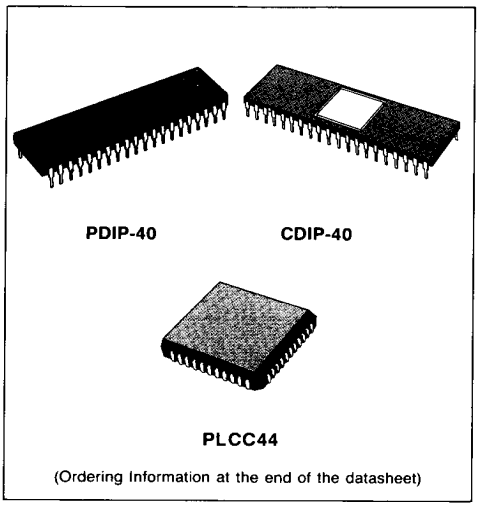
Z-FIO/FIFO INPUT/OUTPUT INTERFACE UNIT

DESCRIPTION

The Z8038 FIO (FIFO Input/Output Interface Unit) is a 128-byte buffer that interfaces two CPUs or a CPU and a peripheral device. Multiple FIOs can be used to create a 16-bit or wider data path, or two can be connected to form a 256-byte FIFO RAM buffer.

The FIO manages data transactions by assuming one of 12 operating modes, using one or more of the following signal configurations : Z-BUS™ high-byte microprocessor, Z-BUS low-byte microprocessor, non-Z-BUS microprocessor, interlocked 2-wire handshake I/O, and 3-wire handshake I/O. These configurations interface dissimilar CPUs or CPUs and peripheral devices running at different speeds or under different protocols. This allows asynchronous data transactions and byte-per-cycle DMA operation and cuts I/O overhead by as much as two orders of magnitude. Figure 1 and 2 illustrate logic functions and pin configuration.



PDIP-40 CDIP-40

PLCC44

(Ordering Information at the end of the datasheet)

Figure 1 : FIO Logic Functions.

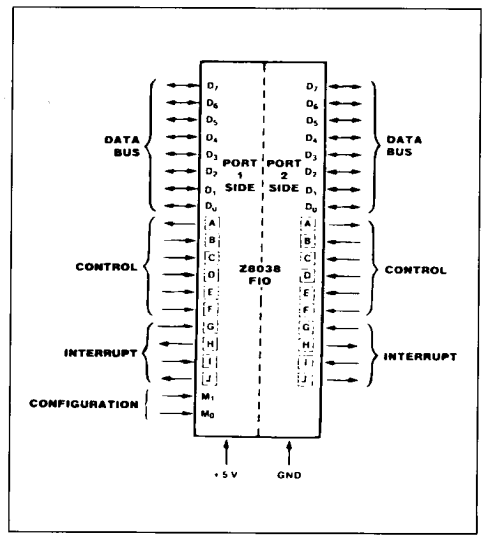


Figure 2 : Dual in Line Pin Connection.

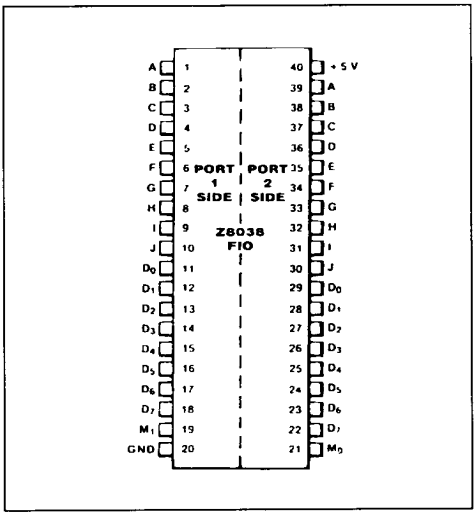
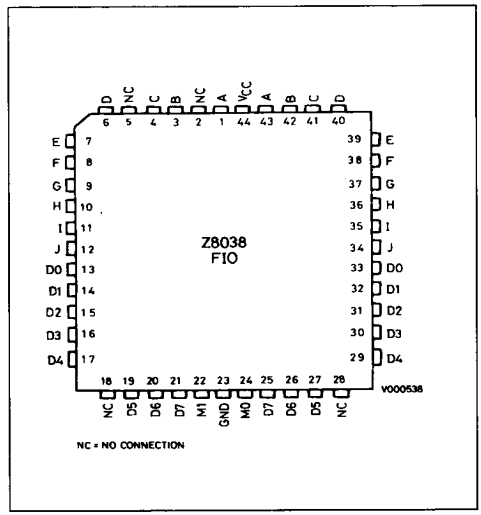


Figure 2a: Chip Carrier Pin Connection.



ARCHITECTURAL DESCRIPTION

The FIO provides an asynchronous, 128-byte FIFO buffer between two CPUs or between a CPU and a peripheral device. Figure 3 shows the general architecture of the FIO. Two or more FIOs can be used in parallel to create a 16-bit or larger interface. Figure 4 shows a 32-bit interface constructed from four FIOs. Two FIOs can be combined to create 256 bytes of buffer space, and additional buffer space can be provided by adding one or more Z8060 FIFO buffers. Figure 5 shows a 512-byte buffer constructed from two FIOs and two FIFOs.

The two ports (1 and 2) are controlled through 16 programmable, directly accessible registers. These registers specify the operating mode of the FIO and provide a message register for CPU-to-CPU communication without involving the FIFO buffer.

The FIO supports DMA operation by facilitating variably-sized block transfers and data transactions to or from memory each machine cycle. Since devices can write to or read from either side of the FIO buffer asynchronously, as well as enable interrupts upon specified conditions, system I/O overhead can be significantly reduced.

ARCHITECTURAL DESCRIPTION (continued)

Figure 3 : FIO Functional Block Diagram.

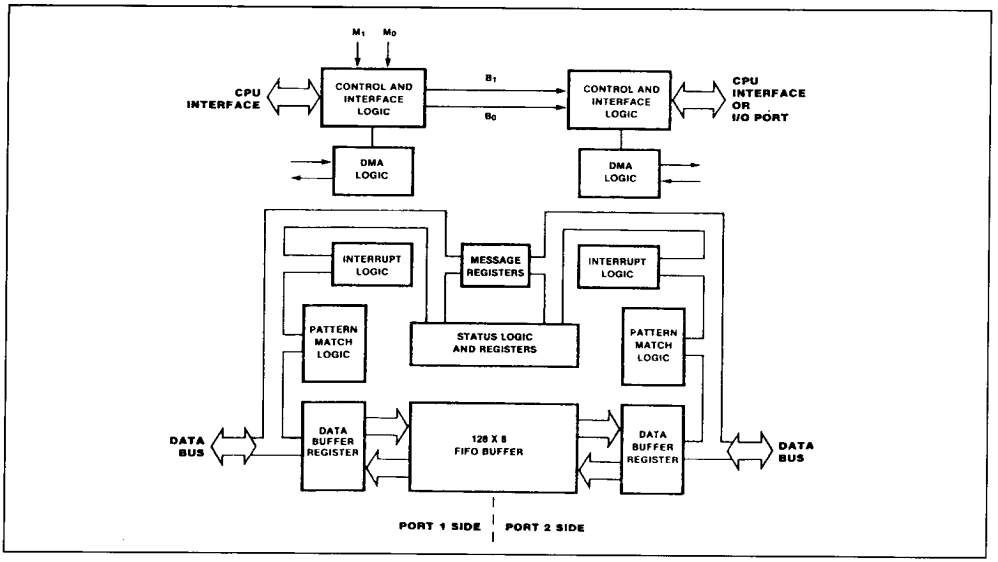
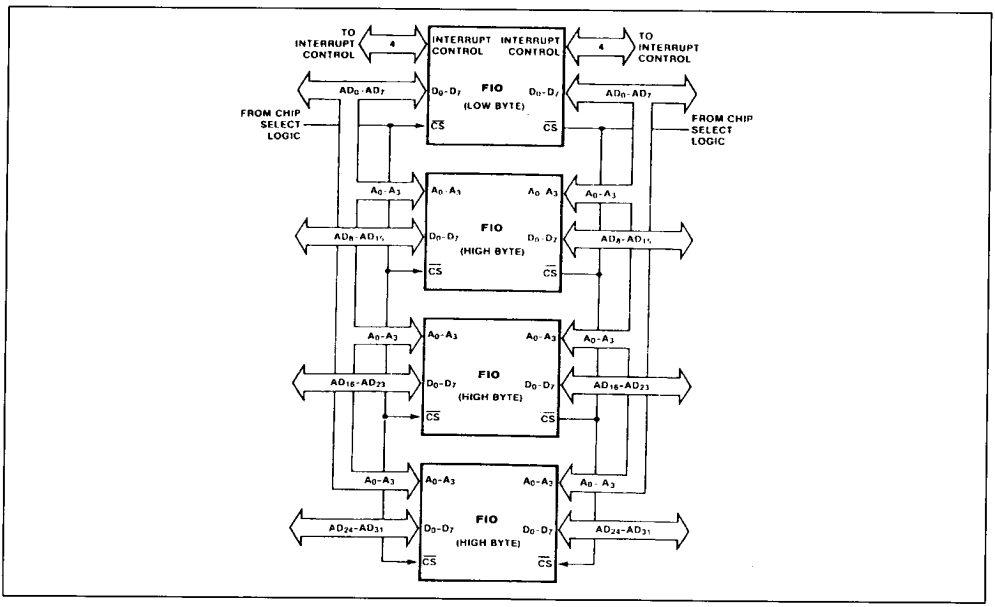
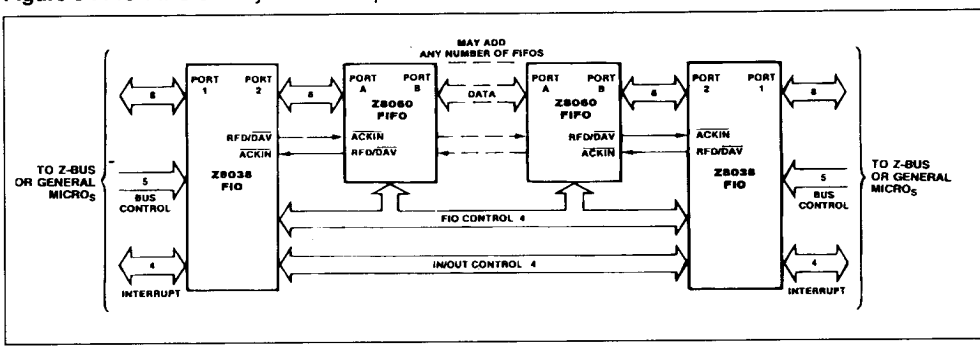


Figure 4 : FIO 32-Bit Width Expansion.



ARCHITECTURAL DESCRIPTION (continued)

Figure 5 : FIO-FIFO 512-Byte Buffer Expansion.



FUNCTIONAL DESCRIPTION

The FIO manages data transfers by assuming one of the 12 operating modes listed in Table 1. These modes are various combinations of the five interfacing protocols.

The five interfacing protocols are Z-BUS high-byte microprocessor, Z-BUS low-byte microprocessor, non-Z-BUS microprocessor (a generalized microprocessor interface), 2-wire interlocked handshake, and 3-wire handshake. Pins A through J carry the signals used in these interfaces ; Table 1-2 shows the pin assignments for each of the interfaces. Also see Appendix A for all 12 pin assignments.

Sixteen programmable registers control each port. Once the operating mode is specified, the internal registers are programmed to specify the direction of data transfer, the transfer of inter-CPU message bytes, external interrupt control, and various internal interrupt conditions.

The FIO improves I/O transaction efficiency by providing seven sources of interrupt generation. The FIO also provides an Interrupt vector that can be programmed to identify the reason for the interrupt.

Table 1 : Operating Modes.

Mode	M1	M0	B1	B0	Port 1	Port2
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-Wie HS
3	0	0	1	1	Z-BUS Low Byte	2-Wire HS
4	0	1	0	0	Z-BUS Low 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 HS
7	0	1	1	1	Z-BUS High Byte	2-Wire HS
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 HS
11	1	0	1	1	Non Z-BUS	2-Wire HS

FUNCTIONAL DESCRIPTION (continued)

Table 2 : Pin Assignments.

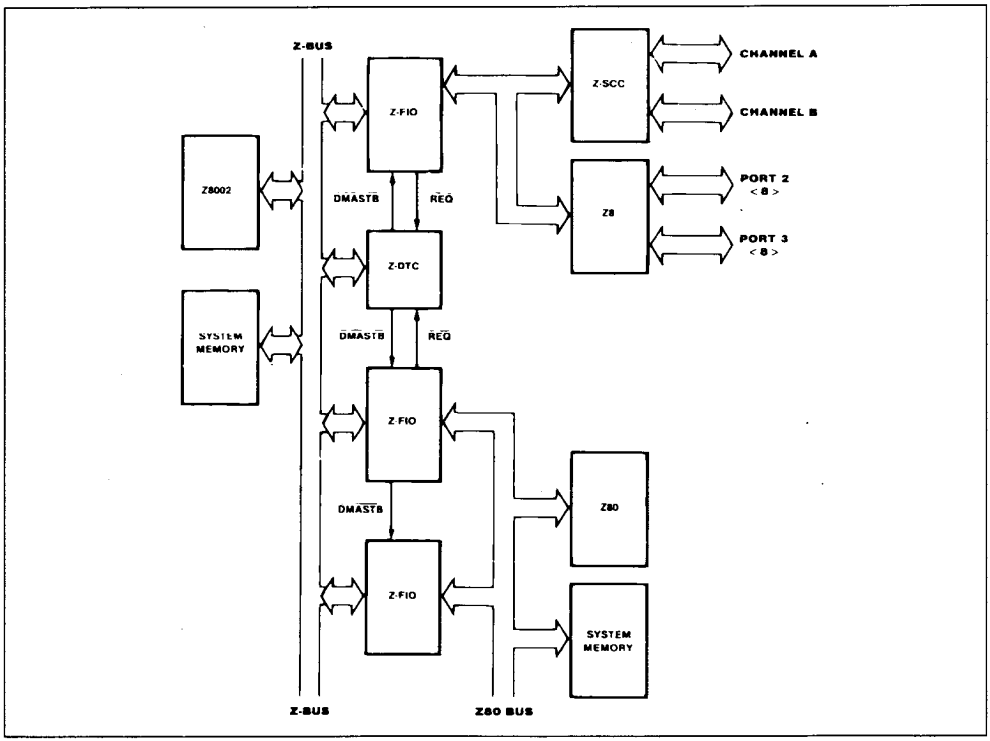
	Z-BUS Low Byte	Z-BUS High Byte	Non Z-BUS	Interlocked HS Port *	3-Wire HS Port *
A	$\overline{\text{REQ/WAIT}}$	$\overline{\text{REQ/WAIT}}$	$\overline{\text{REQ/WAIT}}$	RFD/$\overline{\text{DAV}}$	RFD/$\overline{\text{DAV}}$
B	$\overline{\text{DMASTB}}$	$\overline{\text{DMASTB}}$	$\overline{\text{DACK}}$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

FIO IN DISTRIBUTED NETWORKS

In typical interfacing applications, such as the distributed processing system shown in Figure 6, the Port 1 CPU (Z8002) loads a series of bytes to the FIO's Port 1 registers. This block of control information characterizes the FIO's signal configuration for a specific transaction. The Port 2 CPU can then specify Port 2 operating conventions. The Port 1 CPU remains in control, dynamically reading or writing to the Port 1 registers, either to cause a single change or to replace an entire block of control bytes as specified.

FUNCTIONAL DESCRIPTION (continued)

Figure 6 : Using FIOs in a Distributed Network.



ARCHITECTURAL DESCRIPTION

As figure 3 shows, the FIO architecture is organized as two sets of interface logic and registers. Once programmed, these registers define the two groups of signal lines (A-J) that provide interfacing and timing data to control data flow into and out of the FIO. Table 3 shows how control signals are mapped to these pins in each of the five interfacing protocols.

This chapter also describes the 16 registers in each side of the FIO : the Control registers, Data Buffer register, Byte Count registers, Pattern Match registers, Message registers, and the Interrupt Status registers. All registers, are directly addressable, although in certain operating modes registers must be addressed indirectly.

REGISTER ARCHITECTURE

Thirty-two registers (16 for each port) control the operating modes and pin signals of the FIO. All registers are addressable and can be read from or written to, except the Byte Count and Message In registers, which are read only. All 16 registers are used by Port 1 ; Port 2 uses all 16 except for Control Register 2. These registers are organized into six groups that control similar functions or signals.

The control registers define the operating mode for both ports, as well as control device-level interrupts, some functional interrupts, addressing, Reset, Request/Wait, Data Direction, and Clear. The Data Buffer register buffers data into and out of the 128-byte FIFO. The Byte Count registers monitor the number of bytes left in the FIFO buffer and provides a compare byte for interrupt generation when a specified value is reached during data transfers. The Pattern Mathc registers hold a bit pattern that is compared with the byte in the data buffer. If the bit patterns match, the FIO requests an interrupt. They also provide a mask byte, which forces true comparisons for any bit set. The Message registers transfer byte messages between Port 1 and Port 2 CPUs when in the CPU-to-CPU interface modes. The Interrupt Status registers control generation of interrupt requests. The following sections describe these registers in detail.

Register Addressing Z-BUS High and Low Byte. The Right Justify Address (RJA) bit in Control Register 0 specifies the bits used in register addressing. In the Z-BUS low-byte configuration, when RJA is 0 (the default condition), Address/Data lines AD_1-AD_4 carry the register address. When RJA is 1, Address/Data lines AD_0-AD_3 carry the register address, allowing CPUs to place a 4-bit address on the lowest nibble of the Address/Data lines. Table 4 describes FIO register addressing.

The Z-BUS high-byte configuration is normally used with another FIO in the Z-BUS low-byte configuration, forming a 16-bit interface. In this configuration, Address/Data lines AD_0-AD_3 or AD_1-AD_4 are wired to pins G-J, as appropriate, and the RJA bit does not affect register addressing.

PINS COMMON TO BOTH SIDES

Pin Signals	Pin Names	Pin Numbers	Signal Description
M_0 M_1	M_0 M_1	21 19	M1 and M0 Program Port 1 Side CPU Interface
+ 5 V_{dc}	+ 5 V_{dc}	40	DC Power Source
GND	GND	20	DC Power Ground

REGISTER ARCHITECTURE (continued)

Table 3 : Pin Description.

Z-BUS LOW BYTE MODE

Pin Signals	Pin Names	Pin Numbers Port 1	Pin Numbers Port 2	Signal Description
AD_0-AD_7	D_0-D_7	11-18	29-22	Multiplexed Bidirectionnal Address/data Lines, Z-BUS Compatible
$\overline{REQ}/\overline{WAIT}$ (request/wait)	A	1	39	Output, Active Low, REQUEST (ready) Line for DMA Transfer ; 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. Acknowledge 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.

REGISTER ARCHITECTURE (continued)

Z-BUS HIGH BYTE MODE

Pin Signals	Pin Names	Pin Numbers Port 1	Pin Numbers Port 2	Signal Description
AD_0-AD_7	D_0-D_7	11-18	29-22	Multiplexed Bidirectionnal Address/data Lines, Z-BUS Compatible
$\overline{REQ}/\overline{WAIT}$ (request/wait)	A	1	39	Output, Active Low, REQUEST (ready) Line for DMA Transfer ; 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.
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.

REGISTER ARCHITECTURE (continued)

NON-Z-BUS MODE

Pin Signals	Pin Names	Pin Numbers Port 1	Pin Numbers Port 2	Signal Description
D_0-D_7	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 ; 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 D0-D7 ; Active low indentifies Data byte on D0-D7.
$\overline{INTACK}$ (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.

REGISTER ARCHITECTURE (continued)

PORT 2– I/O PORT MODE

Pin Signals	Pin Names	Pin Numbers	Mode	Signal Description
D_0-D_7	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 FIO 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 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_i	H	32	2-Wire HS 3-Wire HS	Output Line from D_1 of Control Register 3.
$\overline{Output\ Enable}$	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

REGISTER ARCHITECTURE (continued)

Table 4 : Register Addressing.

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 / RJA = 1	AD_7-AD_5 / AD_7-AD_4	AD_4 / AD_3	AD_3 / AD_2	AD_2 / AD_1	AD_1 / AD_0	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

Non-Z-BUS (Nonmultiplexed) Microprocessor. In the non-Z-BUS microprocessor configuration, the FIO connects to the system data bus via D_0-D_7 but not to the address bus. The CPU must provide address information on the data bus and the FIO must distinguish between control and data bytes. The C/$\overline{D}$ line (pin F) is pulled Low to indicate that the byte on Data lines D_0-D_7 should go into the Data Buffer register. When the C/$\overline{D}$ line is held High, incoming bytes are interpreted as control bytes. To write to any Control register, C/$\overline{D}$ is forced High ; the address of the destination register is written to the Register Pointer, then read or written to the target register.

Port 2 I/O Mode. When Port 2 is programmed for either I/O Port mode, none of the Port 2 side Control registers can be programmed. The only register available is Port 2's Data Buffer register.

Control Registers. Control Registers 0-3 define FIO operation according to their programming. The Port 1 Control registers must always be programmed ; the Port 2 Control registers are functional only if Port 2 is in a CPU interface mode, and not if Port 2 is in an I/O interface mode. Even when Port 2 interfaces to a CPU, the Port 2 interface is disabled unless the Port 1 CPU enables Port 2 by setting bit 0 in the Port 1 Control Register 2. Figure 7 shows the four Control registers.

Unless noted in the following descriptions, the register architectures of Port 1 and Port 2 are identical.

Control Register 0. Control Register 0 controls the Master Interrupt Enable, Disable Lower Daisy Chain, Nonvectored Interrupt, Vector Includes Status, Port 2 Operating Mode Select, Right Justify Address, and Reset functions. All bits except D_0 are forced to 0 by Reset. Each of these functions is programmed by a single bit, except for the Port 2 Operating Mode Select, which requires two bits to program one of four possible interfaces. When programming bits in this register, be careful not to accidentally change the Port 2 Operating Mode Control bits, B_1 and B_0.

If reset, bit 7 (Master Interrupt Enable) disables all interrupts from the FIO. If set, individual interrupts can be enabled by setting the appropriate IE (Interrupt Enable) bit in the Interrupt Status registers.

REGISTER ARCHITECTURE (continued)

Bit 6 (Disable Lower Daisy Chain), if set, disables interrupt generation by devices with a lower priority on the Z-BUS interrupt daisy chain by forcing IEO Low.

Bit 5 (No Vector On Interrupt), if set, floats the data bus outputs at high impedance during an active Interrupt Acknowledge signal. The interrupt source can be identified by polling the Interrupt Status registers' Interrupt Pending bits, or by testing to find which IUS bit was set. Do not set this bit when using the nonvectored interrupt on the Z8000.

Bit 4 (Vector Includes Status), if set, codes the reason for the interrupt request into the Interrupt vector, where it can be read by the CPU.

Bits 3 and 2 program the Port 2 operating mode, as described in Table 1. In this table, bit 3 is shown as B_1 and bit 2 is shown as B_0. (See also figure 7).

Bit 1 (Right Justify Address), if reset, specifies that bits 1-4 of the Address/Data bus address the internal FIO registers. This convention is compatible with the Z8001 and Z8002 byte I/O addressing modes and is the default state. When this bit is set, it specifies that bits 0-3 of the bus address the internal FIO registers. This convention supports other CPUs such as the Z8. Since the address of Control Register 0 is 00_H, the setting of RJA has no effect on addressing this register in any mode. In the non-Z-BUS configurations, it is forced to 1. This bit has no effect in Z-BUS High Byte mode, since the address information comes in on pins G, H, I, and J.

Bit 0 (Reset), if set, forces the FIO into a reset state, with all Control registers cleared and all Port 2 outputs at high impedance. This bit must be reset before initialization can take place.

Control Register 1. As shown in figure 7, bits in Control Register 1 freeze the Byte Count register value, indicate transmitted or received messages in the Message registers, stop request transfers upon pattern match or start request on Byte Count register true match, and enable Wait/Request signals. All bits in this register are cleared by reset.

Bit 7 of Control Register 1 is not used and must be programmed to 0. This bit always reads 0.

Bit 6 (Freeze Byte Count), if set, freezes the present value in the Byte Count register so the CPU can obtain a stable value to read. Since bytes can be transferred to and from FIO RAM while such a read is in progress, the frozen value in the Byte Count register might not reflect the ongoing byte count within FIO RAM. Bit 6 is reset upon completion of the CPU read of the Byte Count register. The ongoing count appears in the Byte Count register after the read.

Bit 5 (Message Register Out Full), if set, indicates that the CPU has placed a message in its Message Out register. This bit is reset when the receiving CPU reads the message in its Message In register. This bit is the other CPU's message IP bit and is a read-only bit.

Bit 4 (Message Register Interrupt Under Service), if set, indicates that the other CPU has received a message in its Message In register. This bit is the message IUS (Interrupt Under Service) bit of the other CPU and is a read-only bit.

Bit 3 (Stop Request On Pattern Match), if set, forces the request line High upon a true match between the current byte in the data buffer and the byte loaded into the Pattern Match register. DMA operation halts when the Request line is forced High.

Bit 2 (Start Request On Byte Count), if set, forces the Request line Low upon a true match between the current count in the Byte Count register and the value loaded into the Byte Count Compare register. DMA operation starts when the Request line is forced Low. See figures 27 and 28 for more details.

Bit 1 (Request/Wait), if set, selects the request function for use in high-speed data transfers, such as DMA transactions. If bit 1 is reset, the Wait function is selected (and the output is open drain) for use in lower-speed data transfers and CPU synchronization.

Bit 0 (Request/Wait Enable), if set, enables the Request/Wait signals (see bit 1) during CPU-to-CPU transactions. When reset, the Request/Wait pin is at high impedance.

Control Register 2. Only bits 0 and 1 are used in Control Register 2 (figure 7). Bits 7 through 2 are not used and should be programmed to 0. These bits always read 0. Bits 0 and 1 are cleared by reset. Only Port 1 has a Control Register 2. When the Port 2 CPU reads its Control Register 2, it will always read 00_H.

Bit 1 (Port 2 Side Handshake Enabled), if set, enables Port 2 I/O operation when bits 2 and 3 of Control Register 0 specify one of the I/O Handshake modes shown in Table 1. When this bit is reset, the handshake are disabled ; RFD/DAV is forced High.

Bit 0 (Port 2 Enabled), if set, allows the Port 2 CPU to program the Port 2 registers-when Port 2 is programmed as a CPU interface-and to assume control of Port 2 operation. The Port 2 side is at high impedance until this bit equals 1.

Control Register 3. Bits 7-4 in Control Register 3 enable and control the Clear and Data Direction si-

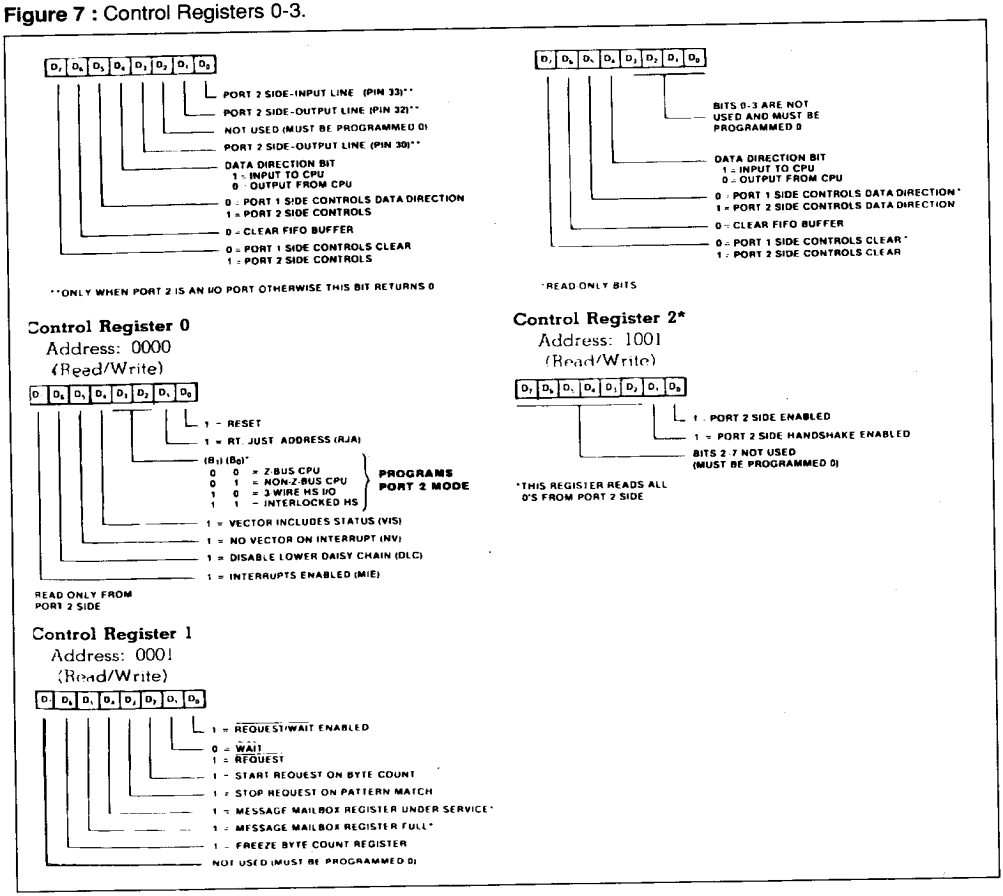
REGISTER ARCHITECTURE (continued)

gnals (figure 7) ; bits 3, 1, and 0 are simple I/O bits for the Port 1 CPU only. Bits 3-0 on the Port 2 side always read 0. Bit 2 is not used and must be programmed to 0 ; it will always read 0. All bits in this register are cleared to 0 by reset. Bits 7 and 5 can be programmed only by the Port 1 CPU. These bits are read-only in the Port 2 Control Register 3.

Bit 7 (Clear, Port 2/Port 1), for CPU-to-CPU operation if set, specifies that Port 2 CPU controls the Clear signal (bit 6). If bit 7 is reset, Port 1 CPU controls the Clear signal. If Port 2 is programmed as an I/O port, this bit programs pin 35 as either input or output (0 = output, 1 = input). This bit can be programmed only by the Port 1 CPU. This is a read-only bit for the Port 2 CPU.

Bit 6 (Clear FIFO Buffer), if reset, clears the FIO RAM buffer. Bit 7 controls which CPU issues Clear signals. (See preceding paragraph). This bit becomes a read-only bit by the CPU not in control of the Clear function.

Figure 7 : Control Registers 0-3.



REGISTER ARCHITECTURE (continued)

Bit 5 (Data Direction, Port 2/Port 1), for CPU-to-CPU operation, if set, specifies that Port 2 CPU controls the direction of data flow through FIO RAM. If bit 5 is reset, Port 1 CPU controls the Data Direction signal. If Port 2 is programmed as an I/O port, this bit programs pin 34 as either an input or an output (0 = output, 1 = input). This bit can be programmed only by the Port 1 CPU. This is read-only bit for the Port 2 CPU.

Bit 4 (Data Direction), if set, specifies that data moves from the FIO into the CPU (1 = input to CPU ; 0 = output from CPU). If bit 4 is reset, data moves from the controlling CPU into the FIO. When Port 2 is an I/O port, a zero (0) on the Data Direction pin (pin 34) means that Port 2 is an output handshake port. Conversely when the Data Direction pin is a one (1), Port 2 is an input handshake port.

Bits 3 and 1 are Port 2 outputs (pins 30 and 32, respectively) when Port 2 is programmed as an I/O port. These bits always read 0 for the Port 2 CPU.

Bit 2 is not used and must be programmed to 0. This bit always reads 0.

Bit 0 reads pin 33, a Port 2 input, when Port 2 is programmed as an I/O port. When Port 2 is not programmed as an I/O port, this bit reads 0. This bit always reads 0 for the Port 2 CPU.

Data Buffer Register. The Data Buffer register (figure 8) latches the byte written to or read from FIO RAM. When the Pattern Match register contains a compare byte, it is compared with the bytes passing through the Data Buffer register.

Byte Count Registers. There are two Byte Count registers (figure 8). One contains the ongoing count of bytes present in FIO RAM ; the other holds a byte value that is compared with this ongoing count.

Byte Count Register. The Byte Count register contains an ongoing count of the number of bytes in FIO RAM. This value is the number of bytes written into RAM minus the number of bytes read from RAM. The largest value contained in this register is 80_H (128 decimal). This value can be frozen by setting bit D_6 in Control Register 1. This halts the count so an accurate read can be accomplished. This is a read-only register.

Byte Count Compare Register. The Byte Count Compare register is loaded with a byte value that is

Figure 8 : Data Buffer and Byte Count Registers.

