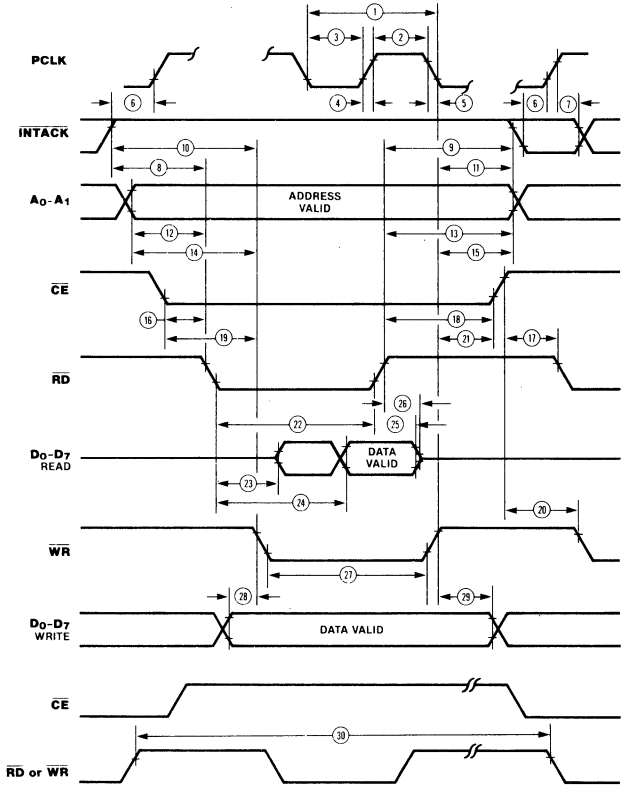
f = 1 MHz, over specified temperature range.

CPU Interface Timing

Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TcPC	PCLK Cycle Time	250	4000	ns	
2	TwPCh	PCLK Width (High)	105	2000	ns	
3	TwPCl	PCLK Width (Low)	105	2000	ns	
4	TrPC	PCLK Rise Time		20	ns	
5	TfPC	PCLK Fall Time		20	ns	
6	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	100		ns	
7	ThIA(PC)	$\overline{INTACK}$ To PCLK ↑ Hold Time	0		ns	
8	TsIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time	200		ns	1
9	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		ns	
10	TsIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time	200		ns	
11	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		ns	
12	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		ns	
13	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		ns	
14	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		ns	
15	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		ns	
16	TsCEl(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time	0		ns	1
17	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time	100		ns	1
18	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time	0		ns	1
19	TsCEl(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		ns	
20	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		ns	
21	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		ns	
22	TwRDl	$\overline{RD}$ Low Width	390		ns	1
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		ns	
24	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		255	ns	
25	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		ns	
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay		70	ns	2
27	TwWRl	$\overline{WR}$ Low Width	390		ns	
28	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	0		ns	
29	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		ns	
30	Trc	Valid Access Recovery Time	1000*		ns	3

1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is measured to the time when the output has changed 0.5 V with minimum ac load and maximum dc load.
3. Trc is 1μS or 3 TcPC, whichever is longer.

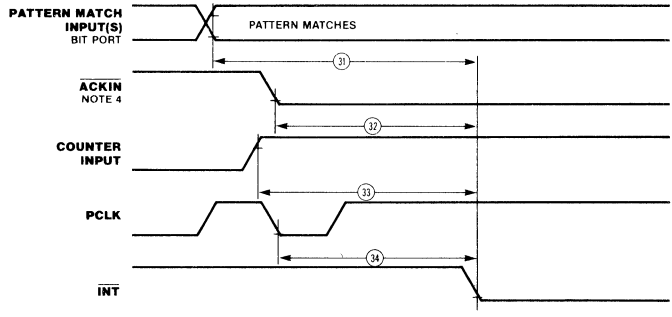
CPU Interface Timing



Interrupt Timing

Number	Symbol	Parameter	Min	Max	Units	Notes*
31	TdPM(INT)	Pattern Match to INT Delay (Bit Port)		2	TcPC + ns	
32	TdACK(INT)	$\overline{ACKIN}$ to $\overline{INT}$ Delay (Port with Handshake)		10	TcPC + ns	4
33	TdCI(INT)	Counter Input to $\overline{INT}$ Delay (Counter Mode)		2	TcPC + ns	
34	TdPC(INT)	PCLK to $\overline{INT}$ Delay (Timer Mode)		3	TcPC + ns	

4. The delay is from $\overline{DAV}$ ↓ for 3-Wire Input Handshake.
The delay is from DAC ↑ for 3-Wire Output Handshake.



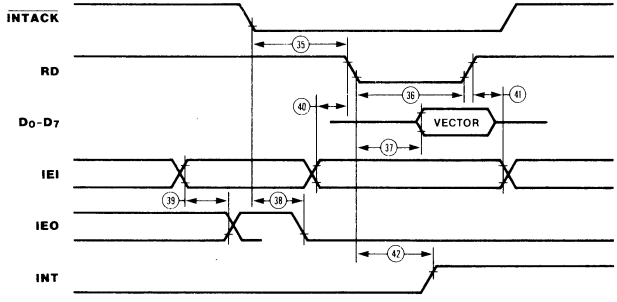
Interrupt Acknowledge Timing

Number	Symbol	Parameter	Min	Max	Units	Notes*
35	TsIA(RDA)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Setup Time	350		ns	5
36	TwRDA	$\overline{RD}$ (Acknowledge Width)	350		ns	
37	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		255	ns	
38	TdIA(IEO)	$\overline{INTACK}$ ↓ to IEO ↓ Delay		350	ns	5
39	TdIEI(IEO)	IEI to IEO Delay		150	ns	5
40	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	100		ns	5
41	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	100		ns	
42	TdRDA(INT)	$\overline{RD}$ ↓ (Acknowledge) to $\overline{INT}$ ↑ Delay		600	ns	

NOTES:

5. The parameters for the devices in any particular daisy chain must meet the following constraint: The delay from $\overline{INTACK}$ ↓ to $\overline{RD}$ ↓ must be greater than the sum of TdIA(IEO) for the highest priority peripheral, TsIEI(RDA) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.

*Timings are preliminary and subject to change.

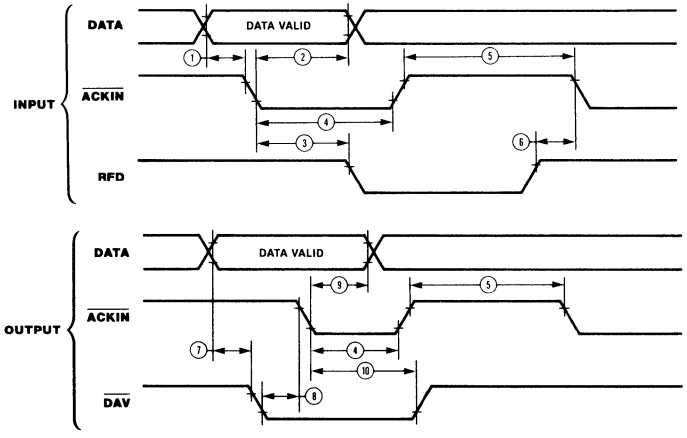


Handshake Timing

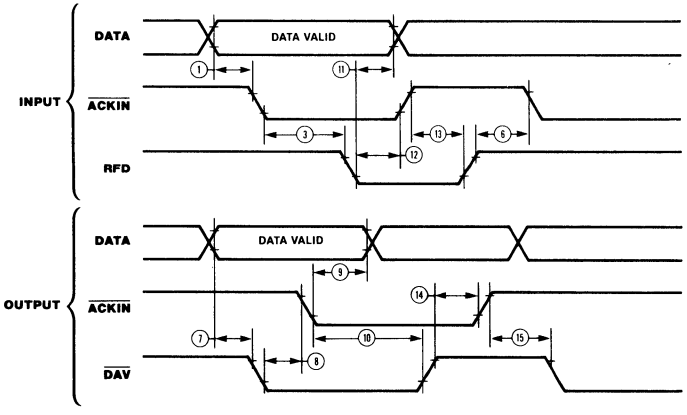
Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TsDI(ACK)	Data Input to $\overline{ACKIN}$ ↓ Setup Time	0		ns	
2	ThDI(ACK)	Data Input to $\overline{ACKIN}$ ↓ Hold Time—Strobed Handshake			ns	
3	TdACKf(RFD)	$\overline{ACKIN}$ ↓ to RFD ↓ Delay	0		ns	
4	TwACKl	$\overline{ACKIN}$ Low Width—Strobed Handshake			ns	
5	TwACKh	$\overline{ACKIN}$ High Width—Strobed Handshake			ns	
6	TdRFDr(ACK)	RFD ↑ to $\overline{ACKIN}$ ↓ Delay	0		ns	
7	TsDO(DAV)	Data Out to $\overline{DAV}$ ↓ Setup Time	25		ns	1
8	TdDAVf(ACK)	$\overline{DAV}$ ↓ to $\overline{ACKIN}$ ↓ Delay	0		ns	
9	ThDO(ACK)	Data Out to $\overline{ACKIN}$ ↓ Hold Time	2		TcPC	
10	TdACK(DAV)	$\overline{ACKIN}$ ↓ to $\overline{DAV}$ ↑ Delay	2		TcPC	
11	ThDI(RFD)	Data Input to RFD ↓ Hold Time—Interlocked Handshake	0		ns	
12	TdRFDf(ACK)	RFD ↓ to $\overline{ACKIN}$ ↑ Delay—Interlocked Handshake	0		ns	
13	TdACKr(RFD)	$\overline{ACKIN}$ ↑ ($\overline{DAV}$ ↑) to RFD ↑ Delay—Interlocked and 3-Wire Handshake	0		ns	
14	TdDAVr(ACK)	$\overline{DAV}$ ↑ to $\overline{ACKIN}$ ↑ (RFD ↑)—Interlocked and 3-Wire Handshake	0		ns	
15	TdACK(DAV)	$\overline{ACKIN}$ ↑ (RFD ↑) to $\overline{DAV}$ ↓ Delay—Interlocked and 3-Wire Handshake	0		ns	
16	TdDAVIf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay—Input 3-Wire Handshake	0		ns	
17	ThDI(DAC)	Data Input to DAC ↑ Hold Time—3-Wire Handshake	0		ns	
18	TdDACOr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay—Input 3-Wire Handshake	0		ns	
19	TdDAVIr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay—Input 3-Wire Handshake	0		ns	
20	TdDAVOf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay—Output 3-Wire Handshake	0		ns	
21	ThDO(DAC)	Data Output to DAC ↑ Hold Time—3-Wire Handshake	2		TcPC	
22	TdDACIr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay—Output 3-Wire Handshake	2		TcPC	
23	TdDAVOr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay—Output 3-Wire Handshake	0		ns	

NOTES:
1. This time can be extended through the use of the deskew timers. *Timings are preliminary and subject to change.

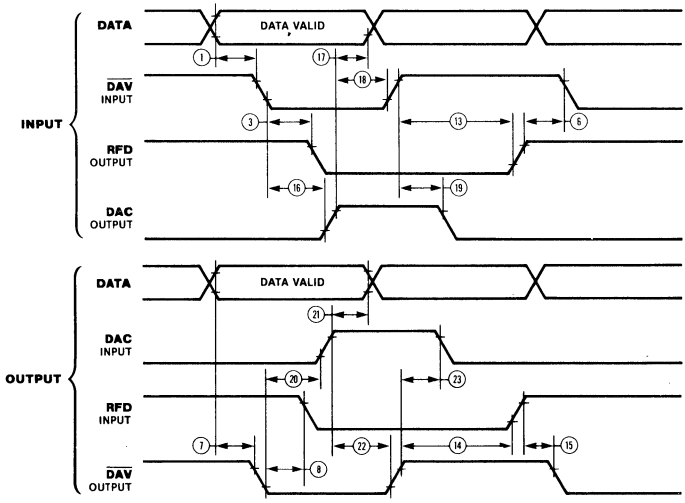
Strobed Handshake



Interlocked Handshake



3-Wire Handshake

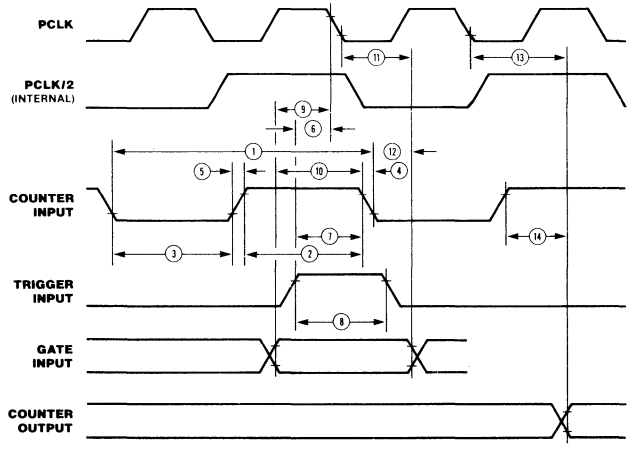


Counter/Timer Timing

Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TcCI	Counter Input Cycle Time	500		ns	
2	TCIh	Counter Input High Width	230		ns	
3	TwCIl	Counter Input Low Width	230		ns	
4	TfCI	Counter Input Fall Time		20	ns	
5	TrCI	Counter Input Rise Time		20	ns	
6	TsTI(PC)	Trigger Input to PCLK ↓ Setup Time (Timer Mode)			ns	1
7	TsTI(CI)	Trigger Input to Counter Input ↓ Setup Time (Counter Mode)			ns	1
8	TwTI	Trigger Input Pulse Width (High or Low)			ns	
9	TsGI(PC)	Gate Input to PCLK ↓ Setup Time (Timer Mode)			ns	1
10	TsGI(CI)	Gate Input to Counter Input ↓ Setup Time (Counter Mode)			ns	1
11	ThGI(PC)	Gate Input to PCLK ↓ Hold Time (Timer Mode)			ns	1
12	ThGI(CI)	Gate Input to Counter Input ↓ Hold Time (Counter Mode)			ns	1
13	TdPC(CO)	PCLK to Counter Output Delay (Timer Mode)			ns	
14	TdCI(CO)	Counter Input to Counter Output Delay (Counter Mode)			ns	

NOTES:
1. These parameters must be met to guarantee trigger or gate are valid for the next counter/timer cycle.

*Timings are preliminary and subject to change.

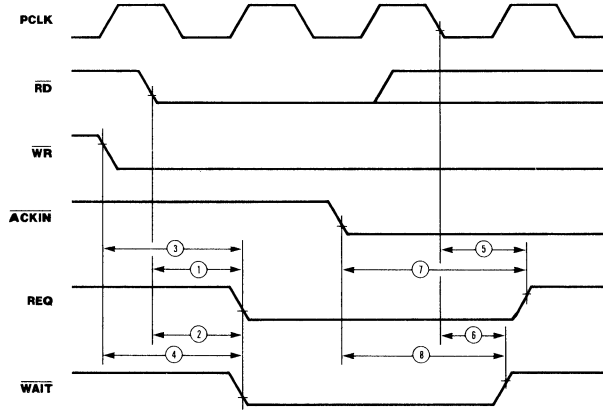


REQUEST WAIT Timing

Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TdRD(REQ)	$\overline{RD}$ ↓ to REQ ↓ Delay			ns	
2	TdRD(WAIT)	$\overline{RD}$ ↓ to $\overline{WAIT}$ ↓ Delay			ns	
3	TdWR(REQ)	$\overline{WR}$ ↓ to REQ ↓ Delay			ns	
4	TdWR(WAIT)	$\overline{WR}$ ↓ to $\overline{WAIT}$ ↓ Delay			ns	
5	TdPC(REQ)	PCLK ↓ to REQ ↑ Delay			ns	
6	TdPC(WAIT)	PCLK ↓ to $\overline{WAIT}$ ↑ Delay			ns	
7	TdACK(REQ)	$\overline{ACKIN}$ ↓ to REQ ↑ Delay			TcPC + ns	1
8	TdACK(WAIT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ ↑ Delay			TcPC + ns	1

NOTES:
1. The delay is from $\overline{DAV}$ ↓ for 3-Wire Input Handshake.
   The delay is from DAC ↑ for 3-Wire Output Handshake.

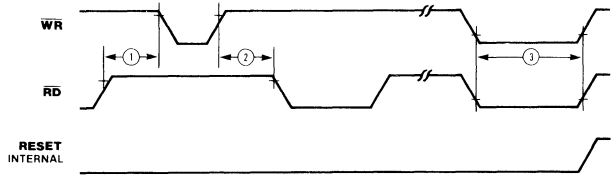
*Timings are preliminary and subject to change.



Reset Timing

Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TdRD(WR)	Delay from $\overline{RD}$ ↑ to $\overline{WR}$ ↓ for No Reset	50		ns	
2	TdWR(RD)	Delay from $\overline{WR}$ ↑ to $\overline{RD}$ ↓ for No Reset	50		ns	
3	TwRES	Minimum Width of $\overline{RD}$ and $\overline{WR}$ both Low for Reset	250		ns	

*Timings are preliminary and subject to change.

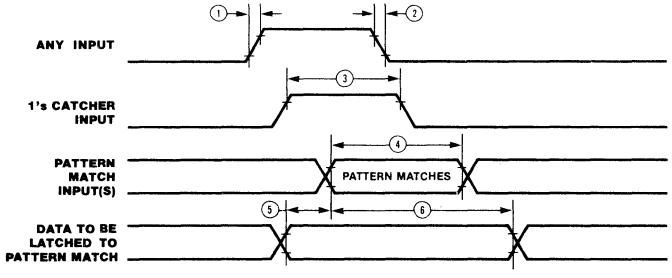


Miscellaneous Port Timing

Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TrI	Any Input Rise Time		100	ns	
2	TfI	Any Input Fall Time		100	ns	
3	Twl's	1's Catcher High Width	250		ns	1
4	TwPM	Pattern Match Input Valid (Bit Port)	750		ns	
5	TsPMD	Data Latched on Pattern Match Setup Time (Bit Port)	0		ns	
6	ThPMD	Data Latched on Pattern Match Hold Time (Bit Port)	1000		ns	

NOTES:
1. If the input is programmed inverting, a Low-going pulse of the same width will be detected.

*Timings are preliminary and subject to change.

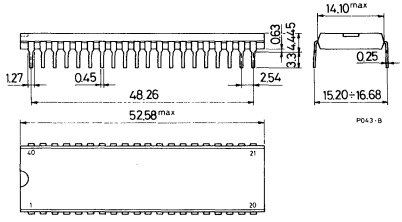


Ordering Information

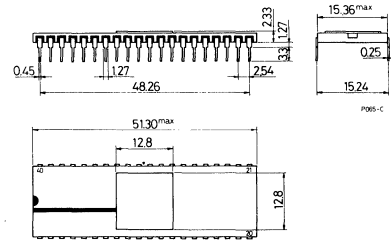
Type	Package	Temp	Clock	Description
Z8536 B1	Plastic 40 pin	0/+70°C	4MHz	Z8536 Counter/Timer and Parallel I/O Unit.
B6	Plastic 40 pin	-40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D2	Ceramic 40 pin	-55/+125°C		
D6	Ceramic 40 pin	-40/+85°C		
Z8536A B1	Plastic 40 pin	0/+70°C	6MHz	
B6	Plastic 40 pin	-40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D6	Ceramic 40 pin	-40/+85°C		

Packages

Plastic



Ceramic



FIFO Input/Output Interface Unit

Z8538 FIO

Features

- 128-byte FIFO buffer provides asynchronous bidirectional CPU/CPU or CPU/peripheral interface, expandable to any width in byte increments by use of multiple Z8060 FIO's
- Interlocked 2-Wire or 3-Wire Handshake logic port mode; Z-BUS or non-Z-BUS interface.
- Pattern-recognition logic stops DMA transfers and/or interrupts CPU; preset byte count can initiate variable-length DMA transfers.
- Seven sources of vectored/nonvectored interrupt which include pattern-match, byte count, empty or full buffer status; a dedicated "mailbox" register with interrupt capability provides CPU/CPU communication.
- $\overline{\text{REQUEST}}/\overline{\text{WAIT}}$ lines control high-speed data transfers.
- All functions are software controlled via directly addressable read/write registers.

General Description

The Z8538 FIO provides an asynchronous 128-byte FIFO buffer between two CPUs or between a CPU and a peripheral device. This buffer interface expands to a 16-bit or wider data path and expands in depth to add as many Z8060 FIFOs (and an additional FIO) as are needed.

The FIO manages data transfers by assuming Z-BUS, non-Z-BUS microprocessor (a generalized microprocessor interface), Interlocked 2-Wire Handshake, and 3-Wire Handshake operating modes. These modes interface dissimilar CPUs or CPUs and peripherals running under differing speeds or protocols, allowing asynchronous data transactions and improving I/O overhead by as much as two orders of magnitude. Figures 1 and 2 show how the signals controlling these operating modes are mapped to the FIO pins.

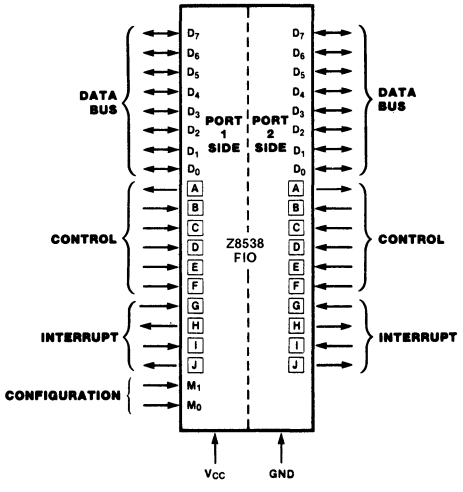


Figure 1. Logic Functions

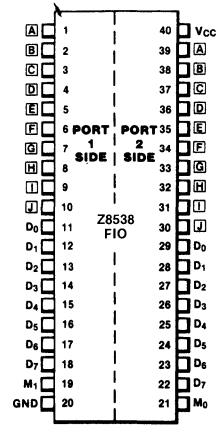


Figure 2. Pin Configuration

General Description (Continued)

The FIO supports the Z-BUS interrupt protocols, generating seven sources of interrupts upon any of the following events: a write to a message register, change in data direction, pattern match, status match, over/underflow error, buffer full and buffer empty status. Each interrupt source can be enabled or disabled, and can also place an interrupt vector on the port address/data lines.

The data transfer logic of the FIO has been specially designed to work with DMA (Direct Memory Access) devices for high-speed transfers. It provides for data transfers to or from memory each machine cycle, while the DMA device generates memory address and control signals. The FIO also supports the variably sized block length, improving system throughput when multiple variable length messages are transferred amongst several sources.

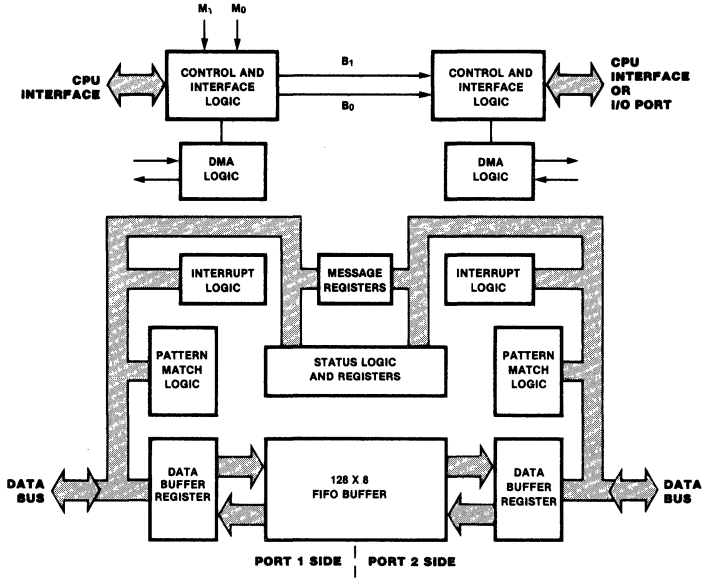


Figure 3. FIO Block Diagram

Functional Description

Operating Modes. Ports 1 and 2 operate in any of twelve combinations of operating modes, listed in Table 2. Port 1 functions in either the Z-BUS or non-Z-BUS microprocessor modes, while Port 2 functions in Z-BUS, non-Z-BUS, Interlocked 2-Wire Handshake, and 3-Wire Handshake modes. Table 1 describes the signals and their corresponding pins in each of these modes.

The pin diagrams of the FIO are identical, except for two pins on the Port 1 side, which select that port's operating mode. Port 2's operating mode is programmed by two bits in Port 1's Control register 0. Table 2 describes the combinations of operating modes; Table 3 describes the control signals mapped to pins A-J in the five possible operating modes.

Functional Description (Continued)

Control Signal Pins	Z-BUS Low Byte	Z-BUS High Byte	Non-Z-BUS	Interlocked HS Port*	3-Wire HS Port*
A	REQ/WT	REQ/WT	REQ/WT	RFD/DAV	RFD/DAV
B	DMASTB	DMASTB	DACK	ACKIN	DAV/DAC
C	DS	DS	RD	FULL	DAC/RFD
D	R/W	R/W	WR	EMPTY	EMPTY
E	CS	CS	CE	CLEAR	CLEAR
F	AS	AS	C/D	DATA DIR	DATA DIR
G	INTACK	A_0	INTACK	IN_0	IN_0
H	IEO	A_1	IEO	OUT_1	OUT_1
I	IEI	A_2	IEI	OE	OE
J	INT	A_3	INT	OUT_3	OUT_3

*2 side only.

Table 1. Pin Assignments

Mode	M_1	M_0	B_1	B_0	Port 1	Port 2
0	0	0	0	0	Z-BUS Low Byte	Z-BUS Low Byte
1	0	0	0	1	Z-BUS Low Byte	Non-Z-BUS
2	0	0	1	0	Z-BUS Low Byte	3-Wire Handshake
3	0	0	1	1	Z-BUS Low Byte	2-Wire Handshake
4	0	1	0	0	Z-BUS High Byte	Z-BUS High Byte
5	0	1	0	1	Z-BUS High Byte	Non-Z-BUS
6	0	1	1	0	Z-BUS High Byte	3-Wire Handshake
7	0	1	1	1	Z-BUS High Byte	2-Wire Handshake
8	1	0	0	0	Non-Z-BUS	Z-BUS Low Byte
9	1	0	0	1	Non-Z-BUS	Non-Z-BUS
10	1	0	1	0	Non-Z-BUS	3-Wire Handshake
11	1	0	1	1	Non-Z-BUS	2-Wire Handshake

Table 2. Operating Modes

Functional Description (Continued)

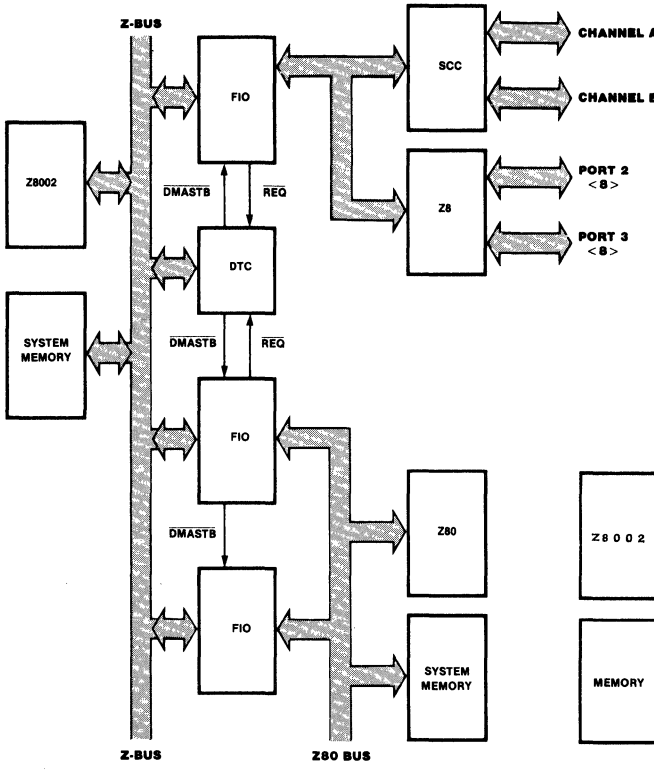


Figure 4. CPU to CPU Configuration

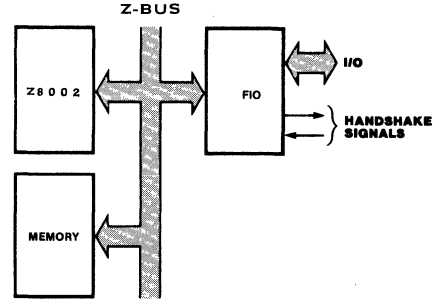


Figure 5. CPU to I/O Configuration

Pins Common To Both Sides

Pin Signals	Pin Names	Pin Numbers	Signal Description
M_0	M_0	21	M_1 and M_0 program Port 1 side CPU interface
M_1	M_1	19	
+5 Vdc	+5 Vdc	40	DC power source
GND	GND	20	DC power ground

Z-BUS Low Byte Mode

Pin Signals	Pin Names	Pin Numbers Port 1	Pin Numbers Port 2	Signal Description
AD_0-AD_7 (Address/Data)	D_0-D_7	11-18	29-22	Multiplexed bidirectional address/data lines, Z-BUS compatible.
$\overline{REQ}/\overline{WAIT}$ (Request/Wait)	A	1	39	Output, active Low, REQUEST (ready) line for DMA transfer; $\overline{WAIT}$ line (open-drain) output for synchronized CPU and FIO data transfers.
$\overline{DMASTB}$ (Direct Memory Access Strobe)	B	2	38	Input, active Low. Strobes DMA data to and from the FIFO buffer.
$\overline{DS}$ (Data Strobe)	C	3	37	Input, active Low. Provides timing for data transfer to or from FIO.
R/$\overline{W}$ (Read/Write)	D	4	36	Input; active High signals CPU read from FIO; active Low signals CPU write to FIO.
$\overline{CS}$ (Chip Select)	E	5	35	Input, active Low. Enables FIO. Latched on the rising edge of $\overline{AS}$.
$\overline{AS}$ (Address Strobe)	F	6	34	Input, active Low. Addresses, $\overline{CS}$ and $\overline{INTACK}$ sampled while $\overline{AS}$ Low.
$\overline{INTACK}$ (Interrupt Acknowledge)	G	7	33	Input, active Low. Acknowledges an interrupt. Latched on the rising edge of $\overline{AS}$.
IEO (Interrupt Enable Out)	H	8	32	Output, active High. Sends interrupt enable to lower priority device IEI pin.
IEI (Interrupt Enable In)	I	9	31	Input, active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (Interrupt)	J	10	30	Output, open drain, active Low. Signals FIO interrupt request to CPU.

Table 3. Signal/Pin Descriptions

Z-BUS High Byte Mode (Continued)

Pin Signals	Pin Names	Pin Numbers Port 1	Pin Numbers Port 2	Signal Description
AD_0-AD_7 (Address/Data)	D_0-D_7	11-18	29-22	Multiplexed bidirectional address/data lines, Z-BUS compatible.
$\overline{REQ}/\overline{WAIT}$ (Request/Wait)	A	1	39	Output, active Low, REQUEST (ready) line for DMA transfer; $\overline{WAIT}$ line (open-drain) output for synchronized CPU and FIO data transfers.
$\overline{DMASTB}$ (Direct Memory Access Strobe)	B	2	38	Input, active Low. Strobes DMA data to and from the FIFO buffer.
$\overline{DS}$ (Data Strobe)	C	3	37	Input, active Low. Provides timing for transfer of data to or from FIO.
$R/\overline{W}$ (Read/Write)	D	4	36	Input, active High. Signals CPU read from FIO; active Low signals CPU write to FIO.
$\overline{CS}$ (Chip Select)	E	5	35	Input, active Low. Enables FIO. Latched on the rising edge of $\overline{AS}$.
$\overline{AS}$ (Address Strobe)	F	6	34	Input, active Low. Addresses, $\overline{CS}$ and $\overline{INTACK}$ are sampled while $\overline{AS}$ is Low.
A_0 (Address Bit 0)	G	7	33	Input, active High. With A_1, A_2, and A_3, addresses FIO internal registers.
A_1 (Address Bit 1)	H	8	32	Input, active High. With A_0, A_2, and A_3, addresses FIO internal registers.
A_2 (Address Bit 2)	I	9	31	Input, active High. With A_0, A_1, and A_3, addresses FIO internal registers.
A_3 (Address Bit 3)	J	10	30	Input, active High. With A_0, A_1, and A_2, addresses FIO internal registers.

Table 3. Signal/Pin Descriptions (Continued)

Non-Z-BUS Mode

Pin Signals	Pin Names	Pin Numbers Port 1	Pin Numbers Port 4	Signal Description
D_0-D_7 (Data)	D_0-D_7	11-18	29-22	Bidirectional data bus.
$\overline{REQ}/\overline{WT}$ (Request/Wait)	A	1	39	Output, active Low, REQUEST (ready) line for DMA transfer; $\overline{WAIT}$ line (open-drain) output for synchronized CPU and FIO data transfer.
$\overline{DACK}$ (DMA Acknowledge)	B	2	38	Input, active Low. DMA acknowledge.
$\overline{RD}$ (Read)	C	3	37	Input, active Low. Signals CPU read from FIO.
$\overline{WR}$ (Write)	D	4	36	Input, active Low. Signals CPU write to FIO.
$\overline{CE}$ (Chip Select)	E	5	35	Input, active Low. Used to select FIO.
$C/\overline{D}$ (Control/Data)	F	6	34	Input, active High. Identifies control byte on D_0-D_7; active Low identifies data byte on D_0-D_7.
$\overline{INTACK}$ (Interrupt Acknowledge)	G	7	33	Input, active Low. Acknowledges an interrupt.
IEO (Interrupt Enable Out)	H	8	32	Output, active High. Sends interrupt enable to lower priority device IEI pin.
IEI (Interrupt Enable In)	I	9	31	Input, active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (Interrupt)	J	10	30	Output, open drain, active Low. Signals FIO interrupt to CPU.

Table 3. Signal/Pin Descriptions (Continued)

Port 2 – I/O Port Mode

Pin Signals	Pin Names	Pin Numbers	Mode	Signal Description
D_0-D_7 (Data)	D_0-D_7	29-22	2-Wire HS* 3-Wire HS	Bidirectional data bus.
RFD/$\overline{DAV}$ (Ready for Data/Data Available)	A	39	2-Wire HS 3-Wire HS	Output, RFD active High. Signals peripherals that FIO is ready to receive data. $\overline{DAV}$ active Low signals that FIO is ready to send data to peripherals.
$\overline{ACKIN}$ (Acknowledge Input)	B	38	2-Wire HS	Input, active Low. Signals FIO that output data is received by peripherals or that input data is valid.
$\overline{DAV}$/DAC (Data Available/Data Accepted)	B	38	3-Wire HS	Input; $\overline{DAV}$ (active Low) signals that data is valid on bus. DAC (active High) signals that output data is accepted by peripherals.
FULL	C	37	2-Wire HS	Output, open drain, active High. Signals that FIO buffer is full.
DAC/RFD (Data Accepted/Ready for Data)	C	37	3-Wire HS	Direction controlled by internal programming. Both active High. DAC (an output) signals that FIO has received data from peripheral; RFD (an input) signals that the listeners are ready for data.
EMPTY	D	36	2-Wire HS 3-Wire HS	Output, open drain, active High. Signals that FIFO buffer is empty.
$\overline{CLEAR}$	E	35	2-Wire HS 3-Wire HS	Programmable input or output, active Low. Clears all data from FIFO buffer.
DATA DIR (Data Direction)	F	34	2-Wire HS 3-Wire HS	Programmable input or output. Active High signals data input to Port 2; Low signals data output from Port 2.
IN_0	G	33	2-Wire HS 3-Wire HS	Input line to D_0 of Control Register 3.
OUT_1	H	32	2-Wire HS 3-Wire HS	Output line from D_1 of Control Register 3.
$\overline{OE}$ (Output Enable)	I	31	2-Wire HS 3-Wire HS	Input, active Low. When Low, enables bus drivers. When High, floats bus drivers at high impedance.
OUT_3	J	30	2-Wire HS 3-Wire HS	Output line from D_3 of Control register 3.

*Handshake

Table 3. Signal/Pin Descriptions (Continued)

Reset

The FIO can be reset under either hardware or software control by one of the following methods:

- By forcing both $\overline{AS}$ and $\overline{DS}$ Low simultaneously in Z-BUS mode (normally illegal).
- By forcing $\overline{RD}$ and $\overline{WR}$ Low simultaneously in non-Z-BUS mode.
- By writing a 1 to the Reset bit in Control register 0 for software reset.

In the Reset state, all control bits are cleared to 0. Only after clearing the Reset bit (by writing a 0 to it) can the other command bits be programmed. This action is true for both sides of the FIO when programmed as a CPU interface.

For proper system control, when Port 1 is reset, Port 2 is also reset. In addition, all Port 2's outputs are floating and all inputs are ignored. To initiate the data transfer, Port 2 must be enabled by Port 1. The Port 2 CPU can determine when it is enabled by reading Control register 0, which reads "floating" data bus if not enabled and "01_H" if enabled.

CPU Interfaces

The FIO is designed to work with both Z-BUS- and non-Z-BUS-type CPUs on both Port 1 and Port 2. The Z-BUS configuration interfaces CPUs with time-multiplexed address and data information on the same pins. The Z8001, Z8002, and Z8 are examples of this type of CPU. The $\overline{AS}$ (Address Strobe) pin is used to latch the address and chip select information sent out by the CPU. The R/$\overline{W}$ (Read/Write) pin and the $\overline{DS}$ (Data Strobe) pin are used for timing reads and writes from the CPU to the FIO (Figures 6 and 7).

The non-Z-BUS configuration is used for CPUs where the address and data buses are separate. Examples of this type of CPU are the Z80 and 8080. The $\overline{RD}$ (Read) and $\overline{WR}$ (Write) pins are used to time reads and writes from the CPU to the FIO (Figures 9 and 10). The C/$\overline{D}$ (Control/Data) pin is used to directly access the FIFO buffer (C/$\overline{D}$ = 0) and to access the other registers (C/$\overline{D}$ = 1). Read and write to all

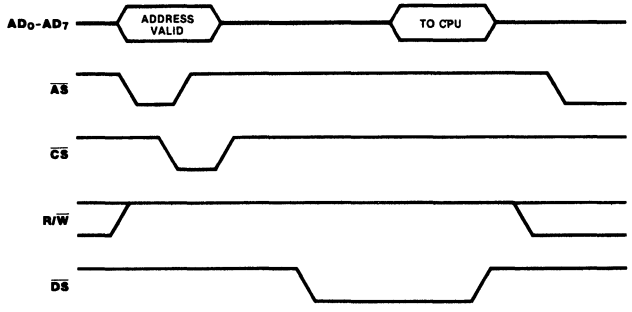


Figure 6. Z-BUS Read Cycle Timing

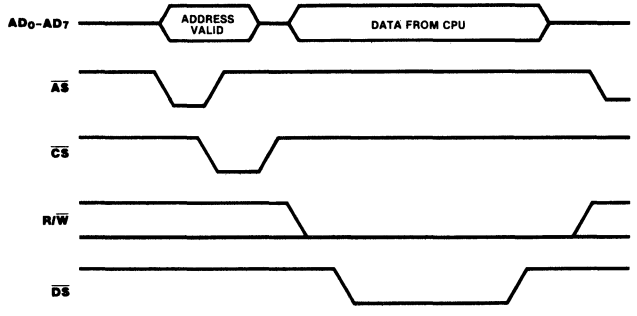


Figure 7. Z-BUS Write Cycle Timing

CPU Interfaces (Continued)

registers except the FIFO buffer[1] are two-step operations, described as follows (Figure 8). First, write the address (C/$\overline{D}$ = 1) of the register to be accessed into the Pointer Register (State 0); second, read or write (C/$\overline{D}$ = 1) to the register pointed at previously (State 1). Continuous status monitoring can be performed in State 1 by continuous Control Read operations (C/$\overline{D}$ = 1).

[1]The FIFO buffer can also be accessed by this two-step operation.

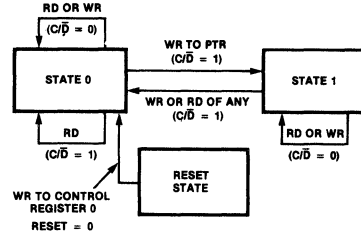


Figure 8. Register Access in Non-Z-BUS Mode

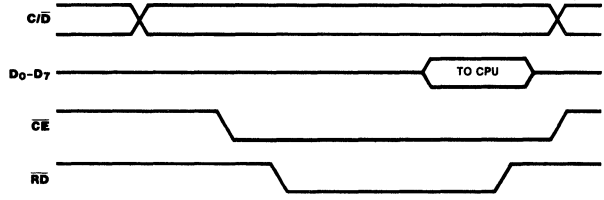


Figure 9. Non-Z-BUS Read Cycle Timing

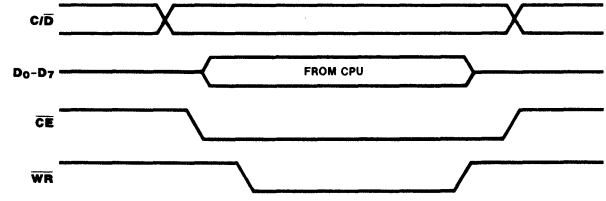


Figure 10. Non-Z-BUS Write Cycle Timing

$\overline{\text{WAIT}}$ Operation

When data is output by the CPU, the $\overline{\text{REQ}}$/$\overline{\text{WT}}$ ($\overline{\text{WAIT}}$) pin is active (Low) only when the FIFO buffer is full, the chip is selected, and the FIFO buffer is addressed. $\overline{\text{WAIT}}$ goes inactive when the FIFO buffer is not full.

When data is input by the CPU, the $\overline{\text{REQ}}$/$\overline{\text{WT}}$ pin becomes active (Low) only when the FIFO buffer is empty, the chip is selected, and the FIFO buffer is addressed. $\overline{\text{WAIT}}$ goes inactive when the FIFO buffer is not empty.

Interrupt Operation

The FIO supports the prioritized daisy chain interrupt protocol for both Z-BUS and non-Z-BUS operating modes (for more details refer to the Z-BUS Summary).

Each side of the FIO has seven sources of interrupt. The priorities of these devices are fixed in the following order (highest to lowest): Mailbox Message, Change in Data Direction, Pattern Match, Status Match, Overflow/ Underflow Error, Buffer Full, and Buffer Empty. Each interrupt source has three bits that control how it generates the interrupt. These bits are Interrupt Pending (IP), Interrupt Enable (IE), and Interrupt Under Service (IUS).

In addition, each side of the FIO has an interrupt vector and four bits controlling the FIO interrupt logic. These bits are Vector Includes Status (VIS), Master Interrupt Enable (MIE), Disable Lower Chain (DLC), and No Vector (NV).

A typical Interrupt Acknowledge cycle for Z-BUS operation is shown in Figure 11 and for non-Z-BUS operation in Figure 12. The only difference is that in Z-BUS mode, $\overline{\text{INTACK}}$ is latched by $\overline{\text{AS}}$, and in non-Z-BUS mode $\overline{\text{INTACK}}$ is not latched.

When MIE = 1, reading the vector always includes status, independent of the state of the VIS bit. In this way, when VIS = 0, all information can be obtained with one additional read, thus conserving vector space. When MIE = 0, reading the vector register returns the unmodified base vector so that it can be verified.

In non-Z-BUS mode, IPs do not get set while in State 1. Therefore, in order to minimize interrupt latency, the FIO should be left in State 0.

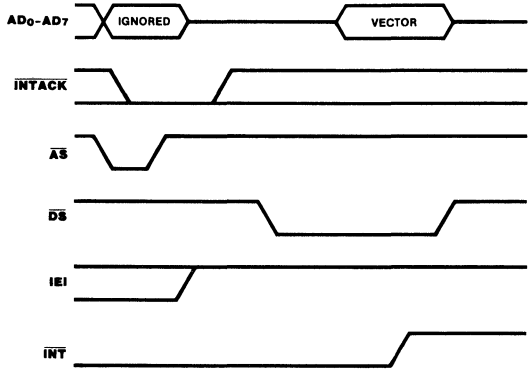


Figure 11. Z-BUS Interrupt Acknowledge Cycle

Interrupt Operation (Continued)

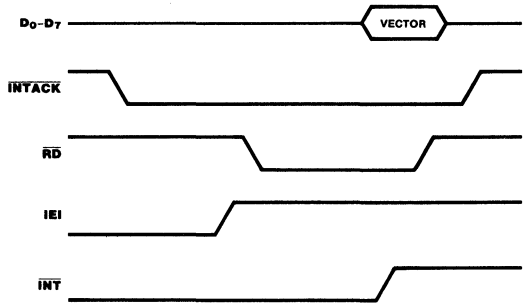


Figure 12. Non-Z-BUS Interrupt Acknowledge Cycle

CPU to CPU Operation

DMA Operation. The FIO is particularly well suited to work with a DMA in both Z-BUS and non-Z-BUS modes. A data transfer between the FIO and system memory can take place during every machine cycle on both sides of the FIO simultaneously.