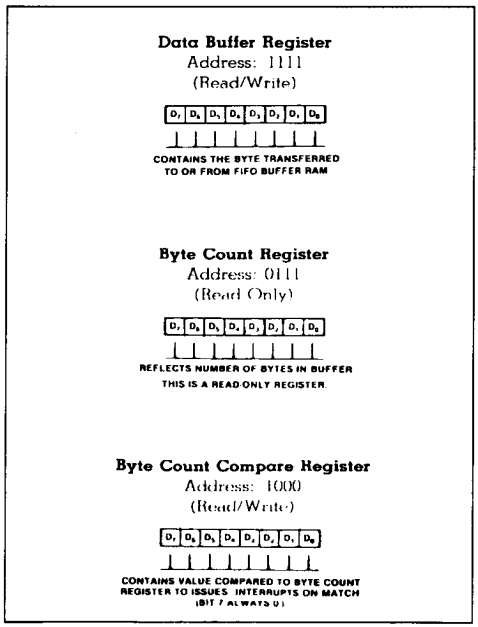
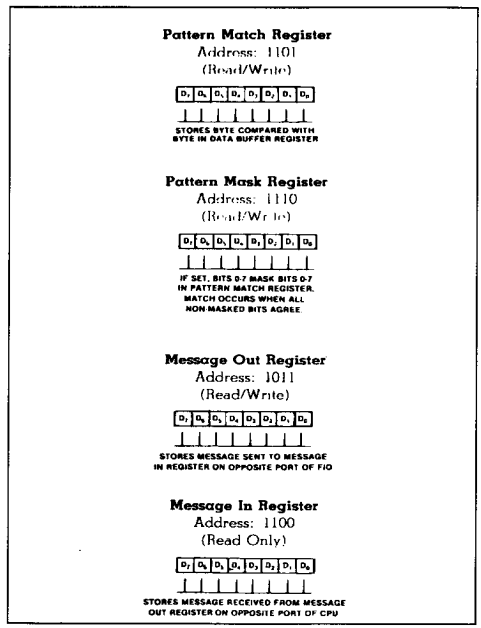


Figure 9 : Pattern Match and Message Registers.



REGISTER ARCHITECTURE (continued)

continuously compared with the ongoing byte count in the Byte Count register. The largest programmable value is $7F_H$ (127 decimal). Bit 7 always reads 0. If the Byte Count Compare Interrupt bit is set (see Interrupt Status Register 2), an interrupt occurs upon a true match.

Pattern Match Registers. The Pattern Match register contains a byte for comparison with the byte in the Data Buffer. The Pattern Mask register contains a byte pattern used to force a true match.

The Pattern Match and Data Buffer registers are in an unknown state on power-up or after reset and may therefore match. This match may set the Pattern Match IP and the Pattern Match flag.

Pattern Match Register. The Pattern Match register (figure 9) contains a byte for comparison with the byte in the Data Buffer register. As each byte in a data transaction passes through the Data Buffer register, it is compared with the value programmed in the Pattern Mathc register. Upon a true match, an interrupt can be generated if enabled in Interrupt Status Register 1. One or more bits in the Pattern Match register can be masked by the value in the Pattern Mask register.

Pattern Mask Register. The Pattern Mask register (figure 9) contains a bit pattern used to force a true match between the pattern in the Pattern Match register and the pattern in the Data Buffer register. If a Pattern Mask bit is set, the corresponding bit in the Pattern Match register always matches the corresponding bit in the Data Buffer register. A match occurs when all nonmasked bits agree. All 1s in this register forces a pattern match. This register is cleared to 0 by reset.

Message Registers. The Message registers (figure 9) transfer messages between CPUs.

Message Out Register. The CPU sends a message by writing the byte into its Message Out register which also puts it in the other CPU's Message In register. When this is done, the receiving CPU receives an interrupt request (if the IE bit is enabled in Interrupt Status Register 0). The Message Out register of one CPU is the Message In register for the other CPU.

Message In Register. The CPU receiving a message byte also receives an interrupt request (if the IE bit is enabled in Interrupt Status Register 0), which signifies that a message has appeared in its Message In register. This is a read-only register.

Interrupt Registers. There are four Interrupt Status registers, shown in figure 10, which control and monitor the FIO internal interrupt functions. Prioritized interrupts can occur if they are enabled. The CPU can read the appropriate status register to see if a specific interrupt condition is enabled ; other bits in the Interrupt Status registers show if an interrupt is pending or if it under service. Each interrupt condition is described by three Interrupt Status bits : IE (Interrupt Enabled), IP (Interrupt Pending) and IUS (Interrupt Under Service).

Interrupt Status Register 0. In Interrupt Status Register 0, only bits 7, 6, and 5 are used. Bits 4-0 must be programmed to 0 ; these bits always read 0. All bits are cleared to 0 by reset.

Bit 7 is the Message Interrupt Under Service (IUS) bit and, when set, indicates that a Message interrupt is under service by the receiving CPU.

Bit 6 is the Message Interrupt Enable (IE) bit and, when set, indicates that a message interrupt will be issued when a message is received in the Message In register.

Bit 5 is the Message Interrupt Pending (IP) bit and, when set, indicates that a message interrupt to the receiving CPU is pending. This bit is reset when the read of the Message In register is completed.

Interrupt Status Register 1. Interrupt Status Register 1 controls and monitors Data Direction Change interrupts and Pattern Match interrupts. It also shows if a true pattern match has taken places. All bits except D_1 and D_0 are cleared by reset. Bits D_1 and D_0 may be a 1 or 0 depending upon whether a match condition exists or not.

Bits 7, 6 and 5 show the status of Data Direction Change interrupts and are cleared by reset. When bit 7 is set, a Data Direction Change interrupt is under service. When bit 6 is set, Data Direction Change interrupts are enabled. When bit 5 is set, a Data Direction Change interrupt is pending. Bits 7 and 5 can also be set or reset by program command. Bit 6 is programmed set or reset.

Bit 4 is not used and must be programmed to 0. This bit always reads 0.

Bits 3, 2, and 1 show the status of Pattern Match interrupts. When bit 3 is set, a Pattern Match interrupt is under service. When bit 2 is set, Pattern Match interrupts are enabled. When bit 1 is set, a Pattern Match interrupt is pending. Bits 3 and 1 can also be

REGISTER ARCHITECTURE (continued)

set or reset by program command. Bit 2 is programmed set or reset. Because of a possible match condition on power-up or reset, the match IP bit should be cleared before enabling the Pattern Match IE bit.

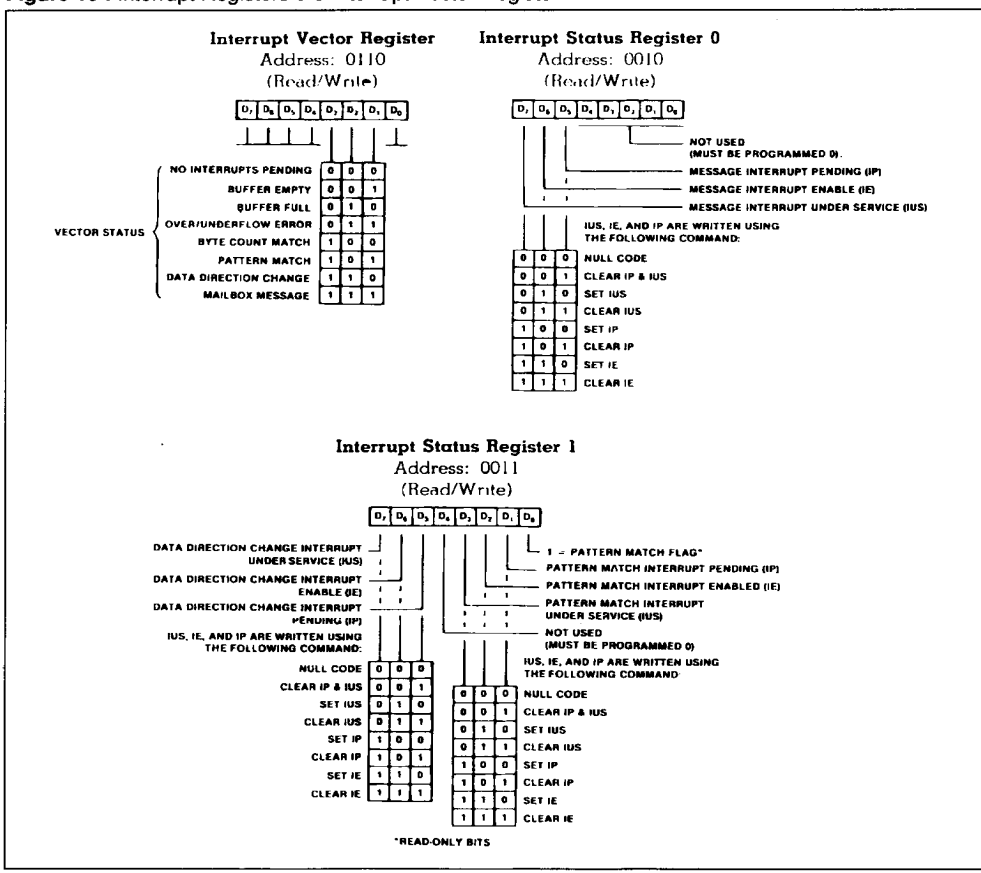
Bit 0 (Pattern Match Flag) is 1 whenever a true pattern match exists between the Data Buffer and the Pattern Match register. The Data Buffer register and the Pattern Match register initialize in an undefined state during reset or power-up. Consequently, the Pattern Match Flag bit can be set, indicating a false pattern match. If Pattern Match interrupts are not going to be used, this bit can be put in a known state, 1, by writing FF_H to the Pattern Mask register.

Interrupt Status Register 2. Interrupt Status Register 2 controls and monitors Byte Count Compare and Overflow and Underflow Error interrupts. This register is cleared to 0 by reset.

Bits 7, 6, and 5 show the status of Byte Count Compare interrupts. When bit 7 is set, a Byte Count Compare interrupt is under service. When bit 6 is set, Byte Count Compare Interrupts are enabled. When bit 5 is set, a Byte Count Compare Interrupt is pending. Bits 7 and 5 can also be set or reset by program command. Bit 6 is programmed set or reset.

Bit 4 is set whenever an attempt is made to write into a full FIO (Overflow Error). This bit is cleared when

Figure 10 : Interrupt Registers 0-3 Interrupt Vector Register.



REGISTER ARCHITECTURE (continued)

the Error IP bit is cleared. This bit is not set when the Wait function is programmed.

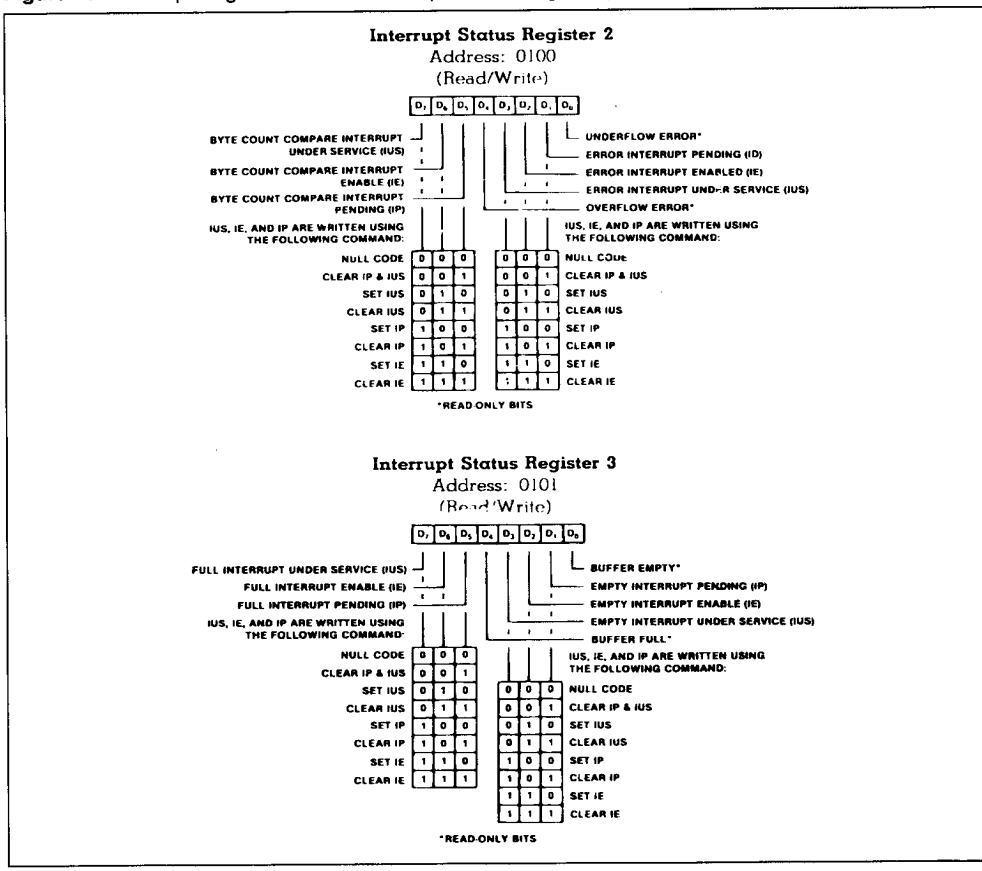
Bits 3, 2, and 1 show the status of Error interrupts. These bits control overflow and underflow errors ; the CPU must read bits 0 and 4 to determine whether overflow or underflow conditions apply. When bit 3 is set, an Error interrupt is under service. When bit 2 is set, Error interrupts are enabled. When bit 1 is set, an Error interrupt is pending. Bits 3 and 1 can also be set or reset by program command. Bit 2 is programmed set or reset.

Bit 0 is set whenever an attempt is made to read from an empty FIO (underflow error). This bit is cleared when the Error IP is cleared. This bit is not set when the Wait function is programmed.

Interrupt Status Register 3. Interrupt Status Register 3 controls and monitors Buffer Full and Buffer Empty interrupts. All bits except D_0 are cleared by reset.

Bits 7, 6, and 5 show the status of Buffer Full interrupts. When bit 7 is set, a Buffer Full interrupt is under service. When bit 6 is set, Buffer Full interrupts are enabled. When bit 5 is set, a Buffer Full interrupt

Figure 10 : Interrupt Registers 0-3 and Interrupt Vector Register (continued).



REGISTER ARCHITECTURE (continued)

is pending. Bits 7 and 5 can also be set or reset by program command. Bit 6 is programmed set or reset. In CPU-to-I/O operation for handshake, the full IP bit is set when the buffer is full and the Full pin (pin 37) is High. For the 3-wire handshake the Buffer Full IP is set when the FIO buffer is full.

Bit 4 is a 1 when the FIO buffer is full in CPU-to-CPU operation. In CPU-to-I/O operation for 2-wire handshake, bit 4 is 1 when the buffer is full and the Full pin (pin 37) is High. For the 3-wire handshake this bit is a 1 when the FIO buffer is full.

Bits 3, 2, and 1 show the status of Buffer Empty interrupts. When bit 3 is set, a Buffer Empty interrupts is under service. When bit 2 is set, Buffer Empty interrupts are enabled. When bit 1 is set, a Buffer Empty interrupt is pending. In CPU-to-I/O operation, the Empty IP bit is set when the buffer is empty and the Empty pin (pin 37) is High.

Bit 0 is 1 when the FIO buffer is empty in CPU-to-CPU operation. In CPU-to-I/O operation, bit 0 is 1 when the buffer is empty and when the Empty pin (pin 36) is High.

Interrupt Vector Register. The Interrupt Vector register (figure 10) holds a byte used as the address of an interrupt service routine (or of a table indexing into such groups of routines). This register can be used in two ways. The Interrupt Vector register can be programmed with a byte address that is gated onto the address bus lines during the Interrupt Acknowledge cycle. This provides a direct vector to a generalized interrupt service routine. If bit 4 of Control Register 0 (Vector Includes Status) is set, however, the Interrupt Vector register includes a 3-bit status code reflecting the reason for the interrupt request as the contents of bits 3, 2, and 1. When MIE is 1, other than during an Interrupt Acknowledge cycle, the Interrupt Vector register always reflects the FIO status in these bits, regardless of whether or not the Vector Includes Status bit is set. When MIE is 0, the base vector can be read back. This provides a means to read what was written to this register (also see table 5). The bit codes are :

Code Bits 3	2	1	Encoded Data
1	1	1	Received Message in Message In Register
1	1	0	Change in Data Transaction Direction
1	0	1	Valid Pattern Match
1	0	0	Valid Byte Count Compare
0	1	1	Overflow or Underflow Error
0	1	0	Buffer Full
0	0	1	Buffer Empty
0	0	0	No Interrupts Pending

Table 5 : Data Read from Interrupt Vector Register.

	MIE = 1 VIS = 1	MIE = 1 VIS = 0	MIE = 0 VIS = 1 or 0
Interrupt Acknowledge Read (INTACK = 0)	Status With Vector	No Status With Vector (base vector)	Interrupts Are Disabled (high impedance)
Non Interrupt Acknowledge Read (INTACK = 1)	Status With Vector	Status With Vector	No Status With Vector (base vector)

FUNCTIONAL DESCRIPTION

The FIO provides an interface between two CPUs or between a CPU and an I/O device. It interfaces to both Z-BUS and non-Z-BUS systems and emulates 2-wire and 3-wire I/O protocols. Figure 11 shows possible FIO operations.

These interfacing tasks require different sets of control signals. After removing the FIO from its reset state and programming the various registers, the FIO assumes one of the 12 operating modes shown in Table 1. In each mode, Port 2 presents one of five possible groups of signals on pins A-J. These signals correspond to the Z-BUS high-byte configuration, Z-BUS low-byte configuration, non-Z-BUS (general microprocessor) configuration, 2-wire handshake configuration, and 3-wire handshake configuration. Port 1 presents one of three groups of signals on pins A-J, corresponding to Z-BUS high-byte, Z-BUS low-byte, or non-Z-BUS configurations.

This chapter describes the signals used in FIO data transactions ; discusses reset operation and uses for the 12 operating modes ; and explains how control signals form interfaces to other devices, how DMA transactions occur, and how the FIO handles internal interrupts and participates in the system-level, daisy-chain interrupt protocols.

INTERFACING SIGNAL

The FIO manages data transfers with control signals, some issued from the bus master CPU, satellite CPU, DMA device, or I/O device, some created from external logic, and some issued from within the FIO itself. There are five basic signal configurations that combine to form 12 operating modes, as shown in Table 1. The control signals used in each configuration are shown mapped to their respective pins in figure 2. The signals fall into various groups, as described in the following sections. Reset is not actually a control signal, but since the condition is caused by control signal states (or programming) it is treated as such here.

Z-BUS High-Byte and Low-Byte. The Z-BUS high-byte and low-byte microprocessor configurations are normally used together to form a 16-bit multiplexed address/data bus interface that is compatible with the Z-BUS. The Z-BUS low-byte microprocessor configuration can be used by itself as an 8-bit interface to a multiplexed bus. The Z-BUS high-byte microprocessor configuration can also be used this way, but it lacks interrupt interfacing signals.

Signals. In the Z-BUS cofigurations, the FIO uses the following set of signals :

Request/Wait, DMA Strobe, Data Strobe, Read/Write, Chip Select, Address Strobe, and the interrupt arbitration signals Interrupt Acknowledge, Interrupt Request, Interrupt Enable In, and Interrupt Enable Out.

$\overline{REQ}/\overline{WAIT}$ *(Request/Wait, output, 3-state, active Low).* $\overline{REQ}$ is sent to a DMA device to begin a DMA transfer, and $\overline{WAIT}$ (an open drain output) tells the CPU that the FIO buffer is full (if data is output from the CPU to the FIO) or that the FIO buffer is empty (if data is input to the CPU from the FIO).

$\overline{DMASTB}$ *(DMA Strobe, input, active Low).* This signal is received by the FIO from a DMA controller. When Low, it acts like a Read/Write signal, strobing data to or from the Data Buffer register directly during DMA transactions.

$\overline{DS}$ *(Data Strobe, input, active Low).* When Low, $\overline{DS}$ specifies that the multiplexed address/data bus is carrying data rather than address information.

R/$\overline{W}$ *(Read/Write, input).* Read and Write signals specify read and write operations. When R/$\overline{W}$ is Low, writes are enabled. When R/W is High, reads are enabled.

$\overline{CS}$ *(Chip Select, input, active Low).* $\overline{CS}$ is usually a signal derived from the controlling CPU's signals by external logic. When Low, $\overline{CS}$ selects the FIO and enables it for operation. This signal is latched on the rising edge of AS.

$\overline{AS}$ *(Address Strobe, input, active Low).* When low, $\overline{AS}$ specifies that the multiplexed address/data bus is carrying address information rather than data.

Z-BUS Low-Byte Configuration Only. In the Z-BUS low-byte configuration, the FIO uses four interrupt arbitration signals to participate in the daisy-chain prioritized interrupt arbitration scheme.

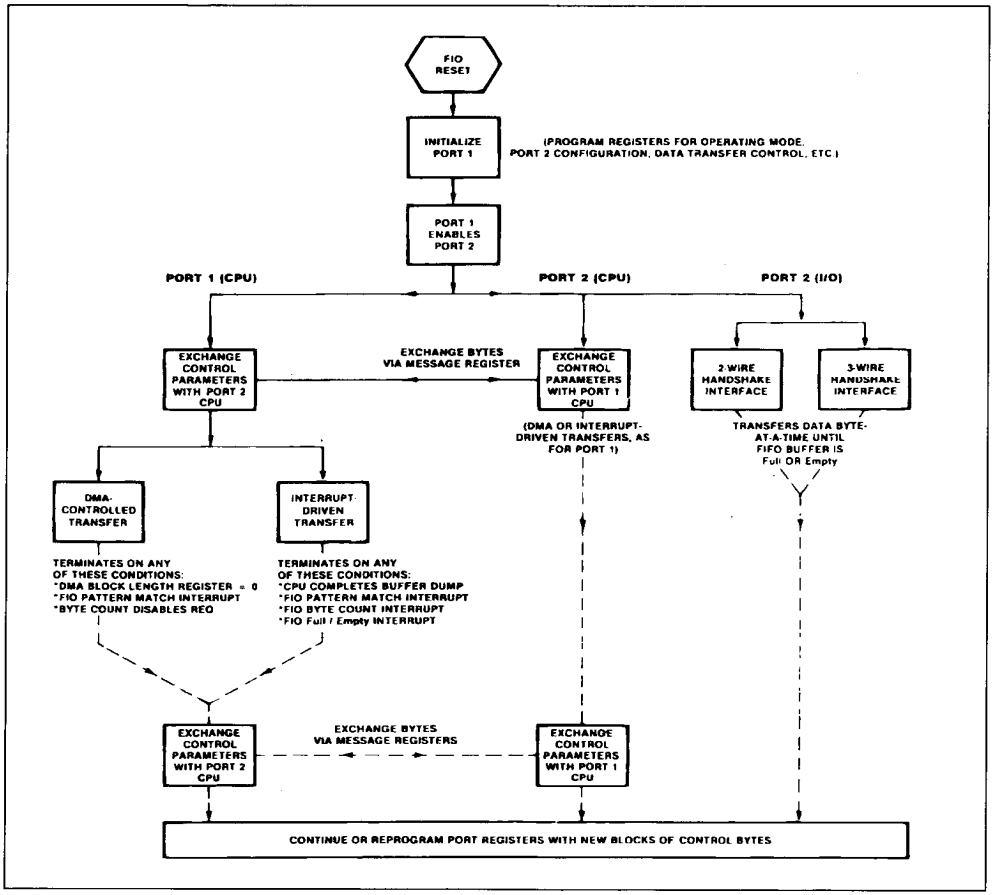
$\overline{INTACK}$ *(Interrupt Acknowledge, input, active Low),* IEI *(Interrupt Enable In, input, active High),* IEO *(Interrupt Enable Out, output, active High),* $\overline{INT}$ *(Interrupt Request, output, open drain, active Low).* $\overline{INTACK}$, IEI, IEO, and $\overline{INT}$ control FIO interrupt operation. $\overline{INT}$ requests an interrupt from the CPU when

INTERFACING SIGNALS (continued)

active (Low) ; INTACK is the interrupt acknowledgement from the CPU. IEI is the input for the daisy-chain, prioritized interrupt enable signal. IEO is the corresponding output for the same signal.

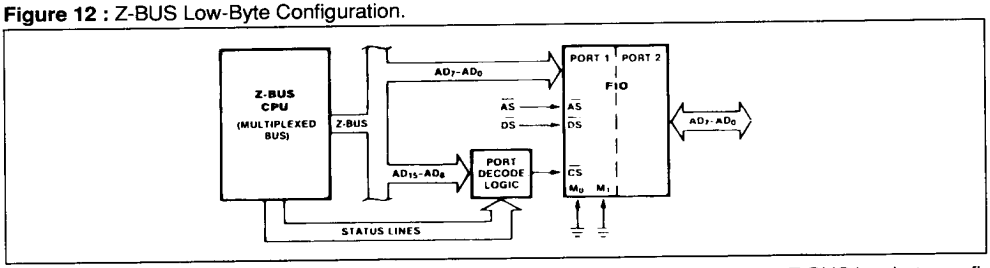
Figure 12 shows a typical use of the Z-BUS low-byte configuration. The M_0 and M_1 pins are shown tied to ground ; this specifies Z-BUS low-byte configuration.

Figure 11 : FIO Operational Summary.



INTERFACING SIGNALS (continued)

Figure 12 : Z-BUS Low-Byte Configuration.



Z-BUS Low Byte Interface Operation. Address/Data lines AD_0-AD_{15} leave the CPU. During the time $\overline{AS}$ is active, signifying that the address/data bus currently carries and address, lines AD_0 – AD_7 carry the 4-bit address of the internal register addressed within the FIO.

Once the Z-BUS low-byte configuration has been specified as shown in Table 1, the $\overline{CS}$ signal is Low, and the address information has been delivered (as shown by the removal of an active $\overline{AS}$ Signal), the $\overline{DS}$ signal goes Low and data appears on AD_0 – AD_7. When bit 0 is set in Control register 0, the FIO reset, all control registers cleared, and all outputs (except IEO on the port 1 side) floating at high impedance, the FIO is ready to program.

Register Addressing. In the Z-BUS low-byte configuration, the FIO allows two methods for register addressing under control of the Right Justify Address bit (bit D_1) of Control Register 0. When bit D_1 is reset (the default condition), address bus lines AD_1 – AD_4 are used for register addressing, and the other Address/Data lines are ignored. This convention assures compatibility with the Z8000 byte I/O addressing protocols. When D_1 is set, address bus lines AD_0-AD_3 are used for register addresses and the other Address/Data lines are ignored. Lines AD_8 – AD_{15} carry the 8-bit port address decoded by the port decode logic along with status to provide an active $\overline{CS}$ signal ; $\overline{AS}$ latches it into the FIO. Table 6 describes Z-BUS low-byte register addressing.

Table 6 : Register Addressing.

Z-BUS Low RJA = 0 / RJA = 1	AD_7-AD_5 / AD_7-AD_4	AD_4 / AD_3	AD_3 / AD_2	AD_2 / AD_1	AD_1 / AD_0	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

INTERFACING SIGNALS (continued)

Reset Operation. In the Z-BUS configurations, the FIO is hardware reset when both AS and DS are forced Low (normally an illegal condition) or it is software reset by writing a 1 to the reset bit in Control Register 0. When Port 1 is reset, Port 2 is also reset. If the Port 1 CPU reads the Port 1 registers during reset, they return all 0s except Control Register 0, which reads 01_H. All of Port 2's signal lines are floating in this state, and all inputs are ignored. If Port 2 is reset by itself, Port 1 is not reset. When the Port 2 CPU reads the Port 2 registers during reset, they return all 0s except Control Register 0, which reads 01_H.

Before data transactions can take place, the FIO must be taken out of the reset state by writing 00_H to Control Register 0. No other bits in this register can be programmed while clearing the reset bit. The Port 1 CPU then enables Port 2 by setting bit D_0 of the Port 1 Control Register 2. The Port 2 CPU can determine when it is enabled by reading its Control Register 0, which is read as a "floating" data bus if not enabled or as 01_H if enabled.

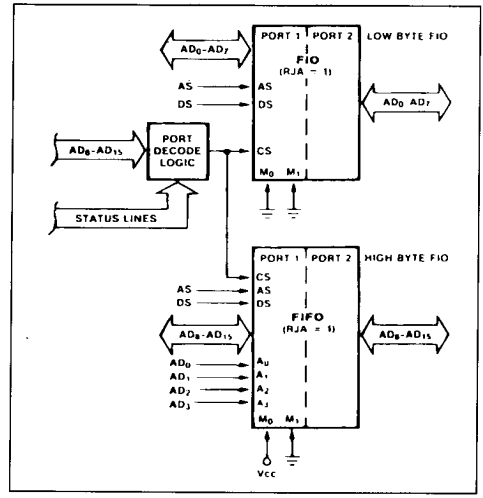
Interrupts. The Z-BUS low-byte configuration supports the prioritized daisy-chain interrupt protocols. INTACK, IEI, IEO, and INT control FIO interrupt operation. INT sends an interrupt signal to the CPU ; INTACK is the interrupt acknowledgement from the CPU. IEI is the input for the daisy-chain prioritized interrupt signal. IEO is the corresponding output for the same signal. When using two FIOs as a 16-bit Z-BUS interface, the low-byte FIO must handle external interrupt transactions, since the corresponding pins on the high-byte FIO are used for internal register addressing. This is not a handicap, since all INT conditions are the same for both FIOs, except for the pattern match condition.

Figure 13 shows a typical use of the Z-BUS low-byte and high-byte configurations with two FIOs, forming a 16-bit multiplexed bus interface. The single FIO in figure 12 runs parallel with another FIO in the Z-BUS high-byte configuration. The Z-BUS high-byte configuration is specified as shown in Table 1.

Z-BUS High Byte Interface Operation. As in the Z-BUS low-byte configuration, Address/Data lines $AD_0 - AD_{15}$ leave the Port 1 CPU. During an active AS signal, lines $AD_0 - AD_7$ carry the address of the internal register addressed within the FIO.

While AS is active, lines $AD_8 - AD_{15}$ carry the port address to the port decode logic, which issues an active CS signal. This signal is also supplied to the high-byte FIO. The RJA bit is not used on the high-byte FIO.

Figure 13 : Z-BUS High-Byte and Low-Byte Configuration.



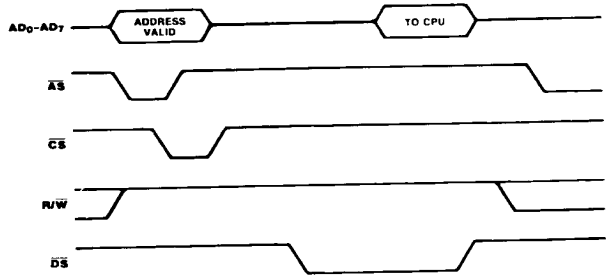
Once the low-byte and high-byte configurations have been specified as shown in Table 1, the CS signal is Low, and the address information has been delivered (as shown by the removal of an active AS signal), the DS signal goes Low, indicating that true data is on the Address/Data lines.

Register Addressing. Both FIOs require the same internal register address, but lines $AD_0 - AD_7$ go to the low-byte FIO only. For this reason, the signals on the four Address/Data pins used to specify FIO internal register addresses must be supplied to pins $A_0 - A_3$ of the high-byte FIO, as shown in figure 14. Since pins $AD_0 - AD_3$ are used for addressing, the low-byte FIO must handle external interrupt interfacing. Table 7 describes Z-BUS high byte register addressing.

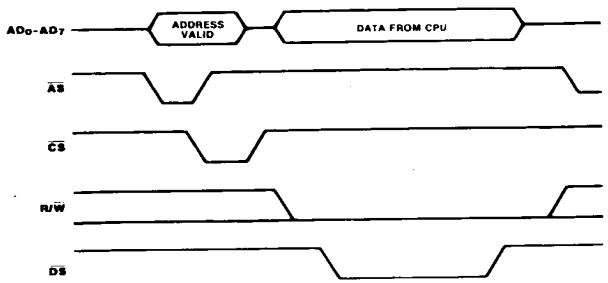
Reset Operation. In the Z-BUS configurations, the FIO is hardware reset when both AS and DS are forced Low (normally an illegal condition) or it is software reset by writing a 1 to the reset bit in Control Register 0. When Port 1 is reset, Port 2 is also reset. If the Port 1 CPU reads the Port 1 registers during reset, they return all 0s except Control Register 0, which reads 01_H. All of Port 2's signal lines are floating in this state, and all inputs are ignored. If Port 2 is reset by itself, Port 1 is not reset. When the Port 2 CPU reads the Port 2 registers during reset, they return all 0s except Control Register 0, which reads 01_H.

INTERFACING SIGNALS (continued)

Figure 14: Z-BUS Configuration Timing.



Z-BUS Read Cycle Timing



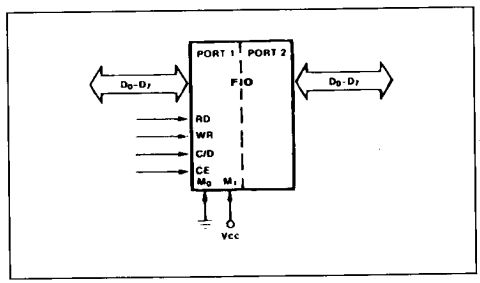
Z-BUS Write Cycle Timing

Before data transactions can take place, the FIO must be taken out of the reset state by writing 00_H to Control Register 0. No other bits in this register can be programmed while clearing the reset bit. The Port 1 CPU then enables Port 2 by setting bit D_0 of the Port 1 Control Register 2. The Port 2 CPU can determine when it is enabled by reading its Control Register 0, which is read as a "floating" data bus if not enabled or as 01_H if enabled.

Z-BUS High - and Low-Byte Programming. The FIOs, like other Z-BUS-compatible peripheral devices, accept byte-serial programming. The CPU must replicate the programming byte sent to $AD_0 - AD_7$ on the $AD_8 - AD_{15}$ address lines, insuring that the same programming byte goes to the high-byte FIO as to the low-byte FIO. The first programming byte is typically sent to Control Register 0. The FIO is then ready to program. Once programming is complete, 16-bit data transfers can take place, with the low-byte FIO transferring the byte on lines $AD_0 - AD_7$, and the high-byte FIO transferring the byte on lines $AD_8 - AD_{15}$.

Non-Z-BUS (nonmultiplexed) **Microprocessor.** The non-Z-BUS (nonmultiplexed) microprocessor configuration interfaces the FIO to a variety of microprocessors that use separate data and address uses, such as the Z80. Figure 15 shows a typical non-Z-BUS configuration. Figure 16 shows non-Z-BUS configuration timing information.

Figure 15 : Non-Z-BUS (nonmultiplexed) Micrpprocessor Configuration.



INTERFACING SIGNALS (continued)

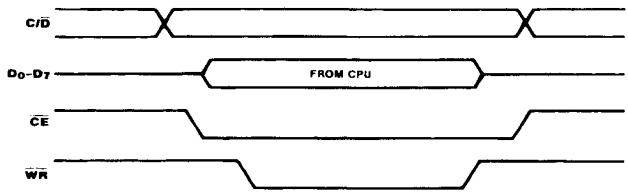
Table 7 : Register Addressing.

Z-BUS High Byte	A_3	A_2	A_1	A_0
Description				
Control Register 0	0	0	0	0
Control Register 1	0	0	0	1
Interrupt Status Register 0	0	0	1	0
Interrupt Status Register 1	0	0	1	1
Interrupt Status Register 2	0	1	0	0
Interrupt Status Register 3	0	1	0	1
Interrupt Vector Register	0	1	1	0
Byte Count Register	0	1	1	1
Byte Count Comparison Register	1	0	0	0
Control Register 2*	1	0	0	1
Control Register 3	1	0	1	0
Message Out Register	1	0	1	1
Message in Register	1	1	0	0
Pattern Match Register	1	1	0	1
Pattern Mask Register	1	1	1	0
Data Buffer Register	1	1	1	1

* Register is only on Port 1 side

Figure 16 : Non-Z-BUS (nonmultiplexed) Microprocessor Configuration Timing.

Non-Z-BUS Read Cycle Timing.



Non-Z-BUS Write Cycle Timing

INTERFACING SIGNALS (continued)

Signals. In the non-Z-BUS (nonmultiplexed) microprocessor configurations, the following signals control data transactions : Request/Wait, DMA Acknowledge, Read, Write, Chip Enable, Control/Data, Interrupt Acknowledge, Interrupt Enable Input, Interrupt Enable Output, and Interrupt Request.

$\overline{\text{REQ}}$/$\overline{\text{WAIT}}$ *(Request/Wait, output, 3-state, active Low).* The request signal is sent to a DMA device to begin a DMA transfer, and the Wait (open drain output) signal tells the CPU that the FIO buffer is full (if data is output from the CPU to the FIO) or that the FIO buffer is empty (if data is input to the CPU from the FIO). Bit D_1 of Control Register 1 selects the Request or Wait signals.

$\overline{\text{DACK}}$ *(DMA Acknowledge, input, active Low).* This signal is received by the FIO from a DMA controller. When Low, it forces the next read or write to the FIO to go through the Data Buffer register, regardless of the C/$\overline{\text{D}}$ conditions.

$\overline{\text{RD}}$ *(Read, input, active Low).* The Read signal tells the FIO that the CPU wants to read data from it.

$\overline{\text{WR}}$ *(Write, input, active Low).* The Write signal tells the FIO that the CPU wants to write data to it.

$\overline{\text{CE}}$ *(Chip Enable, input, active Low).* The Chip Enable signal selects the FIO when the CPU needs to carry out a data transaction involving it. The $\overline{\text{CE}}$ signal is usually generated from CPU address and status signals by external logic.

C/$\overline{\text{D}}$ *(Control/Data, input).* The C/D signal is used to distinguish between control and data bytes.

$\overline{\text{INTACK}}$ *(Interrupt Acknowledge, input, active Low),* IEI *(Interrupt Enable In, input, active High),* IEO *(Interrupt Enable Out, output, active High),* $\overline{\text{INT}}$ *(Interrupt Request, output open drain, active Low).* $\overline{\text{INTACK}}$, IEI, IEO, and $\overline{\text{INT}}$ control FIO interrupt operation. $\overline{\text{INT}}$ requests an interrupt from the CPU when Low, $\overline{\text{INTACK}}$ is the interrupt acknowledgement from the CPU. IEI is the input for the daisy-chain, prioritized interrupt enable signal. IEO is the corresponding output for the same signal.

Register Addressing. In the nonmultiplexed configurations, the FIO connects to the system data bus via $D_0 - D_7$, but not to the address bus. The CPU must provide address information on the data bus, and the FIO must distinguish between address and data bytes. The Control/Data (C/$\overline{\text{D}}$) pin does this with the help of the following programming conventions.

Figure 17 shows the register access state diagram. Before programming either port of the FIO, it should be reset, clearing all control registers. Getting out of the reset state is described in the next section.

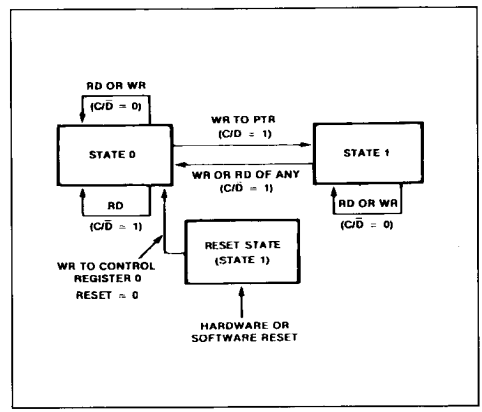
To move data bytes to or from the FIFO buffer, simply lower C/$\overline{\text{D}}$ to 0 (C/$\overline{\text{D}}$ is typically tied to an address pin) and read or write directly into the Data Buffer register. To read or write to all registers (including the Data Buffer register) is a two step operation. First, write the address (C/$\overline{\text{D}}$ = 1) of the register to be accessed into the Pointer Register (state 0 to state 1). The Pointer Register (4 bits) holds the register address. Second, read or write (C/$\overline{\text{D}}$ = 1) to the register addressed previously (state 1 to state 0).

The Pointer Register is not modified or cleared when going from state 1 to state 0. The Pointer Register is only changed by a write (C/$\overline{\text{D}}$ = 1) while in state 0. Consequently, status monitoring can be done while in state 0 by doing a read with C/$\overline{\text{D}}$ = 1.

For example, if the last register address was Control Register 2, then by doing a read with C/$\overline{\text{D}}$ = 1 (stay in state 0) the contents of Control Register 2 will again be read out. This may be useful for doing polling operations.

In the non-Z-BUS configuration, the RJA bit (D_1 of Control Register 0) is set. This specifies that bits 0-3 carry the address of the FIO internal registers and supports such CPUs as the Z80. Table 8 describes the Non-Z-BUS register addressing.

Figure 17 : Register Access State Diagram.



INTERFACING SIGNALS (continued)

Table 8 : Register Addressing.

Non-Z-BUS	D_7-D_4	D_3	D_2	D_1	D_0
Description					
Control Register 0	x	0	0	0	0
Control Register 1	x	0	0	0	1
Interrupt Status Register 0	x	0	0	1	0
Interrupt Status Register 1	x	0	0	1	1
Interrupt Status Register 2	x	0	1	0	0
Interrupt Status Register 3	x	0	1	0	1
Interrupt Vector Register	x	0	1	1	0
Byte Count Register	x	0	1	1	1
Byte Count Comparison Register	x	1	0	0	0
Control Register 2*	x	1	0	0	1
Control Register 3	x	1	0	1	0
Message Out Register	x	1	0	1	1
Message in Register	x	1	1	0	0
Pattern Match Register	x	1	1	0	1
Pattern Mask Register	x	1	1	1	0
Data Buffer Register	x	1	1	1	1

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

Reset Operation. In non-Z-BUS configuration, the FIO is hardware reset by forcing both $\overline{RD}$ and $\overline{WR}$ Low (normally an illegal condition) or by writing a 1 to the reset bit in Control Register 0. After reset is asserted, the only register that can be read from or written to is Control Register 0 (Control Register 0 will read a 01_H). If C/$\overline{D}$ is 1, the next byte writes into Control Register 0. When in the reset state, a write should not be done when C/$\overline{D}$ is 0. The reset state is exited by writing 00_H when C/$\overline{D}$ is 1. When Port 1 is reset, Port 2 is also reset. All of Port 2's signal lines are floating in this state and all inputs are ignored. If the Port 1 or Port 2 CPU (if enabled) reads any of its registers during reset, they return to 01_H (Control Register 0).

Software Reset. To software reset the FIO, it must be in state 1 and the Pointer register must point to Control Register 0. Table 9 outlines a method for resetting the FIO. The left column explains the software procedures used if the port was already reset. The right column shows the procedures used if the port was not reset. This procedure resets the FIO and then removes the reset making the FIO ready for programming. The table assumes that C/$\overline{D}$ is 1.

In Reset. If the port is already reset, reading it will have no effect and will return 01_H, since the reset bit is set. The Write 00_H command writes 00 into Control Register 0, since writes go to that register only during reset. This write takes the port out of the reset state and into state 0. The Write 01_H loads the Pointer Register with Control Register 1's address. This command places the port in state 1. The next write, Write 00_H, puts 00_H into Control Register 1. This has no effect, since the register was cleared during reset.

Not Reset. If the port is not reset, and hence in an unknown state (state 1 or state 0), a read of any kind (when C/$\overline{D}$ is 1) puts the port into state 0. The Write 00_H loads the Pointer register to Control Register 0's address and puts the port into state 1. The Write 01_H command sets the reset bit in Control Register 0 and puts the port into the reset state. The next write (Write 00_H) goes into Control Register 0, which removes the reset condition. At this point, the FIO registers can be programmed.

2-Wire Handshake I/O. The 2-wire handshake I/O configuration (Port 2 side only) is used to interface a CPU (multiplexed or not) to a single I/O device. The action of the FIO must be acknowledged by the other half of the handshake before the next transaction can take place. Figure 18 shows a typical 2-wire handshake configuration.

INTERFACING SIGNALS (continued)

Table 9 : Non-Z-BUS Software Reset Routines.

In Reset	Software	Not Reset
Reset State (state 1)	Read (any register)	State 0 or State 1
Reset State (state 1) (read = 01)	Write 00	State 0 (read = x)
Removes Reset (state 0)	Write 01	Point to Control Register 0 (state 1)
Point to Control Register 1 (state 1)	Write 00	Resets Port Reset State (state 1)
Write 00 to Control Register 1 (state 0)		Removes Reset (state 0)

Note : All Reads and Writes with C/$\overline{D}$ = 1.

While Reset is not a signal issued or received by the FIO, in CPU-to-I/O transactions the I/O port is reset by the CPU Port 1. When Port 1 is reset, all of Port 2's signal lines are floating in this state, and all inputs are ignored. Before data transactions can take place, the Port 1 CPU must enable Port 2 by setting bit D_0 of the Port 1 Control register 2.

The only register that is accessible on the Port 2 side, in this mode, is the Data Buffer register. All the other control registers are inaccessible.

Signals. In the 2-wire handshake I/O configuration, the FIO port uses Ready For Data/Data Available, Acknowledge Input, Clear, Empty, Full, Data Direction, Output Enable, two output lines, and an input line.

RFD/$\overline{DAV}$ *(Ready For Data/Data Available, output).* The Ready For Data signal (active High) is issued from the port and alerts external devices that it is ready to receive data. Data Available (active Low) indicates that the port has data to transmit.

$\overline{ACKIN}$ *(Acknowledge Input, input, active Low).* The Acknowledge Input signal, received by the FIO, notifies the CPU that transmitted data has been accepted by the external device or that data is valid on the bus.

$\overline{CLEAR}$ *(input or output, active Low),* EMPTY *(output/input, open drain, active High),* FULL *(output/input, open drain, active High).* These signals pertain to the condition of the FIO RAM buffer when the FIO performs I/O data transactions. Clear flushes all data from RAM ; when active (High), Empty and Full indicate that the FIFO buffer is empty or full during CPU-to-I/O transactions. The Buffer Full and Buffer Empty bits (D_0 and D_4 of Interrupt Status Register 3, respectively) are set when these pins are High.

DATA DIR *(Data Direction, input or output, active High).* The Data Direction signal allows the CPU or the I/O device to change the direction of data transactions. The control of data direction is programmed by the Port 1 CPU, via the Control register. When the Data Direction pin (pin 34) is Low, Port 2 is in output handshake configuration. Similarly, when the Data Direction pin is High, Port 2 is in input handshake configuration.

$\overline{OE}$ *(Output Enable, input, active Low).* When Output Enable is Low, data is propagated into Port 2 data lines. When OE is High, the data lines are at high impedance.

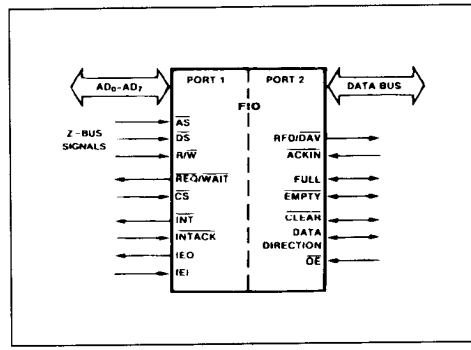
IN_0, IN_2, and OUT_3 are three signal lines that can be used as simple I/O bits.

Once the configuration is specified, two bits in the Port 1 Control Register 2 control the operation of Port 2. If D_0 is set, Port 2 is enabled. If D_1 is set, the Port 2 handshake signals are enabled.

2-Wire Handshake Operation. The FIO's 2-wire handshake is an interlocked handshake in which each action is acknowledged by the other device. This handshake protocol is compatible with the Z8, CIO, FIFO, UPC, and other commercially available parts. Figure 19 gives 2-wire handshake timing information. For both input and output handshake $\overline{ACKIN}$ must be High when the handshake is enabled (D_1 of Control Register 2 is set).

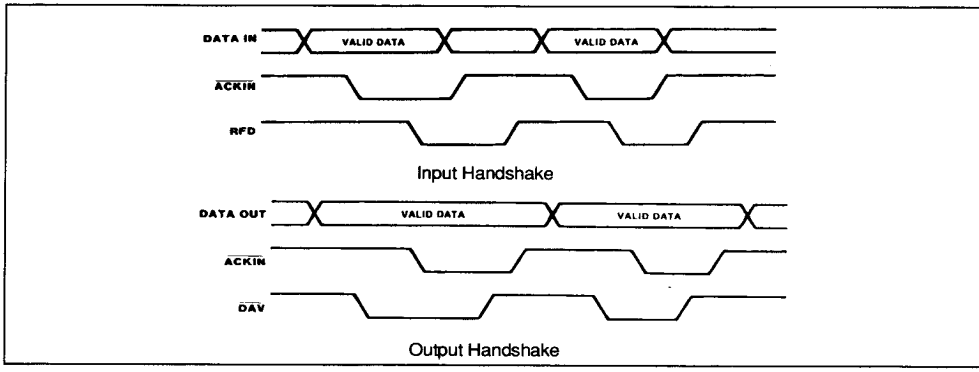
Output Handshake. For output handshake (Data Direction = 0) the $\overline{DAV}$ signal will go Low, indicating that data is available. This can occur only when $\overline{ACKIN}$ is High and there is data in the Data Buffer register. The $\overline{DAV}$ signal will stay Low until the $\overline{ACKIN}$ signal goes Low, indicating that the data has been accepted. The FIO's response to $\overline{ACKIN}$ Low is to bring the $\overline{DAV}$ signal High. At some later time the receiving device will bring $\overline{ACKIN}$ back High, in-

Figure 18 : Wire Handshake I/O Configuration.



INTERFACING SIGNALS (continued)

Figure 19 : 2-Wire Handshake I/O Configuration Timing.



dicating that it is ready for the next data byte. This process continues until the FIFO buffer is empty, the DAV will go High and remain so.

The Enable Handshake bit (D_1 of Control Register 2), when Low, forces the DAV signal High and internally forces ACKIN High. Care should be taken when disabling the handshake operation if the ACKIN signal is Low. Putting this bit Low at the wrong time may violate the handshake interlock and cause improper operation. A known time when this bit can go Low when the FIFO buffer is empty.

Input Handshake. For input handshake (Data Direction = 1), the RFD signal starts out High, indicating that the FIFO buffer is ready for incoming data transfers. Next ACKIN will go Low indicating that data is available on the data bus. The FIO responds by bringing RFD Low, signaling the acceptance of the data. Some time after RFD is Low, ACKIN will again go High. When the FIFO buffer is again ready for a new byte of data, it puts RFD High. This process continues until the FIFO buffer is full, at which time the RFD will go Low and remain so.

The Enable Handshake bit, when Low, forces RFD High and internally forces ACKIN High. Care should be taken when disabling the handshake operation if the ACKIN signal is Low. Putting this bit Low at the wrong time may violate the handshake interlock and cause improper operation. A known time when this bit can go Low when the FIFO buffer is empty.

In CPU-to-I/O transactions, exercise caution when changing data direction and clearing the FIFO buffer. Since DATA DIRECTION and CLEAR are programmed in the same register, they both can be changed during a single write to Control Register 3. Putting the Clear bit Low and changing the Data Direction bit is okay. Putting the Clear bit High and changing the Data Direction bit may cause improper operation. The Data Direction bit should only be changed when the Clear bit is Low or going Low.

The input and output lines, IN_0, OUT_1, and OUT_3 are, respectively, an input to bit D_0 of Port 2 Control Register 3 and outputs from bits D_1 and D_3. D_0 is a read-only bit. These three signal lines are used as simple I/O bits.

Full and Empty Operation. Both the Full and Empty pins are used as bidirectional signals. (See figures 20 and 21). As open drain output lines they can be wire-ANDed with other FIFOs or FIOs to give system status. As inputs, Full and Empty are used to set their respective IPs and also to show the status of the pins.

Figure 20 : Full Pin.

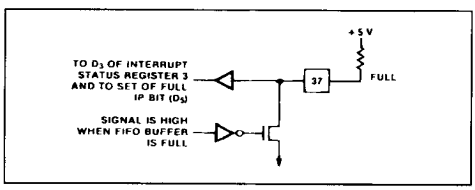
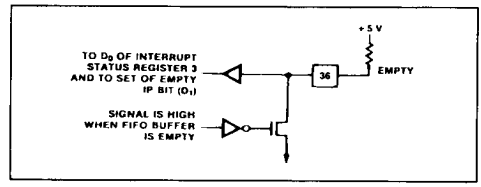


Figure 21 : Empty Pin.



INTERFACING SIGNALS (continued)

Figure 22 : 3-Wire Handshake I/O Configuration.

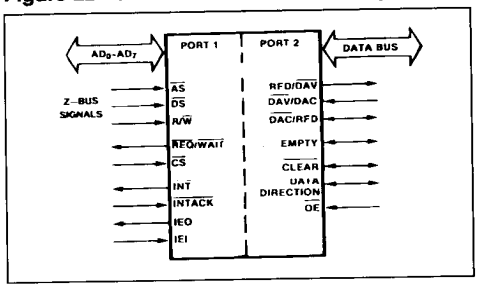


Table 10 : Full and Empty Status.

Number of Bytes in FIFO	Empty	Full
0	High	Low
1-127	Low	Low
128	Low	High

Table 10 shows the Full and Empty signals with only an external pull-up on each pin.

3-Wire Handshake I/O. The 3-wire hand-shake I/O configuration is designed to interface a CPU to many I/O ports simultaneously. Figure 22 shows a typical 3-wire handshake I/O configuration.

While Reset is not a signal issued or received by the FIO, in CPU-to-I/O transactions the I/O port is reset by the action of the CPU port. When Port 1 is reset, Port 2 is also reset. All of Port 2's signal lines are floating in this state, and all inputs are ignored. Before data transactions can take place, the Port 1 CPU must enable Port 2 by setting bit D_0 of the Port 1 Control register 2.

Signals. In the 3-wire handshake I/O configuration, the FIO port uses the following signals : Ready For Data/Data Available, Data Available/Data Accepted, Data Accepted/Ready For Data, Clear, Empty, Data Direction, Output Enable, two output lines, and an input line.

RFD/$\overline{\text{DAV}}$. *(Ready For Data/Data Available, output).* When Port 2 is configured as an input handshake, the Ready For Data signal is issued from the port and, on its rising edge, alerts external devices that it is ready to receive data. When configured as a data output handshake, the falling edge of the Data Available signal indicates tha the port has data to transmit.

DAV/DAC *(Data Available/Data Accepted, input).* In the input handshake configuration, Data Available indicates that data is ready for transmission to the FIO ; in the output handshake configuration, the rising edge of Data Accepted indicates that peripheral devices have received the data.

DAC/RFD *(Data Accepted/Ready For Data, output/input).* The DAC signal is an output when Port 2 is configured as an input handshake and the RFD signal is an input when Port 2 is configured as an output handshake.

$\overline{\text{CLEAR}}$ *(input or output, active Low)*, EMPTY *(output/input open drain, active High).* These two signals pertain to the condition of the FIO RAM buffer when the FIO performs I/O data transactions. $\overline{\text{CLEAR}}$ flushes all data from RAM ; EMPTY indicates, when active (High), that the FIFO buffer is empty during CPU-to-I/O transactions. The buffer empty bit (D_0 of Interrupt Status Register 3) is set when pin 36 is High.

DATA DIR *(Data Direction, input or output, active High).* The Data Direction signal allows the CPU or the I/O device to change the direction of data transactions. The direction is controlled by the Port 1 CPU, via Control Register 3. When the Data direction pin (pin 34) is Low, Port 2 is defined as having an output (source) handshake. When the Data Direction pin is High, Port 2 is defined as having an input (acceptor) handshake.

$\overline{\text{OE}}$ *(Output Enable, input, active Low).* When Output Enable is active (Low), output signals are propagated into Port 2 data lines. When OE is inactive (High), the data lines are at high impedance.

IN_0, IN_2, and OUT_3 are three signal lines available in the I/O modes.

3-Wire Handshake Operation. This configuration is like the handshake protocol used in the IEEE-488 standard. Using this handshake, the lines of many I/O ports can be bused together with external open-drain output drivers to signal when all the ports have accepted data and all are ready for data. Since the data direction of Port 2 can be changed under software control, bidirectional transfers can be performed. When interfacing to a bus such as the IEEE-488, external drivers must be used to meet the IEEE-488 bus specifications. Figure 23 shows the 3-wire handshake I/O timing information.

Output Handshake. For output handshake (Data Direction = 0), the FIO uses one output $\overline{\text{DAV}}$ *(Data Valid)* and two inputs RFD *(Ready For Data)* and DAC *(Data Accepted)* Before any bytes are written into the FIO buffer, the handshake must be enabled *(set D_1 of Control Register 1).* The handshake begins when the RFD input goes High (with DAC Low) indicating that the external device (s) are ready for data. When there is data in the Data Buffer register,

INTERFACING SIGNALS (continued)

the FIO puts the $\overline{DAV}$ signal Low indicating that the data is valid on the bus. The external device (s) now indicate that they are not ready for data by putting RFD Low. When the device (s) have accepted the data, DAC goes High indicating that the data on the bus is no longer needed. The FIO responds by bringing $\overline{DAV}$ High and indicating that the data on the bus may not be valid.

This handshake operation continues until the external device (s) cannot accept any more data ; at this point RFD stays Low, or the FIFO buffer is empty, in which case $\overline{DAV}$ stays High.

When the Enable Handshake bit is Low the $\overline{DAV}$ output is forced High.

Input Handshake. For input handshake (Data Direction = 1) the FIO uses two outputs, RFD (Ready For Data) and DAC (Data Accepted), and one input, $\overline{DAV}$ (Data Valid). The handshake starts by the FIO having RFD High, indicating that it is ready to accept data (DAC is Low). The external device will indicate that data is valid on the bus by forcing $\overline{DAV}$ Low. The FIO responds by bringing RFD Low and then putting DAC High, indicating that it is not ready for data and the data has been accepted. Some time later $\overline{DAV}$ will go High, indicating that the data on the bus is no longer valid. The FIO responds to this condition by bringing DAC back Low. The handshake operation continues until the external device has no more data to send, in which case it leaves $\overline{DAV}$ High, or the FIFO buffer is full, in which case RFD stays Low.

When the Enable Handshake bit is Low, RFD is forced High and DAC is forced Low.

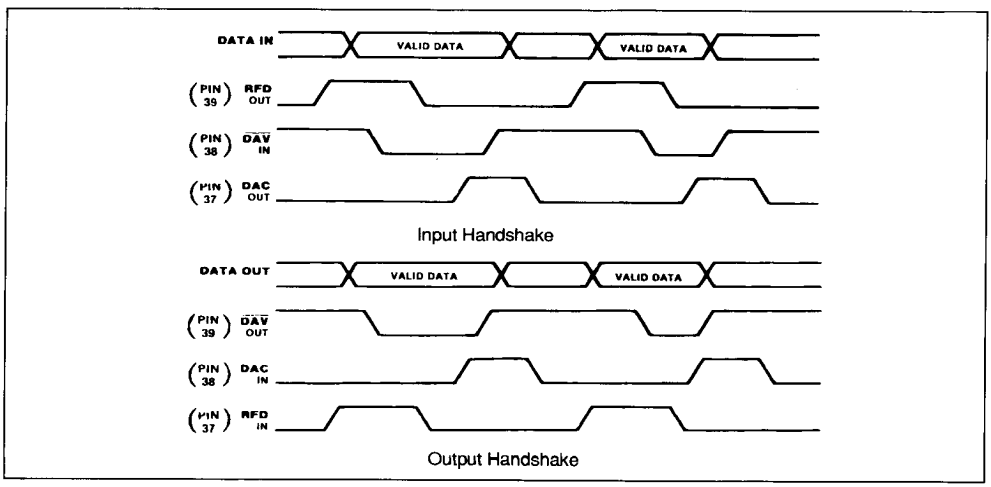
This configuration can be used only with Port 2 and must interface to the I/O device. Therefore, a CPU cannot directly program Port 2 via the Port 2 pins ; Port 2 operation is specified by the Port 1 internal registers. Bits D_2 and D_3 of Port 1 Control Register 0 define the Port 2 configuration. Resetting D_2 and setting D_3 enables Port 2 in the 3-wire handshake configuration.

Once the configuration is specified, two bits in the Port 1 Control Register 2 control the operation of Port 2. If D_0 is set, Port 2 is enabled. If D_1 is set, the Port 2 handshake signals are enabled.

In CPU-to-I/O transactions, care should be taken when changing data direction and clearing the FIFO buffer. Since DATA DIRECTION and CLEAR are programmed in the same register, they both can be changed during a single write to Control Register 3. Putting the Clear bit Low and changing the data direction bit is okay. Putting the Clear bit High and changing the data direction bit may cause improper operation. The data direction bit should only be changed when the Clear bit is Low or going Low.

The input and output lines (IN_0, OUT_1, and OUT_3) are, respectively, an input to bit D_0 of Port 2 Control register 3 and outputs from bits D_1 and D_3. D_0 is a read-only bit. These three signal lines are used for communication between FIOs and FIFOs in a networking environment.

Figure 23 : 3-Wire Handshake I/O Configuration Timing.



INTERFACING SIGNALS (continued)

WAIT OPERATION

Output WAIT. When data is output by the CPU (Data Direction = 0), the REQ/WAIT pin is active (Low) only when the FIFO buffer is full, the chip is selected, and the FIFO buffer is addressed. WAIT goes inactive when the FIFO buffer is not full.

Input WAIT. When data is input by the CPU (Data Direction = 1), the REQ/WAIT pin becomes active (Low) only when the FIFO buffer is empty, the chip is selected, and the FIFO buffer is addressed. WAIT goes inactive when the FIFO buffer is not empty.

The release of the FIO WAIT signal is asynchronous with respect to the CPU WAIT input. The circuit shown in figure 24 synchronizes the FIO WAIT signal with the expected Z8000 CPU WAIT input.

REQUEST (DMA) OPERATION

The FIO works particularly well with DMA devices in both Z-BUS and non-Z-BUS (nonmultiplexed) microprocessor modes. DMA operation can take place on both sides of the FIO simultaneously ; however, DMA operation cannot take place on the Port 2 side when the Port 2 side is in an I/O configuration.

The FIO supports two types of DMA operation, "flyby" and "sequential", both compatible with devices such as the Z80 DMA controller. In the flyby operation, data transfers occur every machine cycle on either on both sides of the FIO. The DMA controller issues a memory address with appropriate control signals and DACK or DMASTB ; the FIO receives or sends the byte strobed into the bus. The byte never passes through the DMA, it merely "flies by" between the memory and the FIO. In sequential operation, the DMA controller reads the byte from memory and then writes it to the FIO buffer or vice versa, addressing the Data Buffer register as a peripheral device. In this operation DACK or DMASTB should be tied to V_{CC}. "Flyby" is not a mode or a bit that is set in the FIO. To use flyby the Request operation must be enabled and the DMASTB/DACK pin is used.

The Request signal is inactive until the Clear bit is inactive (High). For example, when configuring the FIO for DMA operation, bits D_0 and D_1 in the controlling CPU's Control Register 1 can be set while the Clear bit is active without triggering a valid Request signal and initiating the transfer. Once the Clear bit goes inactive (Low), the DMA transfer into the FIO can begin, since the FIFO buffer is empty.

Z-BUS Flyby Operation. In figures 25 and 26, which show Z-BUS FIO-to-memory and memory-to-FIO transfers, the DMASTB signal strobes data to and from the FIO buffer. AS pulses Low to gate the memory address from the DMA into the Address/Data bus. Once AS goes High and DMASTB goes Low, data is read from or written into the FIO buffer directly without having to write the Data Buffer address during each read. DMASTB acts as a read or write signal ; the data direction bit specifies the direction of the transfer (the R/W pin is ignored). There is no effect if the DMA tries to read data when the FIO expects input, or vice versa.

The CS signal is not ignored by the FIO and therefore must be kept invalid (High) during the DMA transfer. This is normally acccomplished by default since the DMA device is addressing memory.

Non-Z-BUS Flyby Operation. In figure 27 and 28, showing non-Z-BUS FIO-to-memory and memory-to-FIO transfers, the DACK signal generated by the DMA device acknowledges the FIO's DMA request. After DACK goes true, the DMA device places the memory address on the memory address bus, and data is gated from memory into the data bus. DACK forces the next read or write to the FIO into the FIO buffer directly without having to write the Data Buffer address during each read (the C/D pin is ignored). The data direction bit specifies the direction of the transfer. There is no effect if the DMA tries to read data when the FIO expects input, or vice versa. The CE signal is not ignored by the FIO and therefore must be kept invalid (High) during the DMA transfer. This is normally accomplished by default since the DMA device is addressing memory.

Request Hysteresis Operation. The FIO has a feature that enhances DMA transactions in both flyby and sequential operation. Figures 29 and 30 are request hysteresis diagrams. These diagrams describe the use of the Byte Count Compare register during DMA transactions and demonstrate how DMA-to-FIO transfers can proceed. To engage this feature, set bit D_2 of Control Register 1.

REQUEST (DMA) OPERATION (continued)

When data is written into the FIO by the DMA controller, the FIO initiates the cycle by issuing an active Request signal. (Refer to figure 29). $\overline{REQ}$ stays active (Low). The FIO is ready to receive bytes from the bus. $\overline{REQ}$ remains active until the buffer is full. It then goes High and remains inactive while the CPU or I/O device reads the bytes from the FIFO buffer until the number of bytes in the buffer is equal to the value programmed into the Byte Count Compare register. $\overline{REQ}$ goes Low once more, and the sequence begins again.

Figure 24 : FIO-to-CPU WAIT Synchronization Circuit.