In Z-BUS mode, the $\overline{\text{DMASTB}}$ pin (DMA Strobe) is used to read or write into the FIFO buffer. The R/$\overline{\text{W}}$ (Read/Write) and $\overline{\text{DS}}$ (Data Strobe) signals are ignored by the FIO; however, the $\overline{\text{CS}}$ (Chip Select) signal is not ignored and therefore must be kept invalid. Figures 13 and 14 show typical timing.

In Non-Z-BUS mode, the $\overline{\text{DACK}}$ pin (DMA Acknowledge) is used to tell the FIO that its DMA request is granted. After $\overline{\text{DACK}}$ goes Low, every read or write to the FIO goes into the FIFO buffer. Figures 15 and 16 show typical timing.

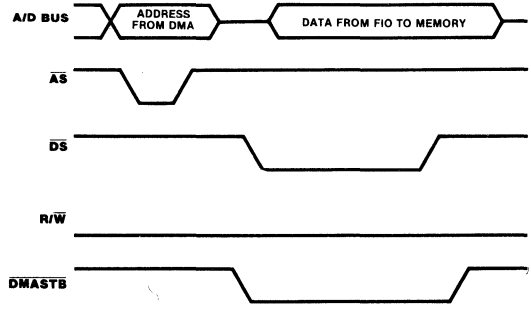


Figure 13. Z-BUS FIO to Memory Data Transaction

CPU to CPU Operation (Continued)

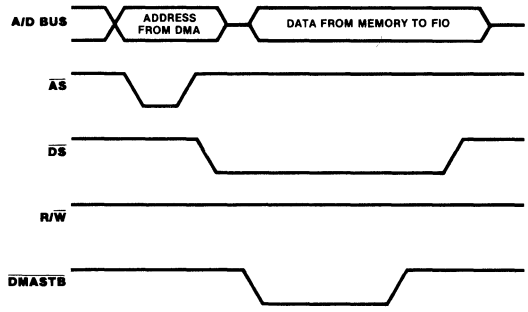


Figure 14. Z-BUS Memory to FIO Data Transaction

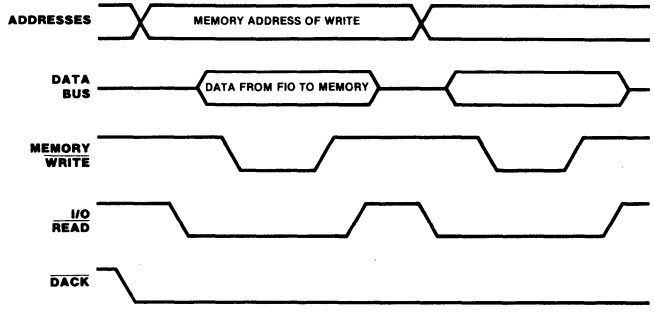


Figure 15. Non-Z-BUS FIO to Memory Transaction

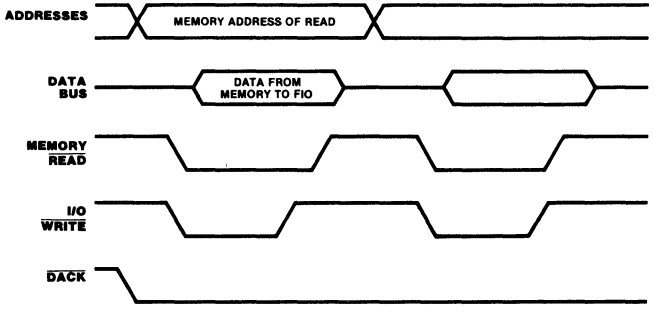
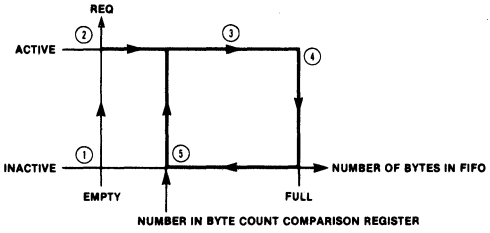


Figure 16. Non-Z-BUS Memory to FIO Data Transaction

CPU to CPU Operation (Continued)

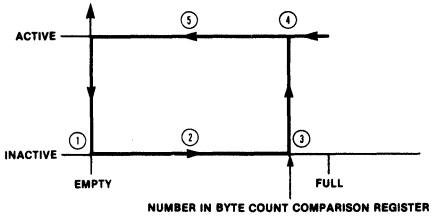
The FIO provides a special mode to enhance its DMA transfer capability. When data is written into the FIFO buffer, the $\overline{REQ}/\overline{WT}$ (REQUEST) pin is active (Low) until the FIFO buffer is full. It then goes inactive and stays inactive until the number of bytes in the FIFO buffer is equal to the value programmed into the Byte Count Comparison register. Then the REQUEST signal goes active and the sequence starts over again (Figure 17).

When data is read from the FIO, the $\overline{REQ}/\overline{WT}$ pin (REQUEST) is inactive until the number of bytes in the FIFO buffer is equal to the value programmed in the Byte Count Comparison register. The REQUEST signal then goes active and stays active until the FIFO buffer is empty. When empty, REQUEST goes inactive and the sequence starts over again (Figure 18).



NOTES:
1. FIFO empty.
2. REQUEST enabled, FIO requests DMA transfer.
3. DMA transfers data into the FIO.
4. FIFO full, REQUEST inactive.
5. The FIFO empties from the opposite port until the number of bytes in the FIFO buffer is the same as the number programmed in the Byte Count Comparison register.

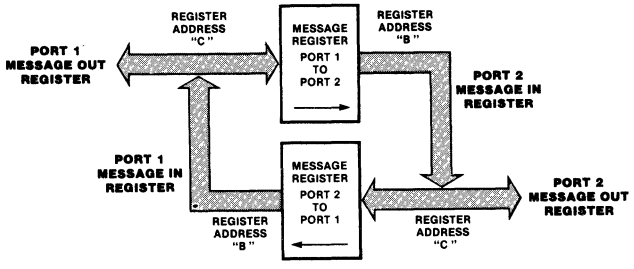
Figure 17. Byte Count Control: Write to FIO



NOTES:
1. FIFO empty.
2. CPU/DMA fills FIFO buffer from the opposite port.
3. Number of bytes in FIFO buffer is the same as the number of bytes programmed in the Byte Count Comparison register.
4. REQUEST goes active.
5. DMA transfers data out of FIFO until it is empty.

Figure 18. Byte Count Control: Read from FIO

Message Registers. Two CPUs can communicate through a dedicated "mailbox" register without involving the 128 × 8 bit FIFO buffer (Figure 19). This mailbox approach is useful for transferring control parameters between the interfacing devices on either side of the FIO without using the FIFO buffer. For example, when Port 1's CPU writes to the



NOTE: Usable only for CPU/CPU interface.

Figure 19. Message Register Operation

CPU to CPU Operation (Continued)

Message Out register, Port 2's message IP is set. If interrupts are enabled, Port 2's CPU is interrupted. Port 2's message IP status is readable from the Port 1 side. When Port 2's CPU reads the data from its Message In register, the Port 2 IP is cleared. Thus, Port 1's CPU can read when the message has been read and can now send another message or follow whatever protocol that is set up between the two CPU's. The same transfer can also be made from Port 2's CPU to Port 1's CPU.

CLEAR (Empty) FIFO Operation. The $\overline{\text{CLEAR}}$ FIFO bit (active Low) clears the FIFO buffer of data. Writing a 0 to this bit empties the FIFO buffer, inactivates the REQUEST line, and disables the handshake (if programmed). The $\overline{\text{CLEAR}}$ bit does not affect any control or data register. To remove the $\overline{\text{CLEAR}}$ state, write a 1 to the $\overline{\text{CLEAR}}$ bit.

In CPU/CPU mode, under program control, only one of the ports can empty the FIFO by writing to its Control Register 3, bit 6. The Port 1 CPU must program bit 7 in Control Register 3 to determine which port controls the $\overline{\text{CLEAR}}$ FIFO operation (0 = Port 1 control; 1 = Port 2 control).

Direction of Data Transfer Operation. The Data Direction bit controls the direction of data transfer in the FIFO buffer. The Data Direction bit is defined as 0 = output from CPU and 1 = input to CPU. This bit reads correctly when read by either port's CPU. For example, if Port 1's CPU reads a 0 (CPU output) in its Data Direction bit, then Port 2's CPU reads a 1 (input to CPU) in its Data Direction bit.

In CPU/CPU mode, under program control, only one of the ports can control the direction of data transfer. The Port 1 CPU must program bit 5 in Control Register 3 to determine which port controls the data direction (0 = Port 1 control; 1 = Port 2 control). Figure 20 shows FIO data transfer options.

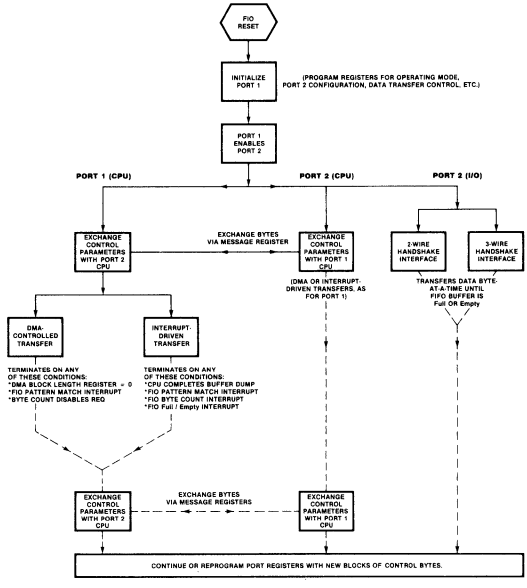


Figure 20. FIO Data Transfer Options

CPU to I/O Operation

When Port 2 is programmed in the Interlocked 2-Wire Handshake mode or the 3-Wire Handshake mode, and Port A is programmed in Z-BUS or non-Z-BUS Microprocessor mode, the FIO interfaces a CPU and a peripheral device. In the Interlocked 2-Wire Handshake mode, RFD/$\overline{DAV}$ and $\overline{ACKIN}$ strobe data to and from Port 2. In the 3-Wire Handshake mode, RFD/$\overline{DAV}$, $\overline{DAV}$/DAC, and DAC/RFD signals control data flow.

Interlocked 2-Wire Handshake. In the Interlocked Handshake, the action of the FIO must be acknowledged by the other half of the handshake before the next action can take place. In output mode, Port 2 does not indicate that new data is available until the external device indicates it is ready for the data. Similarly, in input mode, Port 2 does not indicate that it is ready for new data until the data source indicates that the previous byte of the data is no longer available, thereby acknowledging Port 2's acceptance of the last byte. This allows the FIO to directly interface to a Z8's port, a CIO's port, a UPC's port, another FIO port, or another FIFO Z8060, with no external logic (Figures 21 and 22).

3-Wire Handshake. The 3-Wire Handshake is designed for applications 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 that an input port is ready for new data or that it has accepted the present data. In the 3-Wire Handshake, the rising edge of the RFD status line indicates that the port is ready for data, and the rising edge of the DAC status line indicates that the data has been accepted. With 3-Wire Handshake, the lines of many input ports can be bussed together with open-drain drivers and the output port knows when all of the ports are ready and have accepted the data. This handshake is the same handshake used in the IEEE-488 Instruments. Since the port's direction can be changed under software control, bidirectional IEEE-488-type transfers can be performed. Figures 23 and 24 show the timings associated with 3-Wire Handshake communications.

CLEAR FIFO Operation. In CPU-to-I/O operation, the CLEAR FIFO operation can be performed by the CPU side (Port 1) under software control as previously explained. The CLEAR FIFO operation can also be performed under hardware control by defining the $\overline{CLEAR}$ pin of Port 2 as an input (Control Register 3, bit 7 = 1).

For cascading purposes, the $\overline{CLEAR}$ pin can also be defined as an output (Control Register 3, bit 7 = 0), which reflects the current state of the $\overline{CLEAR}$ FIFO bit. It can then empty other FIOs or initialize other devices in the system.

Data Direction Control. In CPU-to-I/O mode, the direction of data transfer can be controlled by the CPU side (Port 1) under software control as previously explained. The data direction can also be determined by hardware control by defining the Data Direction pin of Port 2 as an input (Control Register 3, bit 5 = 1).

For cascading purposes, the Data Direction pin can also be defined as an output (Control Register 3, bit 5 = 0) pin which reflects the current state of the Data Direction bit. It can then be used to control the direction of data transfer for other FIOs or for external logic.

On the Port 2 side, when data direction is 0, Port 2 is in Output Handshake mode. When data direction is 1, Port 2 is in Input Handshake mode.

CPU to I/O Operation (Continued)

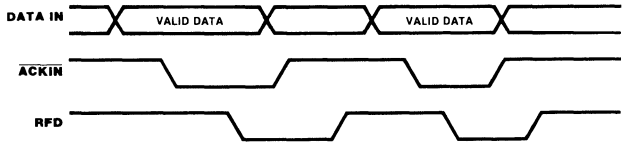


Figure 21. Interlocked Handshake Timing (Imput) Port 2 Side Only

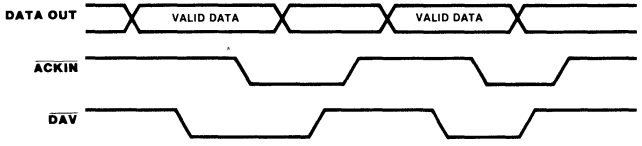


Figure 22. Interlocked Handshake Timing (Output) Port 2 Side Only

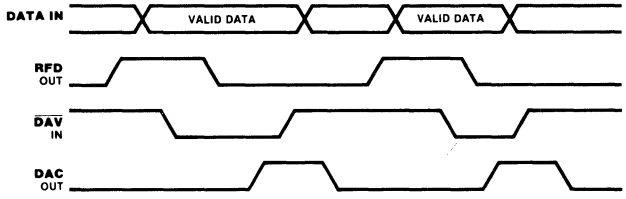


Figure 23. Input (Acceptor) Timing IEEE-488 Port: Port 2 Side Only

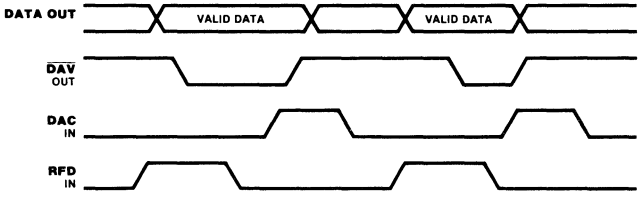


Figure 24. Output (Source) Timing IEEE-488 HS Port: Port 2 Side Only

Programming

The programming of the FIO is greatly simplified by the efficient grouping of the various operation modes in the control registers. Since all of the control registers are read/write, the need for maintaining their image in system memory is eliminated. Also, the read/write feature of the registers aids in system debugging.

Each side of the FIO has 16 registers. All 16 registers are used by the Port 1 side; Control register 2 is not used on the Port 2 side. All registers are addressable 0_H through F_H.

In the Z-BUS Low Byte mode, the FIO allows two methods for register addressing under control of the Right Justify Address (RJA) bit in Control register 0. When RJA = 0, address bus bits 1-4 are used for register addressing and bits 5, 6, and 7 are ignored (Table 4). When RJA = 1, bits 0-3 are used for the register addresses, and bits 4-7 are ignored.

Control Registers. These four registers specify FIO operation. The Port 2 side control registers operate only if the Port 2 device is a CPU. The Port 2 CPU can control interface operations, including data direction, only when enabled by the setting of bit 0 in the Port 1 side of Control Register 2. A 1 in bit 1 of the same register enables the handshake logic.

Interrupt Status Registers. These four registers control and monitor the priority interrupt functions for the FIO.

Interrupt Vector Register. This register stores the interrupt service routine address. This vector is placed on D_0-D_7 when IUS is set by the Interrupt Acknowledge signal from the CPU. When bit 4 (Vector Includes Status) is set in Control Register 0, the reason for the interrupt

Non Z-BUS	D_7-D_4	D_3	D_2	D_1	D_0	
Z-BUS High		A_3	A_2	A_1	A_0	
Z-BUS Low RJA=0	AD_7-AD_5	AD_4	AD_3	AD_2	AD_1	AD_0
Z-BUS Low RJA=1	AD_7-AD_4	AD_3	AD_2	AD_1	AD_0	
Description						
Control Register 0	x	0	0	0	0	x
Control Register 1	x	0	0	0	1	x
Interrupt Status Register 0	x	0	0	1	0	x
Interrupt Status Register 1	x	0	0	1	1	x
Interrupt Status Register 2	x	0	1	0	0	x
Interrupt Status Register 3	x	0	1	0	1	x
Interrupt Vector Register	x	0	1	1	0	x
Byte Count Register	x	0	1	1	1	x
Byte Count Comparison Register	x	1	0	0	0	x
Control Register 2*	x	1	0	0	1	x
Control Register 3	x	1	0	1	0	x
Message Out Register	x	1	0	1	1	x
Message In Register	x	1	1	0	0	x
Pattern Match Register	x	1	1	0	1	x
Pattern Mask Register	x	1	1	1	0	x
Data Buffer Register	x	1	1	1	1	x

x = Don't Care
*Register is only on Port 1 side

Table 4. FIO Register Address Summary

Programming Continued)

is encoded within the vector address in bits 1, 2, and 3. If bit 5 is set in Control register 0, no vector is output by the FIO during an Interrupt Acknowledge cycle. However, IUS is set as usual.

Byte Count Compare Register. This register contains a value compared with the byte count in the Byte Count register. If the Byte Count Compare interrupt is enabled, an interrupt will occur upon compare.

Message Out Register. Either CPU can place a message in its Message Out register. If the opposite side Message register interrupt is enabled, the receiving side CPU will receive an interrupt request, advising that a message is present in its Message In register. Bit 5 in Control Register 1 on the initiating side is set when a message is written. It is cleared when the message is read by the receiving CPU.

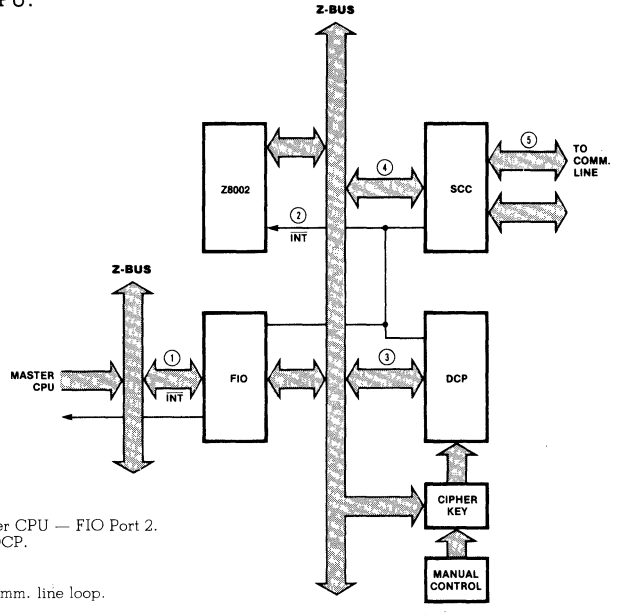
Message In Register. This register receives a message placed in the Message Out register by the opposite side CPU.

Pattern Match Register. This register contains a bit pattern matched against the byte in the Data Buffer register. When these patterns match, a Pattern Match interrupt will be generated, if previously enabled.

Pattern Mask Register. The Pattern Mask register may be programmed with a bit pattern mask that limits comparable bits in the Pattern Match register to non-masked bits (1 = mask).

Data Buffer Register. This register contains the data to be read from or written to the FIFO buffer.

Byte Count Register. This is a read-only register, containing the byte count for the FIFO buffer. The byte count is derived by subtracting the number of bytes read from the buffer from the number of bytes written into the buffer. The count is "frozen" for an accurate reading by setting bit 6 (Freeze Status register) in Control Register 1. This bit is cleared when the Byte Count register read is completed.

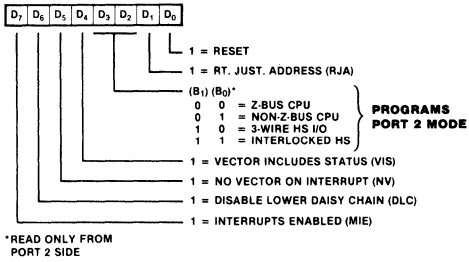


NOTES:
1. Data from master CPU — FIO Port 2.
2. FIO Port 1 — DCP.
3. DCP — RAM.
4. RAM — SCC.
5. SCC — data comm. line loop.

Figure 25. Typical Application: Node Controller

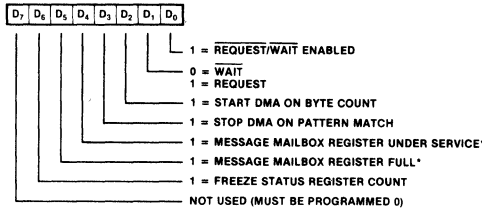
Registers

Control Register 0
Address: 0000
(Read/Write)



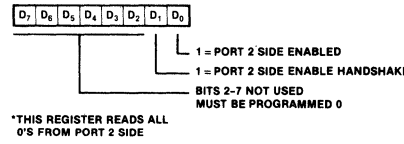
*READ ONLY FROM PORT 2 SIDE

Control Register 1
Address: 0001
(Read/Write)



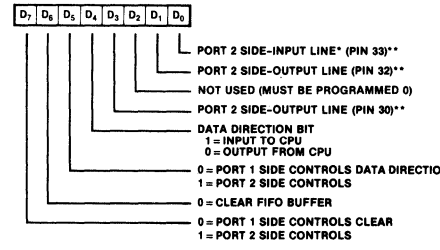
*READ-ONLY BITS

Control Register 2*
Address: 1001
(Read/Write)



*THIS REGISTER READS ALL 0'S FROM PORT 2 SIDE

Control Register 3
Address: 1010
(Read/Write)



*READ-ONLY BITS
**ONLY WHEN PORT 2 IS AN I/O PORT

Figure 26. Control Registers

Interrupt Status Register 0
Address: 0010
(Read/Write)

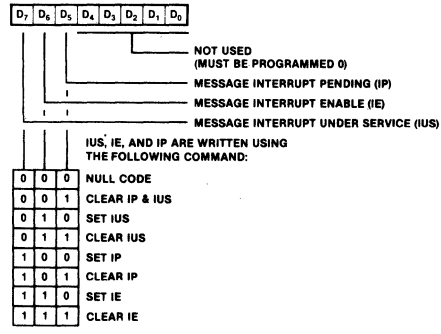
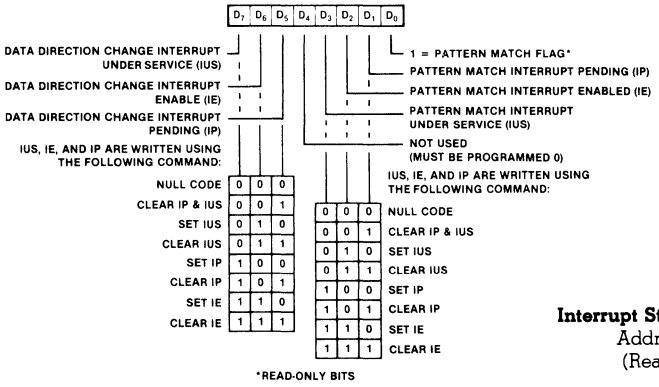


Figure 27. Interrupt Status Registers

Registers (Continued)

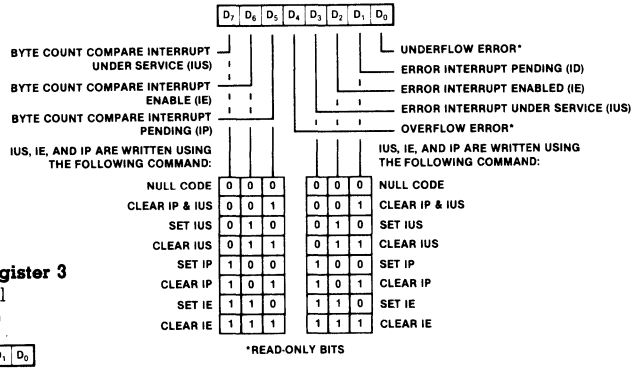
Interrupt Status Register 1

Address: 0011
(Read/Write)



Interrupt Status Register 2

Address: 0100
(Read/Write)



Interrupt Status Register 3

Address: 0101
(Read/Write)

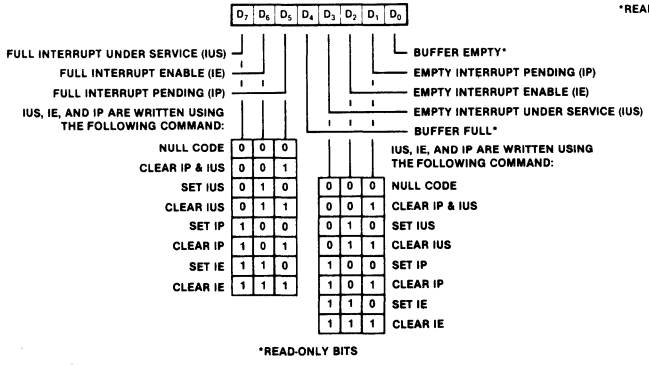


Figure 27. Interrupt Status Registers (Continued)

Registers (Continued)

Byte Count Register
Address: 0111

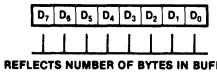


Figure 28. Byte Count Register

Interrupt Vector Register
Address: 0110
(Read/Write)

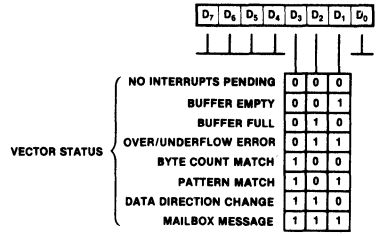


Figure 29. Interrupt Vector Register

Pattern Match Register
Address: 1011
(Read/Write)

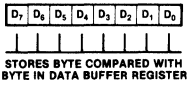


Figure 30. Pattern Match Register

Pattern Mask Register
Address: 1110
(Read/Write)

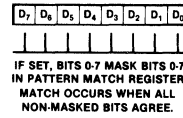


Figure 31. Pattern Mask Register

Data Buffer Register
Address: 1111
(Read/Write)

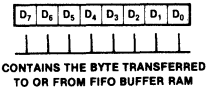


Figure 32. Data Buffer Register

Byte Count Comparison Register
Address: 1000
(Read/Write)

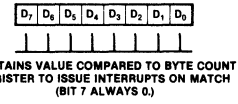


Figure 33. Byte Count Comparison Register

Message Out Register
Address: 1011
(Read/Write)

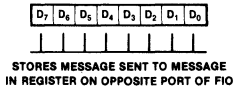


Figure 34. Message Out Register

Message In Register
Address: 1100
(Read Only)

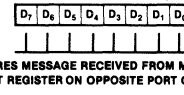


Figure 35. Message In Register

Absolute Maximum Ratings

Voltages on all inputs and outputs with respect to GND . . . . . -0.3 V to +7.0 V

Operating Ambient Temperature . . . . . 0°C to +70°C

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 ≤ V_{CC} ≤ +5.25 V
- GND = 0 V
- T_A as specified in Ordering Information

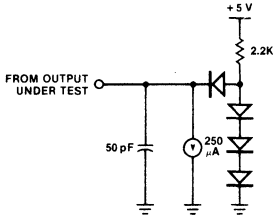


Figure 36. Standard Test Load

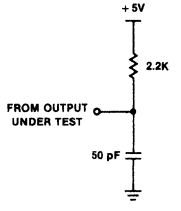


Figure 37. Open-Drain Test Load

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} ≤ +2.4V
I_{OL}	Output Leakage		±10.0	µA	0.4 ≤ V_{OUT} ≤ +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	

Inputs

tr	Any Input Rise Time	100	ns
tf	Any Input Fall Time	100	ns

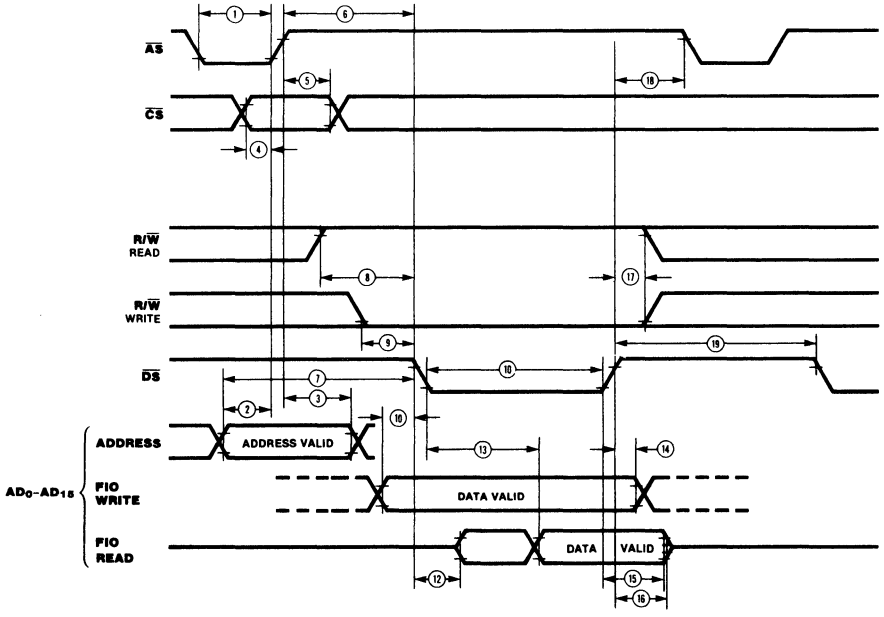
f = 1 MHz, over specified temperature range.

Z-BUS CPU Inteface Timing

Number	Symbol	Parameter	Min	Max	Units	Notes
1	TwAS	$\overline{AS}$ Low Width	70		ns	
2	TsA(AS)	Address to $\overline{AS}$ ↑ Setup Time	10		ns	1
3	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time	50		ns	1
4	TsCSO(AS)	$\overline{CS}$ to $\overline{AS}$ ↑ Setup Time	0		ns	1
5	ThCSO(AS)	$\overline{CS}$ to $\overline{AS}$ ↑ Hold Time	60		ns	1
6	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↑ Delay	60		ns	1
7	TsA(DS)	Address to $\overline{DS}$ ↓	120		ns	
8	TsRWR(DS)	R/$\overline{W}$ (Read) to $\overline{DS}$ ↓ Setup Time	100		ns	
9	TsRWW(DS)	R/$\overline{W}$ (Write) to $\overline{DS}$ ↓ Setup Time	0		ns	
10	TwDS	$\overline{DS}$ Low Width	390		ns	
11	TsDW(DSf)	Write Data to $\overline{DS}$ ↓ Setup Time	30		ns	
12	TdDS(DRV)	$\overline{DS}$ (Read) ↓ to Address Data Bus Driven	0		ns	
13	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		255	ns	
14	ThDW(DS)	Write Data to $\overline{DS}$ ↑ Hold Time	30		ns	
15	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data Not Valid Delay	0		ns	
16	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay		70	ns	2
17	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	60		ns	
18	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		ns	
19	Trc	Valid Access Recovery Time	1000		ns	3

NOTES:
1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is measured to the time when the output has changed 0.5 V from steady state with minimum ac load and maximum dc load.
3. This is the delay from $\overline{DS}$ of one CIO access to $\overline{DS}$ of another FIO access (either read or write).

Z-BUS CPU Inteface Timing (Continued)

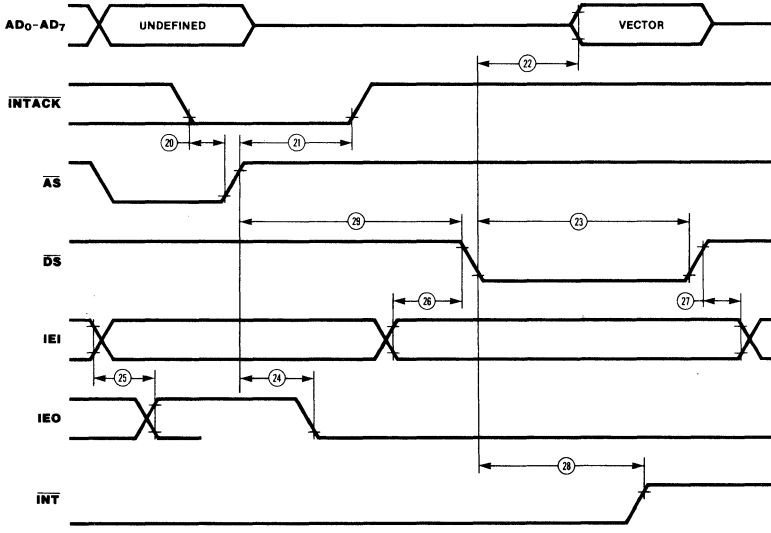


Z-BUS CPU Interrupt Acknowledge Timing

Number	Symbol	Parameter	Min	Max	Units	Notes
20	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Setup Time	0		ns	
21	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		ns	
22	TdDSA(DR)	$\overline{DS}$ (Acknowledge) ↓ to Read Data Valid Delay		360	ns	
23	TwDSA	$\overline{DS}$ (Acknowledge) Low Width	475		ns	
24	TdAS(IEO)	$\overline{AS}$ ↓ to IEO ↓ Delay ($\overline{INTACK}$ Cycle)		350	ns	4
25	TdIEI(IEO)	IEI to IEO Delay		150	ns	4
26	TsIEI(DSA)	IEI to $\overline{DS}$ (Acknowledge) ↓ Setup Time	100		ns	
27	ThIEI(DSA)	IEI to $\overline{DS}$ (Acknowledge) ↓ Hold Time	200		ns	4
28	TdDS(INT)	$\overline{DS}$ ($\overline{INTACK}$ Cycle) to $\overline{INT}$ Delay			ns	
29	TdDCST	Interrupt Daisy Chain Settle Time			ns	4

NOTES:

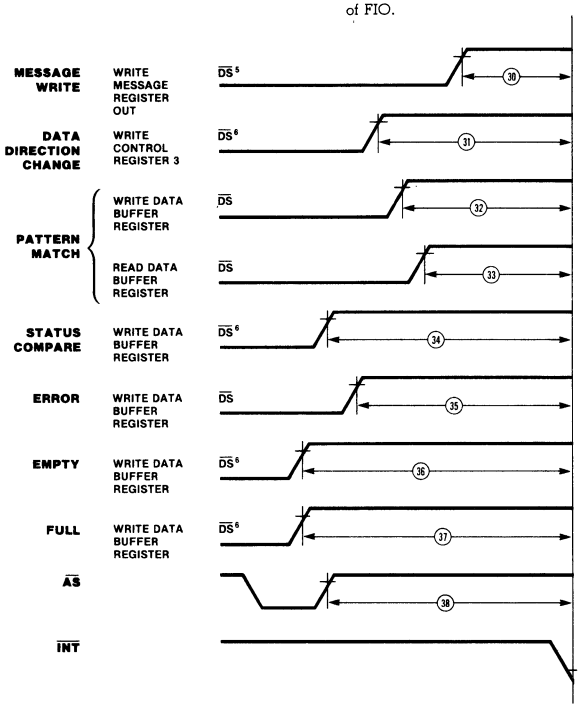
4. The parameters for the devices in any particular daisy chain must meet the following constraint: The delay from $\overline{AS}$ to $\overline{DS}$ must be greater than the sum of TdAS(IEO) for the highest priority peripheral, TsIEI(DSA) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.



Z-BUS Interrupt Timing

Number	Symbol	Parameter	Min	Max	Units	Notes
30	TdMW(INT)	Message Write to $\overline{INT}$ Delay		1	$\overline{AS}$ Cycles + ns	5
31	TdDC(INT)	Data Direction Change to $\overline{INT}$ Delay		1	$\overline{AS}$ Cycles + ns	6
32	TdPMW(INT)	Pattern Match to $\overline{INT}$ Delay (Write Case)		1	$\overline{AS}$ Cycles + ns	
33	TdPMR(INT)	Pattern Match (Read Case) to $\overline{INT}$ Delay		1	$\overline{AS}$ Cycles + ns	
34	TdSC(INT)	Status Compare to $\overline{INT}$ Delay		1	$\overline{AS}$ Cycles + ns	6
35	TdER(INT)	Error to $\overline{INT}$ Delay		1	$\overline{AS}$ Cycles + ns	
36	TdEM(INT)	Empty to $\overline{INT}$ Delay		1	$\overline{AS}$ Cycles + ns	6
37	TdFL(INT)	Full to $\overline{INT}$ Delay		1	$\overline{AS}$ Cycles + ns	6
38	TdAS(INT)	$\overline{AS}$ to $\overline{INT}$ Delay			$\overline{AS}$ Cycles + ns	

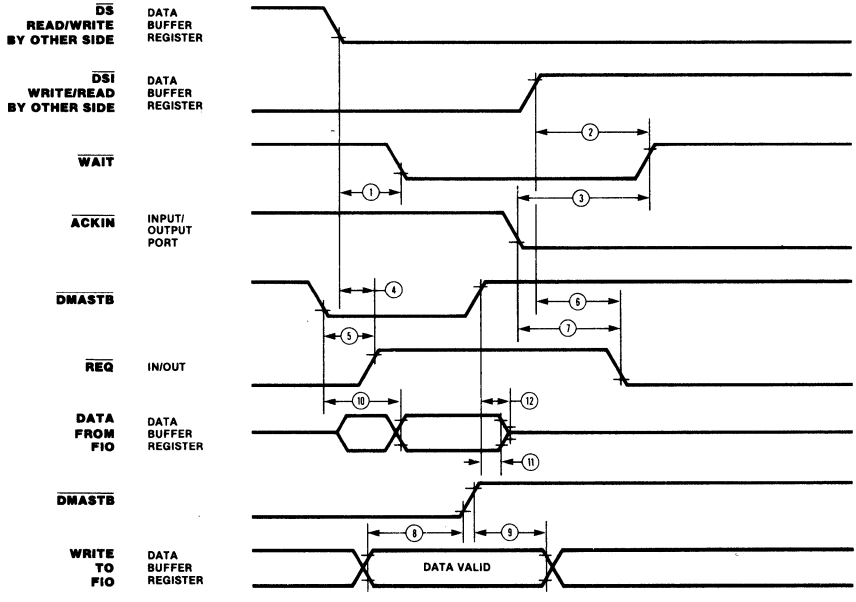
NOTES:
5. Write is from the other side of FIO.
6. Write can be from either side, depending on programming of FIO.



Z-BUS Request/Wait Timing

Number	Symbol	Parameter	Min	Max	Units	Notes
1	TdDS(WAIT)	$\overline{DS}$ ↓ to $\overline{WAIT}$ ↓ Delay			ns	
2	TdDS1(WAIT)	$\overline{DS1}$ ↓ to $\overline{WAIT}$ ↑ Delay			ns	
3	TdACK(WAIT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ ↑ Delay			ns	1
4	TdDS(REQ)	$\overline{DS}$ ↓ to $\overline{REQ}$ ↑ Delay			ns	
5	TdDMA(REQ)	$\overline{DMASTB}$ ↓ to $\overline{REQ}$ ↑ Delay			ns	
6	TdDS1(REQ)	$\overline{DS1}$ ↑ to $\overline{REQ}$ ↓ Delay			ns	
7	TdACK(REQ)	$\overline{ACKIN}$ ↓ to $\overline{REQ}$ ↓ Delay			ns	
8	TdSU(DMA)	Data Setup Time to $\overline{DMASTB}$	200		ns	
9	TdH(DMA)	Data Hold Time to $\overline{DMASTB}$	30		ns	
10	TdDMA(DR)	$\overline{DMASTB}$ ↓ to Valid Data			ns	
11	TdDMA(DRH)	$\overline{DMASTB}$ ↑ to Data Not Valid	0		ns	
12	TdDMA(DR2)	$\overline{DMASTB}$ ↑ to Data Bus Float		70	ns	

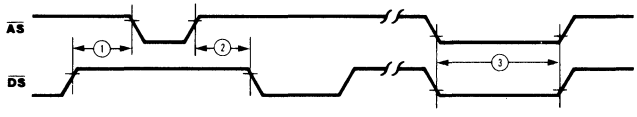
NOTES:
1. The delay is from $\overline{DAV}$ ↓ for 3-Wire Input Handshake. The delay is from DAC ↑ for 3-Wire Output Handshake.



Z-BUS Reset Timing

Number	Symbol	Parameter	Min	Max	Units	Notes
1	TdDSQ(AS)	Delay from $\overline{DS}$ ↑ to $\overline{AS}$ ↓ for No Reset	40		ns	
2	TdASQ(DS)	Delay for $\overline{AS}$ ↑ to $\overline{DS}$ ↓ for No Reset	50		ns	
3	Tw(AS + DS)	Minimum Width of $\overline{AS}$ and $\overline{DS}$ Both Low for Reset	500		ns	1

NOTES:
1. Internal circuitry allows for the reset provided by the Z8 ($\overline{DS}$ held Low while $\overline{AS}$ pulses) to be sufficient.

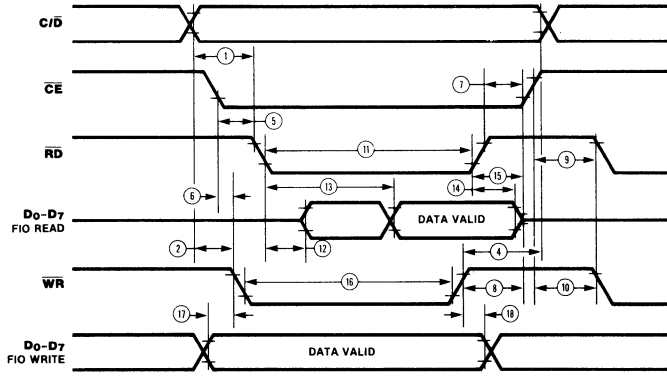


Non-Z-BUS CPU Interface Timing

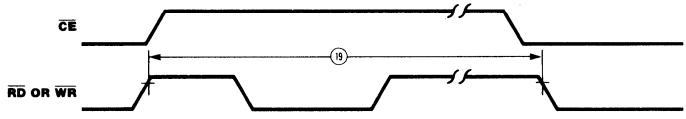
Number	Symbol	Parameter	Min	Max	Units	Notes
1	TsA(RD)	Address Setup to $\overline{RD}$ ↓	80		ns	1
2	TsA(WR)	Address Setup to $\overline{WR}$ ↓	80		ns	
3	ThA(RD)	Address Hold Time to $\overline{RD}$ ↑	0		ns	1
4	ThA(WR)	Address Hold Time to $\overline{WR}$ ↑	0		ns	
5	TsCEI(RD)	$\overline{CE}$ Low Setup Time to $\overline{RD}$	0		ns	1
6	TsCEI(WR)	$\overline{CE}$ Low Setup Time to $\overline{WR}$	0		ns	
7	ThCEI(RD)	$\overline{CE}$ Low Hold Time to $\overline{RD}$	0		ns	1
8	ThCEI(WR)	$\overline{CE}$ Low Hold Time to $\overline{WR}$	0		ns	
9	TsCEh(RD)	$\overline{CE}$ High Setup Time to $\overline{RD}$	100		ns	1
10	TsCEh(WR)	$\overline{CE}$ High Setup Time to $\overline{WR}$	100		ns	
11	TwRDl	$\overline{RD}$ Low Width	400		ns	
12	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		ns	
13	TdRDf(DR)	$\overline{RD}$ ↓ to Valid Data Delay		300	ns	
14	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		ns	
15	TdRD(DRz)	$\overline{RD}$ ↑ to Data Bus Float		70	ns	2
16	TwWRl	$\overline{WR}$ Low Width	400		ns	
17	TsDW(WR)	Data Setup Time to $\overline{WR}$	0		ns	
18	ThDW(WR)	Data Hold Time to $\overline{WR}$	0		ns	
19	Trc	Valid Access Recovery Time	1000		ns	3

NOTES:
1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is measured to the time the output has changed 0.5 V from steady state with minimum ac load and maximum dc load.
3. This is the delay from $\overline{RD}$ ↑ or $\overline{WR}$ ↑ of one FIO access to $\overline{RD}$ ↓ or $\overline{WR}$ ↓ of another FIO access.