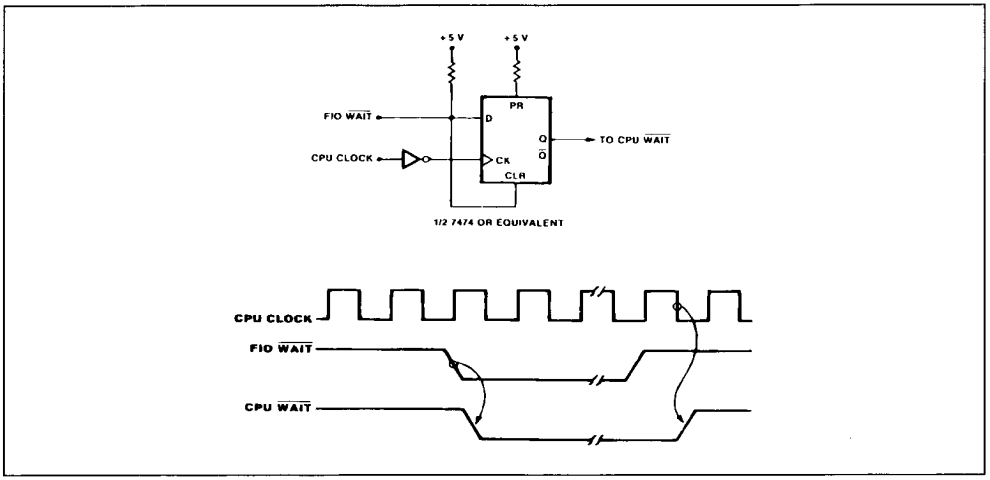
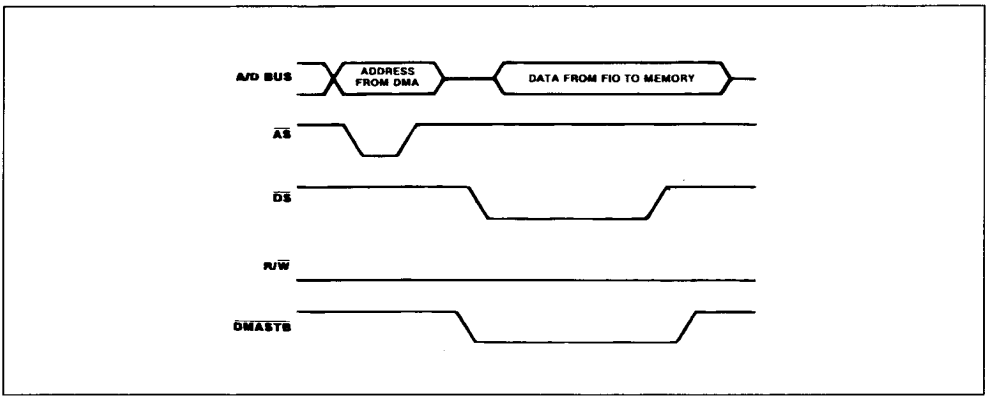


Figure 25 : Z-BUS FIO-to-Memory Data Transactions.



REQUEST (DMA) OPERATION (continued)

Figure 26 : Z-BUS Memory-to-FIO Data Transactions.

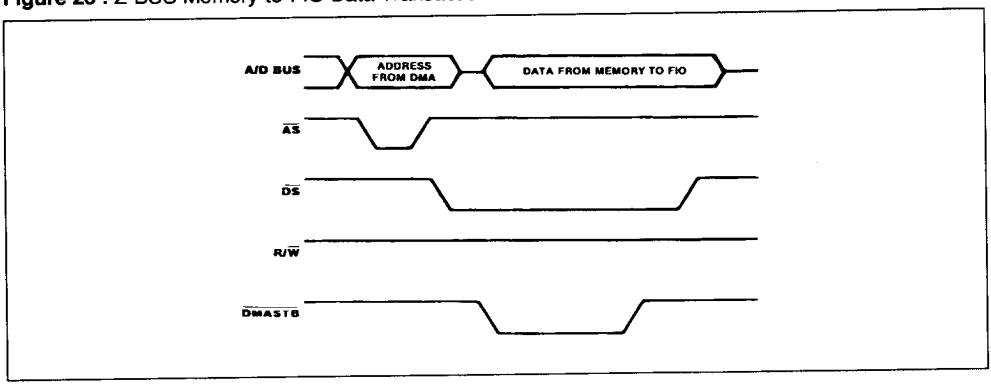
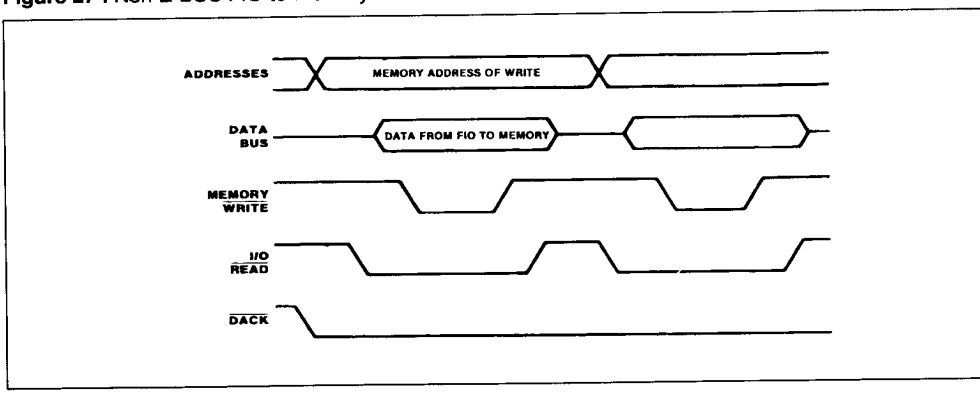


Figure 27 : Non-Z-BUS FIO-to-Memory Data Transactions.



REQUEST (DMA) OPERATION (continued)

Figure 28 : Non-Z-BUS Memory-to-FIO Data Transactions.

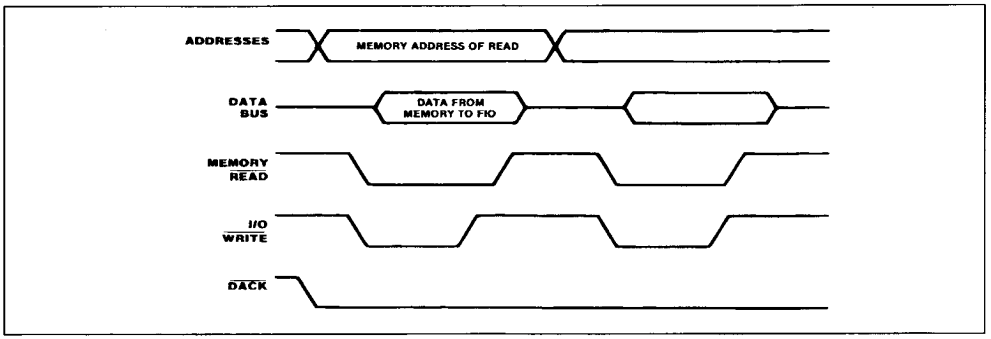
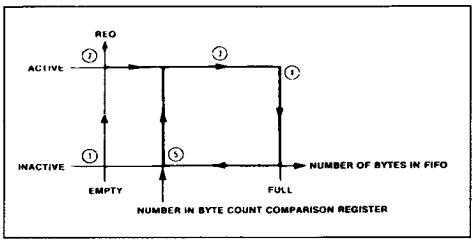
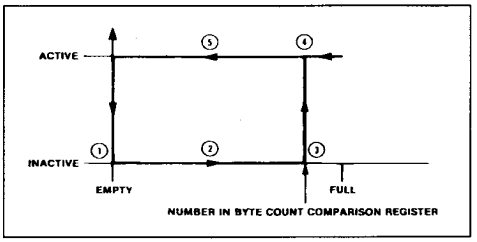


Figure 29 : Request Hysteresis Diagram : Write to FIO.



Notes :
1. FIO 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 30 : Request Hysteresis Diagram : Read from 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 FIO until it is empty.

When data is read from the FIO by the DMA controller, the FIO receives bytes from the CPU. (Refer to figure 30). The Request line is inactive until the number of bytes in the buffer is equal to the value programmed in the Byte Count Compare register. $\overline{REQ}$ then goes Low and remains Low until the DMA controller has emptied the buffer. Then $\overline{REQ}$ goes High and the sequence starts over again. If both ports are operating in CPU modes using DMA, this feature allows DMA operations to continue independently, triggering the Request signals from the two different Byte Count Compare register values.

If both ports are reading and writing at approximately the same rate, the buffer becomes a scratchpad memory for continuous DMA transfers. In this way, the length of the DMA block transfers can be much longer than the FIFO's buffer.

INTERRUPT OPERATION

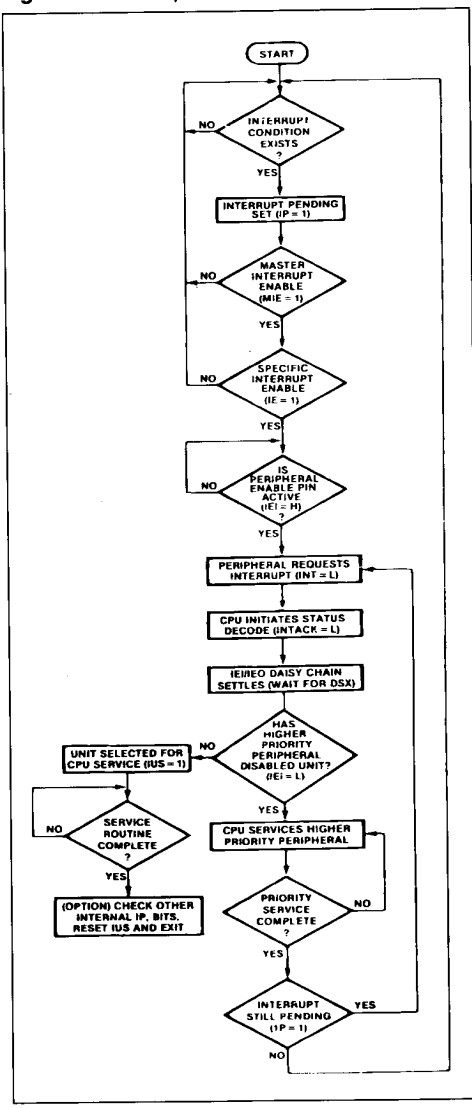
The FIO generates interrupt requests when programmed to do so and when one of seven specified conditions occurs. A complete interrupt cycle consists of an interrupt request followed by an interrupt acknowledge transaction. The CPU acknowledges interrupt request, according to the Z-BUS daisy-chain priority interrupt protocols. The FIO then places an address vector on the bus, providing the starting location of either an interrupt service routine or an index table into a group of such routines. The reason for the interrupt is encoded into the address vector, which becomes a pointer to specialized interrupt service routines.

Figure 31 is a flowchart of a peripheral interrupt sequence and explains the action within a single interrupting device. As the flowchart indicates, the peripheral inherently checks the status of all IP bits only if IEI is active. The IEI line is inactive only if the IEO of a higher-priority device is Low.

The interrupt Status registers specify whether or not interrupts will be generated during true interrupt conditions. If such interrupt conditions occur and the appropriate interrupt is enabled, the FIO communicates the interrupt request to the CPU.

Interrupt Priority. The seven prioritized conditions that can trigger the generation of an interrupt request follow in order of priority : Message Pending, Change in Data Direction, Pattern Match, Byte Count Compare, Overflow/Underflow Error Buffer Full, and Buffer Empty conditions. Each interrupt condition has three corresponding control bits in the Interrupt Status registers : Interrupt Enable bit (IE) enables interrupt generation, Interrupt Pending bit (IP) alerts the CPU that an interrupt is waiting for service, and Interrupt Under Service bit (IUS) shows if the interrupt is being serviced.

Figure 31 : Interrupt Flowchard.



INTERRUPT OPERATION (continued)

Figure 32 : Z-BUS Interrupt Acknowledge Timing.

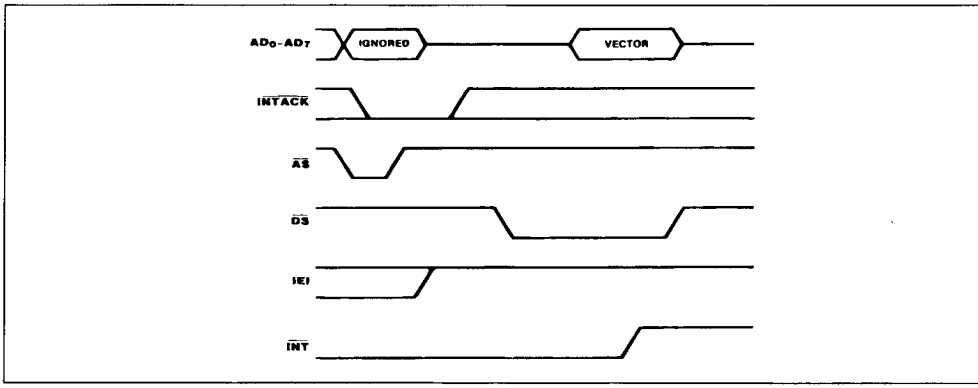
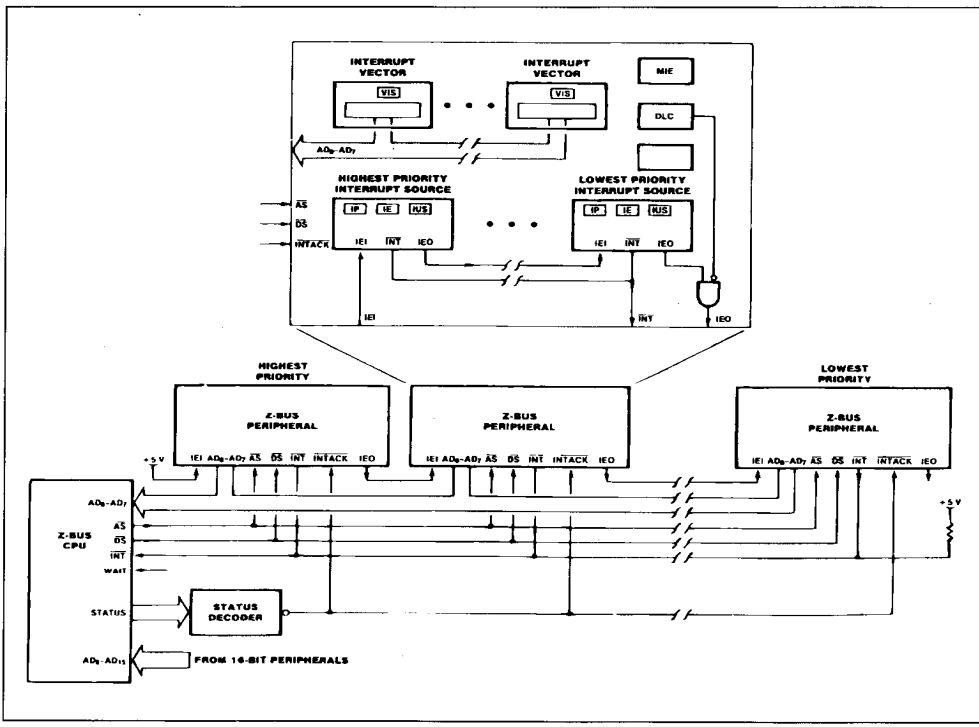


Figure 33 : Z-BUS Interrupt Arbitration.



INTERRUPT OPERATION (continued)

Setting and Clearing of Interrupt Control Bits. The setting and clearing of the IP, IUS, and IE bits is done under the command structure shown in figure 34. In this method the individual bits in a register are not set or cleared directly. This guards against the accidental clearing of an IP bit during a write and the possible loss of an interrupt request. Note the 3-bit code to set the message Interrupt Under Service bit shown beneath the Interrupt Status Register 0 drawing in figure 10. To set this bit, write 01000000 (40_H) to this register. This write does not directly set bit D_6, however. It sets bit D_7, shown in figure 10 as the message IUS bit. When Interrupt Status Register 0 is read following the write, it reads as 10000000 (80_H). These 3-bit groups, then, are described as codes that set or clear the Interrupt Status bits, rather than setting the bit in the corresponding register position. The same format is used with all four Interrupt Status registers.

Refer to the Interrupt Status registers shown in figure 10. The IUS and IP bits can be set two ways-when the condition itself occurs or by program control. These bits are cleared by program control. The IE bit can be set and cleared only by writing to them.

Z-BUS Interrupt Operation. Refer to figures 32 and 33. The FIO generates an interrupt request by lowering the $\overline{INT}$ line, only if such interrupt requests are enabled (IE is 1, MIE is 1), it has an Interrupt Pending (IP = 1), it does not have an Interrupt Under Service (IUS is 0), no higher-priority interrupt is being serviced (IEI is 1), and no interrupt acknowledge transaction is taking place (as indicated by $\overline{INTACK}$ High at the last rising edge of $\overline{AS}$). IEO is not pulled down by the FIO at this time ; IEO continues to follow IEI until an interrupt acknowledge transaction occurs.

Some time after $\overline{INT}$ has been pulled Low, the CPU initiates an interrupt acknowledge transaction. Between the rising edge of $\overline{AS}$ and the falling edge of $\overline{DS}$, the IEI/IEO daisy chain settles. Any Z-BUS peripheral with one of its Interrupts Pending (IP is 1) or one of its Interrupts Under Service (IUS is 1) holds its IEO line Low ; all others make IEO follow IEI.

When $\overline{DS}$ falls, only the highest priority interrupt source with a pending interrupt (IP is 1) has its IEI input High, its IE bit set to 1, and its IUS bit set to 0. This is the interrupt source being acknowledged, and at this point it sets its IUS bit to 1. If its NV bit is 0, the FIO identifies itself by placing its interrupt vector from the Interrupt Vector register on Address/Data lines $AD_0 - AD_7$. If NV is 1, the FIO's $AD_0 - AD_7$ lines remain floating, allowing external logic to supply a vector. (All Z-BUS interrupts require a vector to identify the requesting device).

If the FIO's VIS is 1, the vector also contains status information, coded into bits $D_1 - D_3$, which further describe the source of the interrupt within the FIO's logic. If VIS is 0, the vector held in the FIO is output without status included (base vector). The bit codes are given in Table 11. IPs are set by an AS following the event.

Figure 34 : IP. IUS and IE Command Code.

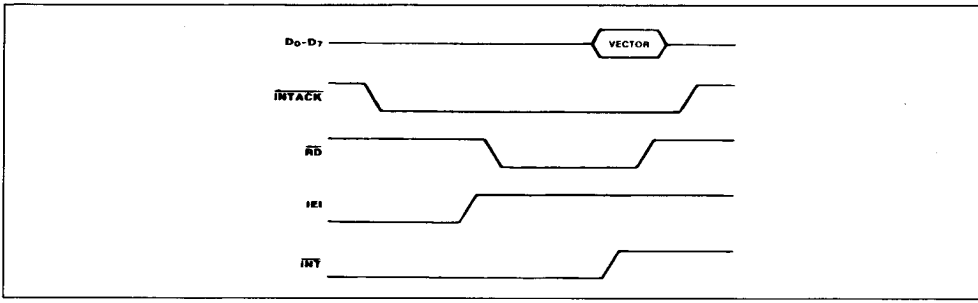
0	0	0	NULL CODE
0	0	1	CLEAR IP & IUS
0	1	0	SET IUS
0	1	1	CLEAR IUS
1	0	0	SET IP
1	0	0	CLEAR IP
1	1	0	SET IE
1	1	1	CLEAR IE

Table 11 : Interrupt Status Bit Codes.

Bits 3	2	1	Codes
1	1	1	Received Message in Message In Register
1	1	0	Change in Data Transaction Direction
1	0	1	Valid Pattern Match
1	0	0	Valid Byte Count Compare
0	1	1	Overflow or Underflow Error
0	1	0	Buffer Full
0	0	1	Buffer Empty
0	0	0	No Interrupts Pending

INTERRUPT OPERATION (continued)

Figure 35 : Non-Z-BUS Interrupt Acknowledge Cycle.



Non-Z-BUS Interrupt Operation. Figure 35 shows the complete non-Z-BUS interrupt acknowledge cycle timing.

When in non-Z-BUS configurations, the IP bit is not set while the port is in state 1. Any interrupt condition that occurs (such as a message received) does not set the corresponding IP bit. Thus, to minimize interrupt latency, the FIO should stay in state 0. The IP bit is set when the port returns to State 0.

The FIO generates an interrupt request (IP is 1 : Interrupt Pending) by lowering the INT line, only if such interrupt requests are enabled (IE is 1, MIE is 1), it has an interrupt pending (IP = 1), it does not have an Interrupt Under Service (IUS is 0), no higher-priority interrupt is being serviced (IEI is 1) and no interrupt acknowledge transaction is taking place. IEO is not pulled down by the FIO at this time ; IEO continues to follow IEI until an interrupt acknowledge transaction occurs.

Some time after $\overline{INT}$ has been pulled Low, the CPU initiates an interrupt acknowledge transaction. Between the falling edge of $\overline{INTACK}$ and the falling edge of $\overline{RD}$, the IEI/IEO daisy chain settles. Any peripheral with one of its Interrupts Pending (IP is 1) or one of its Interrupts Under Service (IUS is 1) holds its IEO line Low ; all others make IEO follow IEI.

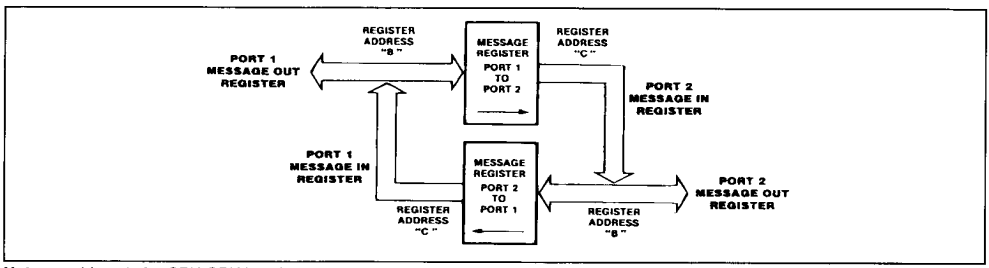
When RD falls, only the highest priority interrupt source with a pending interrupt (IP is 1) has its IEI input High, its IE bit set to 1, and its IUS bit set to 0. This is the interrupt source being acknowledged, and at this point it sets its IUS bit to 1. If its NV bit is 0, the FIO identifies itself by placing its interrupt vector from the Interrupt Vector register on data lines $D_0 - D_7$. If NV is 1, the FIO's $D_0 - D_7$ lines remain floating, allowing external logic to supply a vector.

If the FIO's VIS is 1, the vector also contains status information, coded into bits D_1-D_3, which further describe the source of the interrupt within the FIO's logic. If VIS is 0, the vector held in the FIO is output without status included. The bit codes are in Table 11.

MESSAGE REGISTER OPERATION

The two message registers provide a "mailbox" that is a means for two CPUs use to communicate with each other around the FIO's buffer. Each port's CPU is alerted when the opposite port's CPU sends a message.

Figure 36 : Message Register Operation.



Note : usable only for CPU/CPU interface.

MESSAGE REGISTER OPERATION

Figure 36 describes the message flow between two message registers. When Port 1's CPU writes to the Message Out register which is also Port 2's Message In register, Port 2's Message Interrupt Pending bit is set. If the Message Interrupt bit is enabled, Port 2's CPU is interrupted. Port 2's Message IP bit can be read from the Port 1 side (D_5 of Control Register 1) ; when the Port 2 CPU reads the data from its Message In register, the Port 2 IP is cleared. The Port 1 CPU can therefore tell when the message has been read and can now send another message or follow whatever protocol that it set up between the two CPUs. The same transfer can be made from Port 2's CPU to the Port 1 CPU.

ERROR OPERATION

In CPU configurations, the FIO detects overflow or underflow errors. Overflow is defined as performing a write to the Data Buffer register when the FIFO buffer is full, with the data direction bit at 0. This write is ignored, and the Byte Count register does not change value.

Underflow is defined as a read of the Data Buffer register when the FIFO buffer is empty, with the data direction bit set to 1. The contents of such a read are undefined and the Byte Count register does not change value.

Since the error conditions are mutually exclusive, one error Interrupt Pending bit (D2 of Interrupt Status Register 2) serves for both error conditions, with bit D_0 as 1 indicating an underflow error and bit D_4 as 1 indicating an overflow error. When either condition occurs, the error IP bit is set, regardless of the state of the error Interrupt Enable bit, and the appropriate Over/Underflow bit is set as well. The error IP bit is reset under program control, and the Over/Underflow bits are reset simultaneously.

If the Wait function is enabled, no error conditions should occur.

The error logic detects errors only for its own CPU. For example, if Port 1 is receiving data (Data Direction bit is 0), only overflow errors will be detected. If Port 2 has an underflow error, it will not set Port 1's error IP bit.

PATTERN MATCH OPERATION

The Pattern Match register contains a byte to compare with the byte in the data buffer. The Pattern Mask register contains a bit pattern used to mask bits used in the comparison.

Pattern Match Register. The Pattern Match register (figure 9) contains a byte for comparison with the byte in the Data Buffer register. As each byte in a data transaction passes through the Data Buffer register, it is compared with the value programmed in the Pattern Match register. Upon a true match, an interrupt can be generated if enabled in Interrupts Status register 1. One or more bits in the Pattern Match register can be masked by the value in the Pattern Mask register.

Pattern Mask Register. The Pattern Mask register (figure 9) contains a bit pattern used to force a true match between the pattern in the Pattern Match register and the pattern in the Data Buffer register. If a pattern mask bit is set, the corresponding bit in the Pattern Match register always matches with the corresponding bit in the Data Buffer register. All 1s in this register forces a pattern match.

BYTE COUNT COMPARE OPERATION

The Byte Count Compare register holds a byte that is continuously compared with the count in the Byte Count register. If the Byte Count Compare Interrupt Enable bit (D_6) is set in Interrupt Status register 2, an interrupt occurs upon a true match. The comparison operation is not affected by the Freeze Byte Count bit.

If the byte being programmed in the Byte Count Compare register equals the number of bytes that are currently in the FIFO, then the byte count IP bit is set. The range of valid values for the comparison is from 0 to 127 ($7F_H$).

BYTE COUNT REGISTER FREEZE OPERATION

The Byte Count register tracks the number of bytes read from and written to the FIO's FIFO buffer. Since the count changes with each read or write cycle, a provision is made for "freezing" the count so the CPU can read the register's value. When set, bit 6 in Control Register 1 freezes the current value in Byte Count register.

If there are reads or writes after the freeze bit is set, a read of the Byte Count register shows the frozen value and does not reflect the reads or writes that occur between the setting of the freeze bit and the read of the Byte Count register. The read clears the freeze bit. Until another read or write takes place, the frozen value in the Byte Count register is not updated with the current value.

For example, if the Byte Count register shows 10 transactions when the freeze bit is set, and there are five more transactions between the time the freeze bit is set and the time the Byte Count register is read, the read will return the frozen value, 10. If no further transactions (after the five) take place, another read

BYTE COUNT REGISTER FREEZE OPERATION (continued)

of the Byte Count register will still return 10 ; if a read or write takes place, the current value (10 + 5 ± any other transactions) is returned.

$\overline{\text{CLEAR}}$ AND DATA DIRECTION OPERATION

Care should be taken when changing the data direction and clearing the FIFO buffer. Changing the Data Direction pin/bit when the $\overline{\text{CLEAR}}$ pin/bit is High and the FIFO buffer is not empty will cause improper operation. When changing the Data Direction pin/bit the FIFO buffer must be empty. Putting the $\overline{\text{CLEAR}}$ pin/bit Low assures an empty FIFO buffer.

CPU-TO-CPU Operation. Clear Operation. Figure 37, showing $\overline{\text{CLEAR}}$ operation in CPU-to-CPU operation. When bit D_7 of Port 1 is 0, then Port 1 has control over the Clear Bit. When bit D_6 is 1, the FIFO buffer is not clear and is therefore ready to read or write data. If D_7 is then set to 1, and control of the clear function transfers to Port 2. It should be noted that if the Port 2 side is reset when it has control of the $\overline{\text{CLEAR}}$ bit, the $\overline{\text{CLEAR}}$ bit is also reset (0).

Data Direction Operation. Figure 38 shows the data direction operation for CPU-to-CPU operation. When bit D_5 is 0, Port 1 controls data direction. When D_5 is 1, Port 2 controls data direction. Bit D_4 of the appropriate CPU controls the actual transaction direction, relative to the controlling CPU. When D_4 is 0, data is output from the CPU, and when D_4 is 1, data is input to the CPU. An inverter added to the logic ensures an accurate operation for each side. It should be noted that if the Port 2 side is reset when it has control of the Data Direction bit, the Data Direction bit is also reset. Thus, Port 2's Data Direction bit is 0 and Port 1's Data Direction bit is 1.

CPU-TO-I/O Operation. Clear Operation. Figure 39 shows CLEAR operation in CPU-to-I/O operation. In these transactions, bit D_7 controls whether the FIFO buffer is cleared by resetting D_6 in Control Register 3 or by a system CLEAR signal input on pin 35. If D_7 is 0, Port 1's bit D_6 of Control Register 3 clears the FIO when it is 0. If D_7 is 1, the FIO clears when a system CLEAR signal (active Low) is applied to pin 35.

Data Direction Operation. Figure 40 shows the data direction logic. When bit D_5 is 0, Port 1's bit D_6 in Control Register 3 controls data direction. When D_5 is 1, data direction control is via a system signal received at pin 34. The data direction bit and pin 34 are always the same (no inversion) logic level.

Regardless of whether Port 1 or Port 2 has control of the data direction, Pin 34 determines Port 2 handshake. Pin 34 Low (0) defines the Output Handshake operation ; Pin 34 High (1) defines the Input Handshake operation.

Figure 37 : $\overline{\text{CLEAR}}$: CPU-to-CPU Operation.

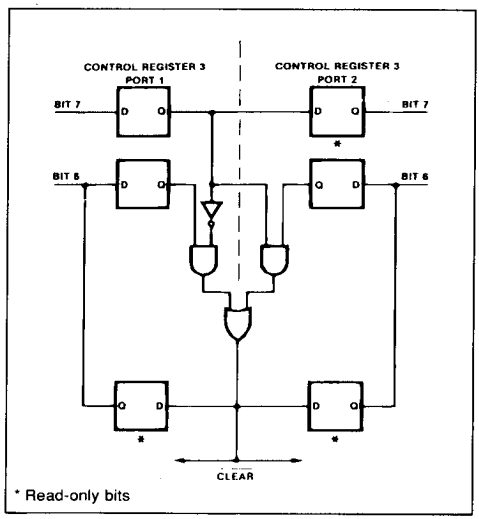


Figure 38 : Data Direction : CPU-to-CPU Operation.

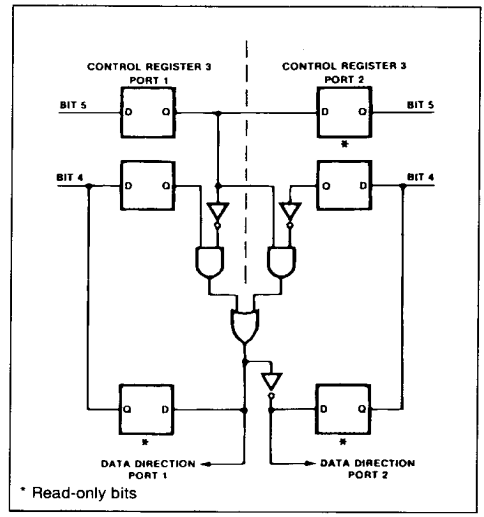


Figure 39 : $\overline{\text{CLEAR}}$: CPU-to-I/O Operation.

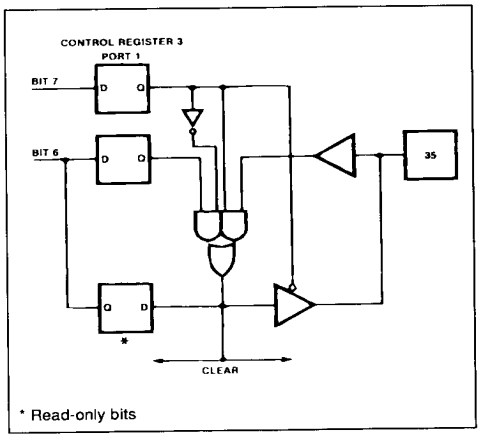
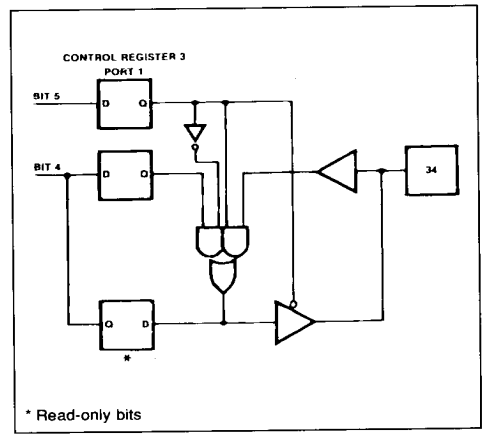


Figure 40 : Data Direction : CPU-to-I/O Operation.