Non-Z-BUS CPU Interface Timing(Continued)



Non-Z-BUS CPU Interface Timing



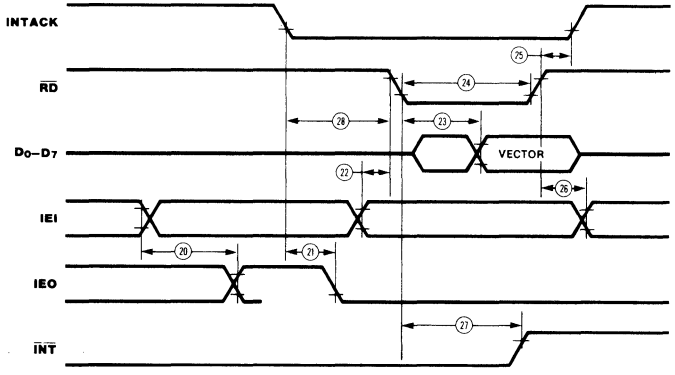
Non-Z-BUS Interface Timing

Non-Z-BUS Interrupt Acknowledge Timing

Number	Symbol	Parameter	Min	Max	Units	Notes
20	TdIEI(IEO)	IEI to IEO Delay	150		ns	4
21	TdI(IEO)	$\overline{\text{INTACK}}$ ↓ to IEO ↓ Delay	350		ns	4
22	TsIEI(RDA)	IEI Setup Time to $\overline{\text{RD}}$ (Acknowledge)	200		ns	4
23	TdRD(DR)	$\overline{\text{RD}}$ ↓ to Vector Valid Delay		300	ns	
24	TwRDI(IA)	Read Low Width (Interrupt Acknowledge)	400		ns	
25	ThIA(RD)	$\overline{\text{INTACK}}$ ↑ to $\overline{\text{RD}}$ ↑ Hold Time	30		ns	
26	ThIEI(RD)	IEI Hold Time to $\overline{\text{RD}}$ ↑	100		ns	
27	TdRD(INT)	$\overline{\text{RD}}$ ↑ to $\overline{\text{INT}}$ ↑ Delay			ns	
28	TdDCST	Interrupt Daisy Chain Settle Time			ns	4

NOTES:

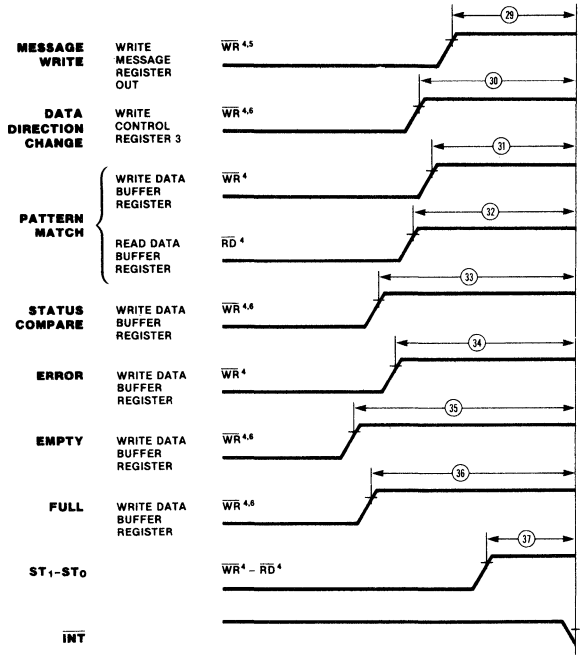
4. The parameter for the devices in any particular daisy chain must meet the following constraint: The delay from $\overline{\text{INTACK}}$ ↓ to $\overline{\text{RD}}$ ↓ must be greater than the sum of TdINA(IEO) for the highest priority peripheral, TsIEI(RD) for the lowest priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.



Non-Z-BUS Interrupt Timing

Number	Symbol	Parameter	Min	Max	Units	Notes
29	TdMW(INT)	Message Write to $\overline{INT}$ Delay			ns	5,6
30	TdDC(INT)	Data Direction Change to $\overline{INT}$ Delay			ns	5,7
31	TdPMW(INT)	Pattern Match (Write Case) to $\overline{INT}$ Delay			ns	5
32	TdPMR(INT)	Pattern Match (Read Case) to $\overline{INT}$ Delay			ns	5
33	TdSC(INT)	Status Compare to $\overline{INT}$ Delay			ns	5,7
34	TdER(INT)	Error to $\overline{INT}$ Delay			ns	5,7
35	TdEM(INT)	Empty to $\overline{INT}$ Delay			ns	5,7
36	TdFL(INT)	Full to $\overline{INT}$ Delay			ns	5,7
37	TdS0(INT)	State 0 to $\overline{INT}$ Delay			ns	

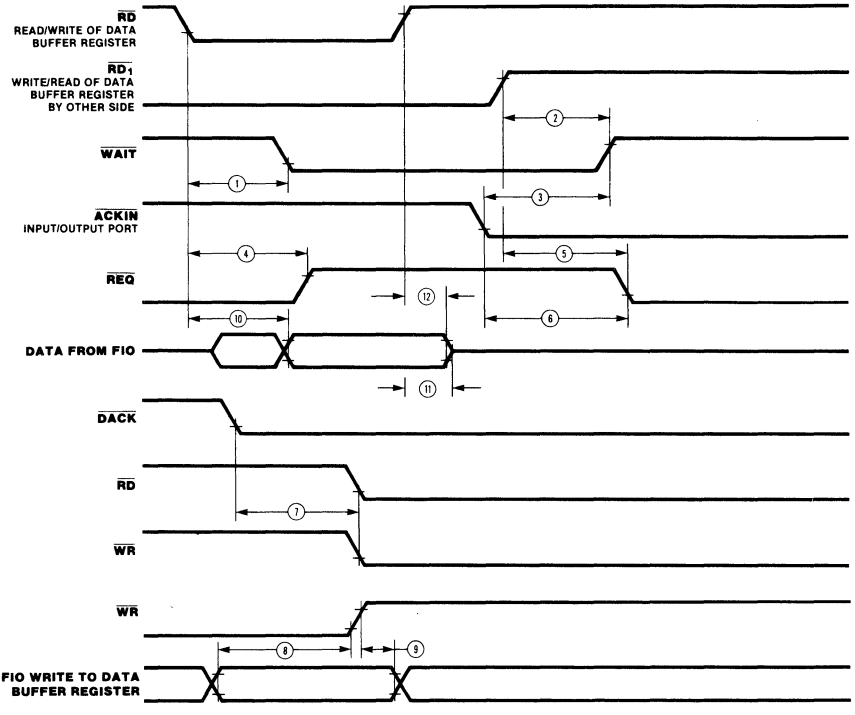
NOTES:
5. Delay number is valid for State 0 only.
6. Write is from other side of FIO.
7. Write can be from either side, depending on programming of FIO.



Non-Z-BUS Request/Wait Timing

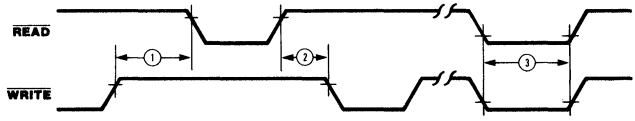
Number	Symbol	Parameter	Min	Max	Units	Notes
1	TdRD(WT)	$\overline{RD}$ ↓ to $\overline{WAIT}$ Active			ns	
2	TdRD1(WT)	$\overline{RD1}$ ↓ to $\overline{WAIT}$ Inactive			ns	
3	TdACK(WT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ Inactive			ns	1
4	TdRD(REQ)	$\overline{RD}$ ↓ to $\overline{REQ}$ Inactive			ns	
5	TdRD1(REQ)	$\overline{RD1}$ ↓ to $\overline{REQ}$ Active			ns	
6	TdACK(REQ)	$\overline{ACKIN}$ ↓ to $\overline{REQ}$ Active			ns	
7	TdDAC(RD)	$\overline{DACK}$ ↓ to $\overline{RD}$ ↓ or $\overline{WR}$ ↓			ns	
8	TSU(WR)	Data Setup Time to $\overline{WR}$			ns	
9	Th(WR)	Data Hold Time to $\overline{WR}$			ns	
10	TdDMA	$\overline{RD}$ ↓ to Valid Data			ns	2
11	TdDMA(DRH)	$\overline{RD}$ ↑ to Data Not Valid	0		ns	2
12	TdDMA(DRZ)	$\overline{RD}$ ↑ to Data Bus Float		70	ns	2

NOTES:
1. The delay is from $\overline{DAV}$ ↓ for 3-Wire Input Handshake. The delay is from DAC ↑ for 3-Wire Input Handshake.
2. Only when $\overline{DACK}$ is active.



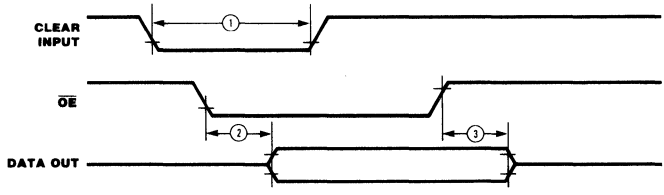
Non-Z-BUS Reset Timing

Number	Symbol	Parameter	Min	Max	Units
1.	TdWR(RD)	Delay from $\overline{WR}$ ↑ to $\overline{RD}$ ↓	100		ns
2.	TdRD(WR)	Delay from $\overline{RD}$ ↑ to $\overline{WR}$ ↓	100		ns
3.	TwRD + WR	Width of $\overline{RD}$ and $\overline{WR}$, both Low for Reset	500		ns



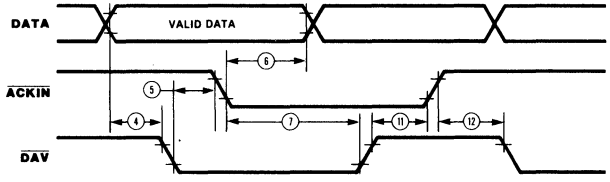
Port 2 Side Operation

Number	Symboi	Parameter	Min	Max	Units
1.	TwCLR	Width of Clear to Reset FIFO	700		ns
2.	TdOE(DO)	$\overline{OE}$ ↓ to Data Bus Driven	0		ns
3.	TdOE(DRZ)	$\overline{OE}$ ↑ to Data Bus Float			ns

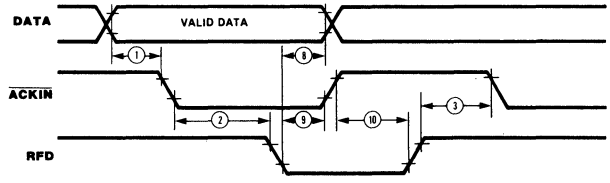


FIO 2-Wire Handshake Timing

Number	Symbol	Parameter	Min	Max	Units
1	TsDI(ACK)	Data Input to $\overline{ACKIN}$ ↓ to Setup Time			ns
2	TdACKf(RFD)	$\overline{ACKIN}$ ↓ to RFD ↓ Delay	0		ns
3	TdRFDr(ACK)	RFD ↑ to $\overline{ACKIN}$ ↓ Delay	0		ns
4	TsDO(DAV)	Data Out to $\overline{DAV}$ ↓ Setup Time	25		ns
5	TdDAVf(ACK)	$\overline{DAV}$ ↓ to $\overline{ACKIN}$ ↓ Delay	0		ns
6	ThDO(ACK)	Data Out to $\overline{ACKIN}$ Hold Time			ns
7	TdACK(DAV)	$\overline{ACKIN}$ ↓ to $\overline{DAV}$ ↑ Delay	0		ns
8	ThDI(RFD)	Data Input to RFD ↓ Hold Time	0		ns
9	TdRFDf(ACK)	RFD ↓ to $\overline{ACKIN}$ ↑ Delay	0		ns
10	TdACKr(RFD)	$\overline{ACKIN}$ ↑ ($\overline{DAV}$ ↑) to RFD ↑ Delay—Interlocked and 3-Wire Handshake	0		ns
11	TdDAVr(ACK)	$\overline{DAV}$ ↑ to $\overline{ACKIN}$ ↑ (RFD ↑)	0		ns
12	TdACKr(DAV)	$\overline{ACKIN}$ ↑ to $\overline{DAV}$ ↓	0		ns



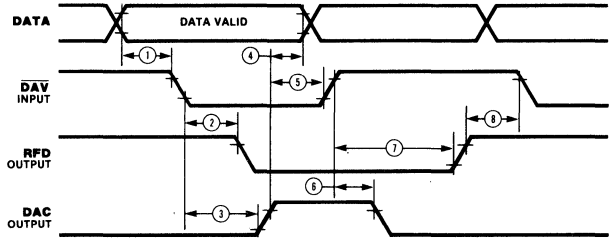
2-Wire Handshake (Port 2 Side Only) Output



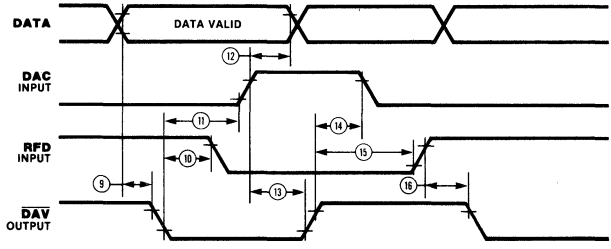
2-Wire Handshake (Port 2 Side Only) Input

3-Wire Handshake Timing

Number	Symbol	Parameter	Min	Max	Units
1	TsDI(DAV)	Data Input to $\overline{DAV}$ ↓ Setup Time			ns
2	TdDAVIf(RFD)	$\overline{DAV}$ ↓ to RFD ↓ Delay	0		ns
3	TdDAVf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay	0		ns
4	ThDI(DAC)	Data In to DAC ↑ Hold Time	0		ns
5	TdDACIr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay	0		ns
6	TdDAVIr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay	0		ns
7	TdDAVIr(RFD)	$\overline{DAV}$ ↑ to RFD ↑ Delay	0		ns
8	TdRFDI(DAV)	RFD ↑ to $\overline{DAV}$ ↓ Delay	0		ns
9	TsDO(DAC)	Data Out to $\overline{DAV}$ ↓			ns
10	TdDAVOf(RFD)	$\overline{DAV}$ ↓ to RFD ↓ Delay	0		ns
11	TdDAVOf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay	0		ns
12	ThDO(DAC)	Data Out to DAC ↑ Hold Time			ns
13	TdDACOr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay			ns
14	TdDAVOr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay	0		ns
15	TdDAVOr(RFD)	$\overline{DAV}$ ↑ to RFD ↑ Delay	0		ns
16	TdRFDO(DAV)	RFD ↑ to $\overline{DAV}$ ↓ Delay	0		ns



3-Wire Handshake Input



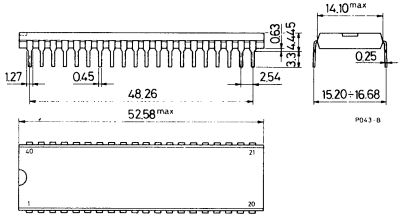
3-Wire Handshake Output

Ordering Information

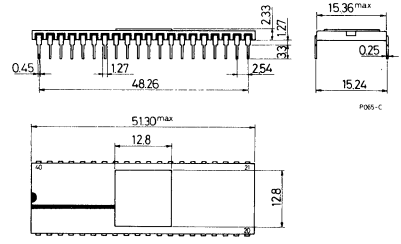
Type	Package	Temp	Clock	Description
Z8538 B1	Plastic 40 pin	0/+70°C	4MHz	Z8538 FIO FIFO Input/Output Interface Unit.
B6	Plastic 40 pin	-40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D2	Ceramic 40 pin	-55/+125°C		
D6	Ceramic 40 pin	40/+85°C		
Z8538A B1	Plastic 40 pin	0/+70°C	6MHz	
B6	Plastic 40 pin	40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D6	Ceramic 40 pin	-40/+85°C		

Packages

Plastic



Ceramic



Universal Peripheral Controller

Z8590 UPC

Features

- Complete slave microcomputer, for distributed processing use.
- Unmatched power of Z8 architecture and instruction set.
- Three programmable I/O ports, two with optional 2-Wire Handshake.
- Six levels of priority interrupts from eight sources: six from external sources and two from internal sources.
- Two programmable 8-bit counter/timers each with a 6-bit prescaler. Counter/Timer T0 is driven by an internal source, and Counter/Timer T1 can be driven by internal or external sources. Both counter/timers are independent of program execution.
- 256-byte register file, accessible by both the master CPU and UPC, as allocated in the UPC program.
- 2K bytes of on-chip ROM for efficiency and versatility.

General Description

The Z8590 Universal Peripheral Controller (UPC) is an intelligent peripheral controller for distributed processing applications (Figure 3). The UPC unburdens the host processor by assuming tasks traditionally done by the host (or by added hardware), such as performing arithmetic, translating or formatting data, and controlling I/O devices. Based on the Z8 microcomputer architecture and instruction set, the UPC contains 2K bytes of internal program ROM, a 256-byte register file, three 8-bit I/O ports, and two counter/timers.

The UPC offers fast execution time, an effective use of memory, and sophisticated interrupt, I/O, and bit manipulation. Using a powerful and extensive instruction set combined with an efficient internal addressing scheme, the UPC speeds program execution and efficiently packs program code into the on-chip ROM.

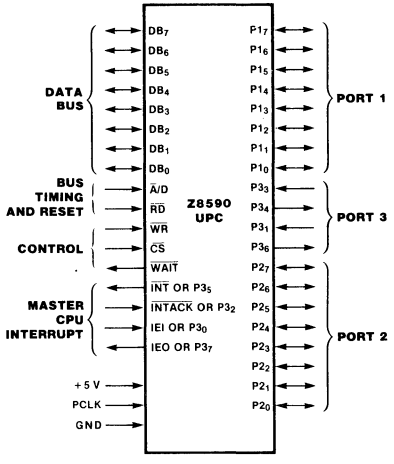


Figure 1. Z8590 UPC Logic Functions

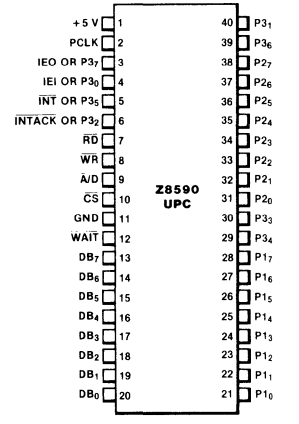


Figure 2. Z8590 UPC Pin Configuration

General Description (Continued)

An important feature of the UPC is an internal register file containing I/O port and control registers accessed both by the UPC program and indirectly by its associated master CPU. This architecture results in both byte and programming efficiency, because UPC instructions can operate directly on I/O data without moving it to and from an accumulator. Such a structure allows the user to allocate as many general purpose registers as the application requires for data buffers between the CPU and peripheral devices. All general-purpose registers can be used as address pointers, index registers, data buffers, or stack space.

The register file is logically divided into 16 groups, each consisting of 16 working registers. A Register Pointer is used in conjunction with short format instructions, resulting in tight, fast code and easy task switching.

Communication between the master CPU and the register file takes place via one group of 19 interface registers addressed directly by both the master CPU and the UPC, or via a block transfer mechanism. Access by the master CPU is controlled by the UPC to allow independence between the master CPU and UPC software.

The UPC has 24 pins that can be dedicated to I/O functions. Grouped logically into three 8-line ports, they can be programmed in many combinations of input or output lines, with or without handshake, and with push-pull or open-drain outputs. Ports 1 and 2 are bit-programmable; Port 3 has four fixed inputs and four outputs.

To relieve software from coping with real-time counting and timing problems, the UPC has two 8-bit hardware counter/timers, each with a fixed divide-by-four, and a 6-bit programmable prescaler. Various counting modes may be selected.

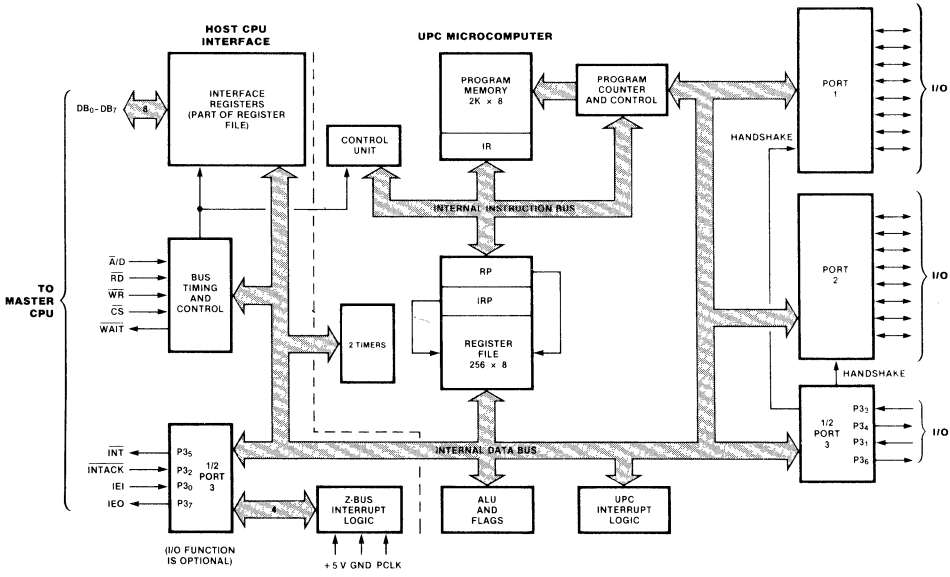


Figure 3. Functional Block Diagram

General Description (Continued)

In addition to the 40-pin standard configuration, the UPC is available in four special configurations:

- A 64-pin RAM development version with external interface for up to 4K bytes of RAM and 36 bytes of internal ROM permitting down-loading from the master CPU.
- A Protopack RAM version with a socket for up to 2K bytes of RAM, with 36 bytes of internal ROM permitting down-loading from the master CPU.
- A 64-pin ROM development version with external interface for up to 4K bytes of ROM and no internal ROM.
- A Protopack ROM version with a socket for 2K bytes of ROM and no internal ROM.

This range of versions and configurations makes the UPC compatible with most system peripheral device control considerations.

Pin Description

$\overline{A}$/D. *Address/Data* (input). A Low on this pin defines information on the data bus as an address. A High defines the information as data.

$\overline{CS}$. *Chip Select* (input, active Low). A Low enables the UPC to accept address or data information from the master CPU during a write cycle or to transmit data to the master CPU during a read cycle. This line is usually generated from higher bits of the address lines.

DB_0-DB_7. *Data Bus* (bidirectional). This bus is used to transfer address and data information between the master CPU and the UPC.

$P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$. *I/O Port Lines* (bidirectional, TTL compatible). These 24 lines are divided into three 8-bit I/O ports and may be configured in the following ways under program control:

$P1_0$-$P1_7$. *Port 1* (input/output—as output it can be push-pull or open-drain). Bit-programmable Parallel I/O.

$P2_0$-$P2_7$. *Port 2* (input/output—as output, it can be push-pull or open-drain). Bit-programmable Parallel I/O.

$P3_0$-$P3_7$. *Port 3* (four inputs, four outputs). Parallel I/O, handshake control, timer I/O, or interrupt control.

PCLK. *Clock* (input). TTL-compatible clock input, 4 MHz maximum. This signal does not need to be related to the master CPU clock.

$\overline{RD}$. *Read* (input, active Low). A Low enables the master CPU to read information from the UPC. Raising the voltage on this pin above V_{DD} will force the UPC into test mode.

$\overline{WAIT}$. *Wait* (output, active Low, open-drain). When the CPU accesses the UPC register file, this signal requests the master CPU to wait until the UPC can complete its part of the transaction.

$\overline{WR}$. *Write* (input, active Low). A Low on this pin enables the master CPU to write information to the UPC. A simultaneous Low on $\overline{RD}$ and $\overline{WR}$ resets the UPC. It is held in reset as long as $\overline{WR}$ is Low.

Functional Description

Address Space. On the 40-pin UPC, all address space is committed to on-chip memory. There are 2048 bytes of mask-programmed ROM and 256 bytes of register file. I/O is memory-mapped to three registers in the register file. Only the Protopack and 64-pin versions of the UPC can access external program memory. See the section entitled

Functional Description (Continued)

"Special Configurations" for complete descriptions of the Protopack and 64-pin versions.

Program Memory. Figure 4 is a map of the 2K on-chip program ROM. Even though the architecture allows addresses from 0 to 4K, behavior of the device above program address 2047 (7FFH) is not defined. The first 12 bytes of program memory are reserved for the UPC interrupt vectors. For the Protopack and 64-pin versions, the address space is extended to 4096 bytes. In the RAM versions, addresses 0CH through 2FH are reserved for on-chip ROM.

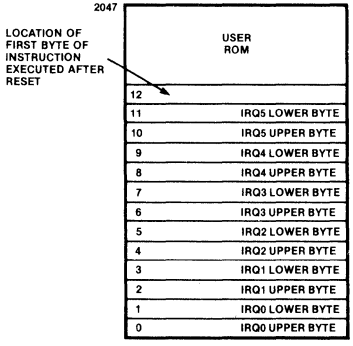


Figure 4. Program Memory Map

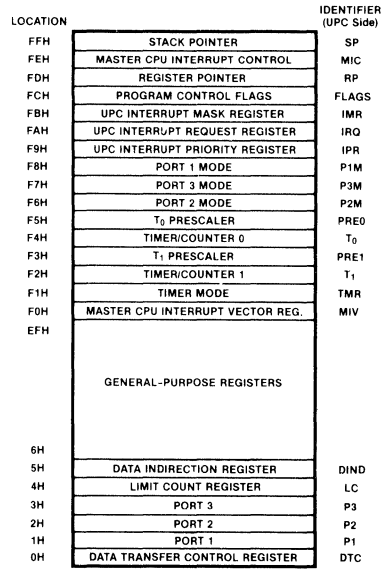


Figure 5. Register File Organization

Register File. This 256-byte file includes three I/O port registers (1-3H), 234 general-purpose registers (6-EEH), and 19 control, status and special I/O registers (0H, 4H, 5H, and F0-FFH). The functions and mnemonics assigned to these register address locations are shown in Figure 5. Of the 256 UPC registers, 19 can be directly accessed by the master CPU; the others are accessed indirectly via the block transfer mechanism.

The I/O port and control registers are included in the register file without differentiation. This allows any UPC instruction to process I/O or control information, thereby eliminating the need for special I/O and control instructions. All general-purpose registers can function as accumulators, address pointers, or index registers. In instruction execution, the registers are read when they are defined as sources and written when defined as destinations.

UPC instructions may access registers directly or indirectly using an 8-bit address mode or a 4-bit address mode and a Register Pointer. For the 4-bit addressing mode, the file is divided into 16 working register groups, each occupying 16 contiguous locations (Figure 6). The Register Pointer (RP) addresses the starting point of the active working-register group, and the 4-bit register designator supplied by the instruction specifies the register within the group. Any instruction altering the contents of the register file can also alter the Register Pointer. The UPC instruction set has a special Set Register Pointer (SRP) instruction for initializing or altering the pointer contents.

Stacks. An 8-bit Stack Pointer (SP), register R255, is used for addressing the stack, residing within the 234 general-purpose

Functional Description (Continued)

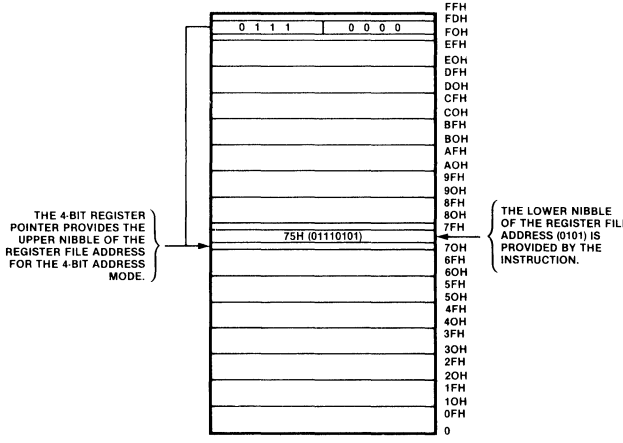


Figure 6. Register Pointer Mechanism

registers, address location 6H through EFH. PUSH and POP instructions can save and restore any register in the register file on the stack. During CALL instructions, the Program Counter is automatically saved on the stack. During UPC interrupt cycles, the Program Counter and the Flag register are automatically saved on the stack. The RET and IRET instructions pop the saved values of the Program Counter and Flag register.

Ports. The UPC has 24 lines dedicated to input and output. These are grouped into three ports of eight lines each and can be configured under software control as inputs, outputs, or special control signals. They can be programmed to provide Parallel I/O with or without handshake and timing signals. All outputs can have active pullups and pulldowns, compatible with TTL loads. In addition, they may be configured as open-drain outputs.

Port 1. Individual bits of Port 1 can be configured as input or output by programming Port 1 Mode register (P1M) F8H. This port is accessed by the UPC program as general register 1H. It is written by specifying address 1H as the destination of any instruction used to store data in the output register. The port is read by specifying address 1H as the source of an instruction.

Port 1 may be placed under handshake control by programming Port 3 Mode register (P3M) F7H. This configures Port 3 pins $P3_3$ and $P3_4$ as handshake control lines $\overline{DAV}_1$ and RDY_1 for input handshake, or RDY_1 and $\overline{DAV}_1$ for output handshake, as determined by the direction (input or output) assigned to bit 7 of Port 1. The Port 3 Mode register also has a bit that programs Port 1 for open-drain output.

Port 2. Individual bits of Port 2 can be configured as inputs or outputs by programming Port 2 Mode register (P2M) F6H. This port is accessed by the UPC program as general register 2H, and its functions and methods of programming are the same as those of Port 1.

Functional Description (Continued)

Port 3 pins $P3_1$ and $P3_6$ are the handshake lines $\overline{DAV}_2$ and RDY_2, with the direction (input or output) determined by the state of bit 7 of the port. The Port 3 Mode register also has a bit used to program Port 2 for open-drain output.

Port 3. This port can be configured as I/O or control lines by programming the Port 3 Mode register. Port 3 is accessed as general register 3H. The directions of the eight data lines are fixed. Four lines, $P3_0$ through $P3_3$, are inputs, and the other four, $P3_4$ through $P3_7$, are outputs. The control functions performed by Port 3 are listed in Table 1.

Counter/Timers. The UPC contains two 8-bit programmable counter/timers, each driven by an internal 6-bit programmable prescaler.

The T1 prescaler can be driven by internal or external clock sources. The T0 prescaler is driven by an internal clock source. Both counter/timers operate independently of the processor instruction sequence to relieve the program from time-critical operations like event counting or elapsed-time calculation. T0 Prescaler register (PRE0) F5H and T1 Prescaler register (PRE1) F3H can be programmed to divide the input frequency of the source being counted by any number from 1 to 64. A counter register (F2H or F4H) is loaded with a number from 1 to 256. The corresponding counter is decremented from this number each time the prescaler reaches end-of-count. When the count is complete, the counter issues a timer interrupt request; IRQ_4 for T0 or IRQ_5 for T1. Loading either counter with a number (n) results in the interruption of the UPC at the nth count.

The counters can be started, stopped, restarted to continue, or restarted from the initial value. They can be programmed to stop upon reaching end-of-count (Single-Pass mode) or to automatically reload the initial value and continue counting (Modulo-n Continuous mode). The counters and prescalers can be read at any time without disturbing their values or changing their counts. The clock sources for both timers can be defined as any one of the following:

- UPC internal clock (4 MHz maximum) divided by four.
- External clock input to Counter/Timer T1 via $P3_1$ (1 MHz maximum).
- Retriggerable trigger input for the UPC internal clock divided by four.
- Nonretriggerable trigger input for the UPC internal clock divided by four.
- External gate input for the UPC internal clock divided by four.

Function	Line	Direction	Signal
Handshake	$P3_1$	In	$\overline{DAV}_2/RDY_2$
	$P3_3$	In	$\overline{DAV}_1/\overline{RDY}_1$
	$P3_4$	Out	$RDY_1/\overline{DAV}_1$
	$P3_6$	Out	$RDY_2/\overline{DAV}_2$
UPC Interrupt Request*	$P3_0$	In	IRQ_3
	$P3_1$	In	IRQ_2
	$P3_3$	In	IRQ_1
Counter/Timer	$P3_1$	In	T_{7N}
	$P3_6$	Out	T_{OUT}
Master CPU	$P3_5$	Out	$\overline{INT}$
	$P3_2$	In	$\overline{INTACK}$
	$P3_0$	In	IEI
	$P3_7$	Out	IEO
Test Mode	$P3_5$	Out	$\overline{A}/D$

*$P3_0$, $P3_1$, and $P3_3$ can always be used as UPC interrupt request inputs, regardless of the configuration programmed.

Table 1. Port 3 Control Functions

Interrupts. The UPC allows six interrupts from eight different sources as follows:

- Port 3 lines $P3_0$, $P3_2$, and $P3_3$.
- The master CPU(3).
- The two counter/timers.

These interrupts can be masked and globally enabled or disabled using Interrupt Mask Register (IMR) FBH. Interrupt Priority Register (IPR) F9H specifies the order of their priority. All UPC interrupts are vectored.

Table 2 lists the UPC's interrupt sources, their types, and their vector locations in program ROM. Interrupt Request IRQ_6 is dedicated to master CPU communications.

Functional Description (Continued)

Name	Source	Vector Location	Comments
IRQ_0	EOM, XERR, LERR	0,1	Internal (R0 Bits 0, 1, 2)
IRQ_1	$\overline{DAV}_1$, IRQ_1	2,3	External ($P3_3$) ↓ Edge Triggered
IRQ_2	$\overline{DAV}_2$, IRQ_2, T_{IN}	4,5	External ($P3_1$) ↓ Edge Triggered
IRQ_3	IRQ_3, IEI	6,7	External ($P3_0$) ↓ Edge Triggered
IRQ_4	T0	8,9	Internal
IRQ_5	T1	10,11	Internal

Table 2. Interrupt Types, Sources, and Vector Locations

Interrupt Requests IRQ_1, IRQ_2, and IRQ_3 are generated on the falling transitions of external inputs $P3_3$, $P3_1$, and $P3_0$. Interrupt Requests IRQ_4 and IRQ_5 are generated upon the timeout of the UPC's two counter/timers. When an interrupt request is granted, the UPC enters an interrupt machine cycle. This cycle disables all subsequent interrupts, saves the Program Counter and Status Flags, and branches to the program memory vector location reserved for that interrupt. This memory location and the next byte contain the 16-bit address of the interrupt service routine for that particular interrupt request.

The UPC also supports polled systems. To accommodate a polled structure, any or all of the interrupt inputs can be masked and the Interrupt Request register polled to determine which of the interrupt requests needs service.

Following any hardware reset operation, an EI instruction must be executed to enable the setting of any interrupt request bit in the IRQ register. Interrupts must be disabled prior to changing the content of either the IPR (F9H) or the IMR (FBH). DI is the only instruction that should be used to globally disable interrupts.

Master CPU Register File Access. There are two ways in which the master CPU can access the UPC register file: direct access and block access.

Direct Access. Three UPC registers—the Data Transfer Control (0H), the Master Interrupt Vector (F0H), and the Master Interrupt Control (FEH)—are mapped directly into the master CPU address space. The master CPU accesses these registers via the addresses shown in Table 3.

UPC Address Decimal	Hex	Identifier	Address
0	0H	DTC	xxx11000
5	5H	DIND	
@5**	@5H**		xxx11111
240	F0H	MIV	xxx10000
254	FEH	MIC	xxx11110
*n		DSC0	xxx00000
n + 1		DSC1	xxx00001
n + 2		DSC2	xxx00010
n + 3		DSC3	xxx00011
n + 4		DSC4	xxx00100
n + 5		DSC5	xxx00101
n + 6		DSC6	xxx00110
n + 7		DSC7	xxx00111
n + 8		DSC8	xxx01000
n + 9		DSC9	xxx01001
n + 10		DSCA	xxx01010
n + 11		DSCB	xxx01011
n + 12		DSCC	xxx01100
n + 13		DSCD	xxx01101
n + 14		DSCE	xxx01110
n + 15		DSCF	xxx01111

x = don't care
*n is the value in the IRP x 16
**Master CPU accesses the register address in Register 5.

Table 3. Master CPU/UPC Register Map

Functional Description (Continued)

The master CPU also has direct access to 16 registers known as the DSC (Data, Status, Command) registers. The DSC registers are numbered 0 through F (DSC0-DSCF). These registers can be any 16 contiguous register file registers beginning on a 16-byte boundary. The base address of the DSC register group is designated by the IRP (I/O Register Pointer), which is bits D_4-D_7 of the Data Transfer Control register (0H). Figure 7 shows how the register address is made up of the 4-bit IRP field, concatenated with the low order 4-bits of the address from the master CPU.

Block Access. The master CPU may transmit or receive blocks of data via address xxx11111. When the master CPU accesses this address, the UPC register pointed to by the Data Indirection register is read or written. The Data Indirection register is incremented, and the Limit Count register is decremented, for example, when the master CPU issues a read or write to address xxx11111 while the Data Indirection register contains the value 33H. The operation causes register 33H to be read or written and the Data Indirection register to be incremented to 34H. This scheme is well suited to Block I/O Instructions and allows the master CPU to efficiently read or write a block of data to or from the UPC.

The Limit Count register (04H) is decremented and is used to control the number of bytes to be transferred by master CPU block accesses. If the master CPU attempts a read or write to the UPC after the Limit Count register reaches 0, the access is not completed, the LERR bit (D_2) of the Data Transfer Control register is set (indicating a limit error), and the LERR error causes an IRQ_0 interrupt request.

The IRP field of the Data Transfer Control register, the Data Indirection register, and the Limit Count register are not directly accessible to the master CPU and therefore must be set by the UPC. This allows the UPC to protect itself from master CPU errors and frees the master CPU from tracking the UPC's internal data layout.

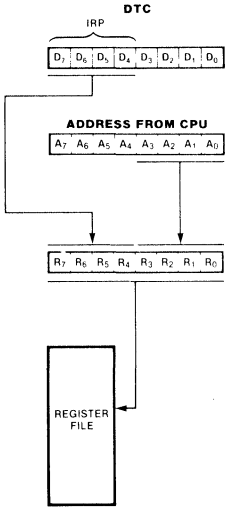


Figure 7. DSC Register Addressing Scheme

Special Configurations

There are two piggyback and two 64-pin versions of the UPC. These versions are identical to the 40-pin UPC with the following exceptions:

- Internal ROM is totally omitted from the 64-pin development (Z8591) and piggyback ROM (Z8593) versions.
- All but 36 bytes of internal ROM are omitted from the 64-pin RAM (Z8592) and piggyback (RAM) (Z8594) versions.
- The memory address and data lines are buffered and brought out to external pins or to the socket on the piggyback.
- Control lines for the external memory are also provided.

Special Configurations (Continued)

The 64-pin version of the UPC allows the user to prototype the system in hardware with an actual UPC device and to develop the code intended to be mask programmed into the on-chip ROM of the 40-pin UPC for the production system. The 64-pin or piggyback RAM versions of the UPC are extremely versatile parts. Memory space can be extended to 4K bytes on the 64-pin version by using external RAM/ROM for all but 36 bytes of the UPC's memory space. This memory can then be down-loaded from the master CPU using a bootstrap program stored in the 36 bytes (C-2F). Figure 8 is a memory map for the 64-pin RAM version.

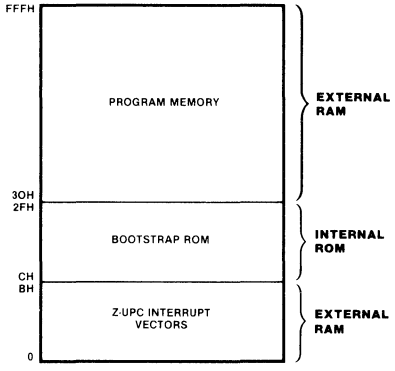


Figure 8. UPC RAM Version Memory Map

64-Pin and Piggyback Pin Functions. Forty of the pins on the 64-pin and piggyback versions have functions identical to those of the 40-pin version. The remaining 24 pins have additional functions described below. (Figures 9 through 11 show the 64-pin and piggyback versions' pin functions and pin assignments.)

A_0-A_{11}. *Program Memory Address Lines* (output). These lines are identical in all 64-pin and RAM versions in the piggyback. They are used to address 4K bytes of external UPC memory.

D_0-D_7. *Program Data* (input). Data is read in from the external memory on these lines. The RAM version also writes external memory through this bus.

IACK. *Interrupt Acknowledge* (output, active High). This signal is active whenever an internal UPC interrupt cycle is in process.

$\overline{MAS}$. *Memory Address Strobe* (output, active Low). This address strobe is pulsed once for each memory fetch to interface with quasi-static RAM.

$\overline{MDS}$. *Memory Data Strobe* (output, active Low). This signal is Low during an instruction fetch or memory write.

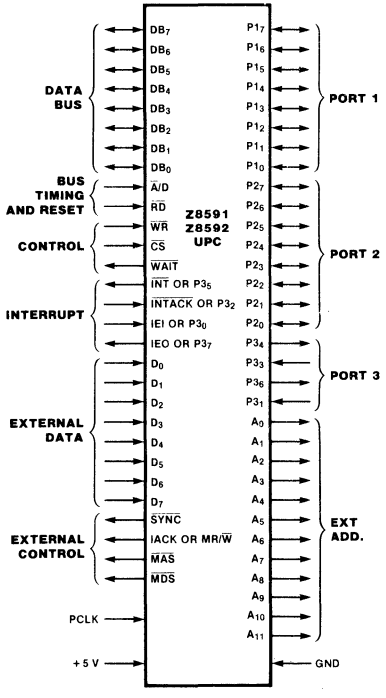


Figure 9. Z8591/Z8592 UPC Logic Functions

Special Configurations (Continued)

MR/$\overline{W}$. *Memory Read/Write* (output RAM versions only). This signal is High when the UPC is fetching an instruction and Low when it is loading external memory.

$\overline{SYNC}$. *Instruction Sync* (output, active Low). This signal is Low during the clock cycle just preceding an opcode fetch.

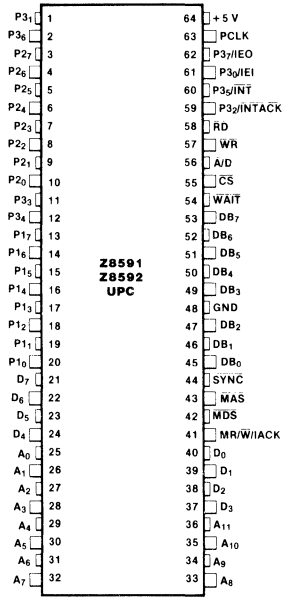


Figure 10. Z8591/Z8592 UPC Pin Configuration

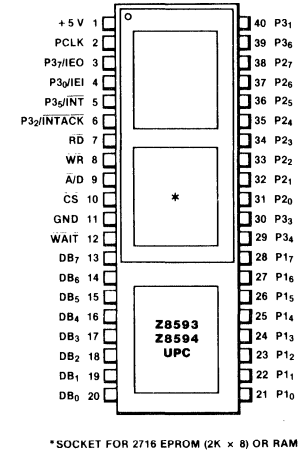


Figure 11. Z8593/Z8594 UPC Piggyback EPROM/RAM Pin Configuration

Addressing Modes

The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

R	Register or working-register address
r	Working-register address only
IR	Indirect-register or indirect working-register address
Ir	Indirect working-register address only
RR	Register pair or working-register pair address
IRR	Indirect register pair or indirect working-register pair address
Irr	Indirect working-register pair only
X	Indexed address
DA	Direct address
RA	Relative address
IM	Immediate

Additional Symbols

dst	Destination location or contents
src	Source location or contents
cc	Condition code (see list)
@	Indirect address prefix
SP	Stack Pointer (control register FFH)
PC	Program Counter
FLAGS	Flag register (control register FCH)
RP	Register Pointer (control register FDH)
IMR	Interrupt Mask register (control register FBH)

Assignment of a value is indicated by the symbol "←". For example,

dst ← dst + src

indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr(n)" is used to refer to bit "n" of a given location. For example,

dst (7)

refers to bit 7 of the destination operand.