INTERFACING THE FIO

The FIO interfaces to many kinds of devices. The Z-BUS high-byte and low-byte configurations accomodate such devices as the Z8000, Z8, and 8085, which require a multiplexed address/data bus. Devices such as the Z80, 8080, and 6800 use separate address and data buses which are supported by the FIO's non-Z-BUS (nonmultiplexed) microprocessor configuration. The 2-wire handshake I/O configuration is the most commonly used parallel I/O protocol, and the 3-wire handshake I/O configuration supports a multi-acceptor protocol.

This chapter introduces interfacing the FIO to other devices, shows how the FIO interfaces to various bus structures, and shows how buffer interfaces larger than 128 bytes can be constructed with FIOs and FIFOs.

INTERFACING CONSIDERATIONS

Z-BUS Interfacing. All Z-BUS memory or I/O transactions use the $\overline{\text{AS}}$ (Address Strobe) and $\overline{\text{DS}}$ (Data Strobe) signals. These signals define the information currently on the bus as address information or data. An active $\overline{\text{AS}}$ signal latches the $\overline{\text{CS}}$ (Chip Select) signal, which is decoded from the correct I/O address and enables the FIO. During an active $\overline{\text{DS}}$ signal, data is available on the bus. If R/$\overline{\text{W}}$ is High during $\overline{\text{DS}}$, data is read from I/O devices of from memory. If R/$\overline{\text{W}}$ is Low, data is written to I/O devices or to memory.

Z-BUS I/O transactions are asynchronous with the CPU clock. I/O transactions are similar to memory transactions, except that I/O transactions are longer. Memory transactions use three clock cycles for completion, but a single wait state is added automatically to I/O transaction timing, and additional wait-states can be inserted by forcing the $\overline{\text{WAIT}}$ line Low for as long as required.

Z8000 Interfacing. When interfacing to the Z8001 or Z8002 (see figure 4.1) in the Z-BUS low-byte configuration, the RJA bit (D_1 of Control Register 0) can be either 1 or 0. When RJA is 0 the FIO takes the address information on AD_1-AD_4. This configuration is compatible with using Z8000 byte I/O (INB, OUTB) instructions. In byte I/O, the AD_0 bit specifies which byte on the data bus is read or written to ($AD_0 = 0$ uses $AD_8 - AD_{15}$; $AD_0 = 1$ uses $AD_0 - D_7$). Typically, all byte I/O devices have odd addresses.

When RJA is 1 the FIO takes the address information on AD_0-AD_3. This configuration is compatible with using Z8000 word I/O instructions (IN, OUT). The word I/O instructions allow AD_0 to be used as an address bit to the FIO. This is useful when using a High and a Low byte FIO to do word transfers. This configuration is also useful for the user who wants consecutive I/O addresses and is not concerned with Z8000 register space.

The Z-BUS high-byte and low-byte configurations, as well as the non-Z-BUS microprocessor configurations, support the daisy-chain priority interrupt

INTERFACING CONSIDERATIONS (continued)

Figure 41 : Z8000 I/O Timing.

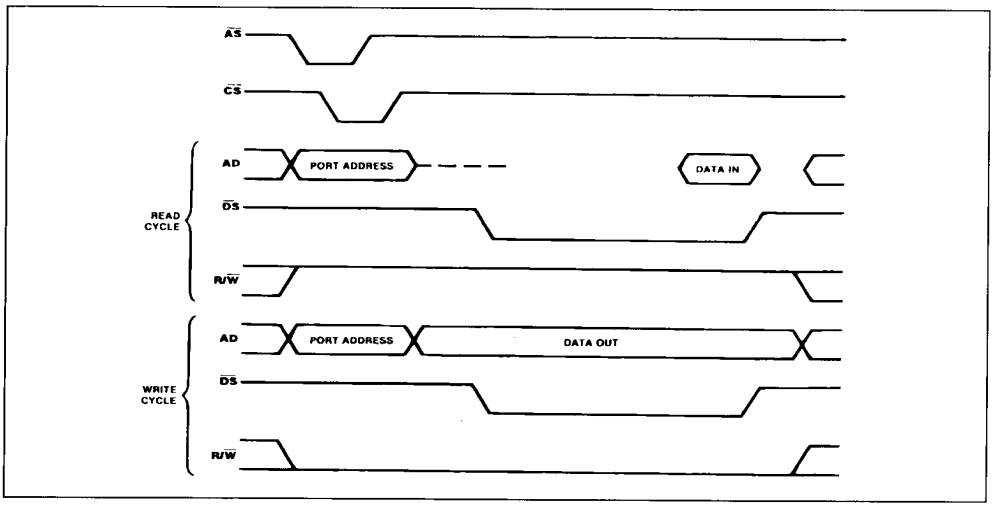
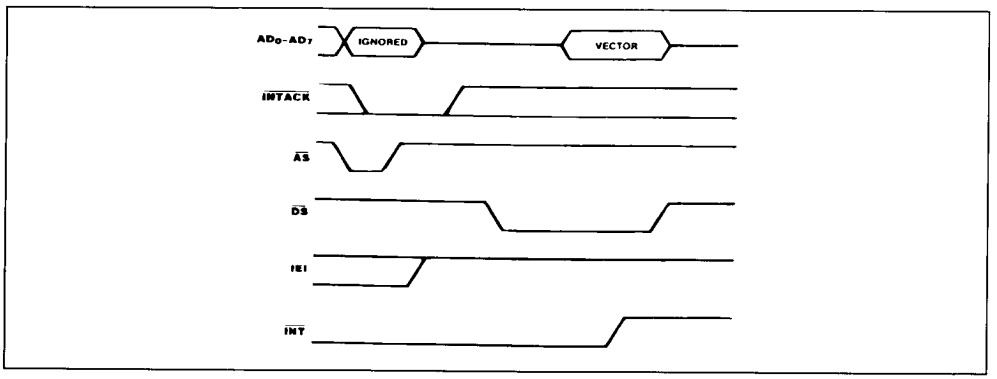


Figure 42 : Z8000 Interrupt Acknowledge Timing.



questing interrupt service to lock other, lower-priority devices out of the interrupt control bus, thus preventing interrupt generation by these devices. Higher-priority devices can still issue interrupt requests that override the original requestor's interrupt service routine.

Additional discrete logic is required to decode the CPU status lines to provide an interrupt acknowledge signal for the FIO. The Z8000 CPU adds five additional wait states (between $\overline{AS}$ rising and $\overline{DS}$ falling) to the interrupt transaction timing (figure 42) as daisy-chain settling time and to allow time for the FIO to place its interrupt vector into the bus.

Non-Z-BUS Interfacing. This interfacing configuration is used with CPUs having separate address and data lines. The FIO is connected only to the data bus of such CPUs, so a signal, C/$\overline{D}$, must distinguish between control bytes and data bytes. In the non-Z-BUS configuration, the addresses of the FIO internal registers must appear on data lines $D_0 - D_3$. Ac-

INTERFACING CONSIDERATIONS (continued)

cordingly, the RJA bit (D_1 of Control Register 0) is always set by the FIO.

In this configuration, the $\overline{CE}$ (Chip Enable) signal must be decoded from the I/O address of the FIO by external logic. Since $\overline{CE}$ is not latched, it must remain true during the entire I/O transaction.

Z80 Interface. I/O Request ($\overline{IORQ}$) and Memory Request ($\overline{MREQ}$) specify I/O or memory transactions ; $\overline{WR}$ and $\overline{RD}$ specify write or read operations. During I/O operation, the Z80 inserts a wait state into the timing to give the FIO time to decode its address and to activate its own $\overline{WAIT}$ line if additional time is needed. Figure 43 shows this timing.

The Z80 samples the $\overline{INT}$ line at the rising edge of the last clock cycle at the conclusion of every instruction. If an interrupt request is detected, the CPU issues an M1 signal without a corresponding $\overline{MREQ}$ signal. The Z80 adds two Wait states to allow the IEI/IEO daisy chain interrupt lines to settle. External logic must generate an $\overline{INTACK}$ signal from these signal conditions. Figure 44 shows various signals from the Z80 and the signals generated from these and sent to the FIO. Figure 45 shows the external logic used to interface the FIO and other 8500 series peripherals to the Z80 bus. When the generated $\overline{INTACK}$ and $\overline{RD}$ are both Low (active), the FIO places its interrupt vector on the bus, if enabled to do so.

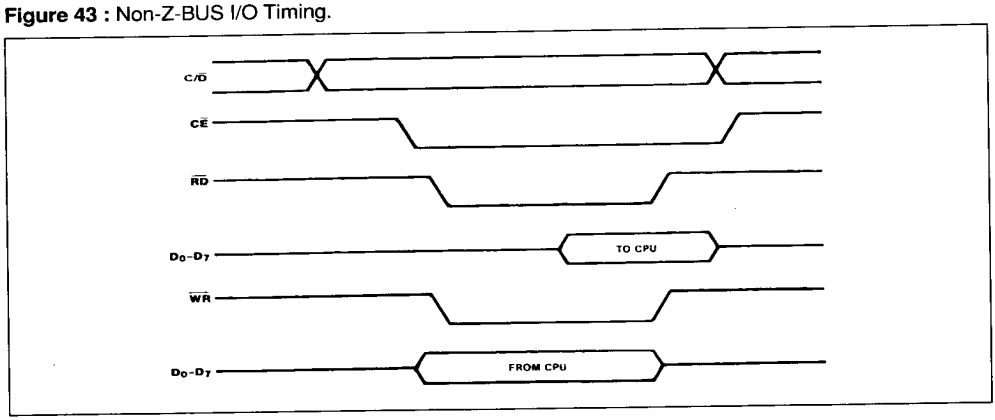
2-Wire (Interlocked) Handshake I/O Interfacing. The 2-wire handshake configuration is usually used when a single I/O device communicates with a CPU ; the 3-wire-handshake configuration (similar to the IEEE-488 standard interface) is used when a CPU is interfacing to a group of devices.

2-Wire Handshake Configuration. During a data transfer into the FIO, RFD/$\overline{DAV}$ is forced High, signaling to the peripheral device that the FIO is ready for data. The peripheral places the data on the bus and acknowledges the transaction with $\overline{ACKIN}$ Low, latching the data into the FIO Data Buffer register on the falling edge. RFD then goes Low until the Data Buffer register is emptied into the buffer memory, at which time RFD goes High (active) if $\overline{ACKIN}$ has gone High (inactive). The cycle repeats until the peripheral is through writing or until the buffer memory is full. When the buffer memory is full, RFD goes Low and remains so.

During a data transfer from the FIO, $\overline{DAV}$ goes active (Low) to signify that data is available from the FIO's Data Buffer register. If $\overline{ACKIN}$ is High, signifying that the previous read from the FIO is completed, the device latches the data from the FIO and forces $\overline{ACKIN}$ Low. The FIO forces $\overline{DAV}$ High when the Data Buffer register is read. $\overline{ACKIN}$ strobes the next byte from the buffer to the Data Buffer register, $\overline{DAV}$ goes Low, and the cycle repeats until the FIO buffer is empty when $\overline{DAV}$ goes High and remains so.

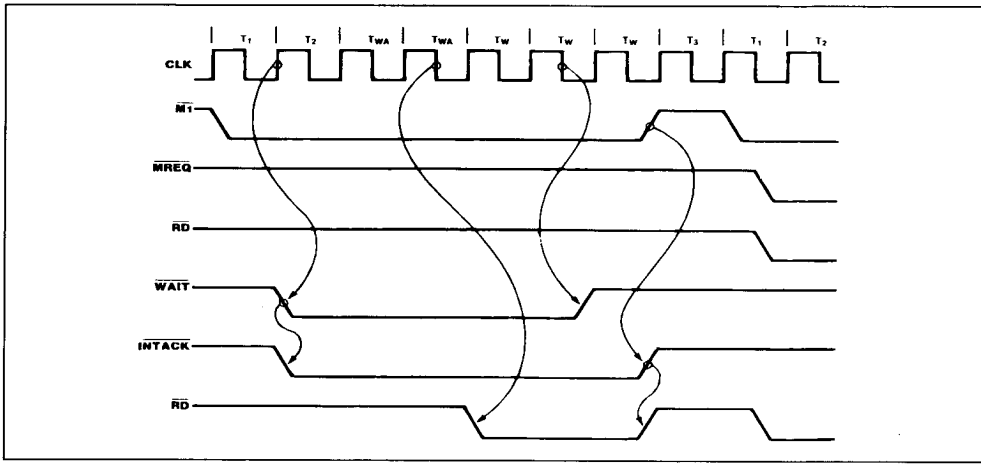
3-Wire Handshake Configuration. During a data transfer into the FIO, the FIO forces RFD High (active) to signal that it is ready for data. The $\overline{DAV}$ input on pin 38 goes Low (active) to strobe the incoming data from the bus lines into the Data Buffer register. The DAC input goes High on pin 37 to show the input device that the data has been accepted.

Figure 43 : Non-Z-BUS I/O Timing.



INTERFACING CONSIDERATIONS (continued)

Figure 44 : Z80 Interrupt Acknowledge Timing.

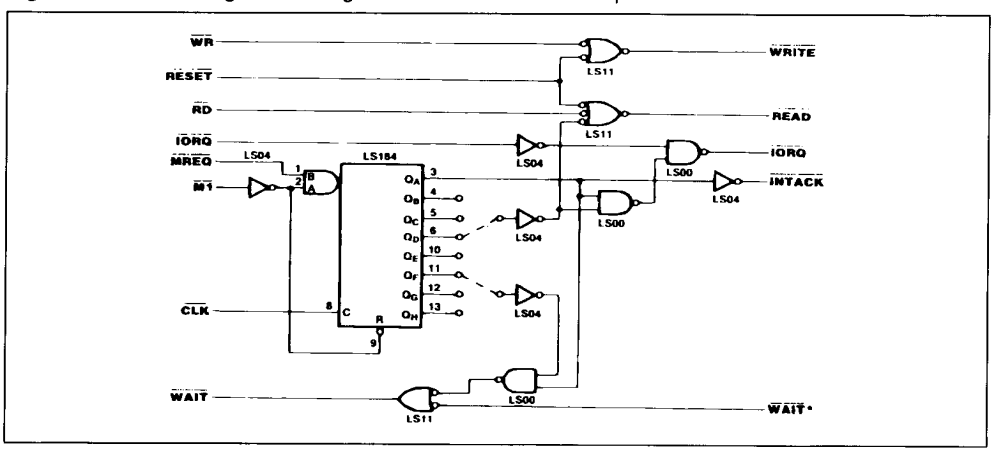


RFD on pin 39 remains Low until the data has been moved from the Data Buffer register to the buffer memory. When RFD goes High again, the cycle repeats until the FIO buffer is full. During a data transfer from the FIO, the RFD signal from the peripheral devices goes High to show they are ready for data. DAV on pin 39 goes Low (active) to show that there is valid data on the bus. The peripheral devices return DAC High (active) on pin 37, signifying that the data has been accepted. When the peripherals are ready for more data, RFD reappears High on pin 37 and the cycle begins again. Output continues until the FIO buffer is empty, which forces DAV High (inactive). DAV stays High until the buffer is reloaded from the CPU port. The peripherals can stop the transaction by holding RFD Low on pin 37.

FIO EXPANSION INTERFACING

FIOs can be combined to create 16-bit or wider interfaces and to create buffers larger than 128 bytes. The following text and illustrations show how FIOs

Figure 45 : External Logic Interfacing the Z80 to 8500 Series Peripherals.



INTERFACING CONSIDERATIONS (continued)

can be combined or used with the Z8060 FIFO buffer to create a buffer size of 512 bytes.

Buffer size can be a critical factor in an I/O-bound system. A larger buffer allows longer intervals between situations that require CPU intervention and can cut I/O overhead significantly. Using a device as powerful as the FIO, which conducts pattern matches and generates interrupts upon a variety of conditions, the CPU spends less time polling I/O devices. The FIO also conducts duplex DMA operation at system speeds, transferring a byte each cycle, since the buffer can be read from and written to simultaneously.

CPU-TO-CPU 256-Byte FIO Buffer Expansion. Figure 46 shows a configuration of FIOs that creates a 256-byte buffer. Port 2 of one FIO is interfaced to Port 2 of the other FIO, and both are programmmed in the 2-wire handshake configuration.

The Empty and Full pins are wire-ORed to communicate these conditions only when both FIOs reflect the same condition.

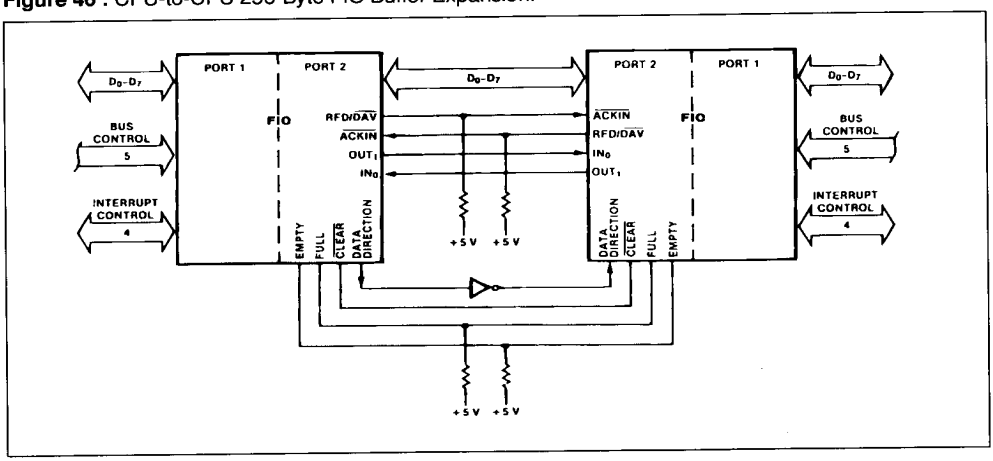
The message register feature does not operate in any of the expanded configurations, so some other provision must be made for communication between two CPUs controlling a multiple-FIO configuration. This can be done by using the IN_0 and OUT_1 signals (mapped to pins 33 and 32, respectively, in the 2-wire handshake configuration). In the 2-wire handshake configuration, the Port 1 CPU can set or reset bit D_1 of Port 1 Control Register 3. The state of this bit is reflected in bit D_0 and can be read by the OUT_1 line from the opposite FIO. Conversely, the opposite FIO can set its own bit D_1, simultaneously setting bit D_0, which can then be read by the first FIO.

The expansion process modifies the operation of the Byte Count register. When data is written to a port, it moves (bubbles) to the first bit position of the buffer opposite the one that is writing. For example, if the left CPU is writing 256 bytes to the FIOs shown in the figure 46, the right FIO buffer will be loaded with 128 bytes before any bytes are stored into the left FIO buffer.

If 100 bytes are then written into this configuration from the left CPU, the right CPU can read its Byte Count register and obtain an accurate read of 100 bytes. If the left FIO reads its Byte Count register, however, it will be empty. If another 100 bytes are then transferred from left to right, the right CPU will read the buffer of its FIO as full (128 bytes loaded) and the left FIO will show 200 minus 128 (or 72) bytes in its buffer. When 256 bytes are written to this configuration, the wire-ORed full line will reflect a buffer-full state by going High.

During a read from these filled buffers, each time a byte is read from the right FIO by its CPU, a byte from the left FIO trickles across to the right FIO. Therefore, the left FIO buffer is emptied before the right FIO begins its countdown to empty. When both buffers are empty, the wire-ORed empty line reflects a buffer-empty state by going High. The empty and full interrupts are modified as well. Empty interrupts

Figure 46 : CPU-to-CPU 256-Byte FIO Buffer Expansion.



FIO EXPANSION INTERFACING (continued)

signify that both buffers are empty and full interrupts signify that both buffers are full.

In this example, the left FIO has control of Data Direction. Its Data Direction pin is an output and the right FIO's Data Direction pin is an input. To ensure proper operation an inverter is placed between the two FIO's Data Direction pins. This provides, for example, that when the left FIO is in Output Handshake (Data DIrection = 0) mode the right FIO will be in Input Handshake mode (Data Direction = 1).

Figure 47 : CPU-to-CPU 512-Byte FIO Buffer Expansion.

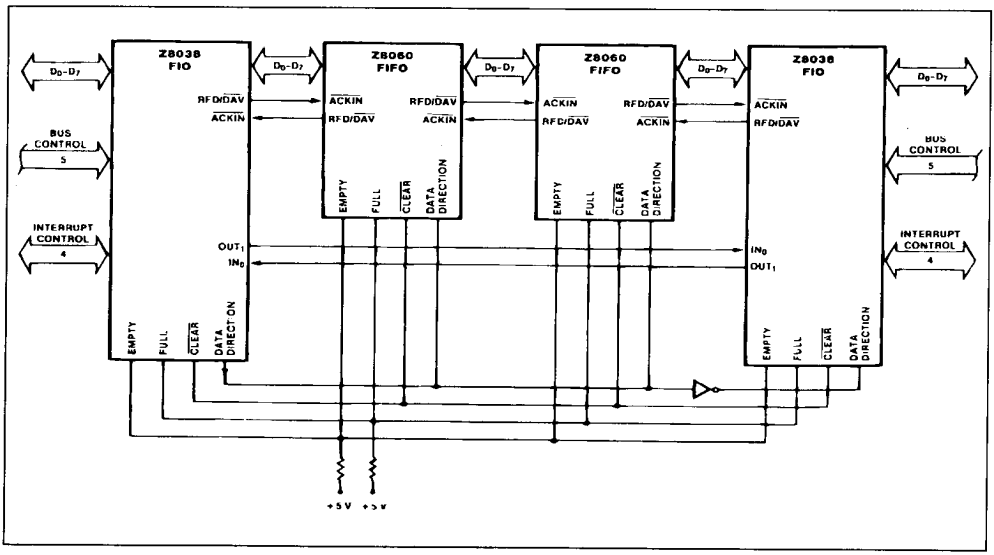
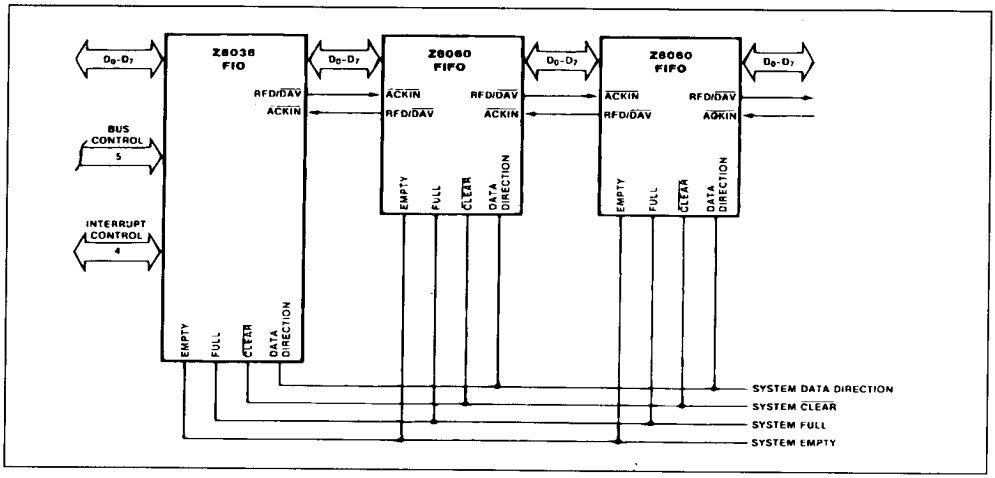


Figure 48 : CPU-to-I/O 384-Byte FIO Buffer Expansion.



FIO EXPANSION INTERFACING (continued)

CPU-TO-CPU 512-Byte FIO-TO-FIFO Buffer Expansion. Figure 47 illustrates a 512-byte buffer interface constructed from two Z8038 FIOs and two Z8060 FIFOs. The Z8060 only operates in the 2-wire hand-shake configuration. The number of FIFOs used in such a configuration is limited only by economics and space.

The FIOs and FIFOs share the same signal configurations used in the previous example, i.e., wire-ORed Full and Empty signals. In this configuration, however, all buffers must be filled (512 bytes) before the Full signal is active, and all buffers must be empty before the Empty signal is active.

CPU-TO-Peripheral 384-Byte FIO Buffer Expansion. Figure 48 illustrates a 384-byte buffer interface constructed from one FIO and two FIFOs. Additional FIFOs can be added as needed, each increasing the buffer capacity. As in the other configurations, the 2-wire handshake configuration must be used.

INTERFACING THE FIO

The FIO manages data transactions between CPUs and between CPUs and peripherals. In some applications, external logic is required to interface the FIO properly, and the software must support the target configuration. This chapter gives two examples of the FIO interfacing to devices : one example shows how the FIO interfaces between the Z8002 CPU and a controller, and the other example shows a Z8000-to-FIO-to-Z80 interface.

FIO INTERFACE BETWEEN Z8002 AND A DISPLAY CONTROLLER

Figure 49 shows a typcial application for the FIO : interfacing a display controller to the Z-BUS. The Z-BUS master, a Z8002 CPU, controls the transactions through Port 1 of the FIO. Port 1 of the FIO is configured in the Z-BUS low byte mode by tying pins 19 (M_1) and 21 (M_0) to ground. In this configuration, the Port 1 Data lines D_0 - D_7 connect direclty to Z-BUS Address/Data lines AD_0 - D_7, and the FIO control registers are directly addressable via these signal lines. The FIO also accepts the Z-BUS interrupt signals directly. The FIO is mapped into I/O locations FFEO - FFFF. Figure 51 shows the software module to interface to the Z8002.

The Z8002 CPU loads the FIO buffer with data for the display controller. When the FIO buffer is full, the WAIT line prevents buffer overflow ; when the buffer is empty, the FIO generates an interrupt request. Clearing the FIO buffer also resets the display controller.

The signal connections between the FIO'd Port 2 and the display controller are straight-forward. The DATA AVAILABLE signal selects the controller when active ; the controller's READY line notifies the FIO's Acknowledge In line when it will accept more data. System RESET and the FIO CLEAR signals are ANDed to create the controller RESET signal.

In this example, the Z8002 CPU adresses the FIO using byte I/O instructions. Therefore, the FIO's RJA bit (D1 of Control Register 0) is 0 and the I/O addresses are all odd.

Figure 49 : FIO to Display Controller.

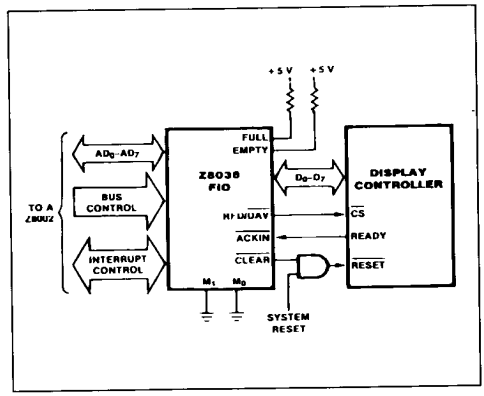
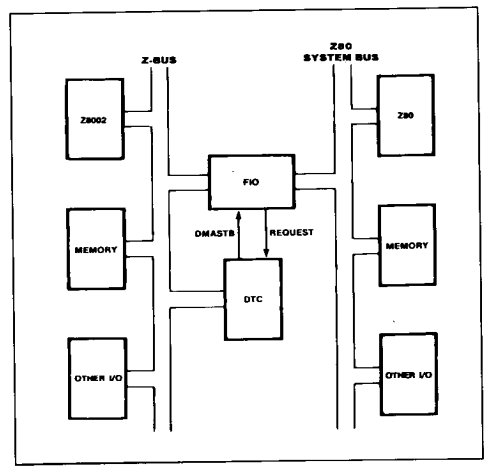


Figure 50 : FIO Interface between the Z8002 and a Display Controller.



FIO INTERFACE BETWEEN Z8002 AND A DISPLAY CONTROLLER (continued)

Figure 51 : Z8002 Initialization Module for the FIO.

```
FIO MODULE                    ! THIS MODULE CONTAINS A PROCEDURE TO INITIALIZE
                              AN FIO FOR USE WITH A DISPLAY CONTROLLER. THE PORT 1 SIDE OF THE
                              FIO IS CONNECTED TO THE LOW-ORDER HALF OF THE Z-BUS, AND THE
                              PORT 2 SIDE IS AN INTERLOCKED HANDSHAKE TO OUTPUT DATA TO THE
                              DISPLAY CONTROLLER. THE FIO IS CHIP SELECTED BY I/O ADDRESSED
                              FFEO  FFFF. !

CONSTANT                      ! FIO REGISTER ADDRESSES !
    CNTL_0 : = % FFE1         CNTL_1 : = % FFE3
    CNTL_2 : = % FFF3         CNTL_3 : = % FFF5
    INT_STATO : = % FFE5      INT_STAT1 : = % FFE7
    INT = STAT2 : = % FFE9    INT_STAT3 : = % FFEB
    INT_VECTOR : = % FFED     FIO = DATA : = % FFFF

    VECTOR : = % F0           ! BASE VECTOR RETURNED DURING INTERRUPT ACK.
                                SEQUENCE !


GLOBAL
    FIO_INIT PROCEDURE        ! FIO INITIALIZATION PROCEDURE !
      ENTRY
        PUSH @ R15, R0        ! SAVE R0 !

        LDB RL0, # % 01
        OUTB # % FFEO RL0     ! RESET FIO !
        CLRB RL0
        OUTB CNTL_0, RL0      ! TURN OFF RESET !

        LDB RL0, # % (2) 10011100
        OUTB CNTL_0, RL0      ! MIE ON, VECTOR INCLUDES STATUS, INTERLOCKED
                                H.S. ON PORT 2 SIDE, RJA = 0 (AD1  AD4) !


        LDB RL0, # % (2) 00000001
        OUTB CNTL_1, RL0      ! WAIT ENABLED !

        LDB RL0, # % (2) 01000000
        OUTB CNTL_3, RL0      ! PORT 1 SIDE CONTROLS CLEAR AND DATA DIRECTION,
                                PORT 1 ACCEPTS CPU OUTPUT, BUFFER CAN HOLD
                                DATA (CLEAR REMOVED) !


        LDB RL0, # VECTOR
        OUTB INT_VECTOR, RL0  ! LOAD BASE INTERRUPT VECTOR, VECTOR RETURNED
                                DURING INTERRUPT ACK. WILL CONTAIN STATUS IN
                                BITS 1, 2, AND 3 !


        LDB RL0, # % (2) 00000011
        OUTB CNTL_2, RL0      ! ENABLE PORT 2 AND PORT 2 HANDSHAKE !

        POP R0, @ R15         ! RESTORE R0 !
        RET
    END FIO_INIT
END FIO
```

FIO INTERFACE BETWEEN Z-BUS AND Z80

Figure 50 shows the FIO interfacing the Z-BUS to the Z80 bus. In this example, the Z-BUS connects to Port 1 of the FIO via Address/Data lines $AD_0 - AD_7$, as in the previous example. The Z8002 CPU controls the CLEAR and DATA DIRECTION functions. The DTC device fills and empties the buffer by DMA transactions to and from Z-BUS memory. Port 1 of the FIO is mapped into I/O locations FFE0 – FFFF. Figure 52 shows the Z80 interrupt acknowledge timing.

The logic necessary to generate this timing is shown in figure 53.

In the software given in figure 54, a Z8000 routine initializes Port 1 to receive data from the Z8002 system. The FIO requests a DMA transaction when the buffer contains 10 or less bytes. The following Z80 program initializes Port 2. The FIO's data port is at address FE, and the control port is at address FF.

Figure 52 : Z80 Interrupt Acknowledge Timing.