Flags

Control Register FCH contains the following six flags:

C	Carry flag
Z	Zero flag
S	Sign flag
V	Overflow flag
D	Decimal-adjust flag
H	Half-carry flag

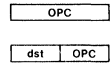
Affected flags are indicated by:

0	Cleared to zero
1	Set to one
*	Set or cleared according to operation
–	Unaffected
X	Undefined

Condition Codes

Value	Mnemonic	Meaning	Flags Set
1000		Always true	—
0111	C	Carry	C = 1
1111	NC	No carry	C = 0
0110	Z	Zero	Z = 1
1110	NZ	Not zero	Z = 0
1101	PL	Plus	S = 0
0101	MI	Minus	S = 1
0100	OV	Overflow	V = 1
1100	NOV	No overflow	V = 0
0110	EQ	Equal	Z = 1
1110	NE	Not equal	Z = 0
1001	GE	Greater than or equal	(S XOR V) = 0
0001	LT	Less than	(S XOR V) = 1
1010	GT	Greater than	[Z OR (S XOR V)] = 0
0010	LE	Less than or equal	[Z OR (S XOR V)] = 1
1111	UGE	Unsigned greater than or equal	C = 0
0111	ULT	Unsigned less than	C = 1
1011	UGT	Unsigned greater than	(C = 0 AND Z = 0) = 1
0011	ULE	Unsigned less than or equal	(C OR Z) = 1
0000		Never true	—

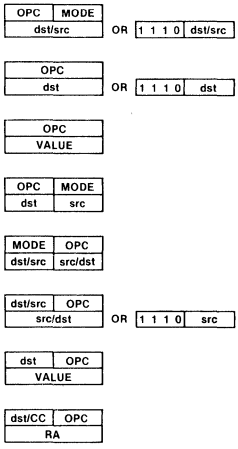
Instruction Formats



CCF, DI, EI, IRET, NOP, RCF, RET, SCF

INC r

One-Byte Instructions



CLR, CPL, DA, DEC, DECW, INC, INCW, POP, PUSH, RL, RLC, RR, RRC, SRA, SWAP

JP, CALL (Indirect)

SRP

ADC, ADD, AND, CP, OR, SBC, SUB, TCM, TM, XOR

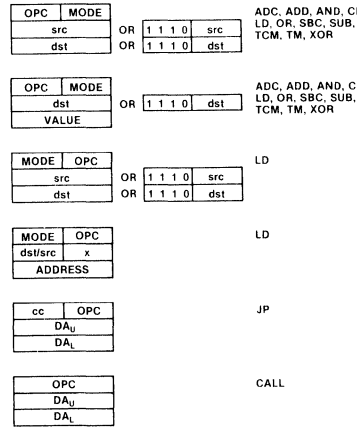
LD, LDE, LDEI, LDC, LDCI

LD

LD

DJNZ, JR

Two-Byte Instructions



ADC, ADD, AND, CP, LD, OR, SBC, SUB, TCM, TM, XOR

ADC, ADD, AND, CP, LD, OR, SBC, SUB, TCM, TM, XOR

LD

LD

JP

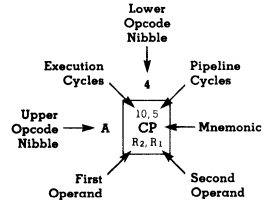
CALL

Three-Byte Instructions

Opcode Map

Lower Nibble (Hex)

Upper Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6,5 **DEC** R1	6,5 **DEC** IR1	6,5 **ADD** r1, r2	6,5 **ADD** r1, Ir2	10,5 **ADD** R2, R1	10,5 **ADD** IR2, R1	10,5 **ADD** R1, IM	10,5 **ADD** IR1, IM	6,5 **LD** r1, R2	6,5 **LD** r2, R1	12/10,5 **DJNZ** r1, RA	12/10,0 **JR** cc, RA	6,5 **LD** r1, IM	12/10,0 **JP** cc, DA	6,5 **INC** r1	
1	6,5 **RLC** R1	6,5 **RLC** IR1	6,5 **ADC** r1, r2	6,5 **ADC** r1, Ir2	10,5 **ADC** R2, R1	10,5 **ADC** IR2, R1	10,5 **ADC** R1, IM	10,5 **ADC** IR1, IM								
2	6,5 **INC** R1	6,5 **INC** IR1	6,5 **SUB** r1, r2	6,5 **SUB** r1, Ir2	10,5 **SUB** R2, R1	10,5 **SUB** IR2, R1	10,5 **SUB** R1, IM	10,5 **SUB** IR1, IM								
3	8,0 **JP** IRR1	6,1 **SRP** IM	6,5 **SBC** r1, r2	6,5 **SBC** r1, Ir2	10,5 **SBC** R2, R1	10,5 **SBC** IR2, R1	10,5 **SBC** R1, IM	10,5 **SBC** IR1, IM								
4	8,5 **DA** R1	8,5 **DA** IR1	6,5 **OR** r1, r2	6,5 **OR** r1, Ir2	10,5 **OR** R2, R1	10,5 **OR** IR2, R1	10,5 **OR** R1, IM	10,5 **OR** IR1, IM								
5	10,5 **POP** R1	10,5 **POP** IR1	6,5 **AND** r1, r2	6,5 **AND** r1, Ir2	10,5 **AND** R2, R1	10,5 **AND** IR2, R1	10,5 **AND** R1, IM	10,5 **AND** IR1, IM								
6	6,5 **COM** R1	6,5 **COM** IR1	6,5 **TCM** r1, r2	6,5 **TCM** r1, Ir2	10,5 **TCM** R2, R1	10,5 **TCM** IR2, R1	10,5 **TCM** R1, IM	10,5 **TCM** IR1, IM								
7	10/12,1 **PUSH** R2	12/14,1 **PUSH** IR2	6,5 **TM** r1, r2	6,5 **TM** r1, Ir2	10,5 **TM** R2, R1	10,5 **TM** IR2, R1	10,5 **TM** R1, IM	10,5 **TM** IR1, IM								
8	10,5 **DECW** RR1	10,5 **DECW** IR1	12,0 **LDE** r1, Irr2	18,0 **LDEI** Ir1, Irr2												6,1 **DI**
9	6,5 **RL** R1	6,5 **RL** IR1	12,0 **LDE** r2, Irr1	18,0 **LDEI** Ir2, Irr1												6,1 **EI**
A	10,5 **INCW** RR1	10,5 **INCW** IR1	6,5 **CP** r1, r2	6,5 **CP** r1, Ir2	10,5 **CP** R2, R1	10,5 **CP** IR2, R1	10,5 **CP** R1, IM	10,5 **CP** IR1, IM								14,0 **RET**
B	6,5 **CLR** R1	6,5 **CLR** IR1	6,5 **XOR** r1, r2	6,5 **XOR** r1, Ir2	10,5 **XOR** R2, R1	10,5 **XOR** IR2, R1	10,5 **XOR** R1, IM	10,5 **XOR** IR1, IM								16,0 **IRET**
C	6,5 **RRC** R1	6,5 **RRC** IR1	12,0 **LDC** r1, Irr2	18,0 **LDCI** Ir1, Irr2				10,5 **LD** r1, x, R2								6,5 **RCF**
D	6,5 **SRA** R1	6,5 **SRA** IR1	12,0 **LDC** r2, Irr1	18,0 **LDCI** Ir2, Irr1	20,0 **CALL*** IRR1		20,0 **CALL** DA	10,5 **LD** r2, x, R1								6,5 **SCF**
E	6,5 **RR** R1	6,5 **RR** IR1		6,5 **LD** r1, Ir2	10,5 **LD** R2, R1	10,5 **LD** IR2, R1	10,5 **LD** R1, IM	10,5 **LD** IR1, IM								6,5 **CCF**
F	6,7 **SWAP** R1	6,7 **SWAP** IR1		6,5 **LD** Ir1, r2		10,5 **LD** R2, IR1										6,0 **NOP**
Bytes per Instruction	2				3				2					3	1	



Legend:
R = 8-Bit Address
r = 4-Bit Address
R1 or r1 = Dst Address
R2 or r2 = Src Address

Sequence:
Opcode, First Operand, Second Operand

Note: The blank areas are not defined.

*2-byte instruction; fetch cycle appears as a 3-byte instruction.

Instruction Summary

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
ADC dst,src dst ← dst + src + C	(Note 1)		1□	*	*	*	*	0	*
ADD dst,src dst ← dst + src	(Note 1)		0□	*	*	*	*	0	*
AND dst,src dst ← dst AND src	(Note 1)		5□	–	*	*	0	–	–
CALL dst SP ← SP - 2 @SP ← PC; PC ← dst	DA IRR		D6 D4	–	–	–	–	–	–
CCF C ← NOT C			EF	*	–	–	–	–	–
CLR dst dst ← 0	R IR		B0 B1	–	–	–	–	–	–
COM dst dst ← NOT dst	R IR		60 61	–	*	*	0	–	–
CP dst,src dst - src	(Note 1)		A□	*	*	*	*	–	–
DA dst dst ← DA dst	R IR		40 41	*	*	*	X	–	–
DEC dst dst ← dst - 1	R IR		00 01	–	*	*	*	–	–
DECW dst dst ← dst - 1	RR IR		80 81	–	*	*	*	–	–
DI IMR (7) ← 0			8F	–	–	–	–	–	–
DJNZ r,dst r ← r - 1 if r ≠ 0 PC ← PC + dst Range: +127, -128	RA		rA r = 0-F	–	–	–	–	–	–
EI IMR (7) ← 1			9F	–	–	–	–	–	–
INC dst dst ← dst + 1	r R IR		rE r = 0-F 20 21	–	*	*	*	–	–
INCW dst dst ← dst + 1	RR IR		A0 A1	–	*	*	*	–	–
IRET FLAGS ← @SP; SP ← SP + 1 PC ← @SP; SP ← SP + 2; IMR (7) ← 1			BF	*	*	*	*	*	*
JP cc,dst if cc is true PC ← dst	DA IRR		cD c = 0-F 30	–	–	–	–	–	–
JR cc,dst if cc is true, PC ← PC + dst Range: +127, -128	RA		cB c = 0-F	–	–	–	–	–	–

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
LD dst,src dst ← src	r r R r X r Ir R R R IR IM	IM R r X r Ir r R IR IM IM R	rC r8 r9 r = 0-F C7 D7 E3 F3 E4 E5 E6 E7 F5	–	–	–	–	–	–
LDC dst,src dst ← src	r Irr	Irr r	C2 D2	–	–	–	–	–	–
LDCI dst,src dst ← src r ← r + 1; rr ← rr + 1	Ir Irr	Irr Ir	C3 D3	–	–	–	–	–	–
LDE dst,src dst ← src	r Irr	Irr r	82 92	–	–	–	–	–	–
LDEI dst,src dst ← src r ← r + 1; rr ← rr + 1	Ir Irr	Irr Ir	83 93	–	–	–	–	–	–
NOP			FF	–	–	–	–	–	–
OR dst,src dst ← dst OR src	(Note 1)		4□	–	*	*	0	–	–
POP dst dst ← @SP SP ← SP + 1	R IR		50 51	–	–	–	–	–	–
PUSH src SP ← SP - 1; @SP ← src		R IR	70 71	–	–	–	–	–	–
RCF C ← 0			CF	0	–	–	–	–	–
RET PC ← @SP; SP ← SP + 2			AF	–	–	–	–	–	–
RL dst 	R IR		90 91	*	*	*	*	–	–
RLC dst 	R IR		10 11	*	*	*	*	–	–
RR dst 	R IR		E0 E1	*	*	*	*	–	–
RRC dst 	R IR		C0 C1	*	*	*	*	–	–
SBC dst,src dst ← dst - src - C	(Note 1)		3□	*	*	*	*	1	*
SCF C ← 1			DF	1	–	–	–	–	–
SRA dst 	R IR		D0 D1	*	*	*	0	–	–

Instruction Summary (Continued)

Instruction and Operation	Addr Mode dst	Addr Mode src	Opcode Byte (Hex)	Flags Affected C Z S V D H
SRP src RP ← src		Im	31	- - - - - -
SUB dst,src dst ← dst - src	(Note 1)		2□	* * * * 1 *
SWAP dst 	R IR		F0 F1	X * * X - -
TCM dst,src (NOT dst) AND src	(Note 1)		6□	- * * 0 - -
TM dst,src dst AND src	(Note 1)		7□	- * * 0 - -
XOR dst,src dst ← dst XOR src	(Note 1)		B□	- * * 0 - -

Note 1

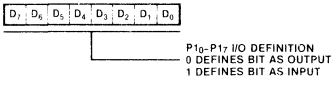
These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a '□' in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

For example, to determine the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

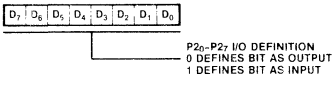
Addr Mode dst	Addr Mode src	Lower Opcode Nibble
r	r	2
r	Ir	3
R	R	4
R	IR	5
R	IM	6
IR	IM	7

Registers

R248 P1M
Port 1 Mode Register
UPC register address (Hex): F8



R246 P2M
Port 2 Mode Register
UPC register address (Hex): F6



R247 P3M
Port 3 Mode Register
UPC register address (Hex): F7

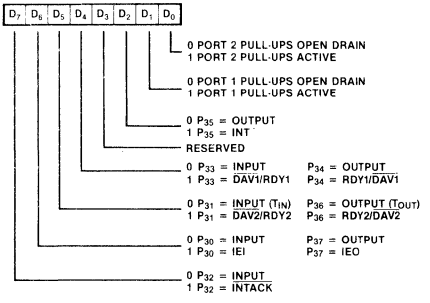
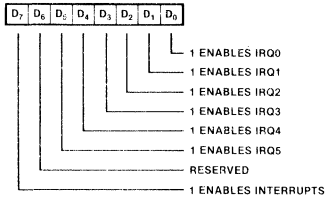


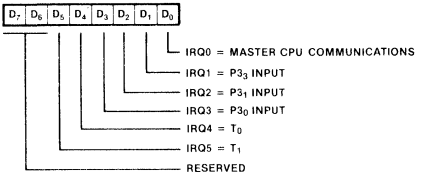
Figure 12. Port Mode Registers

Registers (Continued)

R251 IMR
Interrupt Mask Register
UPC register address (Hex): FB



R250 IRQ
Interrupt Request Register
UPC register address (Hex): FA



R249 IPR
Interrupt Priority Register
UPC register address (Hex): F9 (Write Only)

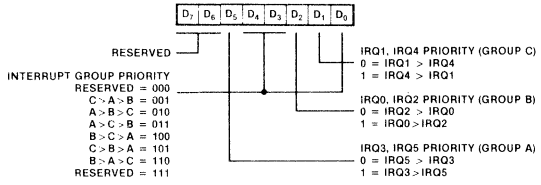
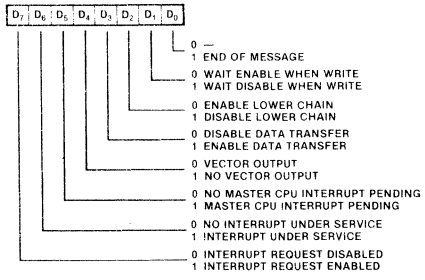


Figure 13. Interrupt Control Registers

R254 MIC
Master CPU Interrupt Control Register
UPC register address (Hex): FE



R240 MIV
Master CPU Interrupt Vector Register
UPC register address (Hex): F0

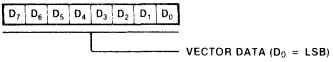
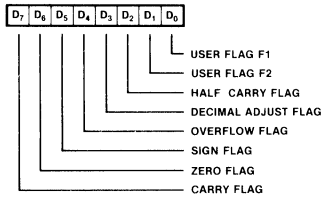


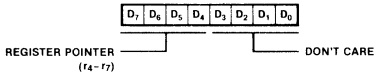
Figure 14. Master CPU Interrupt Registers

Registers (Continued)

R252 FLAGS
Flag Register
UPC register address (Hex): FC



R253 RP
Register Pointer
UPC register address (Hex): FD



R255 SP
Stack Pointer
UPC register address (Hex): FF

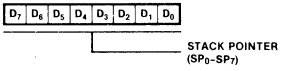
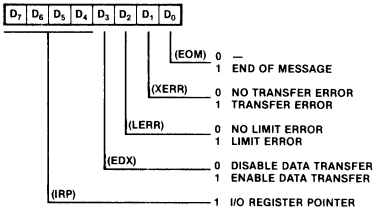
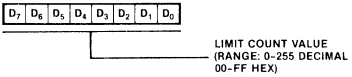


Figure 15. UPC Control Registers

R0 DTC
Data Transfer Control Register
UPC register address (Hex): 00



R4 LC
Limit Count Register
UPC register address (Hex): 04



R5 DIND
Data Indirection Register
UPC register address (Hex): 05

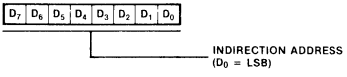
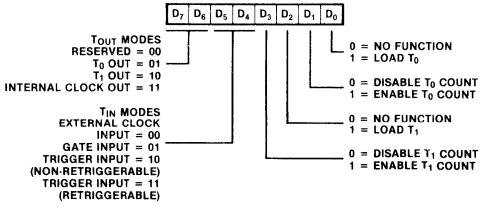


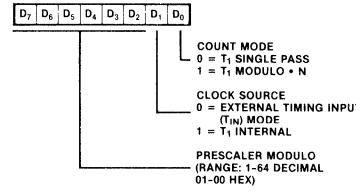
Figure 16. Master CPU-UPC Data Transfer Registers

Registers (Continued)

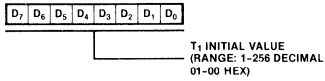
R241 TMR
Timer Mode Register
UPC register address (Hex): F1



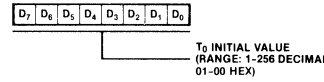
R243 PRE1
Prescaler 1 Register
UPC register address (Hex): F3



R242 T1
Counter/Timer 1 Register
UPC register address (Hex): F2



R244 T0
Counter/Timer 0 Register
UPC register address (Hex): F4



R245 PRE0
Prescaler 0 Register
UPC register address (Hex): F5

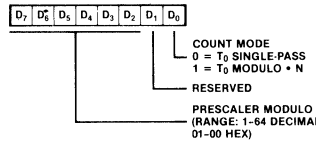


Figure 17. UPC Counter/Timer Registers

Registers (Continued)

Control Register	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Comments
00_H Data Transfer Control Register	X	X	X	X	0	0	0	0	Disable data transfer from master CPU
04_H Limit Count Register	Not Defined								
05_H Data Indirection Register	Not Defined								
$F0_H$ Interrupt Vector Register	Not Defined								
$F1_H$ Timer Mode	0	0	0	0	0	0	0	0	Stops T0 and T1
$F2_H$ T0 Register	Not Defined								
$F3_H$ T0 Prescaler	X	X	X	X	X	X	0	0	Single-Pass mode
$F4_H$ T1 Register	Not Defined								
$F5_H$ T1 Prescaler	X	X	X	X	X	X	0	0	Single-Pass mode External clock source
$F6_H$ Port 2 Mode	1	1	1	1	1	1	1	1	Port 2 lines defined as inputs
$F7_H$ Port 3 Mode	0	0	0	0	X	1	0	0	Port 1, 2 open drain; $P3_5$ = INT; $P3_0$, $P3_1$, $P3_2$, $P3_3$ defined as input; $P3_4$, $P3_6$, $P3_7$ defined as output.
$F8_H$ Port 1 Mode	1	1	1	1	1	1	1	1	Port 1 lines defined as inputs
$F9_H$ Interrupt Priority	Not Defined								
FA_H Interrupt Request	X	X	0	0	0	0	0	0	Reset Interrupt Request
FB_H Interrupt Mask	0	X	X	X	X	X	X	X	Interrupts disabled
FC_H Flag Register	Not Defined								
FD_H Register Pointer	Not Defined								
FE_H Master CPU Interrupt Control Register	0	0	0	0	0	0	0	0	Master CPU interrupt disabled; wait enable when write; lower chain enabled
FF_H Stack Pointer	Not Defined								

NOTE: X means not defined.

Table 4. Control Register Reset Conditions

Absolute Maximum Rating

Voltages on all pins (except V_{BB}) with respect to GND . . . . . -0.5 V to +7.0 V

Operating Ambient Temperature . . . . . 0°C to +70°C

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 reference pin. Standard conditions are as follows:

- +4.75 V ≤ V_{CC} ≤ +5.25 V
- V_{SS} = GND = 0 V
- 0°C ≤ T_A ≤ +70°C

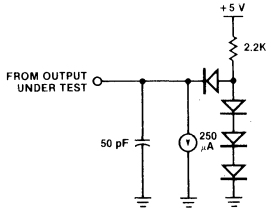


Figure 18. Test Load 1

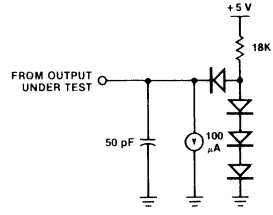


Figure 19. Test Load 2

DC Characteristics

Symbol	Parameter	Min	Max	Unit	Condition	Notes
V_{CH}	Clock Input High Voltage	2.4	V_{CC}	V		
V_{CL}	Clock Input Low Voltage	-0.3	0.8	V		
V_{IH}	Input High Voltage	2.0	V_{CC}	V		
V_{IL}	Input Low Voltage	-0.3	0.8	V		
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -250 μA	1
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = +2.0 mA	1
I_{IL}	Input Leakage	-10	10	μA	0 ≤ V_{IN} ≤ +5.25 V	
I_{OL}	Output Leakage	-10	10	μA	0 ≤ V_{IN} ≤ +5.25 V	
I_{CC}	V_{CC} Supply Current		180	mA		

1. For A_0-A_{11} and D_0-D_7, $\overline{MDS}$, $\overline{SYNC}$, $\overline{MAS}$, and MR/$\overline{W}$/IACK on the 64-pin versions. I_{OH} = 100 μA and I_{OL} = 1.0 mA.