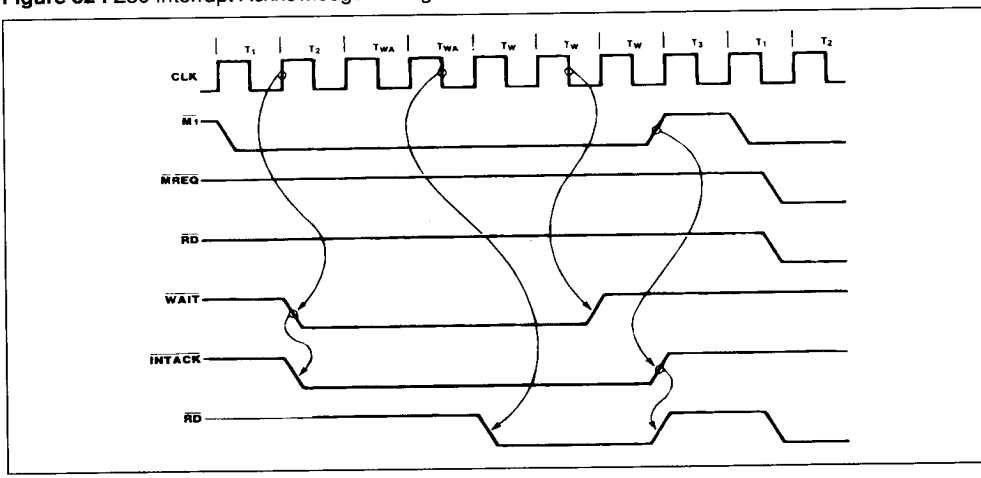
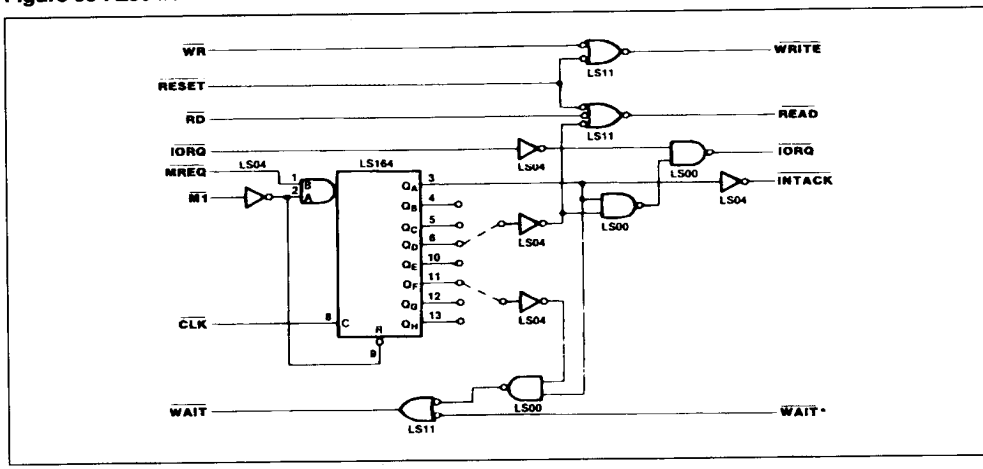


Figure 53 : Z80 INTACK Generation for the FIO.



FIO INTERFACE BETWEEN Z-BUS AND Z80 (continued)

Figure 54 : FIO Port 1 Initialization..

```
FIO_Z8000_TO_Z80 MODULE          ! Z8000 PROGRAM TO INITIALIZE FIO WHICH INTER-
                 FACES Z8000 SYSTEM TO Z80 SYSTEM. THE SYSTEM IS INITIALIZED SO THAT
                 THE Z8000 IS SENDING DATA TO THE Z80 VIA THE FIFO BUFFER. THE Z8000
                 IS IN CONTROL OF DATA DIRECTION AND BUFFER CLEARING. TRANSFERS
                 BETWEEN MEMORY AND THE FIO ON THE Z8000 SIDE ARE HANDLED BY A
                 DMA CONTROLLER. THE Z8000 IS CONNECTED TO THE PORT 1 SIDE OF THE
                 FIO ; THE FIO IS CHIP SELECTED BY I/O PORT ADDRESSES FFE0 THROUGH
                 FFFF, AND IS CONNECTED TO THE LOWER HALF OF THE Z-BUS !

CONSTANT                         ! FIO REGISTER ADDRESSES !
    CNTL_0 : = % FFE1            CNTL_1 : = % FFE3
    CNTL_2 : = % FFF3            CNTL_3 : = % FFF5
    INT_STAT0 : = % FFE5         INT_STAT1 : = % FFE7
    INT_STAT2 : = % FFE9         INT_STAT3 : = % FFEB
    BYTE_CNT : = % FFEF          INT_VECTOR : = % FFED
    FIO_DATA : = % FFFF          DATA_CNT_CMP : = % FFF1
    MSG_IN : = % FFF9            MSG_OUT : = % FFF7
    VECTOR : = % 80              ! BASE VECTOR !
    STARTING_CNT : = 10          ! INITIAL BYTE COMPARISON REGISTER COUNT !

GLOBAL
    FIO_INIT PROCEDURE
      ENTRY
          PUSH @ R15, R0         ! SAVE R0 !

          LDB RL0, # % 01
          OUTB # % FFE0, RL0     ! RESET FIO !
          CLRB RL0
          OUTB CNTL_0, RL0       ! TURN OFF RESET !
          LDB RL0, # % (2) 10010100
          OUTB CNTL_0, RL0       ! MIE ON, VECTOR INCLUDES STATUS, PORT 2 IS NON-
                                   Z-BUS CPU, REG. ADDRESSES SHIFTED LEFT 1 BIT !

          LDB RL0, # % (2) 01000000
          OUTB CNTL_3, RL0       ! PORT 1 SIDE CONTROLS CLEAR AND DATA DIRECTION,
                                   PORT 1 GETS CPU OUTPUT !

          LDB RL0, # VECTOR
          OUTB INT_VECTOR, RL0   ! LOAD BASE VECTOR, VECTOR RETURNED DURING INT.
                                   ACK. INCLUDES STATUS !

          LDB RL0, # % (2) 11000000
          OUTB INT_STAT0, RL0    ! ENABLE MAILBOX REGISTER INTERRUPT !
          LDB RL0, # % (2) 00000001
          OUTB CNTL_2, RL0       ! ENABLE PORT 2 SIDE !
          LDB RL0, # STARTING_CNT
          OUTB DATA_CNT_CMP, RL0 ! LOAD COUNT OF 10 INTO BYTE COUNT COMPARE RE-
                                   GISTER, REQUEST WILL BE MADE TO DMA WHEN BUFF-
                                   ER HAS 10 OR LESS BYTES IN IT !

          LDB RL0, # % (2) 0000011
          OUTB CNTL_1, RL0       ! REQUEST TO DMA ENABLED !
          POP R0, @ R15          ! RESTORE R0 !
          RET
    END FIO_INIT
END FIO_Z8000_TO_Z80
```

FIO INTERFACE BETWEEN Z-BUS AND Z80 (continued)

Figure 55 : FIO Port 2 Initialization.

```
;       THIS Z80 PROGRAM INITIALIZES PORT 2 OF AN FIO
;       USED TO CONNECT A Z8000 AND Z80 SYSTEM TOGETHER
;       WITH THE Z80 RECEIVING DATA FROM THE FIO, VIA
;       AN INTERRUPT WHEN THE FIFO BUFFER IS FULL
;       THE FIO'S DATA PORT IS AT PORT FE HEX, AND THE
;       CONTROL PORT IS AT PORT FF HEX.

FIODAT  EQU     0FEH            ; FIO DATA PORT ADDRESS
FIOCTL  EQU     FIODAT + 1      ; FIO CONTROL PORT ADDR
FIOVEC  EQU     0010H           ; FIO INT. VECTOR ADDR
START :
        IN      A, (FIOCTL)     ; INSURE STATE 0
        XOR     A               ; CLEAR REG A
        OUT     (FIOCTL), A     ; REMOVE RESET
        LD      A, 1            ; POINT TO REG 1
        OUT     (FIOCTL), A
        XOR     A               ; CLEAR REG A
        OUT     (FIOCTL), A     ; CLEAR RESET
        OUT     (FIOCTL), A     ; POINT TO REG 0
        LD      A, 1            ; SET RESET BIT
        OUT     (FIOCTL), A     ; FIO IS RESET NOW
WAITLP :
        IN      A, (FIOCTL)     ; READ REG 0
        CP      000000001       ; CHECK PORT 2 MODE
        JR      Z, WAITLP       ; LOOP UNTIL SET
        XOR     A               ; CLEAR RESET
        OUT     (FIOCTL), A
        LD      HL, FIOLST      ; LOAD INIT LIST PTR
        LD      C, FIOCTL       ; LOAD FIO PORT ADDR
        LD      B, FIOEND-FIOST ; LOAD LIST LENGTH
        OTIR
        RET
FIOLST :
        DEFB    0               ; CTRL REGISTER
        DEFB    10010010B       ; MIE, VIS, RJA
        DEFB    1               ; CTRL REG 1
        DEFB    00000001B       ; NO DMA, WAIT ENABLED
        DEFB    6               ; INT VECTOR REG
        DEFB    FIOVEC. AND. 255 ; LOWER INTERRUPT VECTOR
        DEFB    2               ; INT STATUS REG 0
        DEFB    11000000B       ; SET MESSAGE IE
        DEFB    3               ; INT STATUS REG 1
        DEFB    00001010B       ; CLEAR PATTERN MATCH IP
        DEFB    3
        DEFB    11001100B       ; SET PM IE, SET DDC IE
        DEFB    4               ; INT STATUS REG 2
        DEFB    00001100B       ; SET ERROR IE
        DEFB    5               ; INT STATUS REG 3
        DEFB    11001100B       ; SET FULL IE, EMPTY IE
FIOEND :EQU     $

        END
```

PIN ASSIGNMENTS

Z-BUS Low Byte Mode : Port 1 Side.

Pin Signals	Pin Numbers	Signal Description
AD0-AD7 (address/data)	11-18	Multiplexed Bidirectional Address/data Lines, Z-BUS Compatible
$\overline{REQ}/\overline{WAIT}$ (request/wait)	1	Output, Active Low. $\overline{REQUEST}$ (ready) line for DMA Transfer ; $\overline{WAIT}$ Line (open-drain) Output for Synchronized CPU and FIO Data Transfers.
$\overline{DMASTB}$ (direct Memory Access Strobe)	2	Input, Active Low. Strobes DMA Data to and from the FIFO Buffer.
$\overline{DS}$ (data strobe)	3	Input, Active Low. Provides Timing for Data Transfer to or from FIO.
R/$\overline{W}$ (read/write)	4	Input ; Active high signals CPU read from FIO ; Active low signals CPU write to FIO.
$\overline{CS}$ (chip select)	5	Input, Active Low. Enables FIO. Latched on the Rising Edge of AS.
$\overline{AS}$ (address strobe)	6	Input, Active Low. Addresses, CS and $\overline{INTACK}$ sampled while AS Low.
$\overline{INTACK}$ (interrupt acknowledge)	7	Input, Active Low. Acknowledges an Interrupt. Latched on the Rising Edge of AS.
IEO (interrupt enable out)	8	Output, Active High. Sends interrupt enable to lower priority device IEI pin.
IEI (interrupt enable in)	9	Input, Active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	10	Output, Open Drain, Active Low. Signals FIO interrupt request to CPU.

Z-BUS Low Byte Mode : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
AD0-AD7 (address/data)	29-22	Multiplexed Bidirectional Address/data Lines, Z-BUS Compatible
$\overline{REQ}/\overline{WAIT}$ (request/wait)	39	Output, Active Low. $\overline{REQUEST}$ (ready) line for DMA Transfer ; $\overline{WAIT}$ Line (open-drain) Output for Synchronized CPU and FIO Data Transfers.
$\overline{DMASTB}$ (direct Memory Access Strobe)	38	Input, Active Low. Strobes DMA Data to and from the FIFO Buffer.
$\overline{DS}$ (data strobe)	37	Input, Active Low. Provides Timing for Data Transfer to or from FIO.
R/$\overline{W}$ (read/write)	36	Input ; Active high signals CPU read from FIO ; Active low signals CPU write to FIO.
CS (chip select)	35	Input, Active Low. Enables FIO. Latched on the Rising Edge of AS.
AS (address strobe)	34	Input, Active Low. Addresses, $\overline{CS}$ and $\overline{INTACK}$ sampled while AS Low.
$\overline{INTACK}$ (interrupt acknowledge)	33	Input, Active Low. Acknowledges an Interrupt. Latched on the Rising Edge of AS.
IEO (interrupt enable out)	32	Output, Active High. Sends interrupt enable to lower priority device IEI pin.
IEI (interrupt enable in)	31	Input, Active High. Receives interrupt enable from higher priority device IEO signal.
INT (interrupt)	30	Output, Open Drain, Active Low. Signals FIO interrupt request to CPU.

PIN ASSIGNMENTS (continued)

Non-Z-BUS Mode : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (data)	29-22	Bidirectional Data Bus
$\overline{REQ}/\overline{WAIT}$ (request/wait)	39	Output, Active Low. $\overline{REQUEST}$ (ready) Line for DMA Transfer ; WAIT Line (open-drain) output for Synchronized CPU and FIO Data Transfers.
$\overline{DACK}$ (DMA acknowledge)	38	Input, Active Low. DMA Acknowledge
$\overline{RD}$ (read)	37	Input Active low. Signals CPU read from FIO.
$\overline{WR}$ (write)	36	Input Active Low. Signals CPU write to FIO.
$\overline{CE}$ (chip select)	35	Input, Active Low. Used to Select FIO.
C/$\overline{D}$ (control/data)	34	Input, Active High. Indentifies Data Byte on D0-D7.
$\overline{INTACK}$	33	Input, Active Low. Acknowledges an interrupt
IEO (interrupt enable out)	32	Output, Active high. Sends interrupt enable to lower priority device IEI pin.
IEI (interrupt enable in)	31	Input, Active high. Recives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	30	Output, Open Drain, Active Low. Signals FIO interrupt to CPU.

Figure 56 : Z-BUS-Low-Byte to Z-BUS-Low-Byte.

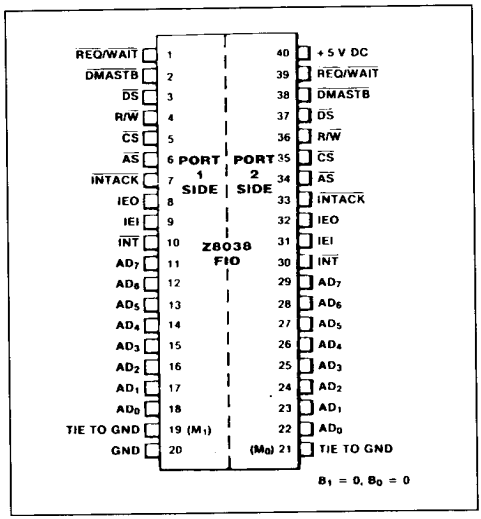
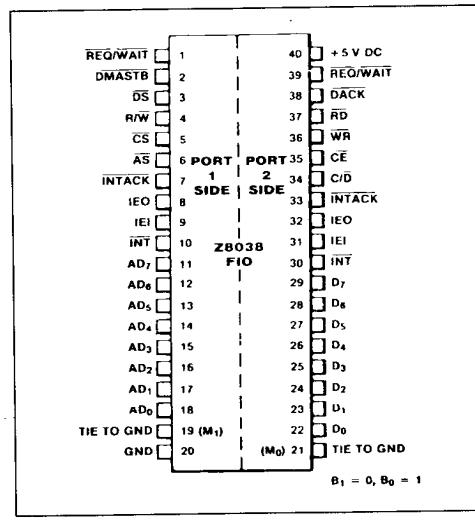


Figure 57 : Z-BUS-Low-Byte to Non-Z-BUS.



PIN ASSIGNMENTS (continued)

Z-BUS Low Byte Mode : Port 1 Side.

Pin Signals	Pin Numbers	Signal Description
AD0-AD7 (address/data)	11-18	Multiplexed Bidirectional Address/data Lines, Z-BUS Compatible
$\overline{REQ}/\overline{WAIT}$ (request/wait)	1	Output, Active Low, $\overline{REQUEST}$ (ready) line for DMA Transfer ; WAIT Line (open-drain) Output for Synchronized CPU and FIO Data Transfers.
$\overline{DMASTB}$ (direct Memory Access Strobe)	2	Input, Active Low. Strobes DMA Data to and from the FIFO Buffer.
$\overline{DS}$ (data strobe)	3	Input, Active Low. Provides Timing for Data Transfer to or from FIO.
R/$\overline{W}$ (read/write)	4	Input ; Active high signals CPU read from FIO ; Active low signals CPU write to FIO.
$\overline{CS}$ (chip select)	5	Input, Active Low, Enables FIO. Latched on the Rising Edge of AS.
$\overline{AS}$ (address strobe)	6	Input, Active Low, Addresses, CS and $\overline{INTACK}$ sampled while AS Low.
$\overline{INTACK}$ (interrupt acknowledge)	7	Input, Active Low. Acknowledges an Interrupt. Latched on the Rising Edge of AS.
IEO (interrupt enable out)	8	Output, Active High. Sends interrupt enable to lower priority device IEI pin.
IEI (interrupt enable in)	9	Input, Active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	10	Output, Open Drain, Active Low. Signals FIO interrupt request to CPU.

PIN ASSIGNMENTS (continued)

3-Wire Handshake : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (data)	29-22	Bidirectional Data Bus
RFD/$\overline{DAV}$ (ready for data/data available)	39	Output, RFD Active High. Signals peripherals that FIO is ready to receive data. DAV active low signals that FIO is ready to send data to peripherals.
$\overline{DAV}$/DAC (data available/data accepted)	38	Input ; $\overline{DAV}$ (active low) signals that data is valid on bus. DAC (active high) signals that ouput data is accepted by peripherals.
DAC/RFD (data accepted/ready for data)	37	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	36	output, Input, Open Drain, Active High. Signals that FIFO buffer is empty.
$\overline{CLEAR}$	35	Programmable Input or Output, Active Low. Clears All Data from FIFO Buffer.
DATA DIR (data direction)	34	Programmable Input or Output. Active High Signals Data Input to Port 2 ; Low Signals Data Output from port 2.
IN_0	33	Input Line to D_0 of Control Register 3
OUT_0	32	Output Line From D_1 of Control Register 3
$\overline{OE}$ (output enable)	31	Input, Active low. When low, enables bus drivers. When high, floats bus drivers at high impedance.
OUT_3	30	Output line from D_3 of Control Register 3

PIN ASSIGNMENTS (continued)

2-Wire Handshake : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (data)	29-22	Bidirectional Data Bus
RFD/$\overline{DAV}$ (ready for data/data available)	39	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)	38	Input, Active Low. Signals FIO that output data is received by peripherals or that input data is valid.
FULL	37	Output, Input, Open Drain, Active High. Signals that FIO buffer is full.
EMPTY	36	Output, Input, Open Drain, Active High. Signals that FIFO buffer is empty.
$\overline{CLEAR}$	35	Programmable Input or Output, Active Low. Clears All Data from FIFO Buffer.
DATA DIR (data direction)	34	Programmable Input or Output. Active High Signals Data Input to Port 2 ; Low Signals Data Output from port 2.
IN_0	33	Input Line to D_0 of Control Register 3
OUT_0	32	Output Line From D_1 of Control Register 3
$\overline{OE}$ (output enable)	31	Input, Active low. When low, enables bus drivers. When high, floats bus drivers at high impedance.
OUT_3	30	Output line from D_3 of Control Register 3

Figure 58 : Z-BUS-Low-Byte to 2-Wire Handshake.

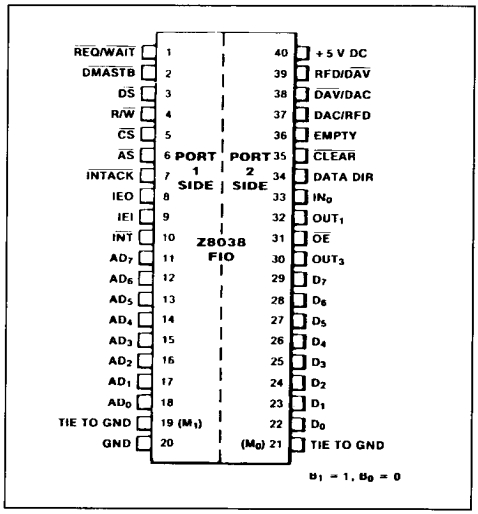
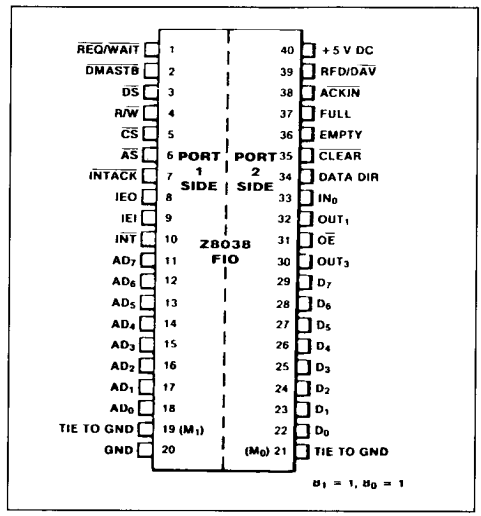


Figure 59 : Z-BUS-Low-Byte to 2-Wire Handshake.



PIN ASSIGNMENTS (continued)

Z-BUS High Byte Mode : Port 1 Side.

Pin Signals	Pin Numbers	Signal Description
AD_0-AD_7 (address/data)	11-18	Multiplexed Bidirectional Address/data Lines Z-BUS Compatible.
$\overline{REQ}/\overline{WAIT}$ (request/wait)	1	Output, Active low, $\overline{REQUEST}$ (ready) line for DMA transfer ; WAIT Line (open-drain) Output for Synchronized CPU and FIO Data transfers.
$\overline{DMASTB}$ (direct memory access strobe)	2	Input, Active low. Strobes DMA data to and from the FIFO Buffer.
$\overline{DS}$ (data strobe)	3	Input, Active low. Provides Timing for Data Transfer to or from FIO.
R/$\overline{W}$ (read/write)	4	Input ; Active high signals CPU read from FIO ; Active low signals CPU write to FIO.
$\overline{CS}$ (chip select)	5	Input, Active Low, Enables FIO. Latched on the Rising Edge of AS.
$\overline{AS}$ (address strobe)	6	Input, Active low. Addresses, $\overline{CS}$ and $\overline{INTACK}$ are sampled while AS is low.
A_0 (address bit 0)	7	Input, Active High. With A_1, A_2 and A_3, Addresses FIO Internal Registers.
A_1 (addresses bit 1)	8	Input, Active high. With A_0, A_2 and A_3, Addresses FIO Internal Registers.
A_2 (addresses bit 2)	9	Input, Active high. With A_0, A_1 and A_3, Addresses FIO Internal Registers.
A_3 (addresses bit 3)	10	Input, Active High. With A_0, A_1, A_3, Addresses FIO Internal Reg

Z-BUS High Byte Mode : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (address/data)	29-22	Multiplexed Bidirectional Address/data Lines Z-BUS Compatible.
$\overline{REQ}/\overline{WAIT}$ (request/wait)`	39	Output, Active low, $\overline{REQUEST}$ (ready) line for DMA transfer ; WAIT Line (open-drain) Output for Synchronized CPU and FIO Data transfers.
$\overline{DMASTB}$ (direct memory access strobe)	38	Input, Active low. Strobes DMA data to and from the FIFO Buffer.
$\overline{DS}$ (data strobe)	37	Input, Active low. Provides Timing for Transfer of Data to or from FIO.
R/$\overline{W}$ (read/write)	36	Input ; Active high Signals CPU read from FIO ; Active low signals CPU write to FIO.
$\overline{CS}$ (chip select)	35	Input, Active Low, Enables FIO. Latched on the Rising Edge of AS.
$\overline{AS}$ (address strobe)	34	Input, Active low. Addresses, $\overline{CS}$ and $\overline{INTACK}$ are sampled while AS is low.
A_0 (address bit 0)	33	Input, Active High. With A_1, A_2 and A_3, Addresses FIO Internal Registers.
A_1 (addresses bit 1)	32	Input, Active high. With A_0, A_2 and A_3, Addresses FIO Internal Registers.
A_2 (addresses bit 2)	31	Input, Active high. With A_0, A_1 and A_3, Addresses FIO Internal Registers.
A_3 (addresses bit 3)	30	Input, Active High. With A_0, A_1, A_2, Addresses FIO Internal Registers.

PIN ASSIGNMENTS (continued)

Non-Z-BUS Mode : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D0-D7 (data)	29-22	Bidirectional Data Bus
$\overline{REQ}/\overline{WAIT}$ (request/wait)	39	Output, Active Low. $\overline{REQUEST}$ (ready) Line for DMA Transfer ; WAIT Line (open-drain) output for Synchronized CPU and FIO Data Transfers.
$\overline{DACK}$ (DMA acknowledge)	38	Input, Active Low. DMA Acknowledge
$\overline{RD}$ (read)	37	Input Active low. Signals CPU read from FIO.
$\overline{WR}$ (write)	36	Input Active Low. Signals CPU write to FIO.
$\overline{CE}$ (chip select)	35	Input, Active Low. Used to Select FIO.
C/$\overline{D}$ (control/data)	34	Input, Active High. Indentifies Control Byte on D_0-D_7 ; Active Low indentifies Data Byte on D_0-D_7.
$\overline{INTACK}$ (interrupt acknowledge)	33	Input, Active Low. Acknowledges an interrupt
IEO (interrupt enable out)	32	Output, Active high. Sends interrupt enable to lower priority device IEI pin.
IEI (interrupt enable in)	31	Input, Active high. Recives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	30	output, Open Drain, Active Low. Signals FIO interrupt to CPU.

Figure 60 : Z-BUS-High-Byte to Z-BUS-High-Byte.

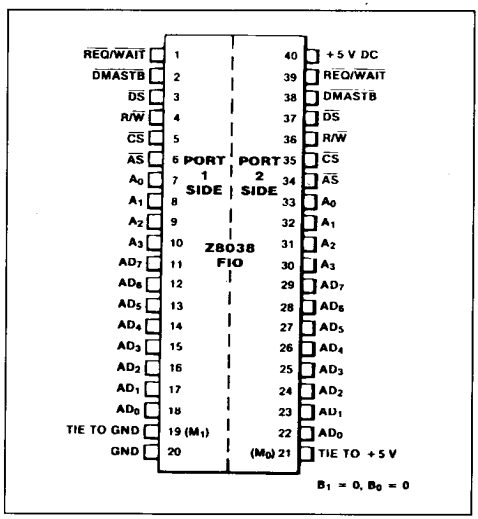
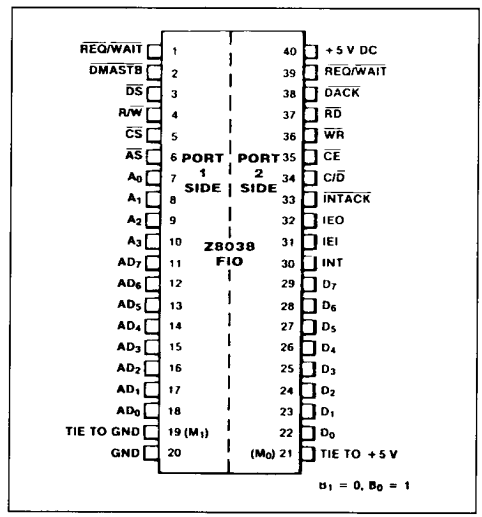


Figure 61 : Z-BUS-High-Byte to Non-Z-BUS.



PIN ASSIGNMENTS (continued)

Z-BUS High Byte Mode : Port 1 Side.

Pin Signals	Pin Numbers	Signal Description
AD0-AD7 (address/data)	11-18	Multiplexed Bidirectional Address/data Lines Z-BUS Compatible.
$\overline{REQ}/\overline{WAIT}$ (request/wait)	1	Output, Active low, $\overline{REQUEST}$ (ready) line for DMA transfer ; WAIT Line (open-drain) Output for Synchronized CPU and FIO Data transfers.
$\overline{DMASTB}$ (direct memory access strobe)	2	Input, Active low. Strobes DMA data to and from the FIFO Buffer.
$\overline{DS}$ (data strobe)	3	Input, Active low. Provides Timing Transfer of Data to or from FIO.
R/$\overline{W}$ (read/write)	4	Input ; Active high signals CPU read from FIO ; Active low signals CPU write to FIO.
$\overline{CS}$ (chip select)	5	Input, Active Low. Enables FIO. Latched on the Rising Edge of AS.
$\overline{AS}$ (address strobe)	6	Input, Active low. Addresses, $\overline{CS}$ and $\overline{INTACK}$ are sampled while AS is low.
A_0 (address bit 0)	7	Input, Active High. With A_1, A_2 and A_3, Addresses FIO Internal Registers.
A_1 (addresses bit 1)	8	Input, Active high. With A_0, A_2 and A_3, Addresses FIO Internal Registers.
A_2 (addresses bit 2)	9	Input, Active high. With A_0, A_1 and A_3, Addresses FIO Internal Registers.
A_3 (address bit 3)	10	Input, Active High. With A_0, A_1, A_2, Addresses FIO Internal Registers.

PIN ASSIGNMENTS (continued)

3-Wire Handshake : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (data)	29-22	Bidirectional Data Bus
RFD/$\overline{DAV}$ (ready for data/data available)	39	Output, RFD Active High. Signals peripherals that FIO is ready to receive data. DAV active low signals that FIO is ready to send data to peripherals.
$\overline{DAV}$/DAC (data available/data accepted)	38	Input ; $\overline{DAV}$ (active low) signals that data is valid on bus. DAC (active high) signals that ouput data is accepted by peripherals.
DAC/RFD (data accepted/ready for data)	37	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	36	Output, Input, Open Drain, Active High. Signals that FIFO buffer is empty.
$\overline{CLEAR}$	35	Programmable Input or Output, Active Low. Clears All Data from FIFO Buffer.
DATA DIR (data direction)	34	Programmable Input or Output. Active High Signals Data Input to Port 2 ; Low Signals Data Output from port 2.
IN_0	33	Input Line to D_0 of Control Register 3
OUT_1	32	Output Line from D_1 of Control Register 3
$\overline{OE}$ (output enable)	31	Input, Active low. When low, enables bus drivers. When high, floats bus drivers at high impedance.
OUT_3	30	Output line from D_3 of Control Register 3

PIN ASSIGNMENTS (continued)

Non-Z-BUS Mode : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (data)	29-22	Bidirectional Data Bus
$\overline{REQ}/\overline{WAIT}$ (request/wait)	39	Output, Active Low. REQUEST (ready) Line for DMA Transfer ; WAIT Line (open-drain) output for Synchronized CPU and FIO Data Transfers.
$\overline{DACK}$ (DMA acknowledge)	38	Input, Active Low. DMA Acknowledge
$\overline{RD}$ (read)	37	Input Active low. Signals CPU read from FIO.
$\overline{WR}$ (write)	36	Input Active Low. Signals CPU write to FIO.
$\overline{CE}$ (chip select)	35	Input, Active Low. Used to Select FIO.
C/$\overline{D}$ (control/data)	34	Input, Active High. Identifies Control Byte on D_0-D_7 ; Active Low identifies Data Byte on D_0-D_7.
$\overline{INTACK}$	33	Input, Active Low. Acknowledges an interrupt
IEO (interrupt enable out)	32	Output, Active high. Sends interrupt enable to lower priority device IEI pin.
IEI (interrupt enable in)	31	Input, Active high. Recives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	30	Output, Open Drain, Active Low. Signals FIO interrupt to CPU.

Figure 64 : Non-Z-BUS to Z-BUS-Low-Byte.

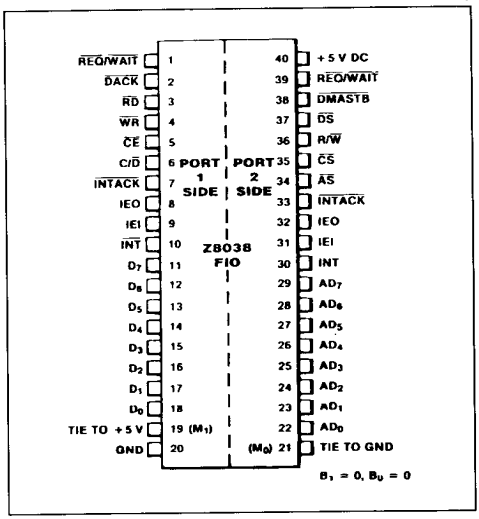
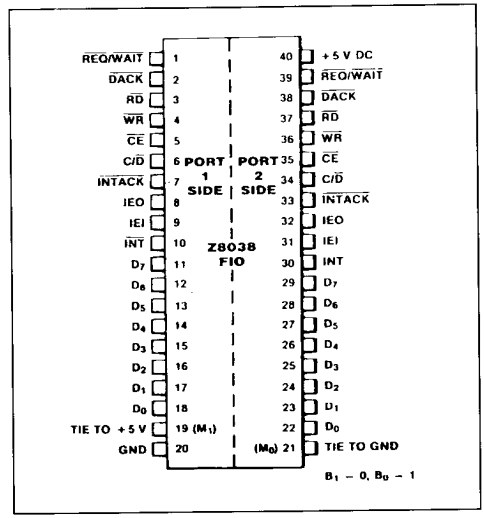


Figure 65 : Non-Z-BUS to Non-Z-BUS.



PIN ASSIGNMENTS (continued)

Z-BUS High Byte Mode : Port 1 Side.

Pin Signals	Pin Numbers	Signal Description
AD0-AD7 (address/data)	11-18	Bidirectional Data Bus
$\overline{REQ}/\overline{WAIT}$ (request/wait)	1	Output, Active Low, $\overline{REQUEST}$ (ready) line for DMA Transfer ; WAIT Line (open-drain) Output for Synchronized CPU and FIO Data Transfers.
$\overline{DACK}$ (DMA acknowledge)	2	Input, Active Low. DMA Acknowledge
$\overline{RD}$ (read)	3	Input, Active Low. Signals CPU read from FIO.
$\overline{WR}$ (write)	4	Input, Active Low. Signals CPU write to FIO.
$\overline{CE}$ (chip select)	5	Input, Active low. Used to Select FIO.
C/$\overline{D}$ (control/data)	6	Input, Active High. Indentifies Control Byte on D_0-D_7 ; Active Low Indentifies Data Byte on D_0-D_7.
$\overline{INTACK}$ (interrupt acknowledge)	7	Input, Active Low. Acknowledges an Interrupt.
IEO (interrupt enable out)	8	Output, Active High. Sends interrupt enable to lower priority device IEI pin.
IEI (interruppt enable in)	9	Input, Active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	10	Output, Open Drain, Active Low. Signals FIO interrupt to CPU.

Z-BUS Low Byte Mode : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (address/data)	29-22	Multiplexed Bidirectional Address/data Lines, Z-BUS Compatible
$\overline{REQ}/\overline{WAIT}$ (request/wait)	39	Output, Active Low, $\overline{REQUEST}$ (ready) line for DMA Transfer ; WAIT Line (open-drain) Output for Synchronized CPU and FIO Data Transfers.
$\overline{DMASTB}$ (direct Memory Access Strobe)	38	Input, Active Low. Strobes DMA Data to and from the FIFO Buffer.
$\overline{DS}$ (data strobe)	37	Input, Active Low. Provides Timing for Data Transfer to or from FIO.
R/$\overline{W}$ (read/write)	36	Input ; Active high signals CPU read from FIO ; Active low signals CPU write to FIO.
$\overline{CS}$ (chip select)	35	Input, Active Low, Enables FIO. Latched on the Rising Edge of AS.
$\overline{AS}$ (address strobe)	34	Input, Active Low, Addresses, $\overline{CS}$ and $\overline{INTACK}$ sampled while AS Low.
$\overline{INTACK}$ (interrupt acknowledge)	33	Input, Active Low. Acknowledges an Interrupt. Latched on the Rising Edge of AS.
IEO (interrupt enable out)	32	Output, Active High. Sends interrupt enable to lower priority device IEI pin.
IEI (interrupt enable in)	31	Input, Active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	30	Output, Open Drain, Active Low. Signals FIO interrupt request to CPU.

PIN ASSIGNMENTS (continued)

2-Wire Handshake : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D0-D7 (data)	29-22	Bidirectional Data Bus
RFD/$\overline{DAV}$ (ready for data/data available)	39	Output, RFD Active High. Signals peripherals that FIO is ready to receive data. DAV active low signals that FIO is ready to send data to peripherals.
$\overline{ACKIN}$ (acknowledge input)	38	Input, Active Low. Signals FIO that output data is received by peripherals or that input data is valid.
FULL	37	Output, Input, Open Drain, Active High. Signals that FIO buffer is full.
EMPTY	36	output, Input, Open Drain, Active High. Signals that FIFO buffer is empty.
CLEAR	35	Programmable Input or Output, Active Low. Clears All Data from FIFO Buffer.
DATA DIR (data direction)	34	Programmable Input or Output. Active High Signals Data Input to Port 2 ; Low Signals Data Output from port 2.
IN_0	33	Input Line to D_0 of Control Register 3
OUT_0	32	Output Line From D_1 of Control Register 3
$\overline{OE}$ (output enable)	31	Input, Active low. When low, enables bus drivers. When high, floats bus drivers at high impedance.
OUT_3	30	Output line from D_3 of Control Register 3

Figure 62 : Z-BUS-High-Byte to 3-Wire Handshake.

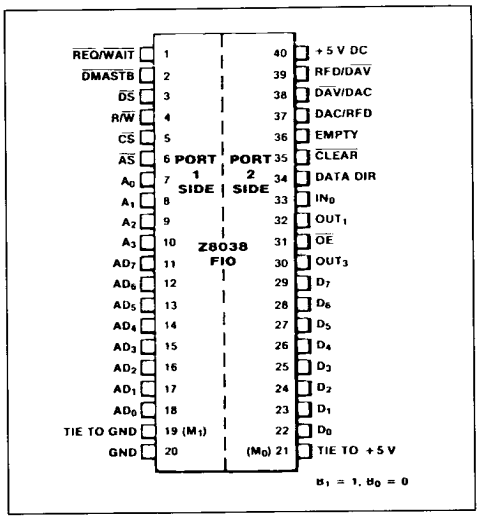
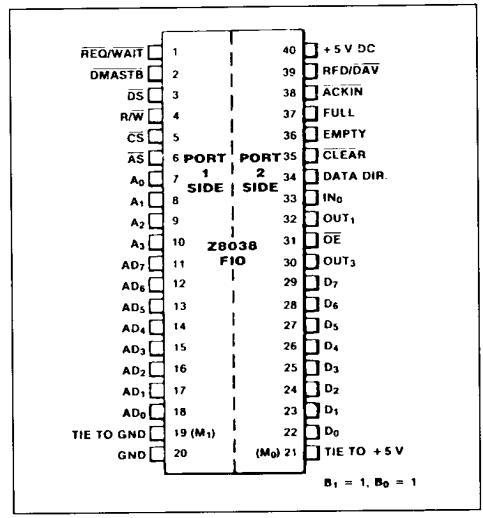


Figure 63 : Z-BUS-High-Byte to 2-Wire Handshake.



PIN ASSIGNMENTS (continued)

Non-Z-BUS Mode : Port 1 Side.

Pin Signals	Pin Numbers	Signal Description
D0-D7 (data)	11-18	Bidirectional Data Bus
$\overline{REQ}/\overline{WAIT}$ (request/wait)	1	Output, Active Low, $\overline{REQUEST}$ (ready) line for DMA Transfer ; WAIT Line (open-drain) Output for Synchronized CPU and FIO Data Transfers.
$\overline{DACK}$ (DMA acknowledge)	2	Input, Active Low. DMA Acknowledge
$\overline{RD}$ (read)	3	Input, Active Low. Signals CPU read from FIO.
$\overline{WR}$ (write)	4	Input, Active Low. Signals CPU write to FIO.
$\overline{CE}$ (chip select)	5	Input, Active low. Used to Select FIO.
C/$\overline{D}$ (control/data)	6	Input, Active High. Indentifies Control Byte on D_0-D_7 ; Active Low Indentifies Data Byte on D_0-D_7.
INTACK (interrupt acknowledge)	7	Input, Active Low. Acknowledges an Interrupt.
IEO (interrupt enable out)	8	Output, Active High. Sends interrupt enable to lower priority device IEI pin.
IEI (interruppt enable in)	9	Input, Active High. Receives interrupt enable from higher priority device IEO signal.
$\overline{INT}$ (interrupt)	10	Output, Open Drain, Active Low. Signals FIO interrupt to CPU.

3-Wire Handshake : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D0-D7 (data)	29-22	Bidirectional Data Bus
RFD/$\overline{DAV}$ (ready for data/data available)	39	Output, RFD Active High. Signals peripherals that FIO is ready to receive data. DAV active low signals that FIO is ready to send data to peripherals.
$\overline{DAV}$/DAC (data available/data accepted)	38	Input ; $\overline{DAV}$ (active low) signals that data is valid on bus. DAC (active high) signals that ouput data is accepted by peripherals.
DAC/RFD (data accepted/ready for data)	37	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	36	Output, Input, Open Drain, Active High. Signals that FIFO buffer is empty.
$\overline{CLEAR}$	35	Programmable Input or Output, Active Low. Clears All Data from FIFO Buffer.
DATA DIR (data direction)	34	Programmable Input or Output. Active High Signals Data Input to Port 2 ; Low Signals Data Output from port 2.
IN_0	33	Input Line to D_0 of Control Register 3
OUT_1	32	Output Line from D_1 of Control Register 3
$\overline{OE}$ (output enable)	31	Input, Active low. When low, enables bus drivers. When high, floats bus drivers at high impedance.
OUT_3	30	Output line from D_3 of Control Register 3

PIN ASSIGNMENTS (continued)

2-Wire Handshake : Port 2 Side.

Pin Signals	Pin Numbers	Signal Description
D_0-D_7 (data)	29-22	Bidirectional Data Bus
RFD/$\overline{DAV}$ (ready for data/data available)	39	Output, RFD Active High. Signals peripherals that FIO is ready to receive data. DAV active low signals that FIO is ready to send data to peripherals.
$\overline{ACKIN}$ (acknowledge input)	38	Input, Active Low. Signals FIO that output data is received by peripherals or that input data is valid.
FULL	37	Output, Input, Open Drain, Active High. Signals that FIO buffer is full.
EMPTY	36	Output, Input, Open Drain, Active High. Signals that FIFO buffer is empty.
$\overline{CLEAR}$	35	Programmable Input or Output, Active Low. Clears All Data from FIFO Buffer.
DATA DIR (data direction)	34	Programmable Input or Output. Active High Signals Data Input to Port 2 ; Low Signals Data Output from port 2.
IN_0	33	Input Line to D_0 of Control Register 3
OUT_0	32	Output Line From D_1 of Control Register 3
$\overline{OE}$ (output enable)	31	Input, Active low. When low, enables bus drivers. When high, floats bus drivers at high impedance.
OUT_3	30	Output line from D_3 of Control Register 3

Figure 66 : Non-Z-BUS to 3-Wire Handshake.

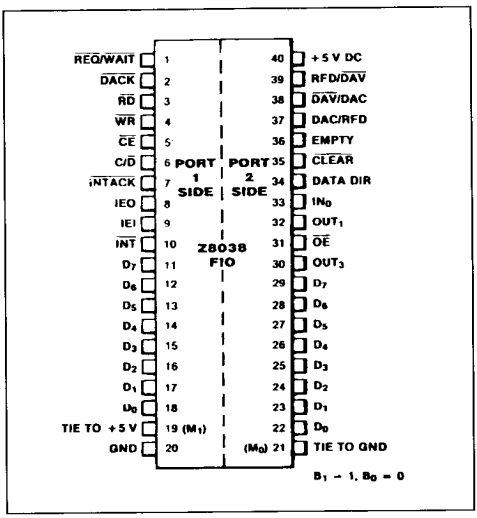
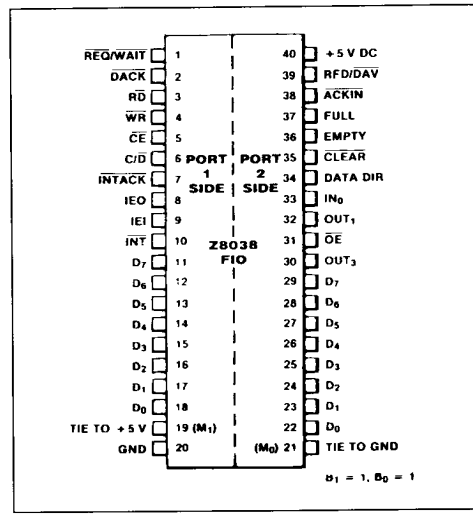
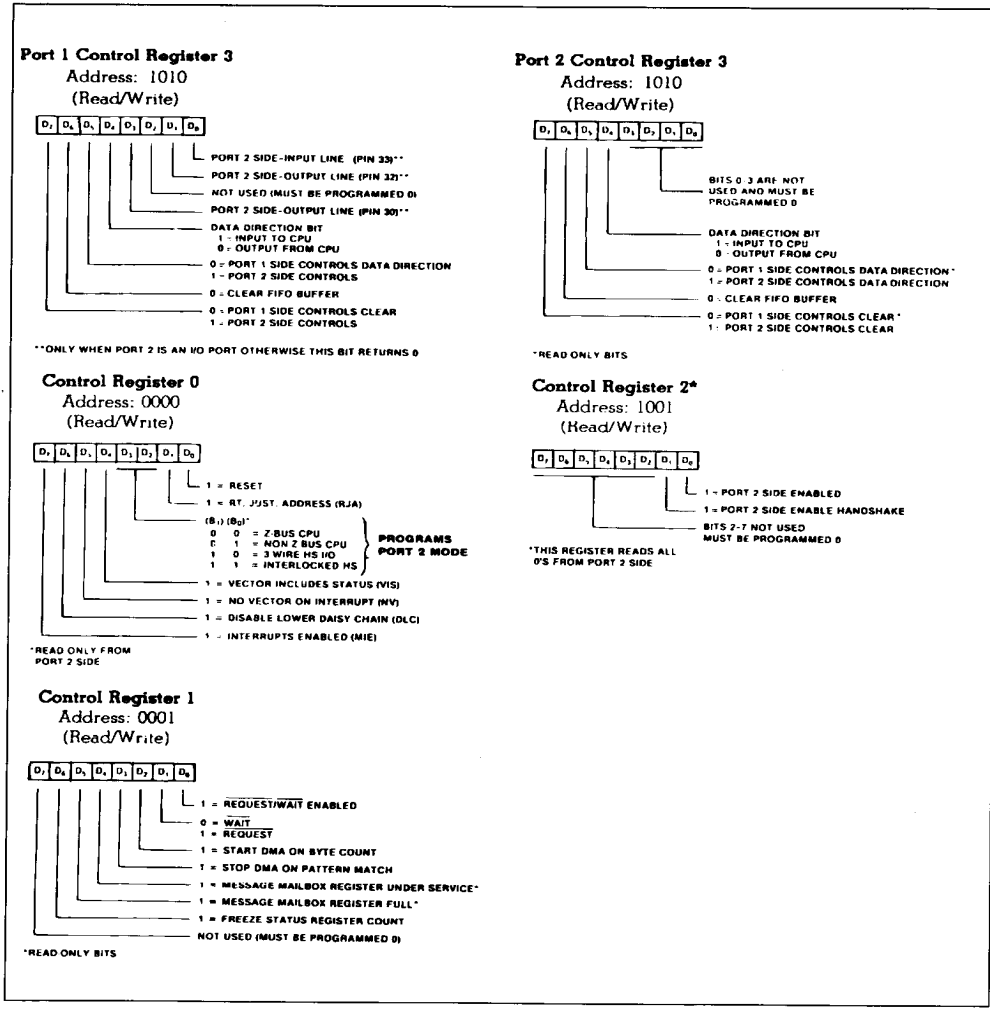


Figure 67 : Non-Z-BUS to 2-Wire Handshake.



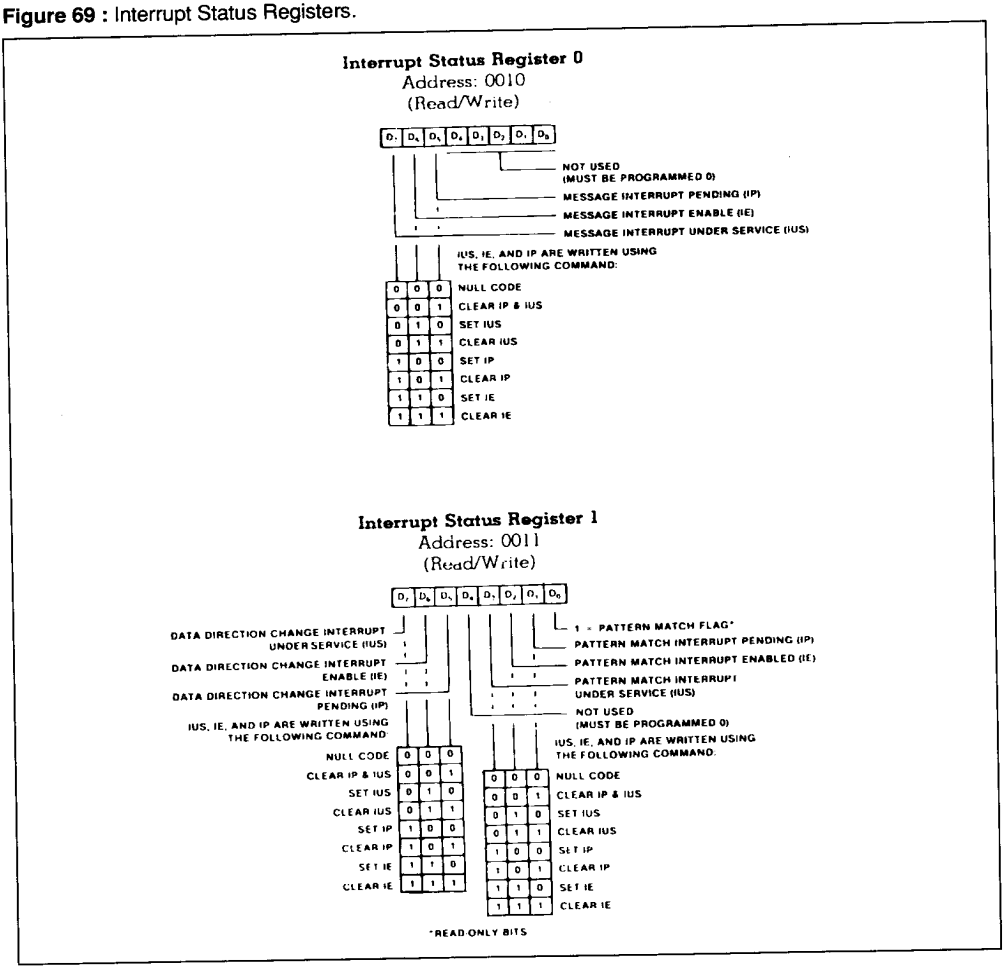
REGISTERS

Figure 68 : Control Registers.



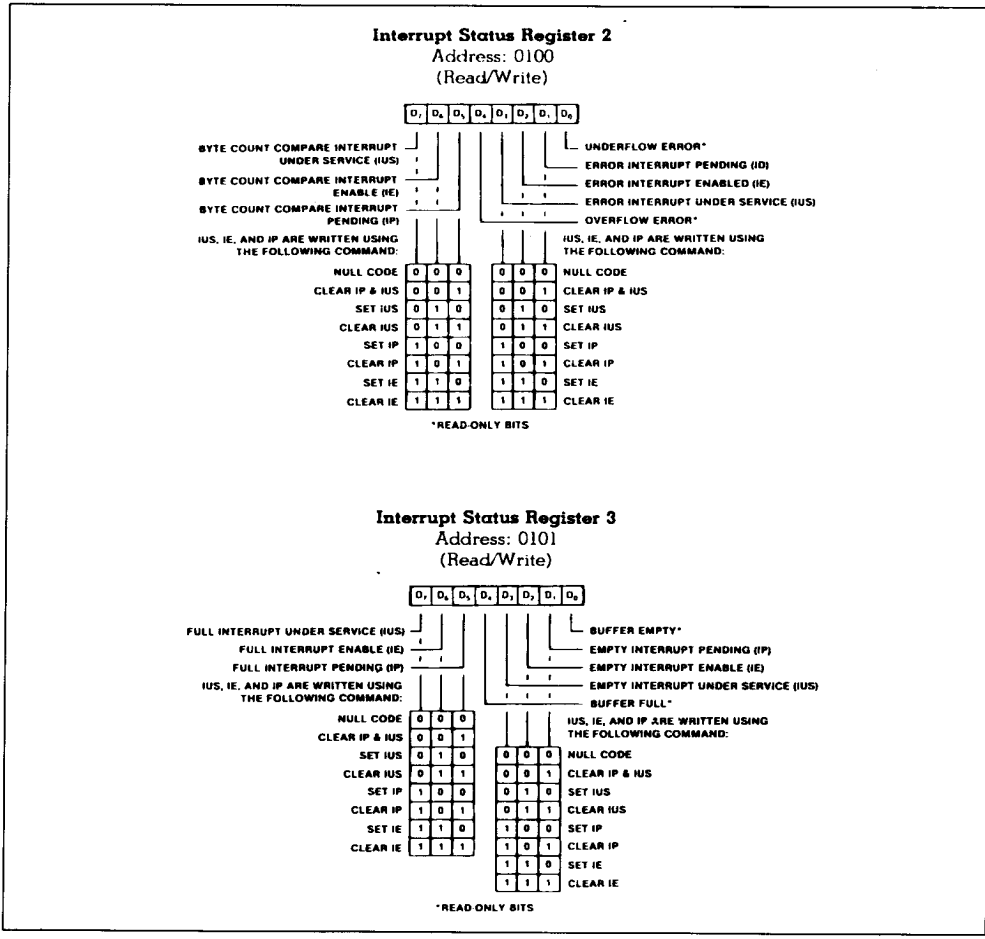
REGISTERS (continued)

Figure 69 : Interrupt Status Registers.



REGISTERS (continued)

Figure 70 : Interrupt Status Registers (continued).



REGISTERS (continued)

Figure 71 : Count Register.

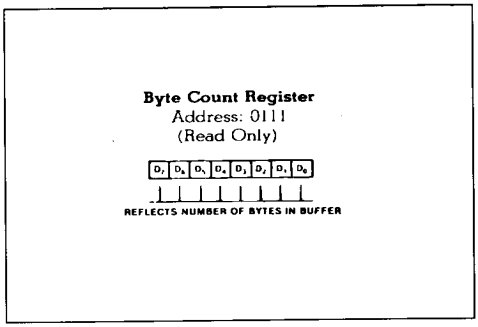


Figure 72 : Interrupt Vector Register.

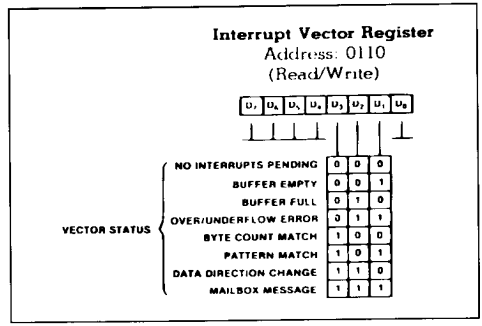


Figure 73 : Pattern Match Register.

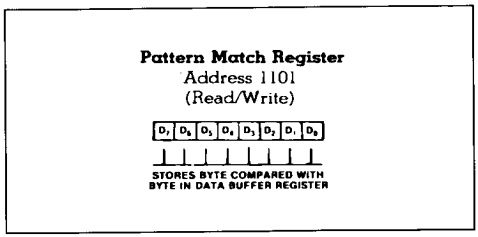


Figure 74 : Pattern Mask Register.

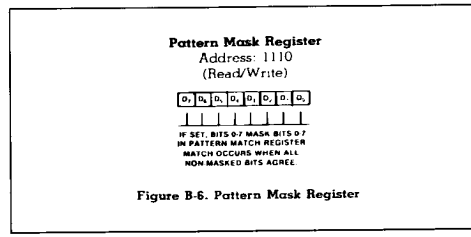


Figure B-6. Pattern Mask Register

Figure 75 : Data Buffer Register.

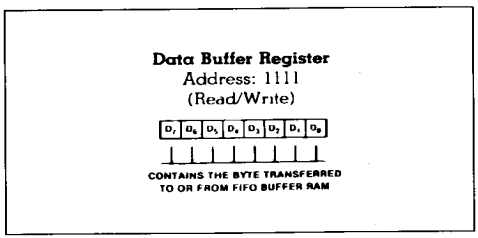


Figure 76 : Byte Count Comparation Register.

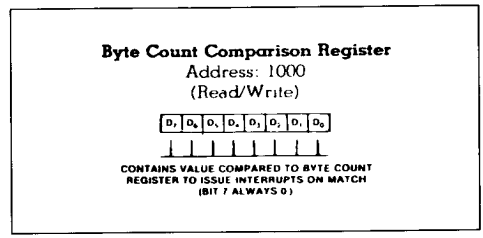


Figure 77 : Message Out Register.

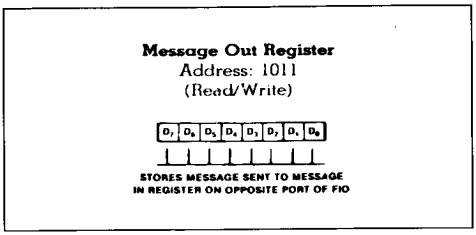
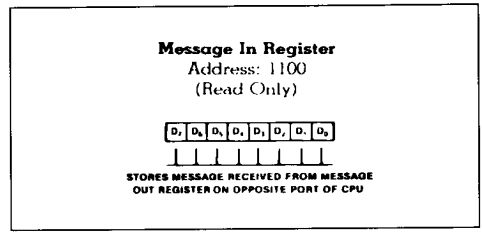


Figure 78 : Message In Register.



ELECTRICAL INFORMATION

ABSOLUTE MAXIMUM RATING

Symbol	Parameter	Test Conditions	Unit
V_I	Voltages on All Inputs and Outputs with Respect to GND	− 0.3 to + 7.0	V
T_A	Operating Ambient Temperature	0 to + 70	°C
T_{stg}	Storage Temperature	− 65 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 ≤ VCC ≤ + 5.25 V
- GND = 0 V
- T_A as specified in Ordering Information

Figure 79 : Standard Test Conditions.

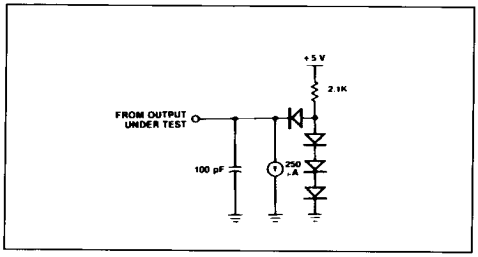
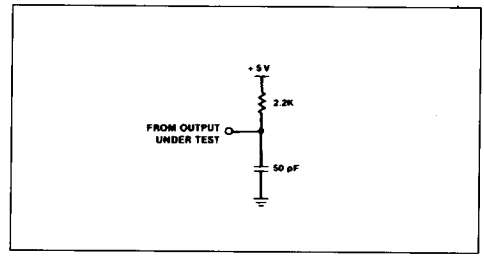


Figure 80 : Open-Drain Test Load.



DC CHARACTERISTICS

Symbol	Parameter	Test Conditions	Min.	Max.	Unit
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	I_{OH} = − 250 mA	2.4		V
V_{OL}	Output Low Voltage	I_{OL} = + 2.0 mA I_{OL} = 3.2 mA		0.4 0.5	V V
I_{IL}	Input Leakage	0.4 = V_{IN} = + 2.4 V		± 10	µA
I_{IL} (M_0, M_1)	Mode Pin Leakage	0.4 = V_{IN} = + 2.4 V		± 100	µA
I_{OL}	Output Leakage	0.4 = V_{OUT} = + 2.4 V		± 10	µA
I_{CC}	V_{CC} Supply Current			250	mA

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

CAPACITANCE

Symbol	Parameter	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Capacitance	Unmeasured Pins Returned to Ground		10	pF
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

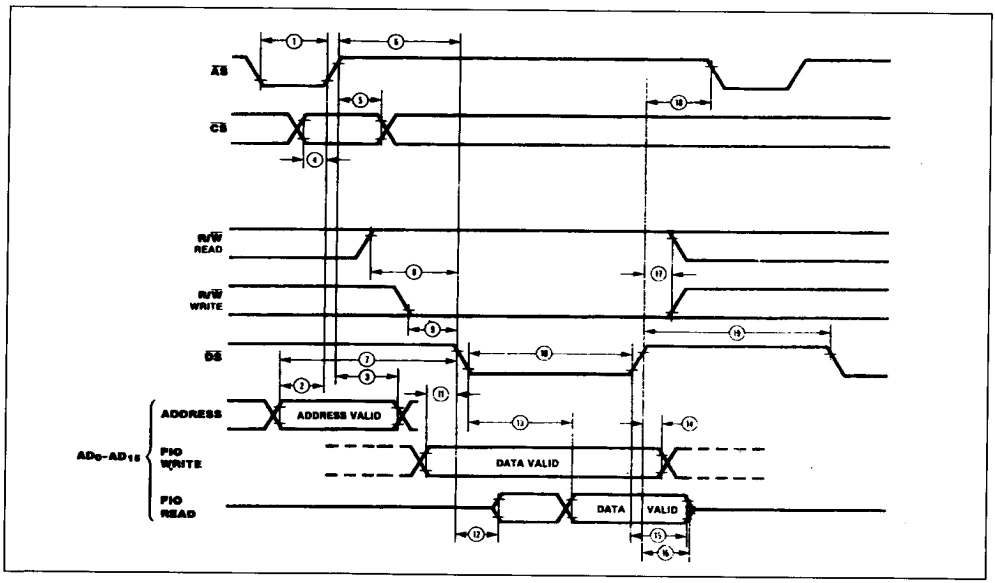
f = 1 MHz, over specified temperature range.

AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TwAS	$\overline{AS}$ Low Width	70		50		1
2	TsA(AS)	Address to $\overline{AS}$ ↑ Setup Time	30		10		1
3	ThA(AS)	Address to $\overline{AS}$ ↑ Hold Time	50		30		1
4	TsCSO(AS)	$\overline{CS}$ to $\overline{AS}$ ↑ Setup Time	0		0		1
5	ThCSO(AS)	$\overline{CS}$ to $\overline{AS}$ ↑ Hold Time	60		40		1
6	TdAS(DS)	$\overline{AS}$ ↑ to $\overline{DS}$ ↓ Delay	60		40		1
7	TsA(DS)	Address to $\overline{DS}$ ↓ (with $\overline{AS}$ ↑ to $\overline{DS}$ ↓ = 60 ns)	120		100		
8	TsRWR(DS)	R/$\overline{W}$ (read) to $\overline{DS}$ ↓ Setup Time	100		80		
9	TsRWW(DS)	R/$\overline{W}$ (write) to $\overline{DS}$ ↓ Setup Time	0		0		
10	TwDS	$\overline{DS}$ Low Width	390		250		
11	TsDW(DSf)	Write Data to $\overline{DS}$ ↓ Setup Time	30		20		
12	TdDS(DRV)	$\overline{DS}$ (read) ↓ to Address Data Bus Driven	0		0		
13	TdDSf(DR)	$\overline{DS}$ ↓ to Read Data Valid Delay		250		180	
14	ThDW(DS)	Write Data to $\overline{DS}$ ↑ Hold Time	30		20		
15	TdDSr(DR)	$\overline{DS}$ ↑ to Read Data not Valid Delay	0		0		
16	TdDS(DRz)	$\overline{DS}$ ↑ to Read Data Float Delay		70		45	2
17	ThRW(DS)	R/$\overline{W}$ to $\overline{DS}$ ↑ Hold Time	55		40		
18	TdDS(AS)	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	50		25		
19	Trc	Valid Access Recovery Time	1000		650		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 a delay from $\overline{DS}$ of one FIO access to $\overline{DS}$ of another FIO access (either read or write).
* All timing references assume 2.0 V for a logic "1" and 0.8 V for a logic "0". All timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 81 : Z-BUS CPU Interface Timing.



AC CHARACTERISTICS (Continued)

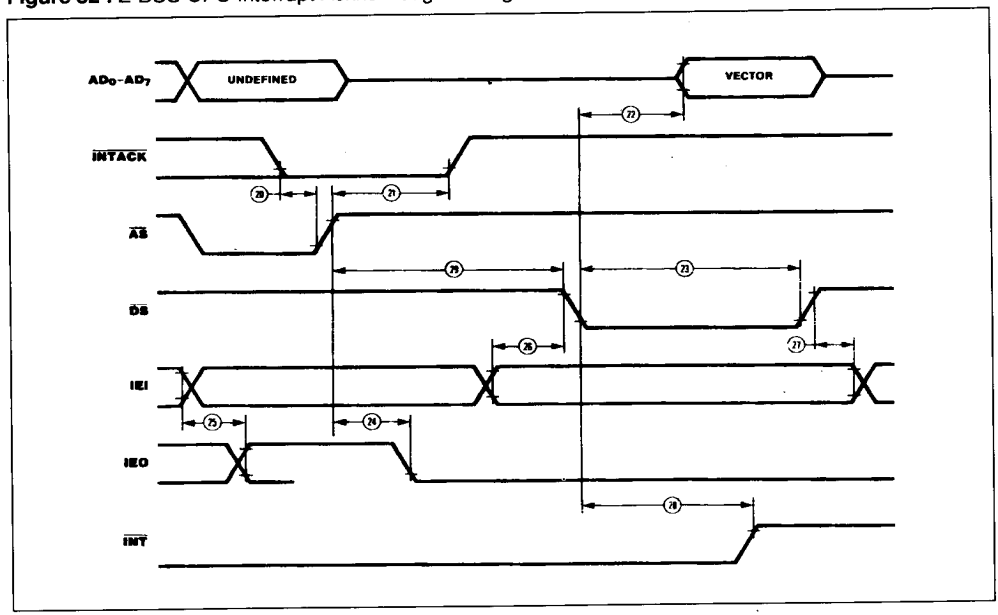
N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
20	TsIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Setup Time	0		0		
21	ThIA(AS)	$\overline{INTACK}$ to $\overline{AS}$ ↑ Hold Time	250		250		
22	TsDSA(DR)	$\overline{DS}$ (acknowledge)↓ to Read Data Valid Delay		250		180	
23	TwDSA	$\overline{DS}$ (acknowledge) Low Width	390		250		
24	TdAS(IEO)	$\overline{AS}$ ↑ to IEO ↓ Delay ($\overline{INTACK}$ cycle)		350		250	4
25	TdIEI(IEO)	IEI to IEO Delay		150		100	4
26	TsIEI(DSA)	IEI to $\overline{DS}$ (acknowledge) ↓ Setup Time	100		70		
27	ThIEI(DSA)	IEI to $\overline{DS}$ (acknowledge) ↑ Hold Time	50		30		4
28	TdDS(INT)	$\overline{DS}$ ($\overline{INTACK}$ cycle) to $\overline{INT}$ Delay		900		800	
29	TdDCST	Interrupt Daisy Chain Settle Time					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.