Master CPU Interface Timing

Number	Symbol	Parameter	Min (ns)	Max (ns)	Notes*
1	TrC	Clock Rise Time		20	
2	TwCh	Clock High Width	105	1855	
3	TfC	Clock Fall Time		20	
4	TwCl	Clock Low Width	105	1855	
5	TpC	Clock Period	250	2000	
6	TsA/D(WR)	$\overline{A}$/D to $\overline{WR}$ ↓ Setup Time	80		
7	TsA/D(RD)	$\overline{A}$/D to $\overline{RD}$ ↓ Setup Time	80		
8	ThA/D(WR)	$\overline{A}$/D to $\overline{WR}$ ↑ Hold Time	30		
9	ThA/D(RD)	$\overline{A}$/D to $\overline{RD}$ ↑ Hold Time	30		
10	TsCSf(WR)	$\overline{CS}$ ↓ to$\overline{WR}$ ↓ SetupTime	0		
11	TsCSf(RD)	$\overline{CS}$ ↓ to $\overline{RD}$ ↓ Setup Time	0		
12	TsCSr(WR)	$\overline{CS}$ ↑ to $\overline{WR}$ ↓ Setup Time	60		
13	TsCSr(RD)	$\overline{CS}$ ↑ to $\overline{RD}$ ↓ Setup Time	60		
14	ThCS(WR)	$\overline{CS}$ to $\overline{WR}$ ↓ Hold Time	0		
15	ThCS(RD)	$\overline{CS}$ to $\overline{RD}$ ↓ Hold Time	0		
16	TsDI(WR)	Data in to $\overline{WR}$ ↓ Setup Time	0		
17	Tw(WR)	$\overline{WR}$ Low Width	390		
18	Tw(RD)	$\overline{RD}$ Low Width	390		
19	ThWR(DI)	Data in to $\overline{WR}$ ↑ Hold Time	0		
20	TdRD(DI)	Data Valid from $\overline{RD}$ ↓ Delay			1
21	ThRD(DI)	Data Valid to $\overline{RD}$ ↑ Hold Time	0		
22	TdRD(DI_Z)	Data Bus Float Delay from $\overline{RD}$ ↑		70	
23	TdRD(DB_A)	$\overline{RD}$ ↓ to Read Data Active Delay	0		
24	TdWR(W)	$\overline{WR}$ ↓ to $\overline{WAIT}$ ↓ Delay		150	
25	TdRD(W)	$\overline{RD}$ ↓ to $\overline{WAIT}$ ↓ Delay		150	
26	TdDI(W)	Data Valid to $\overline{WAIT}$ ↑ Delay	0		

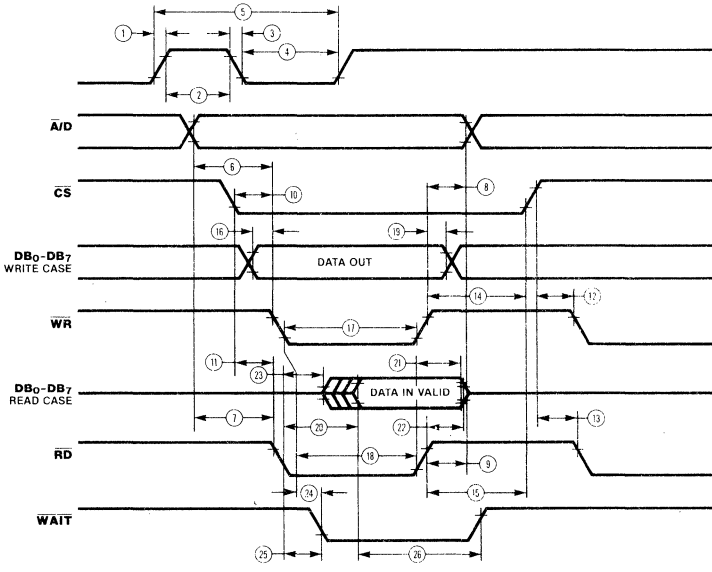
NOTES
1. This parameter is dependent on the state of the UPC at the time of master CPU access.
3. The timing characteristics given reference 2.0 V as High and 0.8 V as Low.
4. All output ac parameters use test load 1.
*Timings are preliminary and subject to change.

Interrupt Acknowledge Transactions

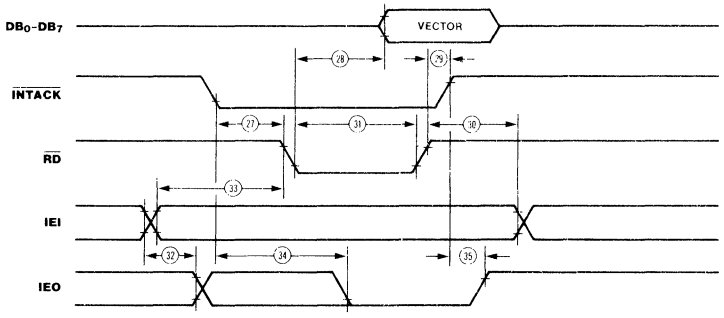
Number	Symbol	Parameter	Min (ns)	Max (ns)	Notes*
27	TsACK(RD)	$\overline{INTACK}$ ↓ to $\overline{RD}$ ↓ Setup Time	90		2
28	TdRD(DI)	$\overline{RD}$ ↓ to Vector Valid Delay		255	
29	ThRD(ACK)	$\overline{RD}$ ↑ to $\overline{INTACK}$ ↑ Hold Time	0		
30	ThIEI(RD)	IEI to $\overline{RD}$ ↓ Hold Time	100		
31	TwRDl	$\overline{RD}$ (Acknowledge) Low Width	255		
32	TdIEI(IEO)	IEI to IEO Delay		120	
33	TsIEI(RD)	IEI to $\overline{RD}$ ↓ Setup Time	150		
34	$TdACK_f$(IEO)	$\overline{INTACK}$ ↓ to IEO ↓ Delay		250	
35	$TdACK_r$(IEO)	$\overline{INTACK}$ ↑ to IEO ↑ Delay		250	

NOTES:
2. In case where daisy chain is not used.
3. The timing characteristics given reference 2.0 V as High and 0.8 V as Low.
4. All output ac parameters use test load 1.
*Timings are preliminary and subject to change.

Master CPU Interface Timing



Interrupt Acknowledge Timig



Handshake Timing

Number	Symbol	Parameter	Min (ns)	Max (ns)	Notes*
1	TsDI(DA)	Data In Setup Time	0		
2	ThDA(DI)	Data In Hold Time	230		
3	TwDA	Data Available Width	175		1,2
4	TdDAL(RY)	Data Available Low to Ready Delay Time	20 0	175	1,2 2,3
5	TdDAH(RY)	Data Available High to Ready Delay Time	 0	150	1,2 2,3
6	TdDO(DA)	Data Out to Data Available Delay Time	50		2
7	TdRY(DA)	Ready to Data Available Delay Time	0	205	2

1. Input Handshake.
2. Test Load 1.
3. Output Handshake.

*Timings are preliminary and subject to change.

Reset Timing

Number	Symbol	Parameter	Min (ns)	Max (ns)	Notes*
1	TdRDQ(WR)	Delay from $\overline{RD}$ ↑ to $\overline{WR}$ ↓ for No Reset	40		
2	TdWRQ(RD)	Delay from $\overline{WR}$ ↑ to $\overline{RD}$ ↓ for No Reset	50		
3	TwRES	Minimum Width of $\overline{WR}$ and $\overline{RD}$ both Low for Reset	250		4

4. Internal reset signal is ½ to 2 clock delays from external reset condition.

*Timings are preliminary and subject to change.

RAM Version Program Memory Timing

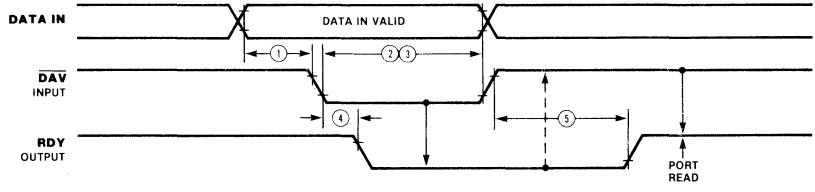
Number	Symbol	Parameter	Min (ns)	Max (ns)	Notes*
1	TwMAS	Memory Address Strobe Width	60		5
2	TdA(MAS)	Address Valid to Memory Address Strobe ↑ Delay	30		5
3	TdMR/W (MAS)	Memory Read/Write to Memory Address Strobe ↑ Delay	30		5
4	TdMDS(A)	Memory Data Strobe ↑ to Address Change Delay	60		
5	TdMDS (MR/W)	Memory Data Strobe ↑ to Memory Read/Write Not Valid Delay	80		
6	Tw(MDS)	Memory Data Strobe Width (Write Case)	160		6
7	TdD0(MDS)	Data Out Valid to Memory Data Strobe ↓ Delay	30		5
8	TdMDS(D0)	Memory Data Strobe ↑ to Data Out Change Delay	30		5
9	Tw(MDS)	Memory Data Strobe Width (Read Case)	230		6
10	TdMDS(DI)	Memory Data Strobe ↓ to Data In Valid Delay		160	7
11	TdMAS(DI)	Memory Address Strobe ↓ to Data In Valid Delay		280	7
12	ThMDS(DI)	Memory Data Strobe ↑ to Data In Hold Time	0		
13	TwSY	Instruction Sync Out Width	160		
14	TdSY(MDS)	Instruction Sync Out to Memory Data Strobe Delay	200		
15	TwI	Interrupt Request via Port 3 Input Width	100		

NOTES:

5. Delay times are specified for an input clock frequency of 4 MHz. When operating at a lower frequency, the increase in input clock period must be added to the specified delay time.
6. Data strobe width is specified for an input clock frequency of 4 MHz. When operating at a lower frequency, the increase in three input clock periods must be added to the specified width. Data strobe width varies according to the instruction being executed.
7. Address strobe and data strobe to data in valid delay times represent memory system access times and are given for a 4 MHz input frequency.
8. All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0."
9. All output ac parameters use test load 2.

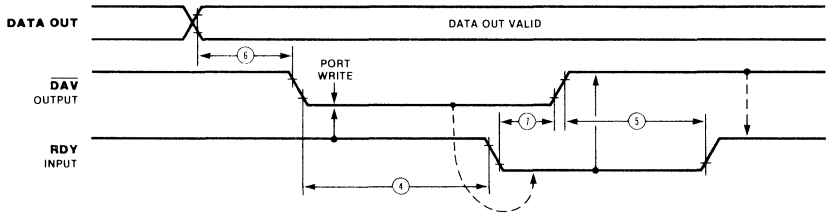
*Timings are preliminary and subject to change.

Handshake Timing



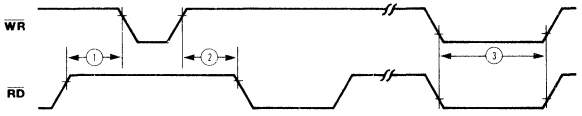
Input Handshake

Handshake Timing (Continued)

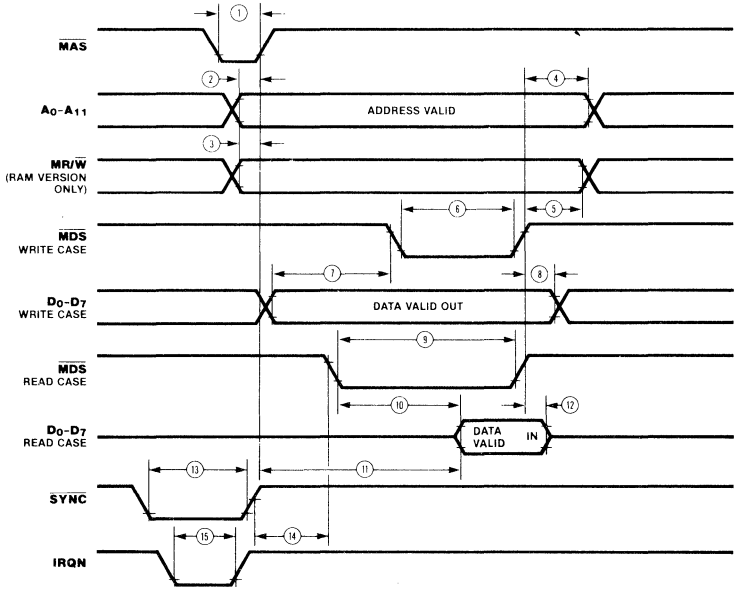


Output Handshake

Reset Timing



RAM Version Program Memory Timing

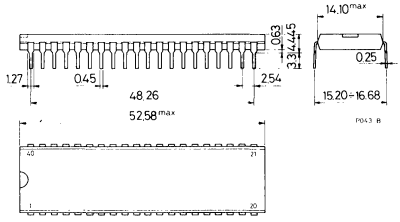


Ordering Information

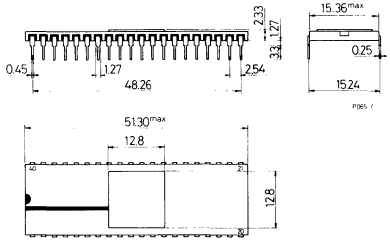
Type	Package	Temp	Clock	Description
Z8590 B1	Plastic 40 pin	0/+70°C	4MHz	Z8590 Universal Peripheral Controller
B6	Plastic 40 pin	-40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D2	Ceramic 40 pin	-55/+125°C		
D6	Ceramic 40 pin	-40/+85°C		
Z8590A B1	Plastic 40 pin	0/+70°C	6MHz	
B6	Plastic 40 pin	-40/+85°C		
D1	Ceramic 40 pin	0/+70°C		
D6	Ceramic 40 pin	-40/+85°C		
Z8591 Q1	QUIP 64	0/+70°C	4MHz	External ROM
Z8591A Q1	QUIP 64	0/+70°C	6MHz	UPC
Z8592 Q1	QUIP 64	0/+70°C	4MHz	External RAM
Z8592A Q1	QUIP 64	0/+70°C	6MHz	UPC
Z8593 R1	PDP	0/+70°C	4MHz	Piggyback EPROM
Z8593A R1	PDP	0/+70°C	6MHz	UPC
Z8594 R1	PDP	0/+70°C	4MHz	Piggyback EPROM
Z8594A R1	PDP	0/+70°C	6MHz	UPC

Packages

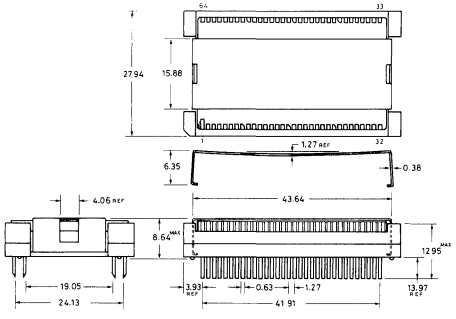
Plastic



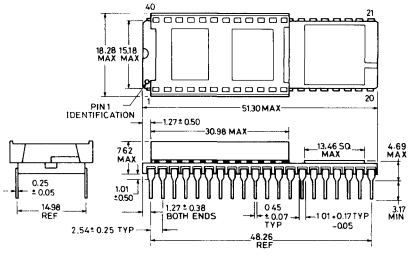
Ceramic



Quip 64



PDP



SGS-ATES GROUP OF COMPANIES

INTERNATIONAL HEADQUARTERS
SGS-ATES Componenti Elettronici SpA
Via C. Olivetti 2 - 20041 Agrate Brianza - Italy
Tel.: 039 - 65551
Telex: 330131 - 330141

BENELUX
SGS-ATES Componenti Elettronici SpA
Benelux Sales Office:
B- 1180 Bruxelles
Winston Churchill Avenue, 122
Tel.: 02 - 3432439
Telex: 24149 B

BRAZIL
Sales Office:
05413 Sao Paulo
Av. Henrique Schaumann Z86 - CJ33
Tel.: 11 - 647245
Telex: 11-21650

DENMARK
SGS-ATES Scandinavia AB
Sales Office:
2730 Herlev
Herlev Torv, 4
Tel.: 02 - 948533
Telex: 35411

EASTERN EUROPE
SGS-ATES Componenti Elettronici SpA
Export Sales Office:
20041 Agrate Brianza - Italy
Via C. Olivetti, 2
Tel.: 039 - 6555287/6555207
Telex: 330131 - 330141

FRANCE
SGS-ATES France S.A.
75643 Paris Cedex 13
Résidence «Le Palatino»
17, Avenue de Choisy
Tel.: 01 - 5842730
Telex: 042 - 250938

HONG KONG
SGS-ATES Singapore (Pte) Ltd.
9th Floor, Block N,
Kaiser Estate, Phase III,
11 Hok Yuen St.,
Hung Hom, Kowloon
Tel.: 3-644251/5
Telex: 63906 ESGIE HX

ITALY
SGS-ATES Componenti Elettronici SpA
Direzione Commerciale Italia
20149 Milano
Via Correggio, 1/3
Tel.: 02-4695651
Sales Offices:
40128 Bologna
Via Corticella, 231
Tel.: 051-326684
00199 Roma
Piazza Gondar, 11
Tel.: 06-8392848/8312777

SINGAPORE
SGS-ATES Singapore (Pte) Ltd.
Singapore 1231
Lorong 4 & 6 - Toa Payoh
Tel.: 2531411
Telex: ESGIES RS 21412

SWEDEN
SGS-ATES Scandinavia AB
19500 Märsta
Bristagatan 16,
Tel.: 0760 - 40120
Telex: 042 - 10932

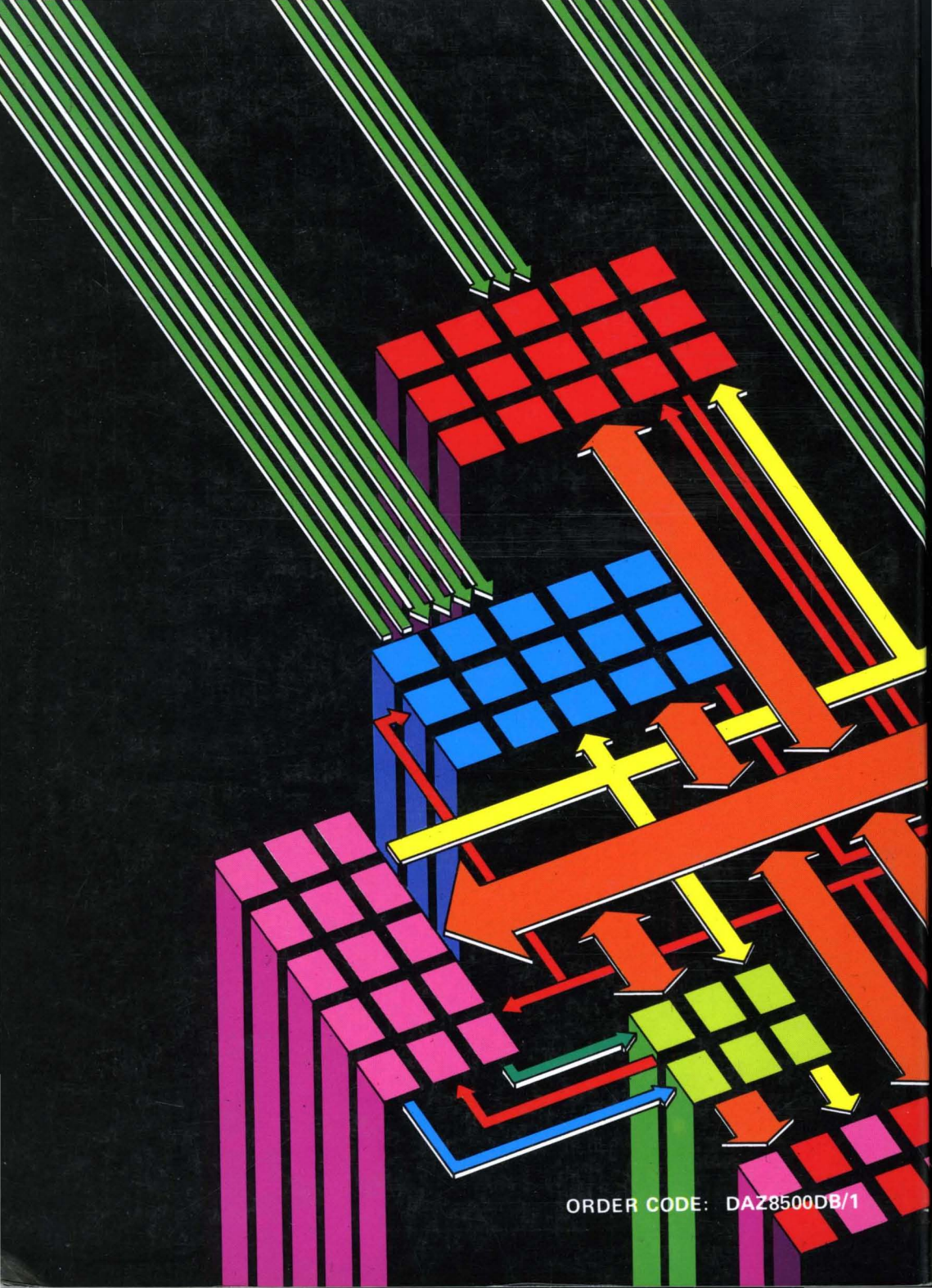
SWITZERLAND
SGS-ATES AG
Swiss Sales Offices
6340 Baar
Oberneuhofstrasse, 2
Tel.: 042 - 315955
Telex: 864915
1218 Grand-Saconnex (Geneve)
Chemin François-Lehmann 22
Tel.: 022 - 986462/3
Telex: 28895

UNITED KINGDOM
SGS-ATES (United Kingdom) Ltd
Aylesbury, Bucks
Planar House, Walton Street
Tel.: 0296 - 5977
Telex: 041 - 83245

U.S.A.
SGS-ATES Semiconductor Corporation
Phoenix, AZ 85022
1000 East Bell Road
Tel.: (602) 867-6100
Telex: SGSPH UR 249976
Sales Offices:
Austin, TX 78757
7113 Burnet Road
Suite 201
Tel.: (512) 458-9182
Dallas, TX 75248
16970, Dallas North Parkway
Suite 702
Tel.: (214) 733-1515
Telex: SGS-ATES DAL 79-5509
Hauppauge, NY 11788
330 Motor Parkway
Suite 100
Tel.: (516) 453-1050
Indianapolis, IND 46141
2346 S. Lynhurst Drive
Suite E-207
Tel.: (317) 241-1116
Santa Clara, CA 95051
2700 Augustine Drive
Suite 209
Tel.: (408) 727-3404
Telex: SGS-ATES SNTA 34-6402
Schaunburg, IL 60196
600 North Meacham Road
Tel.: (312) 490-1890
Telex: SGS CGP DSP 28-2547
Waltham, MA 02154
240 Bear Hill Road
Tel.: (617) 890-6688
Telex: SGS-ATES WHA 92-3495
Woodland Hills, CA 91367
6355 Topanga Canyon Boulevard
Suite 220
Tel.: (213) 716-6600
Telex: SGS-WDHL 18-2863

WEST GERMANY
SGS-ATES Deutschland Halbleiter
Bauelemente GmbH
8018 Grafing bei München
Haidling, 17
Tel.: 08092-690
Telex: 05 27378
Sales Offices:
3012 Langenhagen
Hubertusstrasse, 7
Tel.: 0511 - 772075/7
Telex: 09 23195
8000 München 90
Tegernseer Landstr., 146
Tel.: 089 - 6925100
Telex: 05 215784
8500 Nürnberg 30
Ostendstr., 204
Tel.: 0911 - 572977/78/79
Telex: 0626243
7000 Stuttgart 80
Kalifenweg, 45
Tel.: 0711 - 713091/2
Telex: 07 255545

SGS-ATES GROUP OF COMPANIES
Italy - France - Germany - Malta - Malaysia - Singapore - Sweden - Switzerland - United Kingdom - U.S.A.



ORDER CODE: DAZ8500DB/1