* Timings are preliminary and subject to change.

† Units in nanoseconds (ns).

Figure 82 : Z-BUS CPU Interrupt Acknowledge Timing.

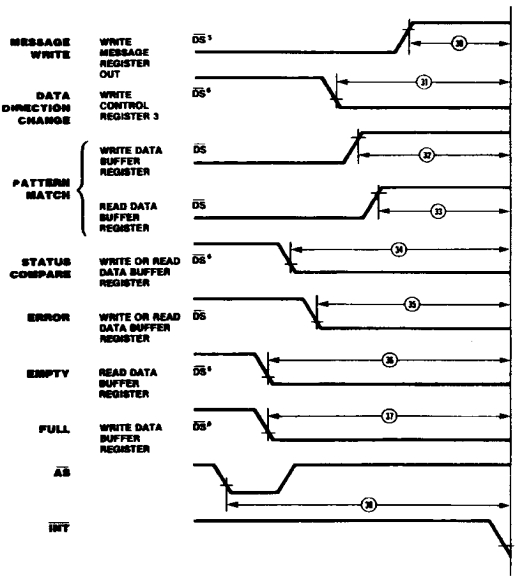


AC CHARACTERISTICS (Continued)

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
30	TdMW(INT)	Message Write to $\overline{INT}$ Delay		1		1	5
31	TdDC(INT)	Data Direction Change to $\overline{INT}$ Delay		1		1	6
32	TdPMW(INT)	Pattern Match to $\overline{INT}$ Delay (write case)		1		1	
33	TdPMR(INT)	Pattern Match (read case) to $\overline{INT}$ Delay		1		1	
34	TdSC(INT)	Status Compare to $\overline{INT}$ Delay		1		1	6
35	TdER(INT)	Error to $\overline{INT}$ Delay		1		1	
36	TdEM(INT)	Empty to $\overline{INT}$ Delay		1		1	6
37	TdFL(INT)	Full to $\overline{INT}$ Delay		1		1	6
38	TdAS(INT)	$\overline{AS}$ to $\overline{INT}$ Delay					

Notes : 5. Write is from the other side of FIO.
6. Write can be from either side, depending on programming of FIO.
* Timing are preliminary and subject to change.
† Units equal to AS Cycles + ns.

Figure 83 : Z-BUS Interrupt Timing.

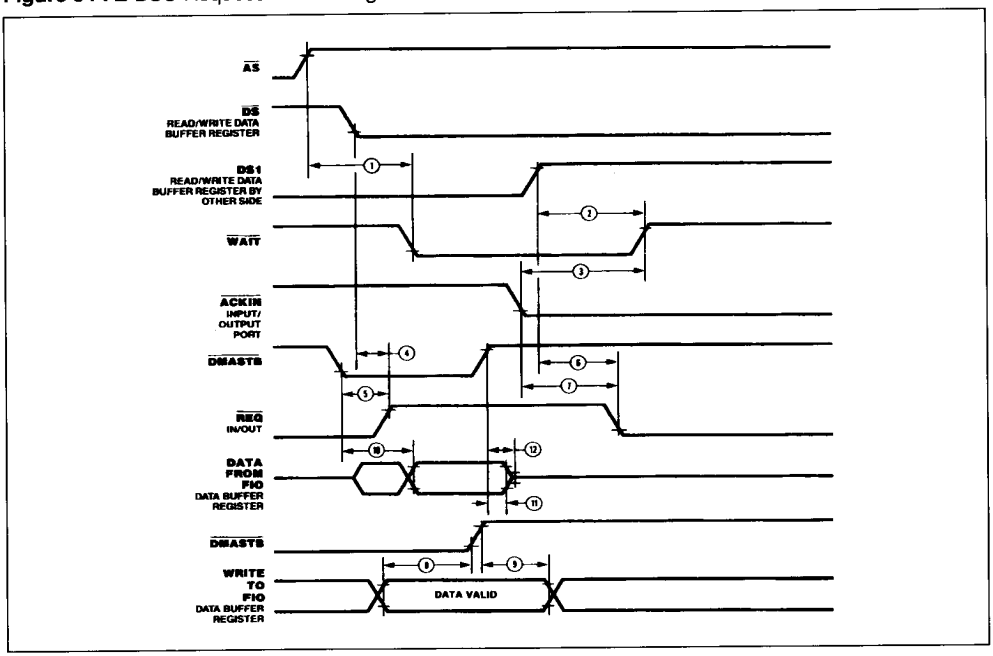


AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TdDS(WAIT)	$\overline{AS}$ ↑ to $\overline{WAIT}$ ↓ Delay		190		160	
2	TdDSI(WAIT)	$\overline{DSI}$ ↑ to $\overline{WAIT}$ ↑ Delay		1000		1000	
3	TdACK(WAIT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ ↑ Delay		1000		1000	1
4	TdDS(REQ)	$\overline{DS}$ ↓ to $\overline{REQ}$ ↑ Delay		350		300	
5	TdDMA(REQ)	$\overline{DMASTB}$ ↓ to $\overline{REQ}$ ↑ Delay		350		300	
6	TdDSI(REQ)	$\overline{DSI}$ ↑ to $\overline{REQ}$ ↓ Delay		1000		1000	
7	TdACK(REQ)	$\overline{ACKIN}$ ↓ to $\overline{REQ}$ ↓ Delay		1000		1000	
8	TdSU(DMA)	Data Setup Time to $\overline{DMASTB}$	200		150		
9	TdH(DMA)	Data Hold Time to $\overline{DMASTB}$	30		20		
10	TdDMA(DR)	$\overline{DMASTB}$ ↓ to Valid Data		150		100	
11	TdDMA(DRH)	$\overline{DMASTB}$ ↑ to Data not Valid	0		0		
12	TdDMA(DR2)	$\overline{DMASTB}$ ↑ to Data Bus Float		70		45	

Notes : 1. The delay is from $\overline{DAV}$ for 3-Wire Input Handshake. The delay is from DAC for 3-Wire Handshake.
* Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 84 : Z-BUS Request/Wait Timing.

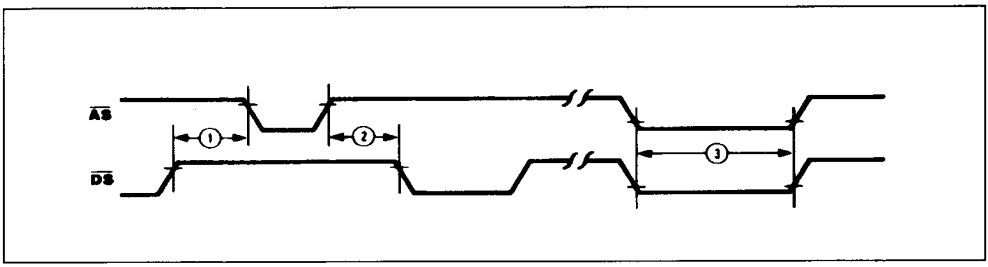


AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TdDSQ(AS)	Delay from $\overline{DS}$ ↑ to $\overline{AS}$ ↓ for no Reset	40		20		
2	TdASQ(DS)	Delay for $\overline{AS}$ ↑ to $\overline{DS}$ ↓ for no Reset	50		30		
3	Tw(AS + DS)	Minimum Width of $\overline{AS}$ and $\overline{DS}$ Both Low for Reset	500		350		1

Notes : 1. Internal circuitry allows for the reset provided by the Z8 (DS held Low while $\overline{AS}$ pulses) to be sufficient.
* Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 85 : Z-BUS Reset Timing.



AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TsA(RD)	Address Setup to $\overline{RD}$ ↓	80		80		1
2	TsA(WR)	Address Setup to $\overline{WR}$ ↓	80		80		
3	ThA(RD)	Address Hold Time to $\overline{RD}$ ↑	0		0		1
4	ThA(WR)	Address Hold Time to $\overline{WR}$ ↑	0		0		
5	TsCEl(RD)	$\overline{CE}$ Low Setup Time to $\overline{RD}$	0		0		1
6	TsCEl(WR)	$\overline{CE}$ Low Setup Time to $\overline{WR}$	0		0		
7	ThCEl(RD)	$\overline{CE}$ Low Hold Time to $\overline{RD}$	0		0		1
8	ThCEl(WR)	$\overline{CE}$ Low Hold Time to $\overline{WR}$	0		0		
9	TsCEh(RD)	$\overline{CE}$ High Setup Time to $\overline{RD}$	100		70		1
10	TsCEh(WR)	$\overline{CE}$ High Setup Time to $\overline{WR}$	100		70		
11	TwRDl	$\overline{RD}$ Low Width	390		250		
12	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		
13	TdRDf(DR)	$\overline{RD}$ ↓ to Valid Data Delay		250		180	
14	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data not Valid Delay	0		0		
15	TdRD(DRz)	$\overline{RD}$ ↑ to Data Bus Float		70		45	2
16	TwWRl	$\overline{WR}$ Low Width	390		250		
17	TsDW(WR)	Data Setup Time to $\overline{WR}$	0		0		
19	Trc(WR)	Write Valid Access Recovery Time	1000		650		
20	Trc(RD)	Read Valid Access Recovery Time	1000 +WR_p		650 +WR_p		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. Recovery time equal to Trc(WR) + write pulse width of the opposite side.
* Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 86 : Non-Z-BUS CPU Interface Timing.

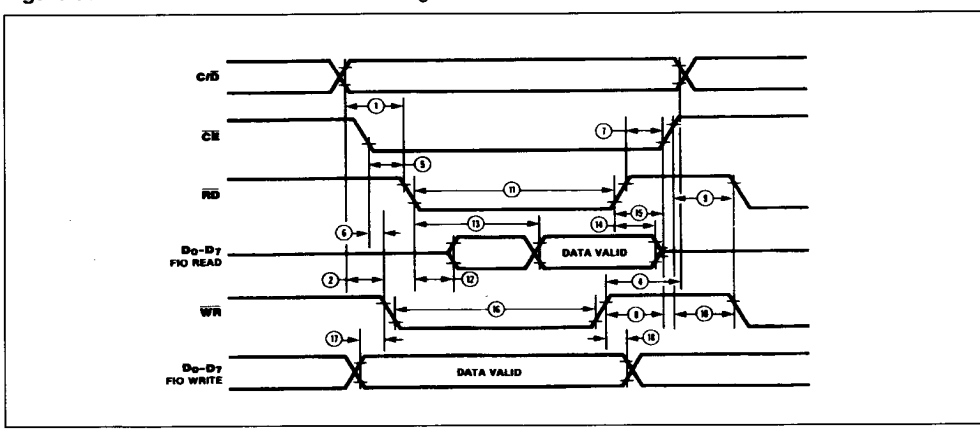
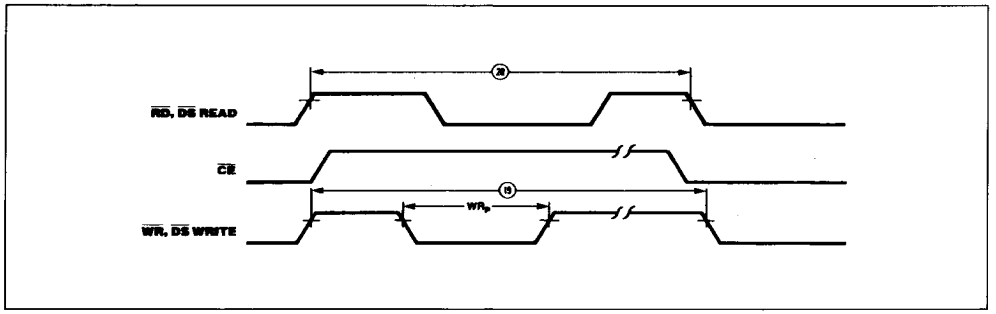


Figure 87 : Z-BUS/Non-Z-BUS Recovery Time.

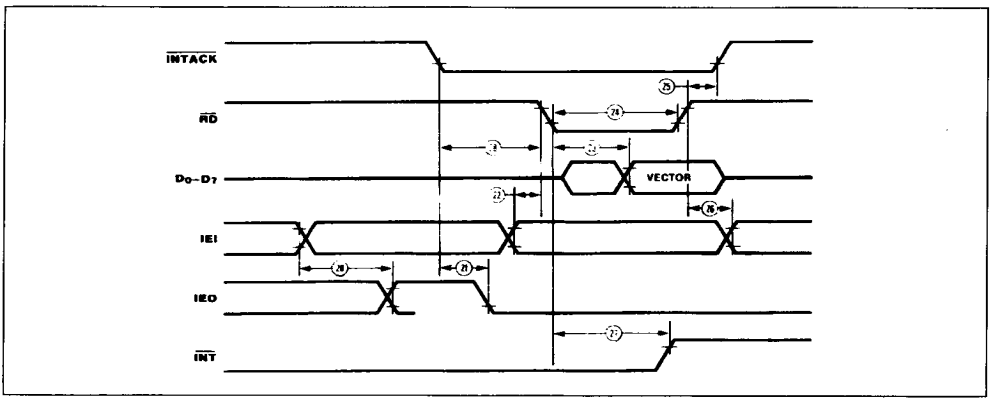


AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
20	TdIEI(IEO)	IEI to IEO Delay		150		100	4
21	TdI(IEO)	$\overline{INTACK}$ ↓ to IEO ↓ Delay		350		250	4
22	TsIEI(RDA)	IEI Setup Time to $\overline{RD}$ (acknowledge)	100		70		4
23	TdRD(DR)	$\overline{RD}$ ↓ to Vector Valid Delay		250		180	
24	TwRDI(IA)	Read Low Width (interrupt acknowledge)	390		250		
25	ThIA(RD)	$\overline{INTACK}$ ↑ to $\overline{RD}$ ↑ Hold Time	30		20		
26	ThIEI(RD)	IEI Hold Time to $\overline{RD}$ ↑	20		10		
27	TdRD(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ ↑ Delay		900		800	
28	TdDCST	Interrupt Daisy Chain Settle Time	350		250		4

Notes : 4. The parameter 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 TdI(IEO) for the highest priority peripheral, TsIEI(RD) for the lowest, priority peripheral, and TdIEI(IEO) for each peripheral separating them in the chain.
† Units in nanoseconds (ns).
* Timings are preliminary and subject to change.

Figure 88 : Non-Z-BUS Interrupt Acknowledge Timing.

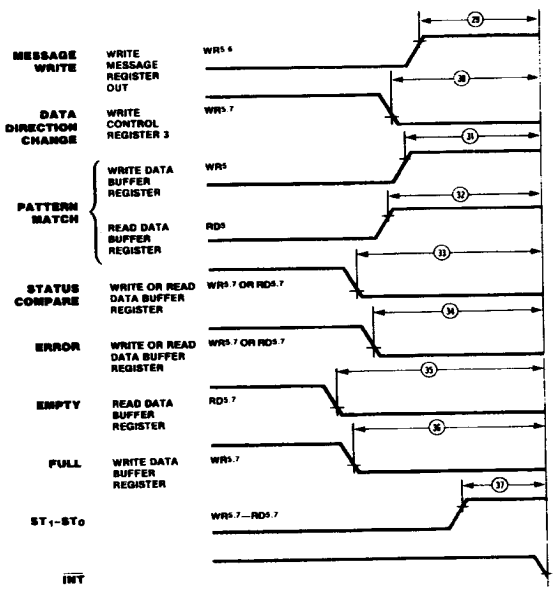


AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
29	TdMW(INT)	Message Write to $\overline{INT}$ Delay					5,6
30	TdDC(INT)	Data Direction Change to $\overline{INT}$ Delay					5,7
31	TdPMW(INT)	Pattern Match (write case) to $\overline{INT}$ Delay					5
32	TdPMR(INT)	Pattern Match (read case) to $\overline{INT}$ Delay					5
33	TdSC(INT)	Status Compare to $\overline{INT}$ Delay					5,7
34	TdER(INT)	Error to $\overline{INT}$ Delay					5,7
35	TdEM(INT)	Empty to $\overline{INT}$ Delay					5,7
36	TdFL(INT)	Full to $\overline{INT}$ Delay					5,7
37	TdSO(INT)	State 0 to $\overline{INT}$ Delay					

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.
* Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 89 : FIO Non-Z-BUS Interrupt Timing.

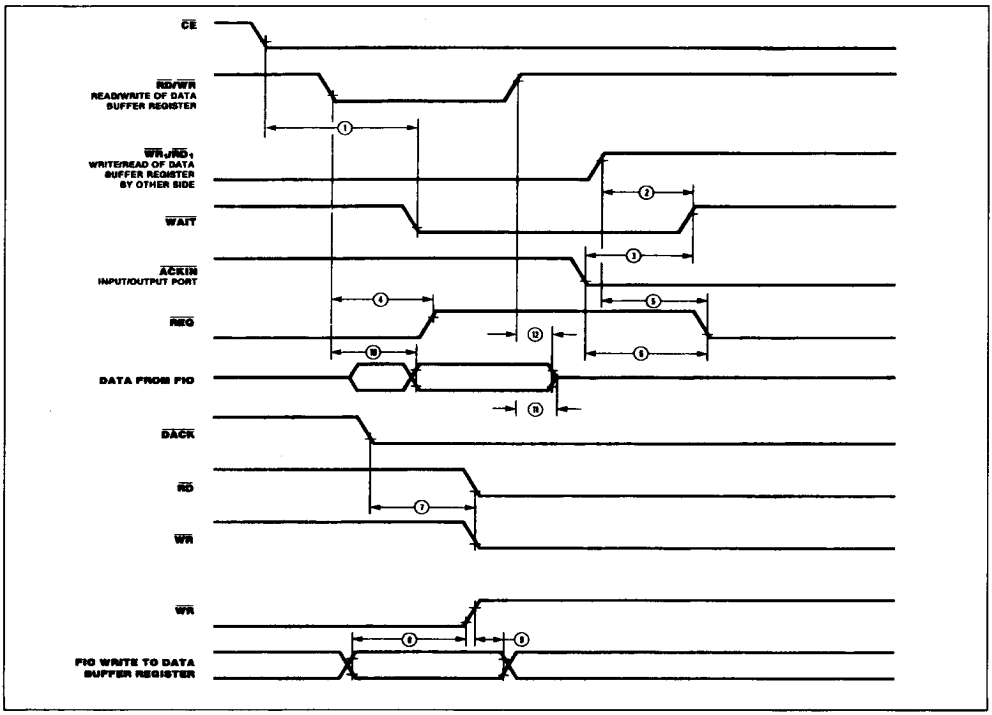


AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TdCE(WT)	$\overline{CE}$ ↓ to $\overline{WAIT}$ Active		200		170	
2	TdRDI(WT)	$\overline{RDI}$ ↑ or $\overline{WRI}$ ↑ to $\overline{WAIT}$ Inactive		1000		1000	
3	TdACK(WT)	$\overline{ACKIN}$ ↓ to $\overline{WAIT}$ Inactive		1000		1000	1
4	TdRD(REQ)	$\overline{RD}$ ↓ or $\overline{WR}$ ↓ to $\overline{REQ}$ Inactive		350		300	
5	TdRDI(REQ)	$\overline{RDI}$ ↑ or $\overline{WRI}$ ↑ to $\overline{REQ}$ Active		1000		1000	
6	TdACK(REQ)	$\overline{ACKIN}$ ↓ to $\overline{REQ}$ Active		1000		1000	
7	TdDAC(RD)	$\overline{DACK}$ ↓ to $\overline{RD}$ ↓ or $\overline{WR}$ ↓	100		80		
8	TSU(WR)	Data Setup Time to $\overline{WR}$	200				
9	Th(WR)	Data Hold Time to $\overline{WR}$	30			20	
10	TdDMA	$\overline{RD}$ ↓ to Valid Data		150		100	2
11	TdDMA(DRH)	$\overline{RD}$ ↑ to Data not Valid	0		0		2
12	TdDMA(DRZ)	$\overline{RD}$ ↑ to Data Bus Float		70		45	2

Notes : 1. The delay is from $\overline{DAV}$ ↓ for 3-Wire Input Handshake. The delay is from DAC ↑ for 3-Wire Output Handshake.
2. Only when $\overline{DACK}$ is active.
* Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 90 : Non-Z-BUS Request/Wait Timing.

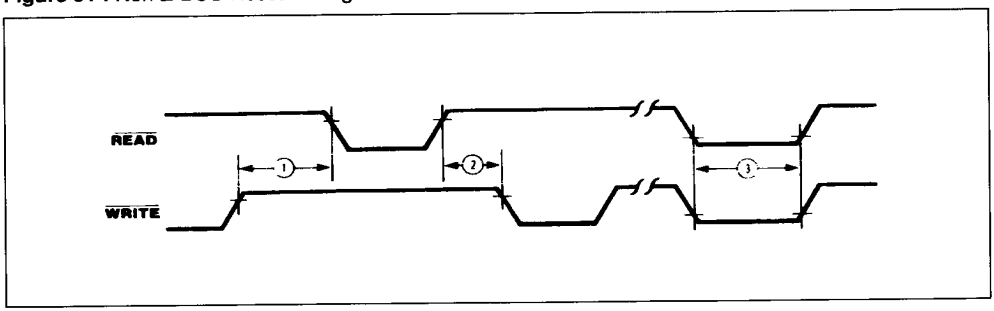


AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TdWR(RD)	Delay from $\overline{WR}$ ↑ to $\overline{RD}$ ↓	100		70		
2	TdRD(WR)	Delay from $\overline{RD}$ ↑ to $\overline{WR}$ ↓	100		70		
3	TwRD + WR	Width of $\overline{RD}$ and $\overline{WR}$, both Low for Reset	500		350		

Notes : * Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 91 : Non-Z-BUS Reset Timing.

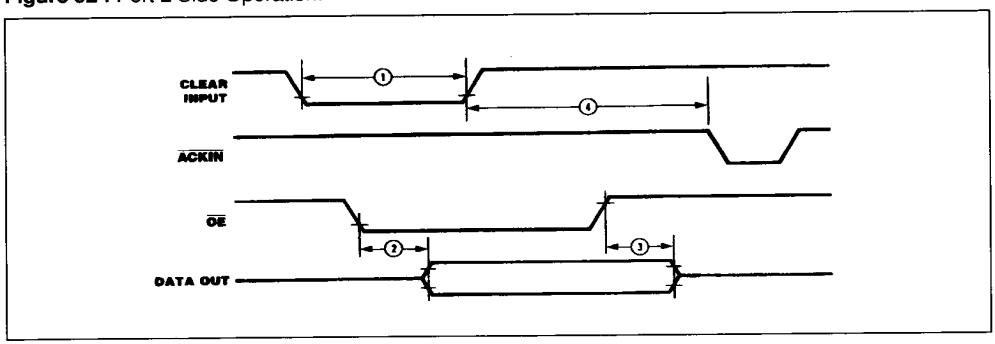


AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TwCLR	Width of Clear to Reset FIFO	700		700		
2	TdOE(DO)	$\overline{OE}$ ↓ to Data Bus Driven		210		210	
3	TdOE(DRZ)	$\overline{OE}$ ↑ to Data Bus Float		150		150	
4	TdCLR(ACK)	$\overline{CLEAR}$ ↑ to $\overline{ACKIN}$ ↓	800		800		

Notes : * Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 92 : Port 2 Side Operation.



AC CHARACTERISTICS

N°	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TsDI(ACK)	Data Input to $\overline{ACKIN}$ ↓ to Setup Time	50		50		
2	TdACKf(RFD)	$\overline{ACKIN}$ ↓ to RFD ↓ Delay	0	500	0	500	
3	TdRFDr(ACK)	RFD ↑ to $\overline{ACKIN}$ ↓ Delay	0		0		
4	TsDO(DAV)	Data Out to $\overline{DAV}$ ↓ Setup Time	50		25		
5	TdDAVf(ACK)	$\overline{DAV}$ ↓ to $\overline{ACKIN}$ ↓ Delay	0		0		
6	ThDO(ACK)	Data Out to $\overline{ACKIN}$ Hold Time	50		50		
7	TdACK(DAV)	$\overline{ACKIN}$ ↓ to $\overline{DAV}$ ↑ Delay	0	500	0	500	
8	ThDI(RFD)	Data Input to RFD ↓ Hold Time	0		0		
9	TdRFDf(ACK)	RFD ↓ to $\overline{ACKIN}$ ↑ Delay	0		0		
10	TdACKr(RFD)	$\overline{ACKIN}$ ↑ ($\overline{DAV}$ ↑) to RFD ↑ Delay-interlocked and 3-wire Handshake	0	400	0	400	
11	TdDAVr(ACK)	$\overline{DAV}$ ↑ to $\overline{ACKIN}$ ↑ (RFD ↑)	0		0		
12	TdACKr(DAV)	$\overline{ACKIN}$ ↑ to $\overline{DAV}$ ↓	0	800	0	800	
13	TdACKf(empty)	$\overline{ACKIN}$ ↓ to Empty	0		0		
14	TdACKf(full)	$\overline{ACKIN}$ ↓ to Full	0		0		
15	TcACK	$\overline{ACKIN}$ Cycle Time	1		1		1

Notes : * Timings are preliminary and subject to change.
† Units in nanoseconds (ns), except as noted.
1. Units in microseconds.

Figure 93 : 2-Wire Handshake (port 2 side only) Output.

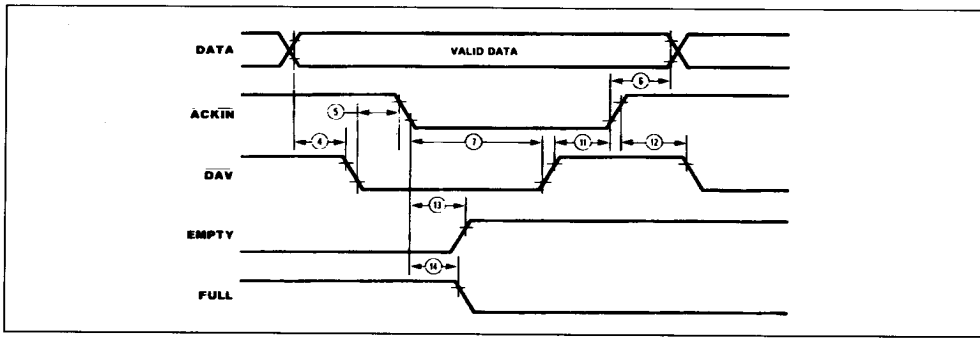
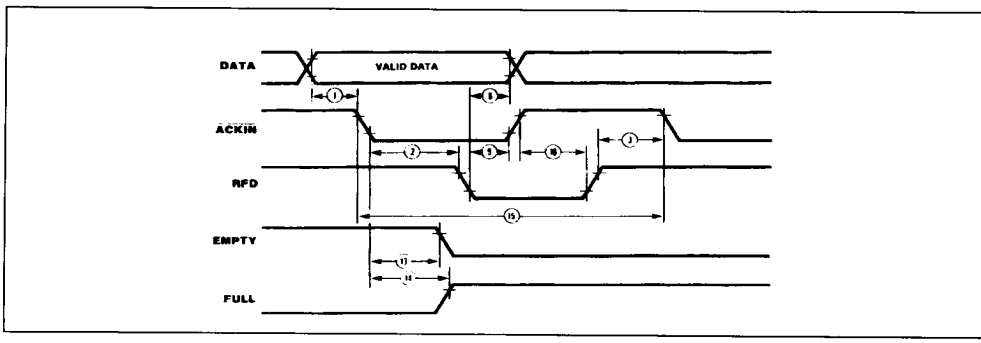


Figure 94 : 2-Wire Handshake (port 2 side only) Input.



AC CHARACTERISTICS (Continued)

N°.	Symbol	Parameter	4 MHz		6 MHz		Notes*†
			Min.	Max.	Min.	Max.	
1	TsDI(DAV)	Data Input to $\overline{DAV}$ ↓ Setup Time	50		50		
2	TdDAVIf(RFD)	$\overline{DAV}$ ↓ to RFD ↓ Delay	0	500	0	500	
3	TdDAVf(DAC)	$\overline{DAV}$ ↓ to $\overline{DAC}$ ↑ Delay	0	500	0	500	
4	ThDI(DAC)	Data in to DAC ↑ Hold Time	0		0		
5	TdDACIr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay	0		0		
6	TdDAVIr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay	0	500	0	500	
7	TdDAVIr(RFD)	$\overline{DAV}$ ↑ to RFD ↑ Delay	0	500	0	500	
8	TdRFDI(DAV)	RFD ↑ to $\overline{DAV}$ ↓ Delay	0		0		
9	TsDO(DAC)	Data Out to $\overline{DAV}$ ↓					
10	TdDAVOf(RFD)	$\overline{DAV}$ ↓ to RFD ↓ Delay	0		0		
11	TdDAVOf(DAC)	$\overline{DAV}$ ↓ to DAC ↑ Delay	0		0		
12	ThDO(DAC)	Data Out to DAC ↑ Hold Time					
13	TdDACOr(DAV)	DAC ↑ to $\overline{DAV}$ ↑ Delay		400		400	
14	TdDAVOr(DAC)	$\overline{DAV}$ ↑ to DAC ↓ Delay	0		0		
15	TdDAVOr(RFD)	$\overline{DAV}$ ↑ to RFD ↑ Delay	0		0		
16	TdRFDO(DAV)	RFD ↑ to $\overline{DAV}$ ↓ Delay	0	800	0	800	

Notes : * Timings are preliminary and subject to change.
† Units in nanoseconds (ns).

Figure 95 : 3-Wire Handshake Input.

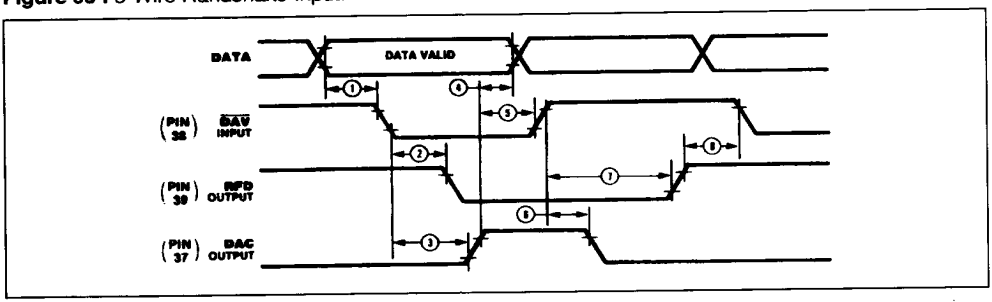
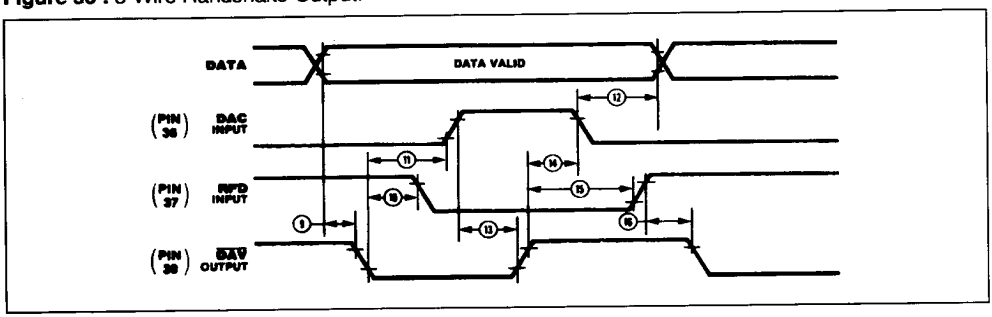


Figure 96 : 3-Wire Handshake Output.



ORDERING INFORMATION

Type	Package	Temp. Range	Clock
Z8038B1V	PDIP-40	0 to + 70 °C	4 MHz
Z8038B6V	PDIP-40	– 40 to + 85 °C	4 MHz
Z8038C1V	PLCC44	0 to + 70 °C	4 MHz
Z8038C6V	PLCC44	– 40 to + 85 °C	4 MHz
Z8038D1N	CDIP-40	0 to + 70 °C	4 MHz
Z8038D6N	CDIP-40	– 40 to + 85 °C	4 MHz
Z8038D2N	CDIP-40	– 55 to + 125 °C	4 MHz
Z8038AB1V	PDIP-40	0 to + 70 °C	6 MHz
Z8038AB6V	PDIP-40	– 40 to + 85 °C	6 MHz
Z8038AC1V	PLCC44	0 to + 70 °C	6 MHz
Z8038AC6V	PLCC44	– 40 to + 85 °C	6 MHz
Z8038AD1N	CDIP-40	0 to + 70 °C	6 MHz
Z8038AD6N	CDIP-40	– 40 to + 85 °C	6 MHz
Z8038AD2N	CDIP-40	– 55 to + 125 °C	6 MHz

Note : PDIP = Plastic DIP ; CDIP = Ceramic Multilayer DIP ; PLCC = Plastic Leaded Chip Carrier.